LIVRABLES EN COMPLEMENT DU RAPPORT SCIENTIFIQUE CAPLOC

CAPLOC

Combinaison de l'Analyse d'images
et de la connaissance de la Propagation
des signaux pour la LOCalisation

Convention 10-MT-CVS-001



- **A** : MEURIE (Cyril), ATTIA (Dhouba), RUICHEK (Yassine).- Livrable de la tâche 1.1 : **Extraction de descripteurs de couleur et de texture pour la caractérisation de l'environnement de réception des signaux GNSS.**- 28 octobre 2013.- 37 p.

- **B** : MEURIE (Cyril), ATTIA (Dhouba), RUICHEK (Yassine).- Livrable de la tâche 1.2 : **Détection de la ligne d'horizon et évaluation du pourcentage de ciel visible par analyse d'images couleur**.- 18 octobre 2012.- 36 p.

- **C** : MEURIE (Cyril), CONIGLIO (Christophe), FLANCQUART (Amaury), ATTIA (Dhouba), RUICHEK (Yassine).- Livrable de la tâche 1.3 : **Segmentation et classification fine de l'environnement de réception des signaux GNSS**.- 18 janvier 2015.- 29 p.

- **D** : AMBELLOUIS (Sébastien), LEFEBVRE (Sébastien), FLANCQUART (Amaury), MOREAU (Julien), RUICHEK (Yassine).- Livrable de la tâche 2.1 : **Calcul du nuage de points 3D.**- 18 octobre 2012.- 39 p.

- **E :** AMBELLOUIS (Sébastien), FLANCQUART (Amaury), MOREAU (Julien), RUICHEK (Yassine).- Livrable de la tâche 2.2 : **Calcul du nuage de points 3D et extraction d'informations**.- 18 janvier 2015.- 55 p.

- **F** : MARAIS (Juliette) et al.- Livrable d'avancement de la tâche 3. **Utilisation de l'image pour la caractérisation des signaux reçus et de la disponibilité globale de la constellation**.- 12 juin 2013.- 27 p.

- **G** : MARAIS (Juliette), AMBELLOUIS (Sébastien), TAY (Sarab), MOREAU (Julien).- Livrable 3.2. **Rapport final de la tâche 3**.- 18 janvier 2015.- 29 p.

- **H** : MARAIS (Juliette), MEURIE (Cyril), AMBELLOUIS (Sébastien), FLANCQUART (Amaury), MOREAU (Julien), ATTIA (Dhouba), RUICHEK (Yassine).- Livrable 4.2. **Bilan des publications**.- 18 janvier 2015.- 14 p.

- **I** : MARAIS (Juliette), MEURIE (Cyril), AMBELLOUIS (Sébastien), FLANCQUART (Amaury), MOREAU (Julien), ATTIA (Dhouba), RUICHEK (Yassine).- Livrable 4.3. **Bilan des résultats et recommandations.**- 18 janvier 2015.- 9 p.

22505

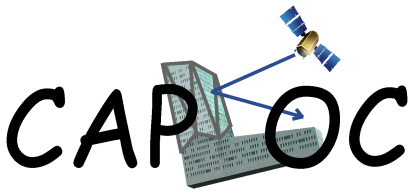
Groupe opérationnel n°2 du Predit 4

« Gestion du trafic et des mobilités »

CAPLOC

Combinaison de l'**A**nalyse d'images
et de la connaissance de la **P**ropagation
des signaux pour la **LOC**alisation

Convention 10-MT-CVS-001



Livrable de la tâche 1.1

Extraction de descripteurs de couleur et de texture pour la caractérisation de l'environnement de réception des signaux GNSS

28 octobre 2013

Cyril MEURIE
IFSTTAR-LEOST
Dhouha ATTIA, Yassine RUICHEK
UTBM/IRTES-SET

Identification

Acronyme du projet	CAPLOC
Titre du projet	Combinaison de l'Analyse d'images et de la connaissance de la Propagation des signaux pour la LOCalisation
Coordinatrice du projet	Juliette MARAIS (IFSTTAR-LEOST)
Date de début de projet	19 octobre 2010
Date de fin de projet	18 octobre 2014
Partenaires	IFSTTAR-LEOST, UTBM-SeT
Type de document	Livrable
Date contractuelle de livraison	18 octobre 2013
Accessibilité	La diffusion de ce document est limitée au programme PREDIT

Gestion des versions

Version	**Date**	**Auteurs**	**Modifications**
1.0	18/10/2013	Cyril MEURIE, Dhouha ATTIA, Yassine RUICHEK	Rédaction
1.1	27/10/2013	Yassine RUICHEK	Corrections
1.2	28/10/2013	Dhouha ATTIA	Corrections
2.0	28/10/2013	Cyril MEURIE	Approbation

Table des matières

1 Introduction **4**

2 La couleur **5**
2.1 La définition physiologique de la couleur 5
2.2 Les modèles de représentation de la couleur 6
2.3 La couleur dans l'image 7

3 La texture **7**
3.1 Attributs géométriques 10
3.2 Attributs basés sur la modélisation spatiale 10
3.2.1 Les modèles autorégressifs multispectraux 10
3.2.2 Les modèles de Markov 10
3.2.3 Les motifs locaux binaires 12
3.3 Attributs spatio-fréquentiels 13
3.3.1 Dans le domaine spatial 13
3.3.2 Dans le domaine fréquentiel 13
3.3.3 Dans le domaine spatio-fréquentiel 13
3.4 Attributs statistiques 13
3.4.1 Attributs statistiques de premier ordre 14
3.4.2 Attributs statistiques de second ordre 15
3.4.3 Attributs statistiques d'ordre supérieur 17
3.5 Attributs issus de la morphologie mathématique 17

4 Les descripteurs de couleur et de texture **17**
4.1 Les descripteurs de couleur 18
4.1.1 Gradient morphologique couleur 18
4.1.2 Gradient de Di-Zenzo 18
4.2 Les descripteurs de texture 20
4.2.1 Gradient basé sur les filtres de Gabor 20
4.2.2 Gradients basés sur les paramètres d'Haralick 21
4.2.3 Gradient basé sur le filtre alterné séquentiel 25

5 Conclusion **31**

Table des figures **32**

Bibliographie **36**

1 Introduction

Dans le projet CAPLOC, l'objectif de la tâche 1 est de développer un système automatique de traitement et d'analyse d'images permettant de caractériser l'environnement de réception des signaux satellitaires et de fournir un ensemble de primitives pertinentes. Pour ce faire, nous utilisons un véhicule de laboratoire instrumenté permettant d'acquérir, pour un parcours donné, des informations de position (grâce à un GPS) et des données images (à partir d'une caméra couleur munie d'un objectif fisheye disposée sur le toit du véhicule). Le travail réalisé dans cette première tâche est donc décomposé en trois sous-tâches rappelées ci-après :

1. Étalonnage et calibrage du système d'acquisition ;
2. Détection de la ligne d'horizon et évaluation du pourcentage de ciel visible par analyse d'images couleur ;
3. Extraction et caractérisation de la texture ;

Dans la sous-tâche 1.1, nous avions initialement projeté de réaliser une première étape de rectification des images afin de corriger les fortes distorsions dues à l'objectif grand angle utilisé. Cependant, comme indiqué, dans le précédent livrable (du 18 octobre 2012), une étude réalisée dans le cadre de travaux antérieurs a permis de montrer que cette opération ajoute un coût calculatoire qui impacte le temps de traitement des images et nous éloigne ainsi de la contrainte temps réel imposée dans la sous-tâche 1.2. Les très bons résultats de détection de la ligne d'horizon et de pourcentage de ciel visible présentés dans le précédent livrable ainsi que les premiers résultats d'extraction de descripteurs couleur/texture présentés dans ce livrable nous amène à conclure que cette sous-tâche n'est finalement pas nécessaire au vue des attentes finales de la tâche 1. Néanmoins, afin de montrer l'avancement de cette tâche, nous avons décidé, non pas de supprimer le livrable correspondant, mais de le remplacer par ce livrable portant sur l'extraction de descripteurs couleur/texture pour la caractérisation de l'environnement de réception des signaux satellitaires.

Dans la sous-tâche 1.2, nous avons cherché à détecter la ligne d'horizon et évaluer le pourcentage de ciel visible présent dans l'image. Pour ce faire, nous avons proposé une stratégie basée sur une simplification de l'image couplée à une classification pixellaire adaptée. Différentes méthodes de simplification d'images issues de la littérature (filtre médian, filtre gaussien, filtre moyenneur, filtre de nagao, reconstruction géodésique par dilatation) ont été testées, tout comme un certain nombre de classifieurs (non)supervisés reconnus pour leur bons résultats (KM-local, FuzzyCmeans, Fisher, SRM, Bayes, Kppv, SVM). Les premiers tests ont été effectués sur quelques images de la base du projet LOCOPROL, afin d'identifier les deux meilleures méthodes de simplification. Nous sommes arrivés à la conclusion que le filtre médian et la reconstruction géodésique par dilatation (RGD) offraient les meilleurs résultats. Nous avons ensuite réalisé une deuxième série de tests sur des images (très hétérogènes et de complexité variée) de la base du projet CAPLOC afin de définir le meilleur couple « méthode de simplification/classifieur utilisé ». Nous avons conclu que l'utilisation d'un filtre médian ou d'une RGD couplée à l'algorithme de Fisher (sur la composante bleue) ou à l'algorithme des Kppv (avec un nombre de $kppv = 6$) offre les meilleurs taux de bonne classification. En effet, nous obtenons un taux de 97,53% pour une combinaison « Médian/Fisher » et de 97.99% pour une combinaison « RGD/Kppv ». La prise en compte du temps de traitement de l'approche proposée nous a fait choisir le couple Médian/Fisher (0.035s au lieu de 15.4s soit un rapport de 1/400). Cependant, nous souhaitons confirmer ce choix par rapport au couple « RGD/Fisher » très bien placé lui aussi (taux = 96.4%) sur un nombre important d'images issues de la prochaine campagne d'acquisition du projet CAPLOC qui est programmée à Lille en octobre 2013. Ces traitements nous ont donc permis

d'identifier précisément les régions de « ciel » dans l'image (respectivement de « non-ciel »), de détecter la ligne d'horizon et de pouvoir après repositionnement des satellites dans l'image, identifier ceux dont les signaux sont reçus via des trajets directs (resp. réfléchis voire bloqués). Les résultats issus de cette sous-tâche ont été utilisés dans la tâche 3 (cf. livrable 3.1) afin de montrer (dans certaines conditions) l'apport de l'analyse d'images pour l'exclusion de certains satellites dans le calcul d'une nouvelle position (se voulant plus précise).

Dans la sous-tâche 1.3, nous cherchons à caractériser l'environnement de réception des signaux satellitaires de manière plus fine (que la classification ciel/non-ciel effectuée dans la tâche 1.2) de sorte à pouvoir i) pondérer l'impact de chacun des satellites dans le calcul de position de localisation (selon s'ils se trouvent dans une classe « ciel », « végétation », « bâtiment », etc) ; ii) caractériser par des attributs pertinents certaines textures rencontrées afin de pouvoir reproduire la texture des plans du modèle 3D (tâche 2). Pour ce faire, nous pensons qu'une segmentation de l'image utilisant conjointement les informations de texture et de couleur faite au préalable, ne pourra qu'améliorer la classification du contenu de l'image en différentes classes (au minimum : « ciel », « végétation », « bâtiment »). Cette stratégie, y compris la caractérisation de textures propres à certains objets d'intérêt présents dans l'image, fait appel à l'utilisation de descripteurs de couleur et de texture décrits dans la thèse de Dhouha ATTIA (doctorante financée par la région Franche-Comté et mis à disposition pour la tâche 1 du projet CAPLOC. Ce livrable reprend une partie de ces travaux de thèse.

Dans ce livrable dédié à la sous-tâche 1.1, nous commençons par présenter les définitions de couleur et de texture qui constituent à elles deux, les deux sources d'information les plus utilisées dans différentes applications de vision par ordinateur. Nous présentons ensuite un état de l'art exhaustif sur les diverses familles d'attributs de texture existantes dans la littérature, tout en soulignant leurs caractéristiques et détaillant leur mode de calcul. En effet, ces attributs sont classés en fonction de leur technique de calcul ainsi que les outils mathématiques mis en œuvre. Cette sélection d'attributs couleur et texture sera utilisée lors de la phase de segmentation permettant ainsi de partitionner en régions homogènes l'image acquise en mobilité et ainsi faciliter la classification en trois classes « ciel », « végétation », « bâtiment »dans le but de caractériser l'environnement de réception des signaux satellitaires. Nous pensons également que l'utilisation de ces descripteurs permettra de caractériser certaines textures rencontrées afin de pouvoir reproduire la texture des plans du modèle 3D (tâche 2). Nous informons le lecteur que ce rapport reprend une partie du manuscrit de la thèse de Dhouha ATTIA (doctorante financée par la région Franche-Comté et mis à disposition pour la tâche 1 du projet CAPLOC) [1].

2 La couleur

Nous commençons par rappeler brièvement la définition physiologique de la couleur, ce qui permettra de comprendre l'interprétation de ce phénomène par le cerveau humain. Ensuite, nous introduisons, avec plus de détails, les différentes représentations de la couleur. Ensuite, nous rappelons le codage de cette source d'information dans une image.

2.1 La définition physiologique de la couleur

La couleur n'est ni une matière, ni une lumière mais plutôt une sensation reçue par l'intermédiaire de l'œil humain et traduite par notre cerveau lors de la vision d'un objet coloré. Par conséquent, la couleur est une sensation physiologique qui dépend d'un ensemble de facteurs :

- la nature de l'objet perçu.
- la lumière incidente qui permet à l'œil de recevoir le message.
- l'œil qui perçoit cette lumière et communique cette sensation au cerveau.

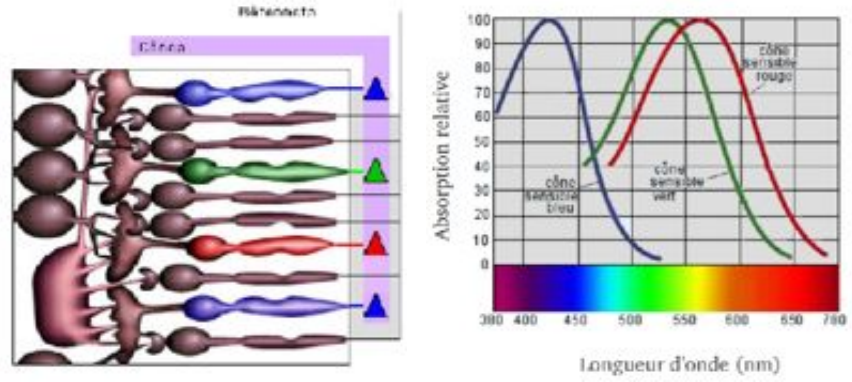
2.2 Les modèles de représentation de la couleur

La couleur joue un rôle fondamental dans la distinction des objets constituant une scène/image, étant donnée l'analogie entre le fonctionnement de l'œil humain et les modèles de la vision humaine. Avant d'introduire chacun de ces modèles, nous rappelons que la théorie tri-chromatique de la lumière a été introduite par Thomas Young en 1801 [2] et complétée dans les travaux de Hermann Von Helmotz en 1804 [2]. Cette théorie précise que la rétine possède trois cônes photorécepteurs couleur sensibles respectivement au bleu, vert et rouge (cf. figure 1(a)). Ces éléments de réception, situés dans la rétine, reçoivent les radiations de la lumière incidente dont les longueurs d'onde doivent appartenir au spectre visible avec des longueurs d'onde comprises entre $380nm$ et $780nm$ et qui répondent chacun selon sa courbe de sensibilité spectrale, notée :

- S pour les *Short wavelength* ou bien B (cône sensible au bleu).
- M *Medium wavelength* ou bien G (cône sensible au vert).
- L *Long wavelength* ou bien R (cône sensible au rouge)

.

Comme le montre la figure 1(b), ces courbes sont caractérisées par des pics appelés "pics d'absorption", qui correspondent respectivement aux longueurs d'onde 440 nm, 545 nm et 570 nm.



(a) Cônes photorécepteurs. (b) Courbes d'absorption.

Figure 1 – Sensibilité spectrale des trois types de cônes photorécepteurs couleur de l'œil humain et des bâtonnets (source : `http://tpelavision.chez-alice.fr/oeil.htm`).

Dans ce qui précède, le modèle *tri-stimulus* a été introduit. Il s'agit du modèle le plus répandu pour le codage des couleurs dans différents espaces. Il existe d'autres modèles de représentation de la couleur dont le principe est rappelé brièvement dans ce qui suit :

- **Le modèle des couleurs opposées**, développé par Hering, consiste à considérer la couleur comme une sensation psychologique. Hering s'éloigne des notions physiques et affirme qu'il n'y a pas trois couleurs fondamentales, mais plutôt quatre primaires correspondantes à des sensations psychologiques et qui sont le rouge, le vert, le bleu et le jaune. D'après le fondateur de ce modèle, le jaune peut avoir une dominante rouge ou une dominante verte, mais en aucun cas une dominante bleue en justifiant sa théorie par les couleurs opposées rouge-vert et bleu-jaune. En suivant la même logique, Hering considère la luminosité comme une opposition blanc-noir [3, 4].
- **Le modèle de Faugeras**, introduit par ce dernier, consiste à transformer la réponse non-linéaire des cônes photorécepteurs en une réponse proportionnelle au logarithme de la lumière incidente. Les composantes couleur fournies par ce modèle sont deux composantes chromatiques et une troisième composante achromatique [5].

– **Le modèle de Pratt**, instauré par ce dernier dans ses travaux [6], est défini contrairement aux autres modèles, par trois nouvelles notions : la teinte, la saturation et la luminance.

Les chercheurs ont instaurés de nouveaux modèles représentant la couleur dont les plus récents ont été proposés en 1998 et 2002 par la CIE [7, 8]. Dans ce qui suit, nous adoptons le modèle couleur le plus répondu : le modèle tri-chromatique.

2.3 La couleur dans l'image

Dans le domaine du traitement d'images, une image couleur donnée I est définie comme étant un ensemble de pixels ayant chacun trois composantes rouge, verte et bleue, suite à la numérisation de la scène par une caméra couleur. Ces composantes peuvent être aussi exprimées dans d'autres espaces couleur. Pour plus amples informations sur ces différents espaces couleur, le lecteur pourra consulter la thèse de Vandenbroucke [9] qui décrit en détail cet aspect.

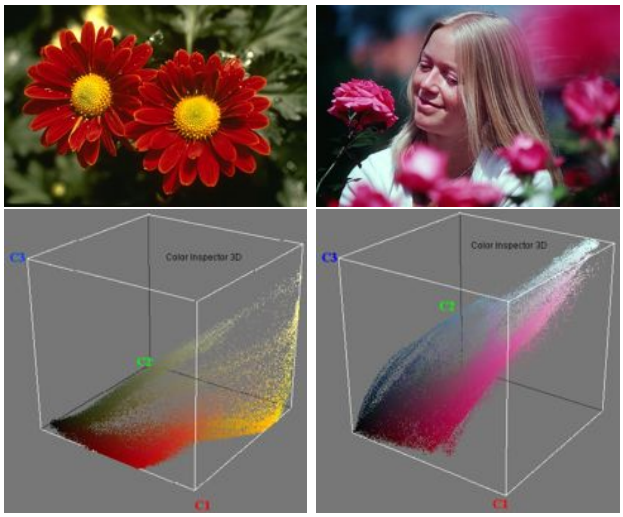


Figure 2 – Représentation 3D de deux images couleur dans un espace muni du repère (C_1,C_2,C_3) à l'aide du logiciel "Color Inspector 3D" (source : `http://www.f4.fhtw-berlin.de/~barthel/ImageJ/ColorInspector//help.htm`,).

D'une manière générale, quel que soit l'espace couleur considéré, un pixel p d'une image est décrit par trois valeurs $C_1(p)$, $C_2(p)$ et $C_3(p)$. Ceci permet de représenter le pixel par un point dans l'espace et de percevoir l'image comme un nuage de points (cf.figure 2).

3 La texture

Au même titre que la couleur, la texture est un facteur fondamental dans la perception de l'environnement et la reconnaissance de ses objets. Contrairement à la couleur, la texture reste difficile à définir de manière précise et générique. Le nombre de définitions possibles données dans la littérature en témoignent [10, 11, 12]. La définition littéraire de la texture est la "répétition spatiale d'un même motif dans différentes directions de l'espace". Cependant, cette explication est parfois considérée insuffisante étant donné qu'elle est indépendante du comportement de l'observateur humain. À titre d'exemple, la texture d'un tissu ou d'un mur de briques (cf. figure 3) peut satisfaire cette définition, ce qui n'est pas toujours le cas pour d'autres textures. La texture possède plusieurs aspects visuels discriminants à savoir l'aspect grossier, fin, granuleux,

régulier ou irrégulier (cf. figure 4). Par conséquent, l'homogénéité de la surface est le critère à considérer pour différencier deux objets de textures différentes.



FIGURE 3 – Exemples de textures répétitives prises de la base de texture VISTEX [13].

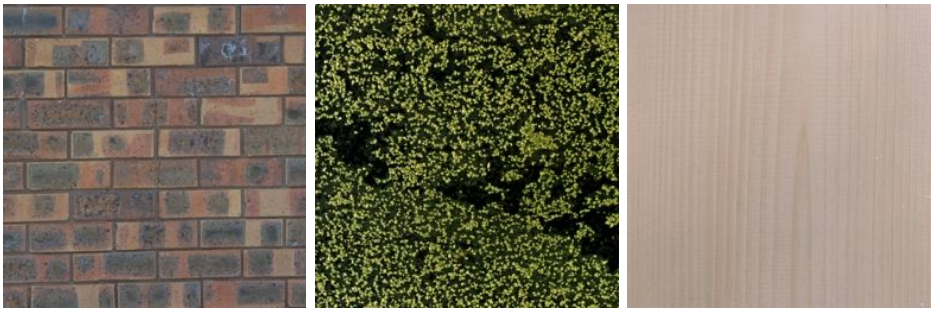
La texture est décrite dans la littérature par deux caractéristiques à savoir les contours (information mono-dimensionnelle) qui sépare les régions homogènes et la surface (information bidimensionnelle) qui définit ces régions homogènes. Cependant, cette description peut être erronée dès qu'un changement de résolution de l'image survient, ce qui entraîne la modification de l'échelle d'observation.



(a) Granuleux (b) Grossier (c) Fin (d) Régulier (e) Irrégulier

FIGURE 4 – Différents types de textures extraites de la base de texture VISTEX [13].

Dans la littérature, un certain nombre de travaux [12, 14] regroupent les approches de description de texture en deux principales familles. La première catégorie repose sur une description macroscopique de la texture (cf. figure 5(a)) basée sur la définition d'éléments primitifs répétitifs. Plus précisément, cette famille considère la texture comme étant un agencement spatial plus ou moins régulier d'éléments de base ou de primitives. Une primitive est une agglomération connexe de pixels ayant des propriétés similaires [15]. Les partisans d'une telle description de texture [11, 12, 16, 15] s'orientent généralement vers des approches spectrales ou structurelles qui sont des approches déterministes pour la caractériser. La deuxième famille d'approches est basée sur une étude microscopique de la texture (cf. figure 5(b)). Ces approches s'intéressent aux surfaces microscopiques isolables disposées généralement de manière anarchique et qui peuvent se répéter de manière discernable comme dans le cas du sable, gravier, herbe, etc. Les partisans [10] d'une telle caractérisation de la texture s'orientent généralement vers des approches probabilistes. Dans la plupart des cas, la texture peut être ni totalement structurelle ni totalement aléatoire [17, 18]. Ce type de texture se caractérise essentiellement par la répétition d'un motif de base selon certaines directions dominantes, sans qu'il y ait vraiment de répartition régulière ou complètement aléatoire. Cette texture est appelée texture orientée. C'est le cas de l'image de la figure 5(c) issue de la base de texture de VISTEX.



(a) Texture macroscopique (b) Texture microscopique (c) Texture orientée

FIGURE 5 – Différentes catégories de texture.

Dans les images naturelles, la texture se présente sous la forme d'une mosaïque de différents types de textures. Par conséquent, la caractérisation d'un nombre infini de textures dans une image naturelle s'avère une tâche plus compliquée et moins précise, avec les outils usuels issus de la vision par ordinateur comparés à la capacité de l'œil humain. TUCERYAN [19] propose un ensemble de propriétés texturales intuitives exploitées dans l'analyse d'images en niveaux de gris. Les outils exploitant ces propriétés permettent d'étudier les impressions visuelles d'un observateur humain, ce qui permet de comprendre et de reproduire au mieux la caractérisation des différentes textures. Ces propriétés sont les suivantes :

- La texture peut être perçue à différents niveaux d'échelle [1] et à différents niveaux de résolution spatiale [2].
- La texture est un arrangement spatial des niveaux de gris des pixels.
- L'étude de la texture demande la définition d'un voisinage spatial ayant une taille qui dépend du type de la texture ou de la surface de son motif de base.
- Si le nombre de motifs de base est suffisamment important dans une région, cette dernière peut être décrite comme texturée.

La texture dans l'image peut être extraite en utilisant des outils mathématiques permettant d'obtenir des attributs de texture. Les paramètres calculés permettent donc de caractériser, et analyser les images en question afin de les segmenter. Dans la littérature, il existe quatre classes connues d'attributs de texture [19, 20] (cf. figure 6) : les attributs géométriques qui sont peu utilisés, les attributs basés sur la modélisation spatiale des textures (tels que les modèles autorégressifs, les modèles de Markov et les motifs locaux binaires (Local Binary Patterns : LBP)), les attributs spatio-fréquentiels (telles que la transformée de Gabor, la transformée en ondelettes et la transformée de Fourrier) et les attributs statistiques (tels que les attributs de premier ordre : la moyenne et la variance, les attributs de second ordre : les histogrammes et les matrices de co-occurrences et les attributs d'ordre supérieur : les matrices de longueurs de plages). Notons qu'un attribut peut satisfaire plusieurs propriétés. Dans ce rapport, nous considérons une cinquième catégorie d'attributs de texture calculés à l'aide des opérateurs de la morphologie mathématique. Nous détaillons dans la suite de ce rapport, chaque catégorie d'attributs.

1. L'échelle correspond ici au champ de vision : c'est la taille en millimètre des côtés de la surface observée
2. La résolution spatiale correspond au nombre de pixels par unité de longueur

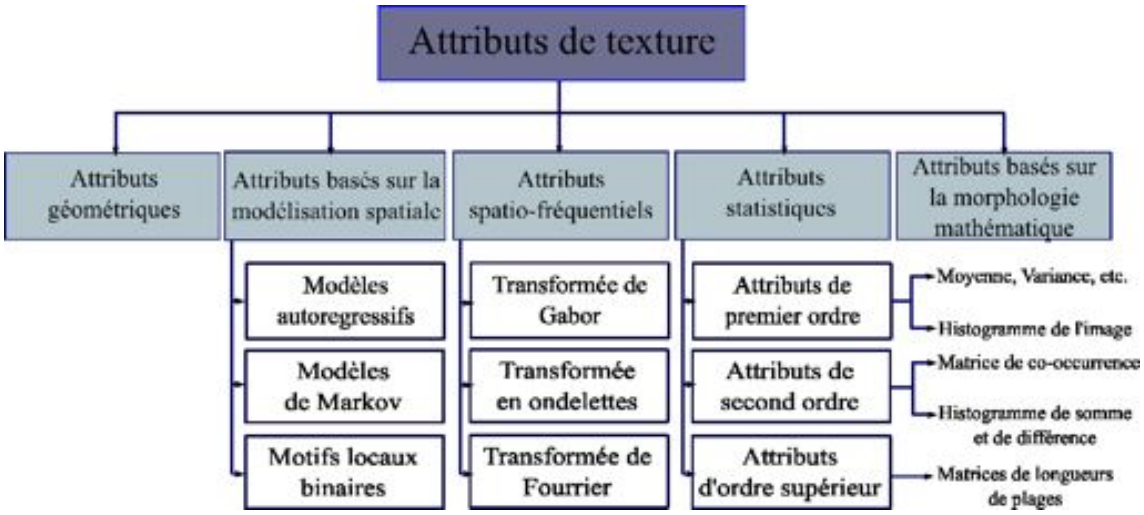


FIGURE 6 – Les différentes classes d'attributs de texture de la littérature.

3.1 Attributs géométriques

La famille d'attributs géométriques est généralement adaptée à la perception macroscopique de la texture. Le calcul de ce type d'attributs commence par chercher des motifs de base (primitifs) de la texture et identifie ensuite des règles de placement de ces primitifs. En effet, les attributs géométriques exigent la régularité de la texture étudiée. Bien que des travaux de ZHENG [21] utilisent ce type d'attributs dans le but de caractériser la texture de la viande, il est à noter que cette famille d'attributs est très peu utilisée vu qu'elle ne prend pas en compte les irrégularités de la texture fortement présentes dans les images naturelles. Par conséquent, nous privilégions les attributs qui sont indépendants des conditions a priori faite sur la géométrie de la texture étudiée.

3.2 Attributs basés sur la modélisation spatiale

Cette catégorie d'attributs dépend essentiellement de la répartition spatiale des couleurs dans l'image. Par conséquent, le calcul d'attributs est effectué en modélisant cette répartition spatiale. Parmi les modèles les plus répandus, nous pouvons citer les trois modèles suivants [20] :

3.2.1 Les modèles autorégressifs multispectraux

Les modèles autorégressifs multispectraux sont issus de l'extension des modèles autorégressifs en niveaux de gris. En effet, en un pixel p, les valeurs des composantes couleur sont obtenues en utilisant des combinaisons linéaires des niveaux des composantes couleur des pixels voisins auxquels un bruit blanc est ajouté. Les attributs de texture sont donc obtenus par estimation des paramètres de la combinaison linéaire. Le modèle Multispectral Simultaneous Auto-Regressive (MSAR) est le modèle le plus répandu pour des applications de segmentation de texture dans une image couleur [22, 23]. Le modèle MSAR est également utilisé pour la classification des bases de texture couleur [24, 25]. D'autres auteurs ont proposé l'extension des modèles autoregressifs, en séparant la luminance de la chrominance afin de caractériser les textures couleur de la base VisTex [26].

3.2.2 Les modèles de MARKOV

Les modèles de MARKOV constituent un cas particulier des modèles autoregressifs. L'intérêt de ces modèles (par rapport aux modèles autoregressifs) réside dans leur stratégie de modélisation

des interactions spatiales entre un pixel et ses voisins, ce qui nécessite la définition d'un système de voisinage. En effet, pour définir les champs de Markov, l'image est modélisée en un ensemble fini S de sites s_i correspondant aux pixels. À chaque site, un descripteur est associé, ce qui permet de décrire l'état du site. Le descripteur peut être le niveau de gris ou une étiquette, mais il peut être aussi une information plus complexe. Par exemple, pour les images couleur, le descripteur peut être une fonction paramétrique, notée *fonction potentiel*, pour laquelle les valeurs pour une clique [3] dépendent des relations entre les niveaux au sein d'une même composante couleur (relations *intra-composante*) et des relations entre les niveaux de deux composantes couleur différentes (relations *inter-composantes*). La définition d'interaction spatiale locale nécessite la structuration des relations spatiales entre ces sites. Un système de cliques est déduit après avoir muni l'ensemble de sites S d'un système de voisinage $\mathcal{V}$ défini par les équations suivantes [27] :

$$\mathcal{V}_s = \{t\} \text{ tels que } \begin{cases} s \notin \mathcal{V}_s \\ t \in \mathcal{V}_s \Rightarrow s \in \mathcal{V}_t \end{cases}$$

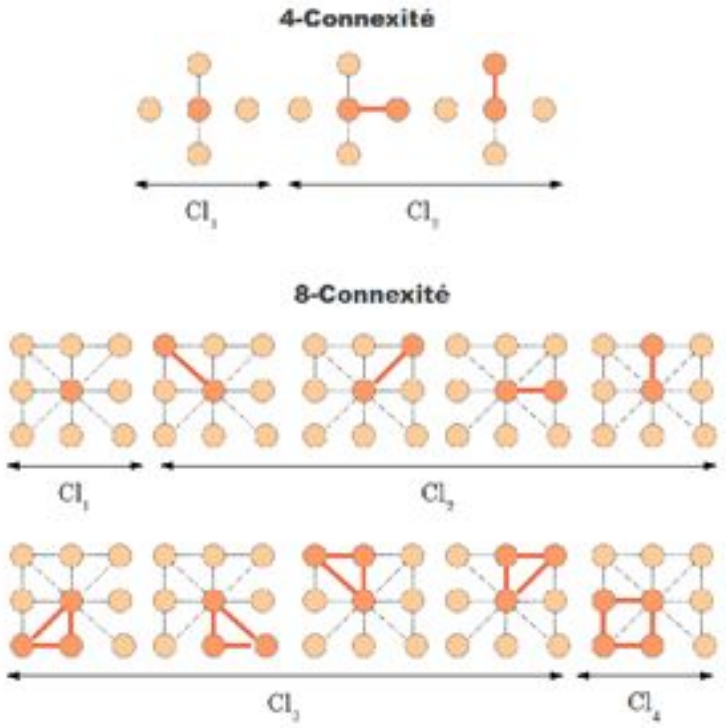


FIGURE 7 – Cliques.

La définition du voisinage permet de définir la notion de clique. En effet, une clique peut être soit un singleton de S, soit un ensemble de sites tous voisins les uns des autres. Dans la littérature, les voisinages les plus répandus dans le cas des champs de Markov sont le 4-voisinage et le 8-voisinage. Pour chaque voisinage, un système de cliques peut être déduit comme illustré dans la figure 7. Les modèles de MARKOV ont été largement utilisés par plusieurs auteurs à savoir KHOTANZAD [25] qui a estimé les paramètres de la *fonction potentiel* en utilisant les moindres carrées afin de caractériser et classer les textures couleur de la base VisTex, HERNANDEZ [24] qui a mis en œuvre l'apport de la couleur lors de l'utilisation des modèles de MARKOV pour calculer

3. Un sous-ensemble de S, relatif au système de voisinage considéré, caractérisé par un ordre qui est son cardinal (le nombre d'éléments le constituant).

des attributs de texture au lieu d'utiliser seulement l'information de luminance, et CHINDARO [28] qui a utilisé des champs de Markov multispectraux en combinant plusieurs espaces couleur.

3.2.3 Les motifs locaux binaires

Les motifs locaux binaires (en anglais Local Binary Pattern : LBP) ont été initialement introduits par HARWOOD et al. [29] mais utilisés par OJALA et al. [30] pour analyser la texture. Cette technique consiste à indexer le motif local dans le voisinage d'un pixel p d'une image I [20, 31]. Plus précisément, les valeurs d'indexation sont obtenues en comparant les niveaux des composantes couleur du pixel courant p aux niveaux des composantes couleur de ses voisins. Le résultat de la comparaison à chaque pixel voisin doit être codé en utilisant un masque connu par le masque de poids. Dans ce qui suit, nous détaillons la technique de calcul des valeurs de LBP en un pixel.

Soit $LBP(p)^{C_kC_{k'}}$ le résultat de l'attribut LBP en un pixel donné p qui traduit la relation spatiale entre le pixel central (en prenant la valeur du niveau de gris de la composante C_k) et ses pixels voisins (en prenant la valeur de leur niveau de gris de la composante couleur $C_{k'}$). Pour obtenir la valeur de $LBP(p)^{C_kC_{k'}}$, la première étape consiste à effectuer un seuillage des niveaux de gris des pixels voisins (de la composante couleur $C_{k'}$) dans un voisinage $\mathcal{V}_p$, en utilisant comme seuil la valeur $C_k(p)$ (cf. équation 1).

$$\forall t \in \mathcal{V}_p \begin{cases} \text{Si} \quad C_{k'}(t) \leq C_k(p) \quad \text{alors } LBP(p)^{C_kC_{k'}} = 0 \\ \text{Si} \quad C_{k'}(t) \geq C_k(p) \quad \text{alors } LBP(p)^{C_kC_{k'}} = 1 \end{cases} \tag{1}$$

où $k, k' \in \{1, 2, 3\}$ pour un espace couleur donné (C_1, C_2, C_3).

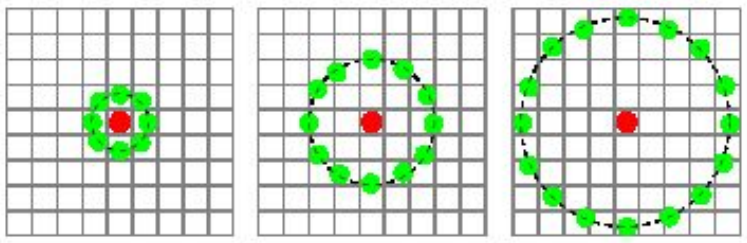


FIGURE 8 – Exemples de voisinages utilisés pour le calcul des LBP : à gauche $N = 8$, au centre $N = 12$ et à droite $N = 16$.

La deuxième étape consiste à appliquer le masque de poids aux résultats de seuillage. En effet, il suffit de multiplier les valeurs des pixels voisins, obtenues après le seuillage, par les valeurs du masque. Le masque utilisé est constitué de valeurs prises entre 2^0 et 2^{N-1}, où N est le nombre des voisins considérés pour le pixel p. Notons que le voisinage du pixel considéré peut être distribué de plusieurs manières. Pour une meilleure illustration des étapes de calcul du LBP (seuillage et masquage), nous présentons un exemple sur la figure **??**. Étant donné une image de texture couleur dans un espace couleur (C_1, C_2, C_3), nous obtenons, pour chaque pixel donné p, neuf coefficients de LBP dont trois caractérisent les relations *intra-composantes* $(LBP(p)^{C_kC_k} \forall k \in \{1, 2, 3\})$ et six caractérisent les relations *inter-composantes* $(LBP(p)^{C_kC_{k'}} \forall k \neq k' \in \{1, 2, 3\})$. Les coefficients de LBP calculés en chaque pixel sont, tout d'abord, rassemblés dans 9 histogrammes. Ces derniers sont, ensuite, concaténés pour obtenir au final un histogramme, noté $LBP(I)$, qui caractérise la texture complète [32].

Tout comme les modèles autorégressifs et les modèles de Markov, les LBP font partie des attributs de texture utilisant la modélisation spatiale. Il existe d'autres attributs caractérisant la texture couleur que nous présentons dans la suite de cette section.

3.3 Attributs spatio-fréquentiels

Les attributs spatio-fréquentiels se divisent en trois catégories : ceux qui se calculent dans le domaine spatial, dans le domaine fréquentiel ou bien dans le domaine spatio-fréquentiel. Nous détaillons dans ce qui suit chaque catégorie :

3.3.1 Dans le domaine spatial

Le calcul d'attributs de texture dans le domaine spatial consiste à caractériser la texture par le nombre de transitions par unité de surface. Par conséquent, contrairement aux textures fines, les textures grossières tendent à avoir une faible densité de transition [19]. Par exemple, les transitions couleur peuvent être détectées par la méthode de Di-Zenzo [33].

3.3.2 Dans le domaine fréquentiel

Le calcul d'attributs de texture dans le domaine fréquentiel consiste à effectuer une transformation du domaine spatial au domaine fréquentiel. Les transformations les plus répandues sont effectuées par une transformée de fourrier et une transformée en cosinus. Celles-ci sont basées sur l'utilisation d'un ensemble de filtres sensibles chacun à une fréquence particulière. Notons que dans le cas des images couleur, le filtrage est effectué sur chaque composante couleur. À partir de l'image résultante (image filtrée), les attributs de texture sont calculés en utilisant des outils statistiques à savoir l'énergie et la variance. Il est à noter que la transformée de fourrier et la transformée en cosinus sont souvent mieux adaptées pour des images ayant des zones homogènes (texture grossière). Par conséquent, l'intégralité de l'information est représentée par les basses fréquences tandis que les hautes fréquences sont réservées à des changements locaux.

3.3.3 Dans le domaine spatio-fréquentiel

Un troisième type d'attributs de texture associant les deux domaines spatial et fréquentiel existe. Parmi les transformations issues de cette catégorie et communément utilisées, nous pouvons citer la transformée de Gabor [34] et la transformée en ondelettes [35].

Maintenant que nous avons rappelé brièvement les définitions des attributs de texture géométriques, les attributs basés sur la modélisation spatiale et les attributs spatio-fréquentiels, nous introduisons dans ce qui suit, une quatrième famille d'attributs de texture qui sont les attributs statistiques.

3.4 Attributs statistiques

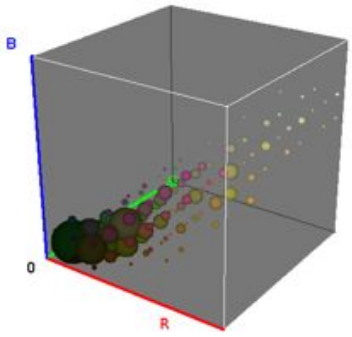
Les attributs statistiques traduisent la relation (en termes de variation de couleur) entre un pixel et ses voisins en tenant compte d'un système de voisinage défini a priori. Les attributs statistiques peuvent être divisés en trois ensembles à savoir les attributs statistiques de premier ordre où seulement le pixel en question est pris en compte (les histogrammes), les attributs statistiques de deuxième ordre (les matrices de co-occurrences et les histogrammes de sommes et de différences) et les attributs statistiques d'ordre supérieure (les matrices de longueurs de plages). Notons que les attributs statistiques sont très utilisés grâce à la pertinence de leurs outils pour caractériser même les textures fines qui ne présentent pas de régularités apparentes. Nous rappelons, dans la suite, les attributs statistiques de texture, de chaque ensemble, les plus communément utilisés.

3.4.1 Attributs statistiques de premier ordre

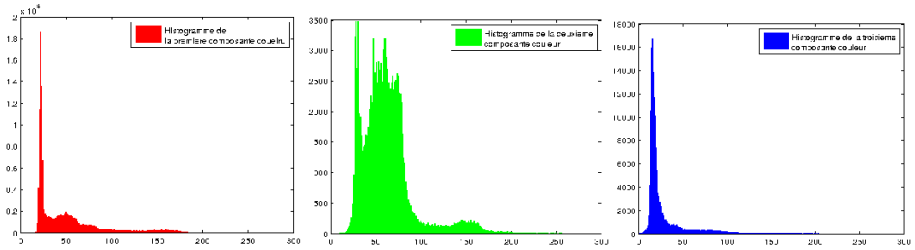
Les attributs statistiques de premier ordre renferment deux ensembles. Le premier ensemble regroupe les statistiques calculées directement sur l'image couleur à savoir la moyenne des composantes couleur, la variance qui mesure la dispersion des composantes couleur autour de leur moyenne, la dissymétrie qui mesure l'asymétrie des composantes couleur à leur moyenne et l'aplatissement qui mesure le degré de convexité ou de concavité des composantes couleur par rapport à leur moyenne. Ces attributs statistiques que nous venons de citer sont peu performants vu qu'ils n'exploitent que l'information couleur de l'image sans tenir compte des interactions spatiales entre les pixels.



(a) Image originale



(b) Histogramme tridimensionnel



(c) Histogrammes mono-dimensionnels calculés pour chaque composante couleur

FIGURE 9 – Illustration des différents histogrammes mono-dimensionnels et tridimensionnel d'une image couleur dans l'espace couleur (R, G, B) (de haut en bas et de gauche à droite : image originale couleur, histogramme tridimensionnel obtenu par le logiciel *Color Inspector 3D* et histogrammes mono-dimensionnels de la première, deuxième et troisième composante couleur).

Le deuxième ensemble d'attributs statistiques de texture de premier ordre sont les histogrammes d'image. Ces derniers sont des outils permettant de décrire la distribution des niveaux de gris des composantes couleur de l'image. Il existe deux formes d'histogramme à savoir l'histogramme mono-dimensionnel calculé pour chaque composante couleur et l'histogramme tridimensionnel représenté généralement sous forme de cellules où chacune d'elles contient l'occurrence de pixels ayant une même couleur. À titre indicatif, si les composantes couleur sont codées sur 8 bits, l'histogramme de l'image contiendra 256^3 cellules. L'exemple de la figure 9 illustre les différents histogrammes mono-dimensionnels (cf. figure 9(c)) et tridimensionnel (cf. figure 9(b)) d'une image couleur dans l'espace couleur (R, G, B).

Dans la pratique et dans la plupart des applications de segmentation d'images, les histogrammes ne sont pas directement utilisés. En effet, des nouveaux paramètres statistiques sont calculés à partir de ces histogrammes à savoir la médiane qui permet de découper l'histogramme en deux parties d'égales populations, le mode qui correspondant au maximum de l'histogramme

et l'intervalle inter-quartile qui caractérise la dispersion du niveau de gris autour de la médiane. Bien que ces attributs sont faciles à déterminer et à exploiter, leur inconvénient majeur réside dans l'exploitation unique de l'information couleur sans tenir compte des interactions spatiales entre pixels de l'image.

3.4.2 Attributs statistiques de second ordre

Les deux attributs statistiques de second ordre les plus répandus sont les matrices de co-occurrences et les histogrammes de somme et de différence. Les matrices de co-occurrences, proposées par HARALICK en 1973 [36], constituent un outil performant pour la description des interactions spatiales entre les pixels. En effet, cet outil permet d'étudier la distribution de l'information couleur, notamment dans le cas des matrices de co-occurrences chromatiques [37], tout en tenant compte de la relation spatiale entre les pixels. Nous nous intéressons dans ce qui suit aux matrices de co-occurrences chromatiques qui ont prouvé leur pertinence et efficacité pour la segmentation d'images, en comparaison avec les matrices monochromatiques.

Les matrices de co-occurrences d'une image couleur I, dans un espace couleur (C_1, C_2, C_3), sont déterminées en calculant l'occurrence d'un couple de niveaux de gris dont la relation spatiale est définie par un vecteur de déplacement $\vec{R}$. Cette relation spatiale peut être aussi définie par un angle θ et une mesure $\|\vec{R}\|$ donnant respectivement la direction et le module du vecteur $\vec{R}$ (la distance spatiale en nombre de pixel entre le pixel à analyser et ses voisins) comme illustré sur la figure 10.

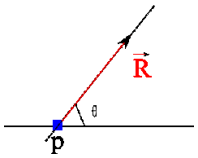


FIGURE 10 – Un vecteur de translation $\vec{R}$ en fonction des paramètres θ et $\|\vec{R}\|$.

Soit $M^{C_k C_{k'}}[I](\vec{R})$, $\forall k, k' \in \{1, 2, 3\}$, la matrice de co-occurrences mesurant l'interaction spatiale entre deux pixels de l'image I. Pour une paire de niveaux de gris (m, n) correspondant à la cellule d'indices (m, n) dans la matrice de co-occurrences, la valeur de cette cellule est égale à l'occurrence des couples de niveaux de gris formés par : le niveau de gris de la composante couleur C_k de I en un pixel p telle que $I^{C_k}(p) = m$ et le niveau de gris d'un pixel obtenu après la translation de p par le vecteur $\vec{R}$ telle que $I^{C_{k'}}(p) = n$ (cf. équation 2).

$$M^{C_k C_{k'}}[I](\vec{R})(m, n) = card\{(I^{C_k}(p), I^{C_{k'}}(p + \vec{R})) = (m, n)\} \quad (2)$$

Dans la littérature, l'angle θ prend souvent ses valeurs dans l'ensemble $\{0^0, 45^0, 90^0, 135^0\}$. La figure 11 illustre des exemples de translation par différents vecteurs $\vec{R}$.

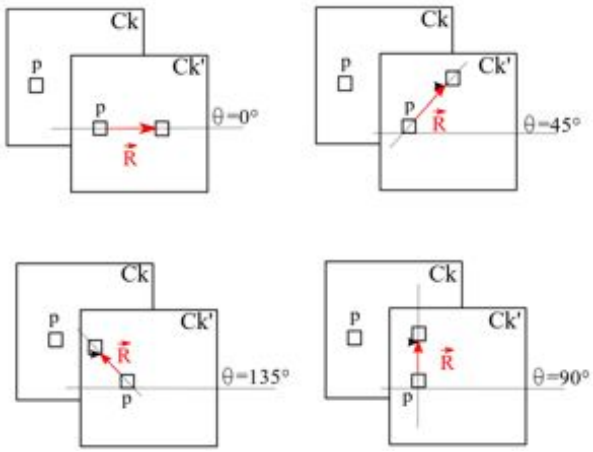


FIGURE 11 – Exemple de différents vecteurs de translation $\vec{R}$.

Nous présentons maintenant, un deuxième attribut statistique de second ordre, moins répandu que les matrices de co-occurrences : les histogrammes de somme et de différence, proposés par UNSER [38]. Tout comme les matrices de co-occurrences, les histogrammes de somme et de différence ont été initialement utilisés pour les images en niveaux de gris, puis étendus dans le cas des images couleur. Cet attribut présente des performances d'analyse du contenu de l'image quasi-similaires en comparaison avec les matrices de co-occurrences, avec l'avantage d'occuper un espace mémoire et un temps de calcul plus réduits.

Le principe de calcul des histogrammes de somme et de différence ressemble à celui des matrices de co-occurrences. Soit I une image couleur codée dans un espace couleur (C_1, C_2, C_3). L'histogramme de somme (respectivement de différence) consiste à calculer l'occurrence de la somme (respectivement de la différence) du niveau de gris dans la composante couleur C_k d'un pixel p et le niveau de gris dans la composante couleur $C_{k'}$ du pixel après une translation $\vec{d}$. Notons $H_S^{C_k C_{k'}}[I](\vec{d})$ l'histogramme de somme et $H_D^{C_k C_{k'}}[I](\vec{d})$ l'histogramme de différence de l'image I entre les composantes couleur C_k et $C_{k'}$, $\forall k, k' \in \{1, 2, 3\}$. Pour une cellule d'indice $m \in [0, 2 \times (N-1)]$ de l'histogramme de somme où N est le niveau de quantification, sa valeur est donnée par l'équation 3. Pour une cellule d'indice $n \in]-(N-1), (N-1)[$ de l'histogramme de différence, sa valeur est donnée par l'équation 4.

$$H_S^{C_k C_{k'}}[I](\vec{d})(m) = card\{I^{C_k}(p) + I^{C_{k'}}(p + \vec{d}) = m\} \tag{3}$$

$$H_D^{C_k C_{k'}}[I](\vec{d})(n) = card\{I^{C_k}(p) - I^{C_{k'}}(p + \vec{d}) = n\} \tag{4}$$

À partir de ces deux outils statistiques de second ordre, nous pouvons obtenir pour une image donnée, une collection de matrices de co-occurrences ou une collection d'histogrammes de somme et de différence pour différents vecteurs de translation. Cependant, dans la pratique, ces collections ne sont pas directement exploitées pour caractériser la texture d'une image. HARALICK [36] propose quatorze nouveaux paramètres calculés à partir des matrices de co-occurrences, tandis que UNSER [38] propose seize nouveaux paramètres calculés à partir des histogrammes de somme et de différence.

Bien que les histogrammes de somme et de différence soient moins coûteux en temps de calcul et en espace mémoire, les matrices de co-occurrences restent l'attribut statistique de second ordre le plus utilisé dans la littérature [39, 40, 41, 42, 43]. Pour cette raison, nous avons fait le choix d'utiliser les matrices de co-occurrences pour caractériser l'information de texture utilisée. Le

lecteur intéressé par les paramètres issus des histogrammes de somme et de différence pourra se référer à [20] pour plus de détails.

3.4.3 Attributs statistiques d'ordre supérieur

L'attribut statistique d'ordre supérieur le plus connu est la matrice de longueurs de plages. Cet attribut est déterminé par le calcul de l'occurrence des niveaux de gris des composantes couleur d'un segment de pixels (au lieu d'un couple de pixels dans le cas des matrices de co-occurrences). Les pixels appartenant au segment considéré doivent être adjacents, consécutifs et avoir le même niveau de gris dans une composante couleur C_k selon une direction donnée [21]. Soit I une image couleur dans l'espace couleur (C_1, C_2, C_3). Une matrice de longueurs de plages $\mathcal{P}^{C_k}[I](\theta, (m, L))$ est calculée en fonction de la direction θ (souvent parmi $\{0^0, 45^0, 90^0$ et $135^0\}$), de la composante couleur C_k ($\forall k \in \{1, 2, 3\}$), du niveau de gris m dans une composante couleur C_k des différents pixels formant le segment considéré et de la longueur L en nombre de pixels de ce segment. Ainsi, une cellule de cette matrice calcule l'occurrence de la suite de pixels adjacents dans une direction θ et ayant le niveau de gris m d'une composante couleur C_k. Tout comme les matrices de co-occurrences et les histogrammes de somme et de différence, les matrices de longueurs de plages ne sont pas directement exploitées. Galloway [44] propose cinq paramètres extraits des matrices de longueurs de plages (détaillés dans les travaux de Porebski [20]).

3.5 Attributs issus de la morphologie mathématique

La caractérisation de la texture dans une image peut être effectuée également grâce à d'autres outils issus de la morphologie mathématique. Leur mode de calcul étant fondé sur la théorie des ensembles, nous avons choisi de regrouper ces outils dans une cinquième catégorie : les attributs de texture issus de la morphologie mathématique. La morphologie mathématique est une nouvelle théorie d'analyse d'images inventée dans les laboratoires de l'école des mines de Paris en 1964. Les attributs issus de la morphologie mathématique s'appuient sur la théorie des ensembles, les treillis, la topologie des ensembles fermés et les probabilités. Ces attributs peuvent ainsi s'appliquer à des structures algébriques variées (des fonctions, des ensembles flous ou des propositions logiques). Ils peuvent avoir, dans certains cas, un caractère aléatoire comme dans le cas d'analyse de textures [45]. La plupart des attributs définis dans la morphologie mathématique sont le résultat de combinaisons d'opérateurs de base de cette théorie à savoir l'érosion, la dilatation, l'ouverture et la fermeture. Parmi les opérateurs de la morphologie mathématique permettant l'étude de la texture dans une image, nous pouvons citer le filtre alterné séquentiel (ASF) et le gradient morphologique.

4 Les descripteurs de couleur et de texture

Maintenant que nous avons rappelé la définition de la couleur et de la texture et présenté les différentes classes d'attributs de texture de la littérature, nous allons détailler les définitions des principaux descripteurs couleur et texture issus de la littérature. Pour ce qui est des gradients couleur, nous avons choisi le gradient de Di-Zenzo et le gradient morphologique couleur (avec un ordre lexicographique) qui restent incontournables. En ce qui concerne les gradients texture, nous avons eu recours aux attributs les plus fréquemment utilisés dans la littérature. Robustes et efficaces, ces attributs ne se contentent pas seulement de réaliser une modélisation spatiale, mais permettent aussi de donner une représentation fréquentielle. C'est la raison pour laquelle nous nous sommes intéressés au filtre de Gabor. Les matrices de co-occurrences constituent également un outil souvent utilisé, ayant prouvé leur efficacité dans la description de l'information texture

présente dans l'image. Nous retenons également le filtre alterné séquentiel (Alternate Sequentiel Filter : ASF) qui est un attribut issu de la morphologie mathématique. Nous utilisons ce filtre composite et complexe car il permet de décrire la texture de façon beaucoup plus fine que les autres attributs (ouverture et fermeture par exemple) de la morphologie mathématique.

4.1 Les descripteurs de couleur

Dans cette section, nous allons définir les descripteurs couleur que nous avons sélectionnés, à savoir le gradient de Di-Zenzo et le gradient morphologique couleur.

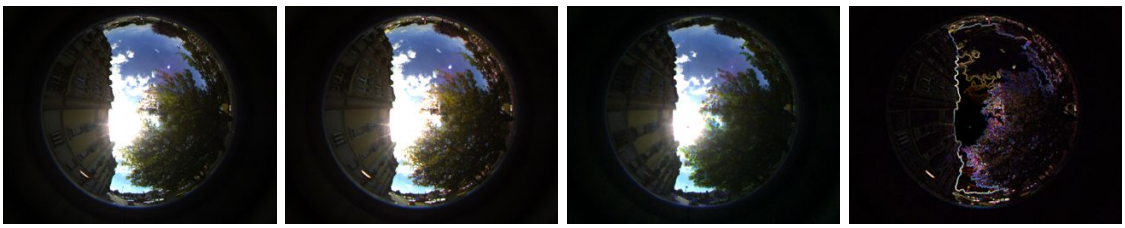
4.1.1 Gradient morphologique couleur

Le gradient morphologique est un opérateur issu de la théorie de la morphologie mathématique [46]. Ce gradient a été défini par Beucher dans le cas discret [47]. D'autres définitions du gradient morphologique (directionnel, régularisé, thik/thin, etc.) ont été présentées dans les travaux de Rivest et al. [48]. L'expression de calcul la plus courante du gradient morphologique est celle proposée par Beucher. Elle est donnée dans le cas discret par l'équation 5. Il s'agit du résidu du dilaté d'une image I et de son érodé.

$$\mathcal{G}_{\mathcal{M}}(I) = \delta_{ES}(I) - \varepsilon_{ES}(I) \tag{5}$$

où ES est un élément structurant donné. δ_{ES} et ε_{ES} sont respectivement les opérations de dilatation et d'érosion.

Dans notre cas, le gradient morphologique a été calculé sur l'image couleur initiale en utilisant un élément structurant ES carré (en 8-connexité), de taille 3×3 pixels et un ordre lexicographique. La figure 12 présente un exemple de gradient morphologique couleur calculé sur une image couleur de la base du projet CAPLOC.



(a) Image originale (b) Image dilatée (c) Image érodée (d) Gradient couleur

Figure 12 – Illustration du gradient morphologique couleur calculé sur une image de la base du projet CAPLOC.

4.1.2 Gradient de Di-Zenzo

Le gradient de Di-Zenzo est un gradient multi-composantes s'appliquant sur des images couleur [33, 49]. Son mode de calcul est composé de deux étapes. La première consiste à calculer séparément sur chaque composante couleur, la dérivée première de l'image. Après cette étape, nous disposons de deux images : une image d'amplitude et une image de direction du gradient (pour chaque composante couleur de l'image). L'amplitude du gradient est obtenue par le maximum de la dérivée en un pixel de l'image de coordonnées x et y tandis que la direction est obtenue par l'équation $artang(\frac{dy}{dx})$. La valeur de l'amplitude du gradient en un point reflète la variation de niveau de gris observée dans l'image en ce point. Cette variation est considérée comme forte

quand la valeur de l'amplitude est élevée. La direction du gradient est orthogonale à la frontière qui passe par le point considéré. La deuxième étape du calcul du gradient de Di-Zenzo consiste à fusionner ces informations selon un critère choisi a priori.

Dans la pratique, l'algorithme de Di-Zenzo est basé sur la recherche des valeurs propres de la matrice M donnée par les équations suivantes [50] :

$$M = \begin{pmatrix} a & b \\ b & c \end{pmatrix}$$

tels que

$$a = \mathcal{G}_{x_1}^2 + \mathcal{G}_{x_2}^2 + \mathcal{G}_{x_3}^2 \tag{6}$$

$$b = \mathcal{G}_{x_1} \times \mathcal{G}_{y_1} + \mathcal{G}_{x_2} \times \mathcal{G}_{y_2} + \mathcal{G}_{x_3} \times \mathcal{G}_{y_3} \tag{7}$$

$$c = \mathcal{G}_{y_1}^2 + \mathcal{G}_{y_2}^2 + \mathcal{G}_{y_3}^2 \tag{8}$$

avec $\mathcal{G}_{x_1}$, $\mathcal{G}_{x_2}$ et $\mathcal{G}_{x_3}$ (respectivement $\mathcal{G}_{y_1}$, $\mathcal{G}_{y_2}$ et $\mathcal{G}_{y_3}$) représentent le gradient selon le déplacement dx (respectivement dy) sur chacune des trois composantes. Les valeurs propres de la matrice M sont utilisées pour calculer le module du gradient (cf. équation 9) et les vecteurs propres associés traduisent les directions du gradient $\theta_{\mathcal{G}}$ (cf. équation 10).

$$\|\mathcal{G}\| = \sqrt{(\lambda_1^{\mathcal{G}} + \lambda_2^{\mathcal{G}})} \tag{9}$$

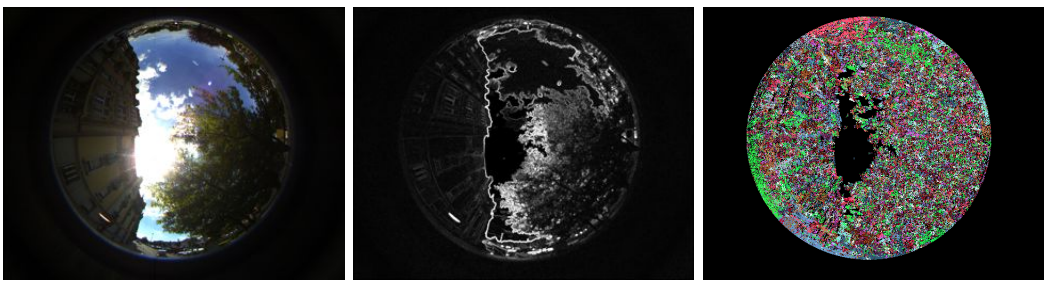
avec

$$\lambda_1^{\mathcal{G}} = \frac{1}{2} \times (a + c + \sqrt{(a-c)^2 - 4 \times b^2})$$

$$\lambda_2^{\mathcal{G}} = \frac{1}{2} \times (a + c - \sqrt{(a-c)^2 - 4 \times b^2})$$

$$\theta_{\mathcal{G}} = \frac{1}{2} \times \arctan(\frac{2 \times b}{(a-c)}) \tag{10}$$

Dans notre cas, le gradient de Di-Zenzo a été calculé sur l'image couleur initiale en utilisant des codes de Freeman dans l'intervalle $[0, \ldots, 7]$. La figure 13 présente un exemple de calcul du gradient de Di-Zenzo sur une image couleur issue du projet CAPLOC.



(a) Image originale (b) Module du gradient (c) Orientation du gradient

Figure 13 – Illustration du gradient de Di-Zenzo sur une image couleur de la base CAPLOC.

4.2 Les descripteurs de texture

Les gradients de texture que nous décrivons dans cette section sont basés sur les attributs de texture faisant partie des attributs de texture les plus communément utilisés dans les applications de la segmentation d'images couleur/texture. Ces gradients sont basés sur les filtres de Gabor, des paramètres d'HARALICK extraits des matrices de co-occurrences et le filtre alterné séquentiel.

4.2.1 Gradient basé sur les filtres de Gabor

La transformation de Gabor est souvent définie comme une transformée de Fourrier avec fenêtrage [34]. Par conséquent, cette transformation permet d'examiner la régularité de la texture dans le domaine fréquentiel tout en tenant compte de sa localisation spatiale. La transformée de Gabor est obtenue par l'équation 12, en appliquant la transformée de Fourrier décrite dans les travaux de [34] avec un fenêtrage gaussien.

Soient I une image initiale de taille $L_x \times L_y$ pixels, $w(x_\alpha - x_0, y_\alpha - y_0)$ la fenêtre gaussienne entre la position spatiale (x_0, y_0) et la position spatiale (x_α, y_α) après une rotation spatiale (d'un angle α) du vecteur $(x, y)^t$ (cf. équation 11).

$$\begin{pmatrix} x_\alpha \\ y_\alpha \end{pmatrix} = A_\alpha \begin{pmatrix} x \\ y \end{pmatrix} \quad \text{avec} \quad A_\alpha = \begin{pmatrix} \cos(\alpha) & \sin(\alpha) \\ -\sin(\alpha) & \cos(\alpha) \end{pmatrix} \tag{11}$$

$$\mathcal{T}_{G_{u_0,v_0}}(x_0, y_0) = \frac{1}{L_x L_y} \sum_{x=0}^{L_x-1} \sum_{y=0}^{L_y-1} I(x,y).w(x_\alpha - x_0, y_\alpha - y_0).\exp\left(-2\pi i \left(\frac{xu_0}{L_x} + \frac{yu_0}{L_y}\right)\right) \tag{12}$$

où x et y sont les coordonnées spatiales telles que $x \in \{0, \ldots, L_x - 1\}$ et $y \in \{0, \ldots, L_y - 1\}$. u_0 et v_0 sont les fréquences spatiales correspondant aux coordonnées spatiales x_0 et y_0.

Dans le cas général, u et v sont les fréquences spatiales des coordonnées spatiales x et y telles que $u \in \{-\frac{L_x}{2}, \ldots, \frac{L_x}{2}\}$ et $v \in \{-\frac{L_y}{2}, \ldots, \frac{L_y}{2}\}$. Elles représentent respectivement les décompositions verticale et horizontale.

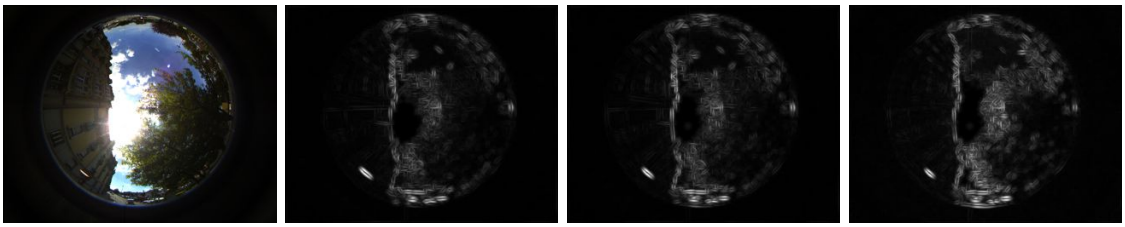
En utilisant l'aspect symétrique de la fenêtre de Gauss $w(x, y)$ (cf. équation 13), la transformée de Gabor peut être décrite par l'équation 14. Le masque de convolution $\mathcal{F}_{G_{u_0,v_0,\alpha(x_0,y_0)}}$ représente le filtre de Gabor.

$$w(x,y) = \frac{1}{2\pi\sigma_x\sigma_y}.\exp\left(-\frac{1}{2}\frac{x^2}{\sigma_x^2}\right).\exp\left(-\frac{1}{2}\frac{y^2}{\sigma_y^2}\right) \tag{13}$$

$$\mathcal{T}_{G_{u_0,v_0}}(x_0, y_0) = I(x_0, y_0) \times \mathcal{F}_{G_{u_0,v_0,\alpha(x_0,y_0)}} \tag{14}$$

La caractérisation de texture d'une image couleur par les filtres de Gabor est obtenue en appliquant ces filtres (sensible chacun à une fréquence donnée) sur chacune des fenêtres balayant les composantes couleur [51, 34]. α prend les valeurs $\{0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3}{4}\pi\}$ et la taille de la fenêtre de Gauss prend les valeurs 21×21 pixels et 35×35 pixels. Pour un filtre donné, l'attribut de texture peut être obtenu en calculant l'énergie, l'entropie ou la variance des coefficients de Gabor générés pour chaque fenêtre. Les expressions de ces quantités sont données dans les travaux de PALM et al. [34]. Dans ce rapport, nous avons choisi, suite à plusieurs tests, l'énergie comme attribut de texture. Le descripteur de texture basé sur les filtres de Gabor est alors obtenu en calculant le gradient morphologique sur chaque couche de texture. À l'issue de cette étape, nous retenons, pour chaque pixel, la valeur maximale parmi toutes les couches de texture. La figure 14 présente

une illustration des résultats (sur les trois composantes couleur) du gradient basé sur les filtres de Gabor.



(a) Image originale (b) $Gradient_R$ Gabor (c) $Gradient_G$ Gabor (d) $Gradient_B$ Gabor

FIGURE 14 – Illustration du descripteur de texture basé sur les filtres de Gabor sur une image couleur issue du projet CAPLOC dans l'espace couleur (R, G, B).

4.2.2 Gradients basés sur les paramètres d'HARALICK

Les matrices de co-occurrences proposées par HARALICK en 1973 [36], constituent un outil performant pour la description des interactions spatiales entre les pixels. En effet, cet outil permet d'étudier la distribution de l'information couleur dans le cas des matrices de co-occurrences chromatiques tout en tenant compte de la relation spatiale entre les pixels de l'image à analyser. Les matrices de co-occurrences d'une image I sont calculées à l'aide de l'équation 2. Dans la majorité des cas, les matrices de co-occurrences ne sont pas directement exploitées pour le calcul des attributs de texture. Ceci s'explique par la difficulté d'interpréter visuellement des surfaces de co-occurrences. Par ailleurs, les matrices de co-occurrences mettent en évidence d'énorme quantité d'informations. C'est pourquoi les utilisateurs extraient de ces matrices des attributs moins coûteux en mémoire. Ainsi, 14 attributs connus sous le nom de paramètres d'HARALICK [36] ont été introduits afin de conserver la pertinence des matrices de co-occurrences.

Dans le cas pratique, les 14 paramètres d'HARALICK ne sont pas tous utilisés. Seulement 6 paramètres (l'énergie, le contraste, la corrélation, la variance, l'homogénéité et l'entropie), davantage interprétables, sont souvent exploités pour le calcul d'attributs de texture. Il reste difficile de donner, même de manière, intuitive une signification précise pour les autres paramètres [36, 20]. Nous présentons ci-après les 14 paramètres d'HARALICK en illustrant sur une image de la base de CAPLOC, ceux qui restent les plus performants et donc les plus utilisés dans la littérature.

Soient I l'image initiale, m et n deux niveaux de gris de I, C_k et C'_k deux composantes couleur de I et $\vec{d}$ le vecteur de déplacement.

a) **Second moment angulaire (Énergie/Uniformité)**

Le second moment angulaire (Second Angular Moment : SAM), appelé aussi énergie, est une mesure qui examine l'homogénéité globale de l'image. Cet attribut atteint de fortes valeurs lorsque la distribution des niveaux de gris est constante ou de forme périodique. Dans le cas particulier où les deux niveaux de gris m et n sont égaux, il s'agit d'une mesure d'uniformité. Dans ce cas, le paramètre SAM est d'autant plus faible que les valeurs non-nulles de la diagonale de la matrice de co-occurrences sont équi-réparties. Plus l'indice est élevé, plus le nombre de niveaux de gris renvoyant à eux-mêmes par la translation $\vec{d}$ est faible [52]. L'expression du paramètre SAM s'écrit par l'équation donnée ci-après :

$$H_1^{C_k,C_{k'},\vec{d}} = \sum_{m=0}^{N-1}\sum_{n=0}^{N-1}\left(M^{C_kC_{k'}}[I](\vec{d})(m,n)\right)^2 \quad (15)$$

b) **Contraste**

Le contraste mesure les variations locales des niveaux de gris (couleurs) présents dans l'image à analyser. Cette mesure est importante lors du passage (réciproque) d'un niveau faible à un niveau élevé. Une valeur faible de cet attribut caractérise une texture plutôt uniforme [53]. Le contraste s'écrit par l'équation donnée ci-après :

$$H_2^{C_k,C_{k'},\vec{d}} = \sum_{\substack{l=0 \\ l=|m-n|}}^{N-1} l^2 \left(\sum_{m=0}^{N-1} \sum_{n=0}^{N-1} M^{C_k C_{k'}}[I](\vec{d})(m,n) \right) \tag{16}$$

c) **Entropie**

La mesure d'entropie comme son nom l'indique examine le désordre que peut présenter l'image [52]. Plus précisément, l'entropie caractérise la dispersion spatiale et spectrale des valeurs non-nulles de la matrice de co-occurrences. Cette mesure prend des valeurs faibles lorsque les couples de pixels sont bien répartis dans l'image et des valeurs importantes lorsque ces couples sont en désordre. L'entropie s'exprime par l'équation donnée ci-après :

$$H_3^{C_k,C_{k'},\vec{d}} = -\sum_{m=0}^{N-1} \sum_{n=0}^{N-1} M^{C_k C_{k'}}[I](\vec{d})(m,n) \log \left(M^{C_k C_{k'}}[I](\vec{d})(m,n) \right) \tag{17}$$

d) **Corrélation**

La corrélation permet de mesurer les dépendances en ligne et en colonne entre les niveaux de composantes couleur dans l'image [36, 54]. L'expression de la corrélation est décrite par l'équation donnée ci-après :

$$H_4^{C_k,C_{k'},\vec{d}} = \frac{(m-\mu_m^M)(n-\mu_n^M) \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} M^{C_k C_{k'}}[I](\vec{d})(m,n)}{\sigma_m^M \sigma_n^M} \tag{18}$$

où μ_m^M, μ_n^M, σ_m^M et σ_n^M constituent respectivement les centres de gravité et les écarts type de $M_m^{C_k C_{k'}}[I](\vec{d})(m)$ et de $M_n^{C_k C_{k'}}[I](\vec{d})(n)$, avec :

$$M_m^{C_k C_{k'}}[I](\vec{d})(m) = \sum_{n=0}^{N-1} M^{C_k C_{k'}}[I](\vec{d})(m,n) \tag{19}$$

et

$$M_n^{C_k C_{k'}}[I](\vec{d})(n) = \sum_{m=0}^{N-1} M^{C_k C_{k'}}[I](\vec{d})(m,n) \tag{20}$$

e) **Variance (Inertie)**

Le cinquième paramètre d'Haralick est la variance, appelé aussi inertie. Intuitivement, cette mesure permet d'examiner la distribution de chaque composante couleur autour de sa moyenne [55]. Autrement dit, la variance mesure l'hétérogénéité de la texture et augmente lorsque les niveaux de gris diffèrent de leur moyenne. Elle est décrite par l'équation donnée ci-après :

$$H_5^{C_k,C_{k'},\vec{d}} = \frac{1}{N^2} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} (m-\mu^M)^2 M^{C_k C_{k'}}[I](\vec{d})(m,n) \tag{21}$$

où μ^M est le centre de gravité des coefficients de la matrice $M^{C_k C_{k'}}[I]$.

f) **Moment des différences inverses (homogénéité)**

Le moment différentiel inverse correspond à une mesure d'homogénéité locale au sein de la texture [54]. En effet, l'homogénéité traduit la forte ou la faible occurrence d'un couple de pixels donné, séparé par l'isométrie $\vec{d}$. Cette mesure est élevée dans les zones de l'image où le niveaux de gris est uniforme ou lorsque l'image présente des motifs de périodes $\|\vec{d}\|$ dans la direction θ. L'expression de ce paramètre est décrite par l'équation donnée ci-après :

$$H_6^{C_k,C_{k'},\vec{d}} = \sum_{m=0}^{N-1}\sum_{n=0}^{N-1} \frac{1}{1+(m-n)^2} M^{C_kC_{k'}}[I](\vec{d})(m,n) \tag{22}$$

Afin d'être exhaustif, nous citons l'expression des autres paramètres d'HARALICK, peu utilisés du fait de leur interprétation difficile.

g) **Moyenne des sommes**

$$H_7^{C_k,C_{k'},\vec{d}} = \sum_{l=0}^{2(N-1)} l.M_{m+n}^{C_kC_{k'}}[I](\vec{d})(l) \tag{23}$$

où

$$M_{m+n}^{C_kC_{k'}}[I](\vec{d})(l) = \sum_{\substack{m'=0\\ m'+n'=l}}^{N-1}\sum_{n'=0}^{N-1} M^{C_kC_{k'}}[I](\vec{d})(m',n')$$

h) **Variance des sommes**

$$H_8^{C_k,C_{k'},\vec{d}} = \sum_{l=0}^{2(N-1)} (l - H_7^{C_k,C_{k'},\vec{d}}).M_{m+n}^{C_kC_{k'}}[I](\vec{d})(l) \tag{24}$$

i) **Entropie des sommes**

$$H_9^{C_k,C_{k'},\vec{d}} = \sum_{l=0}^{2(N-1)} M_{m+n}^{C_kC_{k'}}[I](\vec{d})(l) \log\left(M_{m+n}^{C_kC_{k'}}[I](\vec{d})(l)\right) \tag{25}$$

j) **Variance des différences**

$$H_{10}^{C_k,C_{k'},\vec{d}} = \sum_{l=0}^{N-1} (1-\mu_{m-n}^{M}) M_{m-n}^{C_kC_{k'}}[I](\vec{d})(l) \tag{26}$$

où

$$M_{m-n}^{C_kC_{k'}}[I](\vec{d})(l) = \sum_{\substack{m'=0\\ |m'-n'|=l}}^{N-1} \sum_{n'=0}^{N-1} M^{C_kC_{k'}}[I](\vec{d})(m',n') \quad l=0,1,\ldots,N-1$$

et

$$\mu_{m-n}^{M} = \sum_{l=0}^{N-1} l.M_{m-n}^{C_kC_{k'}}[I](\vec{d})(l)$$

k) **Entropie des différences**

$$H_{11}^{C_k,C_{k'},\vec{d}} = \sum_{l=0}^{N-1} M_{m-n}^{C_kC_{k'}}[I](\vec{d})(l) \log\left(M_{m-n}^{C_kC_{k'}}[I](\vec{d})(l)\right) \tag{27}$$

l) **Informations sur la corrélation**

$$H_{12}^{C_k,C_{k'},\vec{d}} = \frac{H_3^{C_k,C_{k'},\vec{d}} - H_{mn}^1}{max(H_m^1, H_n^1)} \tag{28}$$

où

$$H_{mn}^1 = -\sum_{m'=0}^{N-1}\sum_{n'=0}^{N-1} M^{C_kC_{k'}}[I](\vec{d})(m',n') \log\left(M_m^{C_kC_{k'}}[I](\vec{d})(m') \times M_n^{C_kC_{k'}}[I](\vec{d})(n')\right)$$

$$H_m^1 = -\sum_{m'=0}^{N-1}\sum_{n'=0}^{N-1} M_m^{C_kC_{k'}}[I](\vec{d})(m') \log\left(M_m^{C_kC_{k'}}[I](\vec{d})(m')\right)$$

$$H_n^1 = -\sum_{m'=0}^{N-1}\sum_{n'=0}^{N-1} M_n^{C_kC_{k'}}[I](\vec{d})(n') \log\left(M_n^{C_kC_{k'}}[I](\vec{d})(n')\right)$$

m) **Informations sur la corrélation**

$$H_{13}^{C_k,C_{k'},\vec{d}} = \left(1 - \exp[-2.(H_{mn}^2 - H_3^{C_k,C_{k'},\vec{d}})]\right)^{\frac{1}{2}} \tag{29}$$

où

$$H_{mn}^2 = -\sum_{m'=0}^{N-1}\sum_{n'=0}^{N-1} M_m^{C_kC_{k'}}[I](\vec{d})(m') \times M_n^{C_kC_{k'}}[I](\vec{d})(n') \log\left(M_m^{C_kC_{k'}}[I](\vec{d})(m') \times M_n^{C_kC_{k'}}[I](\vec{d})(n')\right)$$

n) **Coefficient de corrélation maximale**

$$H_{14}^{C_k,C_{k'},\vec{d}} = (\text{Plus grande valeur propre de } \mathcal{Q})^{\frac{1}{2}} \tag{30}$$

où

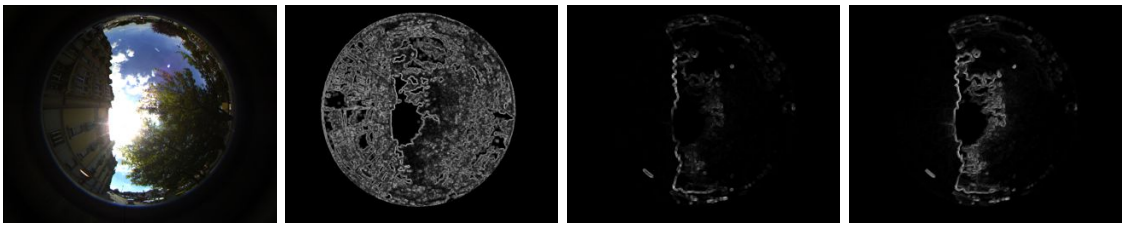
$$\mathcal{Q}(\vec{d})(m,n) = \sum_{l=0}^{N-1} \frac{M^{C_kC_{k'}}[I](\vec{d})(m,l) \times M^{C_kC_{k'}}[I](\vec{d})(n,l)}{M_m^{C_kC_{k'}}[I](\vec{d})(m) \times M_n^{C_kC_{k'}}[I](\vec{d})(n)}$$

Dans ce rapport, les gradients basés sur les paramètres d'Haralick ont été calculés en appliquant un gradient morphologique sur les images d'attributs de texture H_h (énergie, contraste et variance), comme décrit dans l'équation 31. Les matrices de co-occurrences ont été générées en balayant l'image avec une fenêtre (de taille 4×4 pixels) et avec un niveau de quantification $N = 16$. Plusieurs combinaisons de translations ont été testées ($\|\vec{d}\| \in \{3, 5\}$ pixels) dans différentes directions $\theta \in \{0, \ldots, 150\}$.

$$\mathcal{G}_{H_h} = \delta_{ES}(H_h) - \varepsilon_{ES}(H_h) \quad , \forall h \in \{1, \ldots, 14\} \tag{31}$$

où δ_{ES} et ε_{ES} sont respectivement les opérations de dilatation et d'érosion en utilisant l'élément structurant ES.

La figure 15 illustre trois gradients calculés à partir des trois paramètres d'Haralick que nous jugeons opportuns (énergie, contraste et variance), sur une image de la base du projet CAPLOC. Pour une meilleure visualisation, nous n'avons illustré que les gradients dont le résultat est exploitable.



(a) Image originale (b) Gradient Énergie (c) Gradient Contraste (d) Gradient Variance

Figure 15 – Gradients calculés à partir d'attributs d'Haralick sur une image couleur de la base du projet CAPLOC.

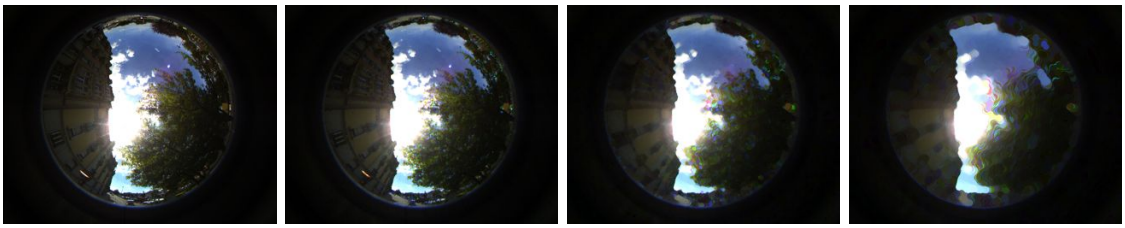
Au vu de la pertinence et l'apport des attributs de texture calculés à partir des matrices de co-occurrences, d'autres paramètres ont été introduits, à savoir le moment diagonal, la différence inverse, la dissimilarité, le troisième moment angulaire, le cluster shade, etc. Ces paramètres sont décrits dans les travaux [56, 20, 57]. Dans ce rapport, nous nous contentons des paramètres d'Haralick.

4.2.3 Gradient basé sur le filtre alterné séquentiel

Le filtre alterné séquentiel (Alternate Sequentiel Filter : ASF) est un outil issu de la morphologie mathématique. Il consiste à appliquer une succession de fermetures φ et d'ouvertures γ avec des éléments structurants $\{ES_q\}_{q \in 1,\ldots,o}$ de taille croissante jusqu'à un ordre o donné [58]. Ce filtre permet de simplifier l'image en conservant ses détails (sans lissage), ce qui permet de caractériser la texture en préservant les contrastes. Posons I l'image initiale, le filtre alterné séquentiel d'ordre p, que nous notons $\mathcal{ASF}_o(I)$, est défini par l'équation suivante :

$$\mathcal{ASF}_o(I) = \varphi_{ES_o}(\gamma_{ES_o}(\ldots \varphi_{ES_2}(\gamma_{ES_2}(\varphi_{ES_1}(\gamma_{ES_1}(I)))))) \tag{32}$$

La figure 16 illustre la simplification d'une image issue de la base du projet CAPLOC par un filtre ASF avec différents ordres.



(a) Image originale (b) $o = 2$ (c) $o = 4$ (d) $o = 8$

Figure 16 – Illustration d'une simplification d'une image issue de la base du projet CAPLOC par un filtre ASF avec différents ordres.

Pour le calcul d'un gradient de texture basé sur un filtre ASF, nous avons suivi l'approche proposée par Angulo [59] et qui consiste à redéfinir ce qu'est une couche de texture en utilisant une analyse multi-échelle locale, puis à introduire une nouvelle définition d'un gradient de texture basé sur la morphologie mathématique. Ce gradient morphologique de texture a été calculé à l'aide d'opérations issues de la morphologie mathématique dont nous détaillons brièvement le principe composé de quatre étapes : le filtrage de l'image initiale, la génération d'une couche de texture, l'analyse granulométrique et finalement le calcul du gradient de texture morphologique.

a) **Filtrage de l'image initiale**

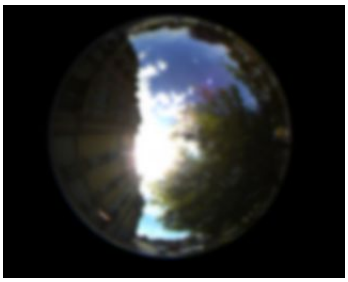
La première étape de calcul du gradient morphologique de texture consiste à appliquer un filtrage sur l'image initiale afin de supprimer les détails les moins significatifs tout en conservant la forme et les contours des objets. Le filtre choisi est un filtre connexe de type nivellement. Ce filtre, issu de la morphologie mathématique, permet de conserver les zones homogènes et de supprimer les pics du bruit dans l'image. Soient I l'image initiale, $I_{\mathcal{N}}$ l'image filtrée par le filtre de nivellement et $I_{Marqueur}$ une image de marqueur donnée. La valeur en un pixel p de l'image filtrée peut être obtenue par itération jusqu'à idempotence (stabilité du résultat) des opérations données par l'équation 33.

$$\mathcal{N}(p) = \begin{cases} \min\left(I(p), \delta_{ES}(I_{Marqueur})(p)\right) & \text{si} \quad I_{Marqueur}(p) < I(p) \\ \max\left(I(p), \varepsilon_{ES}(I_{Marqueur})(p)\right) & \text{si} \quad I_{Marqueur}(p) > I(p) \end{cases} \tag{33}$$

où δ_{ES} est la dilatation morphologique et ε_{ES} est l'érosion morphologique avec un élément structurant ES. L'image $I_{Marqueur}$ est souvent le résultat de la simplification de l'image initiale I. Dans notre cas, $I_{Marqueur}$ est la simplification de I par un filtre $\mathcal{ASF}$.



(a) Image originale



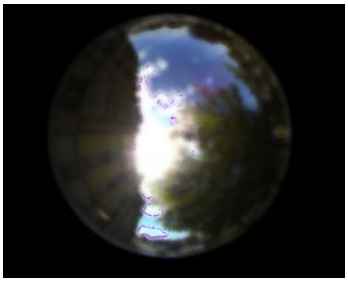
(b) Filtre gaussien ($\sigma = 4$)



(c) Filtre moyenneur ($W = 4$)



(d) Filtre ASF ($p = 4$)



(e) Nivellement gaussien



(f) Nivellement moyenneur



(g) Nivellement ASF

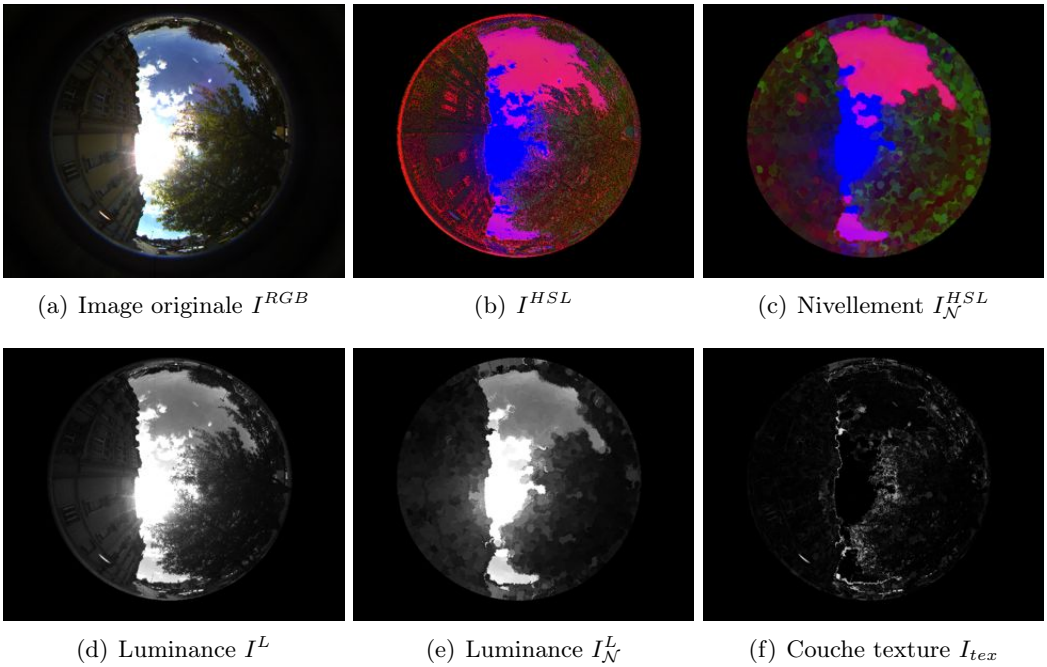
FIGURE 17 – Illustration d'un filtrage de nivellement sur une image de la base du projet CAPLOC bruitée avec un bruit gaussien (de moyenne 5 et d'écart type 10) et avec trois images marqueurs (obtenues avec un filtre gaussien, un filtre moyenneur et un filtre ASF).

La figure 17 illustre les différents résultats obtenus en fonction de l'image de marqueur utilisée. Nous pouvons constater que le filtre de nivellement couplé au filtrage par ASF offre la meilleure simplification de l'image, tout en conservant les contours des objets.

b) **Définition de la couche de texture**

ANGULO [59], tout comme SOFOU et al. [60], ont défini la texture par le résidu de nivellement de la composante de luminance. En effet, ANGULO affirme que le phénomène de texture dans l'image n'est qu'une variation d'intensité lumineuse. Soit I^L la composante de luminance de l'image initiale I^{HSL} présentée dans l'espace couleur (H, S, L) et $I^L_{\mathcal{N}}$ la composante de luminance de l'image résultante du filtrage par nivellement. La formulation mathématique de l'obtention de la couche de texture telle que définie par ANGULO est donnée par l'équation 34 et illustrée dans la figure 18.

$$I_{tex} = I^L - I^L_{\mathcal{N}} \tag{34}$$



(a) Image originale I^{RGB} (b) I^{HSL} (c) Nivellement $I^{HSL}_{\mathcal{N}}$

(d) Luminance I^{L} (e) Luminance $I^{L}_{\mathcal{N}}$ (f) Couche texture I_{tex}

FIGURE 18 – Calcul de la couche de texture.

c) **Analyse granulométrique**

La granulométrie morphologique est une technique permettant l'étude des tailles des objets dans l'image en se basant sur le principe de tamisage [45]. Cette technique permet la sélection successive d'objets de tailles données croissantes, ce qui donne une idée sur la distribution de leur taille dans l'image. Formellement, l'analyse granulométrique consiste à appliquer un ensemble de tamis de tailles croissantes. Ces tamis sont des transformations $(\Psi_\eta)_{\eta\geq 0}$ vérifiant les propriétés ci-dessous et ne dépendant que d'un seul paramètre $\eta \geq 0$. Ces transformations sont souvent des ouvertures morphologiques $(\gamma_\eta)_{\eta\geq 0}$ de tailles croissantes.

- $\Psi_0(I) = I$
- $\forall \eta \geq 0, \forall I_1, I_2 \subset I$, Ψ_η est croissante : $I_1 \subset I_2 \Rightarrow \Psi_\eta(I_1) \subset \Psi_\eta(I_2)$
- $\forall \eta \geq 0, \forall I_1 \subset I$, Ψ_η est anti-extensive : $\Psi_\eta(I_1) \subset I$
- Loi d'absorption : $\forall \eta \geq 0, \forall \varkappa \geq 0$, étant deux tailles différentes, $\Psi_\eta \Psi_\varkappa = \Psi_\varkappa \Psi_\eta = \Psi_{\max(\eta,\varkappa)}$

En considérant les transformations granulométriques comme étant une série d'ouvertures $(\gamma_\eta)_{\eta\geq 0}$ de tailles croissantes, l'analyse granulométrique consiste donc à évaluer chaque ouverture de taille η avec une mesure $\mathcal{S}$ définie comme la somme des valeurs des pixels de l'ouverture (cf. équation 35). Une série de fermetures de tailles croissantes $(\varphi_\eta)_{\eta\geq 0}$ constitue donc une anti-granulométrie (granulométrie pour les objets sombres dans l'image I).

$$\mathcal{S}(\gamma_\eta(I)) = \sum_{p\in\gamma_\eta(I)} \gamma_\eta(I)(p) \tag{35}$$

L'ouverture de taille η, n'étant composée que par les structures claires de I de taille supérieure à η, le calcul de la mesure $\mathcal{S}$ revient donc à calculer le nombre de ces structures claires de tailles supérieures à η. Par analogie, lorsque la mesure S est calculée pour la famille de fermetures $(\varphi_\eta)_{\eta\geq 0}$, ceci revient à calculer dans l'image I, le nombre de structures sombres de

tailles supérieures à η. Par conséquent, nous appelons un spectre de forme $\mathcal{PS}(I, \eta)$ la fonction décrite par l'équation 36 et qui consiste à calculer le nombre de structures lumineuses de taille η dans une image I. Cette fonction indique la répartition de structures associées à chaque taille η. Par conséquent, le spectre de forme peut servir comme un descripteur de texture dans l'image. De manière générale, le spectre de forme $\mathcal{PS}$ est souvent calculé sur la totalité de l'image tandis que, dans une même image, il est possible d'avoir plusieurs textures distinctes. Par conséquent, il est plus intéressant d'appliquer $\mathcal{PS}(I_W, \eta)$ sur des fenêtres de taille W centrées sur les pixels de l'image. Nous appelons image d'énergie locale de taille τ (seuil de la taille de structures claires) l'image formée des spectres de forme locaux de taille τ de chaque pixel de l'image I.

$$\mathcal{PS}(I, \eta) = \begin{cases} \mathcal{S}(\gamma_\eta(I)) - \mathcal{S}(\gamma_{\eta+1}(I)) & \text{si} \quad \eta \geq 0 \\ \mathcal{S}(\varphi_\eta(I)) - \mathcal{S}(\varphi_{\eta+1}(I)) & \text{si} \quad \eta \leq -1 \end{cases} \tag{36}$$

d) **Calcul du gradient morphologique de texture**

Notons le gradient morphologique de texture $\mathcal{G}^{tex}$ comme étant le résidu d'une dilatation morphologie et d'une érosion morphologique avec un élément structurant ES. Soit e_τ l'image d'énergie locale de taille τ calculée sur une couche de texture I_{tex}. Le gradient morphologique de texture est donc défini comme étant le supremum des gradients morphologiques des images d'énergie locales e_τ lorsque τ prend des valeurs dans l'ensemble $\Pi = [-\eta, \eta]$.

$$\mathcal{G}^{tex}(I_{tex})(p) = \bigvee_{\tau \in \Pi} [\delta_{ES}(e_\tau)(p) - \varepsilon_{ES}(e_\tau)(p)] \tag{37}$$

Le gradient de texture basé sur un filtre ASF consiste à appliquer l'analyse multi-échelle locale détaillée précédemment en utilisant comme image de marqueur dans le filtrage par nivellement, l'image simplifiée par le filtre ASF. Le processus de calcul du gradient $\mathcal{G}_{ASF}$ est décrit par la figure 19. L'élément structurant ES choisi pour les différentes opérations (l'ASF, la dilatation, l'érosion et l'ouverture morphologiques) est un disque de rayon adapté à chaque opération. Les fenêtres W centrées sur les pixels p de l'image ont été choisies empiriquement avec une même taille 4×4 pixels. L'ordre choisi pour le filtre ASF est $o = 6$ et l'ensemble $\Pi = \{-2, -4, -6, 2, 4, 6\}$.

Maintenant que nous avons détaillé différents descripteurs de couleur et de texture jugés intéressants, nos travaux jusqu'à la fin du projet consistent : i) à segmenter l'image couleur en tenant compte des informations de texture et de couleur relevées par ces descripteurs de sorte à pourvoir classer plus finement les régions d'intérêts en « ciel », « végétation », « batiments » ; ii) caractériser par des attributs pertinents certaines textures rencontrées afin de pouvoir reproduire la texture des plans du modèle 3D (tâche 2).

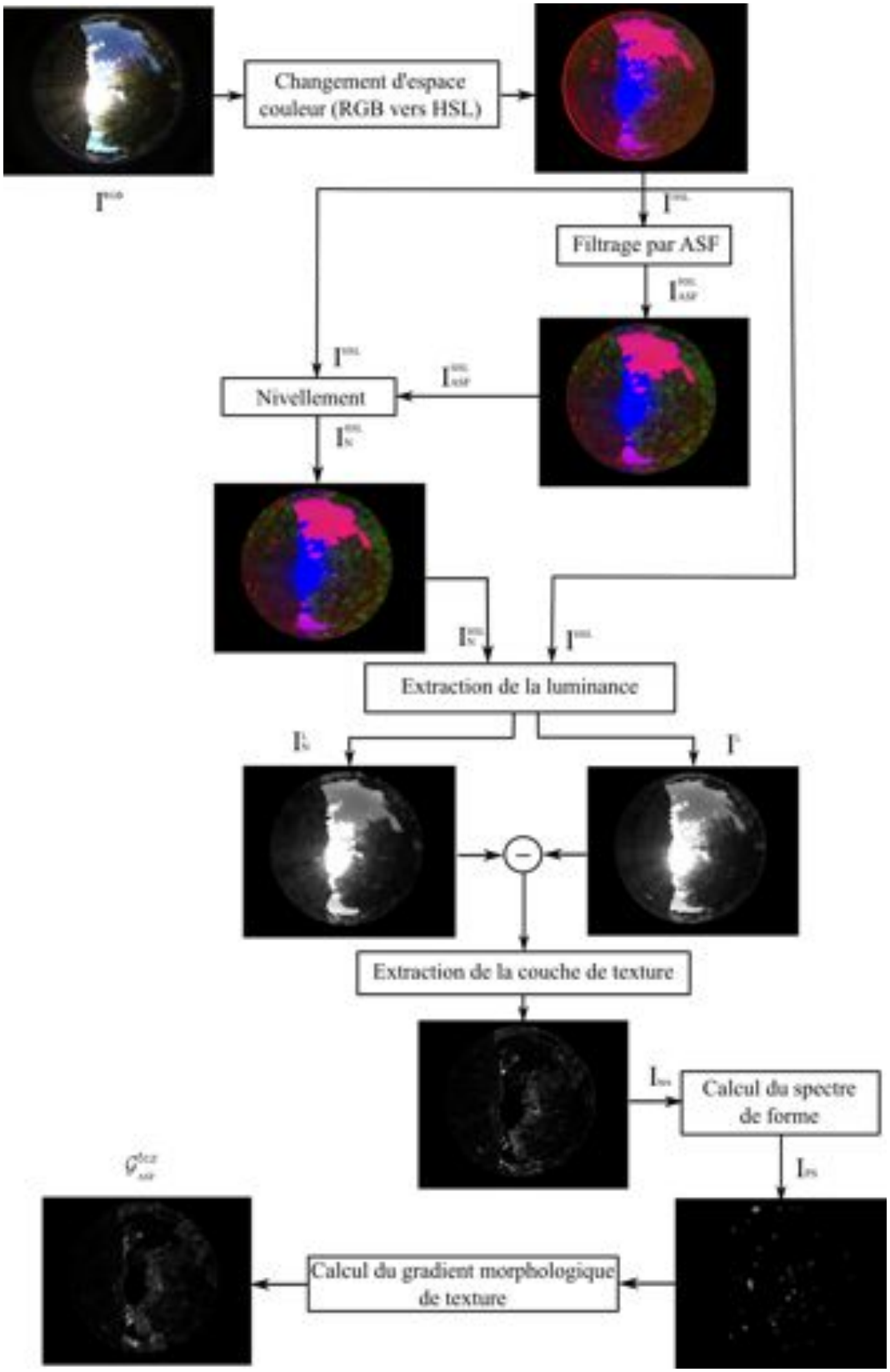


FIGURE 19 – Analyse multi-échelle locale pour le calcul du gradient morphologique de texture $\mathcal{G}_{ASF}$ basé sur le filtrage par ASF.

5 Conclusion

Ce rapport nommé L1.2 et dédié à la sous-tâche 1.1, avait pour objectif initial, de présenter la rectification des images acquises en mobilité afin de corriger les fortes distorsions dues à l'objectif grand angle utilisé. Cependant, comme indiqué, dans le précédent livrable (L1.1 du 18 octobre 2012), une étude réalisée dans le cadre de travaux antérieurs a permis de montrer que cette opération ajoute un coût calculatoire qui impacte le temps de traitement des images et nous éloigne ainsi de la contrainte temps réel imposée dans la sous-tâche 1.2. Les très bons résultats de détection de la ligne d'horizon et de pourcentage de ciel visible présentés dans le précédent livrable ainsi que les premiers résultats d'extraction de descripteurs couleur/texture présentés dans ce livrable nous ont amené à conclure que cette opération de rectification n'est finalement pas nécessaire au vue des attentes finales de la tâche 1. Néanmoins, afin de montrer l'avancement de cette tâche, nous avons décidé, non pas de supprimer le livrable correspondant, mais de remplacer son contenu par les travaux réalisés sur l'extraction de descripteurs couleur/texture pour la caractérisation de l'environnement de réception des signaux satellitaires.

Les travaux actuels et futurs de la tâche 1 ont pour objectif de caractériser l'environnement de réception des signaux satellitaires de manière plus fine (que la classification ciel/non-ciel effectuée dans la tâche 1.2) de sorte à pouvoir i) pondérer l'impact de chacun des satellites dans le calcul de position de localisation (selon qu'ils se trouvent dans une classe « ciel », « végétation », « bâtiment », etc) ; ii) caractériser par des attributs pertinents certaines textures rencontrées afin de pouvoir reproduire la texture des plans du modèle 3D (tâche 2). Nous avons donc, dans ce livrable, présenté un état de l'art exhaustif des descripteurs de couleur et de texture jugés pertinents dans le cadre de notre application et catégorisés parmi les attributs géométriques, les attributs basés sur la modélisation spatiale des textures, les attributs spatio-fréquentiels, les attributs statistiques et ceux issus de la morphologie mathématique. Cette sélection d'attributs couleur et texture est actuellement utilisée dans la définition d'un gradient structurel combinant les informations couleur/texture les plus pertinentes. Ce gradient structurel sera quant à lui utilisé en entrée de l'algorithme de segmentation par croissance de régions. L'image acquise en mobilité et ainsi segmentée (partitionnée en régions homogènes à l'aide de ces informations couleur/texture) facilitera sans aucun doute la classification de l'environnement de réception des signaux satellitaires en différentes classes (au minimum : « ciel », « végétation », « bâtiment »). Ces travaux feront l'objet du dernier livrable de la tâche 1, prévu en fin de projet.

Table des figures

1 Sensibilité spectrale des trois types de cônes photorécepteurs couleur de l'œil humain et des bâtonnets (source : `http://tpelavision.chez-alice.fr/oeil.htm`). 6
2 Représentation 3D de deux images couleur dans un espace muni du repère (C_1,C_2,C_3) à l'aide du logiciel "Color Inspector 3D" (source : `http://www.f4.fhtw-berlin.de/~barthel/ImageJ/ColorInspector//help.htm,`). 7
3 Exemples de textures répétitives prises de la base de texture VisTex [13]. 8
4 Différents types de textures extraites de la base de texture VisTex [13]. 8
5 Différentes catégories de texture. 9
6 Les différentes classes d'attributs de texture de la littérature. 10
7 Cliques. 11
8 Exemples de voisinages utilisés pour le calcul des LBP : à gauche $N = 8$, au centre $N = 12$ et à droite $N = 16$. 12
9 Illustration des différents histogrammes mono-dimensionnels et tridimensionnel d'une image couleur dans l'espace couleur (R, G, B) (de haut en bas et de gauche à droite : image originale couleur, histogramme tridimensionnel obtenu par le logiciel *Color Inspector 3D* et histogrammes mono-dimensionnels de la première, deuxième et troisième composante couleur). 14
10 Un vecteur de translation $\vec{R}$ en fonction des paramètres θ et $\|\vec{R}\|$. 15
11 Exemple de différents vecteurs de translation $\vec{R}$. 16
12 Illustration du gradient morphologique couleur calculé sur une image de la base du projet CAPLOC. 18
13 Illustration du gradient de Di-Zenzo sur une image couleur de la base CAPLOC. 19
14 Illustration du descripteur de texture basé sur les filtres de Gabor sur une image couleur issue du projet CAPLOC dans l'espace couleur (R, G, B). 21
15 Gradients calculés à partir d'attributs d'Haralick sur une image couleur de la base du projet CAPLOC. 25
16 Illustration d'une simplification d'une image issue de la base du projet CAPLOC par un filtre ASF avec différents ordres. 25
17 Illustration d'un filtrage de nivellement sur une image de la base du projet CAPLOC bruitée avec un bruit gaussien (de moyenne 5 et d'écart type 10) et avec trois images marqueurs (obtenues avec un filtre gaussien, un filtre moyenneur et un filtre ASF). 27
18 Calcul de la couche de texture. 28
19 Analyse multi-échelle locale pour le calcul du gradient morphologique de texture $\mathcal{G}_{ASF}$ basé sur le filtrage par ASF. 30

Références

[1] D. Attia : *Segmentation d'images par combinaison adaptative couleur/texture et classification de pixels. Application à la caractérisation de l'environnement de réception des signaux GNSS.* Thèse de doctorat, Université de Technologie de Belfort-Montbéliard, Octobre 2013. 5

[2] B. Pire : Experience de young. *Encyclopædia Universalis.* 6

[3] E. Hering : Zur erklarung der farbenblindheit aus der theorie der gegenfarben. *Lotos, Jb. f. Naturwiss*, 1:76–107, 1880. 6

[4] L. M. Hurvich et D. Jameson : An opponent-process theory of color vision. *Psychological Review*, 64:384–404, 1957. 6

[5] 0. D. Faugeras : Digital color image processing within the framework of a human visual model. *IEEE ASSP*, 27:380–393, 1979. 6

[6] W. Pratt : *Digital image processing.* New York, 1978. 7

[7] CIE : The cie interim colour appearance model ciecam's. Rapport technique, Rapport technique 131, 1998. 7

[8] CIE : A color appearance model for color management systems : Ciecam02. Rapport technique, Rapport technique, 2002. 7

[9] Nicolas Vandenbroucke : *Segmentation d'images couleur par classification de pixels dans des espaces d'attributs colorimétriques adaptés. Application á l'analyse d'images de football.* Thèse de doctorat, Université des sciences et technologies de Lille1, 2000. 7

[10] J. Sklansky : Image segmentation and feature extraction. *IEEE Transactions on Systems, Man, and Cybernetics*, 8:237–247, 1978. 7, 8

[11] H. Tamura, S.Mori et Y. Yamawaki. : Textural features corresponding to visual perception. *IEEE Transactions on Systems, Man, and Cybernetics*, 8:460–473, 1978. 7, 8

[12] R. M. Haralick : Statistical and structural approaches to texture. *Proceedings of the IEEE*, 67:786–804, 1979. 7, 8

[13] VisTex : Vision texture database. Massachusetts Institute of Technology, 1995. 8, 32

[14] J.P. Cocquerez et S. Philipp : *Analyse d ?images : filtrage et segmentation.* 1995. 8

[15] S. Herlidou : *Caractérisation tissulaire en IRM par l ?analyse de texture : Etude du tissu musculaire et de tumeurs intracraniennes.* Thèse de doctorat, Université de Rennes 1, 1999. 8

[16] B. Julesz : Textons, the elements of texture perception and their interactions. *Nature*, 290:91–97, 1981. 8

[17] S. Mavromatis : *Analyse de texture et visualisation scientifique.* Thèse de doctorat, Université de la méditérranée, 2001. 8

[18] A. Hanbury : *Morphologie mathématique sur le cercle unité avec applications aux teintes et aux textures orientées.* Thèse de doctorat, École supérieure des Mines de Paris, 2002. 8

[19] M. Tuceryan et A. K. Jain : Texture analysis. *The Handbook of Pattern Recognition and Computer Vision*, pages 207–248. Editions World Scientific Publishing Co, 1998. 9, 13

[20] A. Porebski, N. Vandenbroucke et L. Macaire : Attributs de texture couleur. *Proceedings de l'Ecole d'Hiver sur l'Imagerie Numérique Couleur (EHINC'09)*, 2009. Ecole d'Ingénieurs du Pas-de-Calais (EIPC), Laboratoire d'Automatique, G'eenie Informatique et Signal (LAGIS UMR CNRS), Univeristé des Sciences et Technologies de Lille. 9, 10, 12, 17, 21, 25

[21] C. Zheng, D.W. Sun et L. Zheng. : A new region-primitive method for classification of colour meat image texture based on size, orientation and contrast. *Meat Science*, 76:620–627, 2007. 10, 17

[22] O.J. Hernandez et A. Khotanzad : Color image segmentation using multispectral random field texture model and color content features. *Journal of Computer Science et Technology*, 4(3):141–146, 2004. 10

[23] J. Bennett et A. Khotanzad : Multispectral random fieldmodels for synthesis and analysis of color images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):327–332, 1998. 10

[24] O.J. Hernandez, J. Cook, M. Griffin, C. De Rama et M. McGovern : Classification of color textures with random field models and neural networks. *Journal of Computer Science et Technology*, 5(3):150–157, 2005. 10, 11

[25] A. Khotanzad et O.J. Hernandez : A classification methodology for color textures using multispectral random field mathematical models. *Mathematical and Computational Applications*, 11(2):111–120, 2006. 10, 11

[26] I.U.H. Qazi, O. Alata, J.C. Burie et C. Fernandez-Maloigne. : Colour spectral analysis for spatial structure characterization of textures in ihls colour space. *Pattern Recognition*, 43(3):663–675, 2009. 10

[27] F. Tupin et M. Sigelle : *Markov Random Fields*. ISTE Ltd and John Wiley & Sons, Digital Signal and Image Processing Series, 2008. 11

[28] S. Chindaro, K. Sirlantzis et M.C. Fairhurst : Ica-based multi-colour space texture classification system. *Electronics Letters*, 42(21):1208–1210, 2006. 12

[29] D. Harwood, T. Ojala, M. Pietikainen, S. Kelman et S. Davis : Texture classification by center symmetric auto correlation using kullback discrimination of distributions. Rapport technique, Computer Vision Laboratory, Center for Automation Research, University of Maryland, College Park, Maryland. CAR-TR-678, 1993. 12

[30] T. Ojala, M. Pietikinen et D. Harwood : A comparative study of texture measures with classification based on feature distributions. *Pattern Recognition*, 29:51–59, 1996. 12

[31] M. Pietikäinen, A. Hadid, G. Zhao et T. Ahonen : *Computer Vision Using Local Binary Patterns*, volume 40, chapitre Local Binary Patterns for Still Images, pages 13–47. Springer-Verlag London Limited, 2011. 12

[32] M. Pietikäinen, T. Mäenpää et J. Viertola : Color texture classification with color histograms and local binary patterns. *In Proceedings of the 2nd International Workshop on Texture Analysis and Synthesis*, pages 109–112, 2002. 12

[33] S. DiZenzo : A note on the gradient of multi-image. *Computer Vision, Graphics, and Image Processing*, 33:116–125, 1986. 13, 18

[34] C. Palm et T. Lehmann : Classification of color textures by gabor filtering. *Machine Graphics & Vision International Journal*, 11:195–219, 2002. 13, 20

[35] P. Carré, N Richard et C. Fernandez-Maloigne : *Analyse et caractérisation des signaux bidimentionnels*, chapitre Représentation en ondelettes. Traité I2C, 2001. 13

[36] R. M. Haralick, K. Shanmugan et I. Dinstein : Textural features for image classification. *IEEE Trans. On Systems, Man, and Cybernetics*, 3(6):610–621, novembre 1973. 15, 16, 21, 22

[37] C. Palm : Color texture calssification by integrative co-occurrence matrices. *Pattern recognition*, 37:965–976, 2004. 15

[38] M. Unser : Sum and difference histogram for texture classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:118–125, 1986. 16

[39] A. Porebski et N. Vandenbroucke L. Macaire : Iterative feature selection for color texture classification. *IEEE International Conference on Image Processing ICIP*, 3:509–512, 2007. 16

[40] L. Tesar, D. Smutek, A. Shimizu et H. Kobatake : Medical image segmentation using cooccurrence matrix based texture features calculated on weighted region. *ACST'07 Proceedings of the third conference on IASTED International Conference : Advances in Computer Science and Technology*, pages 243–248, 2007. 16

[41] C. Meurie, A. Cohen et Y. Ruichek : An efficient combination of texture and color information for watershed segmentation. *In Proceedings of International Conference on Image and Signal Processing ICISP*, pages 147–156, 2010. 16

[42] F. Niea, C. Gaoa, Y. Guoa et M. Ganc : Two-dimensional minimum local cross-entropy thresholding based on co-occurrence matrix. *Computers and Electrical Engineering*, 37:757–767, 2011. 16

[43] A. Dash, P. Kanungo et B.P. Mohanty : A modified gray level co-occurrence matrix based thresholding for object background classification. *International Conference on Communication Technology and System Design 2011, Procedia Engineering*, 30:85–91, 2012. 16

[44] M-M. Galloway : Texture analysis using gray level run lengths. *Computer graphics and image processing*, 4:172–179, 1975. 17

[45] E. Angelini, I. Bloch, Y. Gousseau, S. Ladjal, B. Pesquet-Popescu, M. Sigelle et F. Tupin : Morphologie mathématique. *Méthodes Avancées de Traitement d ?Images SI 343 - MATIM*, chapitre 6, pages 95–126. Polycopié du cours SI343 - MATIM, 2007. 17, 28

[46] L. Najman et H. Talbot : *Morphologie mathématique : Tome 1, Approches déterministes*, volume 1 de *IC2 : Série Traitement du signal et de l'image*. Hermès-Lavoisier, 1 édition, 10 2008. 18

[47] S. Beucher : *Segmentation d'images et morphologie mathématique*. Thèse de doctorat, Ecole des mines de Paris, Juin 1990. 18

[48] J. F. Rivest, P. Soille et S. Beucher : Morphological gradients. *Journal of. Electronic Imaging*, 2:326–336, 1993. Centre de Morphologie Mathématique, Ecole des Mines de Paris. 18

[49] P. Lambert : *Etudes méthodologiques du filtrage et de la segmentation d'images multi-composantes*. Thèse de doctorat, UNIVERSITE DE SAVOIE, juillet 2002. Habilitation a Diriger Des Recherches en Electronique, Electrotechnique et Automatique. 18

[50] J. Dubois : Segmentation par approche contours. Mémoire de D.E.A., Universite de Rouen, 2008. 19

[51] C. Palm, D. Keysers, T. Lehmann et K. Spitzer : Gabor filtering of complex hue saturation images for color texture classification. *Proceedings of the 5 th Joint Conference and Information Science*, pages 45–49, 2000. 20

[52] M. Sharma : *Evaluation of image segmentation and texture extraction methods in scene analysis*. Thèse de doctorat, University of Exeter, 2001. 21, 22

[53] M. Sharma et S. Singh : Evaluation of texture methods for image analysis. *in Proceedings of the 7th Australian and New Zealand Intelligent Information Systems Conference*, pages 117–121, 2001. 22

[54] S. Paquis : *Reconnaissance des surfaces de chaussée par analyse de texture image*. Thèse de doctorat, Université Jean Monnet de Saint-Étienne, 2003. 22, 23

[55] H. Majdoulayne : *Extraction de caractéristiques de texture pour la classification d ?images satellites*. Thèse de doctorat, Université de Toulouse, 2009. 22

[56] H. Anys et D.C. He : Evaluation of textural and multipolarization radar features for crop classification. *IEEE Transactions on Geoscience and Remote Sensing*, 23:1169–1181, 1995. 25

[57] V. Bombardier, E. Schmitt et P. Charpentier : A fuzzy sensor for color matching vision system. *Measurement*, 42:189–201, 2009. 25

[58] Pierre Soille : *Morphological image analysis*. Springer-Verlag Berlin, Heidelberg, 1999. 25

[59] Jesus Angulo : Morphological texture gradients. application to colour+texture watershed segmentation. *Proc. of the 8th International Symposium on Mathematical Morphology*, pages 399–410, octobre 2007. 26, 27

[60] A. Sofou et G. Evan : Coupled geometric and texture pde-based segmentation. *IEEE International Conference on Image Processing*, 2:650–653, 2005. 27

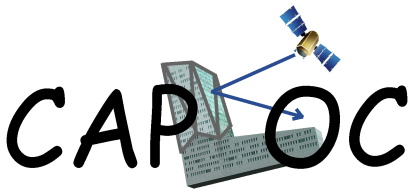
Groupe opérationnel n°2 du Predit 4

« Gestion du trafic et des mobilités »

CAPLOC

Combinaison de l'**A**nalyse d'images
et de la connaissance de la **P**ropagation
des signaux pour la **LOC**alisation

Convention 10-MT-CVS-001



Livrable de la tâche 1.2

Détection de la ligne d'horizon et évaluation du pourcentage de ciel visible par analyse d'images couleur

18 octobre 2012

Cyril MEURIE
Ifsttar-LEOST
Dhouha ATTIA, Yassine Ruichek
UTBM/IRTES-SET

Identification

Acronyme du projet	CAPLOC
Titre du projet	Combinaison de l'Analyse d'images et de la connaissance de la Propagation des signaux pour la LOCalisation
Coordinatrice du projet	Juliette MARAIS (Ifsttar-LEOST)
Date de début de projet	19 octobre 2010
Date de fin de projet	18 octobre 2014
Partenaires	Ifsttar-LEOST, UTBM/IRTES-SET
Type de document	Livrable
Date contractuelle de livraison	18 octobre 2012
Accessibilité	La diffusion de ce document est limitée au programme PREDIT

Gestion des versions

Version	**Date**	**Auteurs**	**Modifications**
1.0	15/10/2012	Cyril MEURIE et Dhouha ATTIA et Yassine RUICHEK et Amaury FLANCQUART	Rédaction
1.1	16/10/2012	Sébastien LEFEBVRE et Dhouha ATTIA	Corrections
1.2	17/10/2012	Juliette MARAIS et Sebastien AMBELLOUIS	Corrections
2.0	18/10/2012	Cyril MEURIE	Approbation

Table des matières

1 Introduction **4**

2 Contexte et État de l'art **5**

3 Protocole d'expérimentation **6**

4 Stratégie proposée **8**

5 Simplification de l'image **9**
5.1 Lissage par filtre médian . . . 9
5.2 Lissage par filtre gaussien . . . 10
5.3 Lissage par filtre moyenneur . . . 10
5.4 Simplification par Reconstruction Géodésique par Dilatation . . . 11
5.5 Lissage par filtre de Nagao . . . 12

6 Classification pixellaire **13**
6.1 Méthodes de classification non-supervisée . . . 14
6.1.1 KMlocal . . . 14
6.1.2 Fuzzy Cmeans . . . 15
6.1.3 Fisher . . . 16
6.1.4 Statistical Region Merging (SRM) . . . 16
6.2 Méthodes de classification supervisée . . . 17
6.2.1 K-Plus proches voisins (Kppv) . . . 17
6.2.2 Bayes . . . 17
6.2.3 Les Supports Vecteurs Machines . . . 18

7 Premiers résultats sur la classification des régions de ciel/non-ciel et sur la détection de la ligne d'horizon **18**
7.1 Méthode d'évaluation de la qualité de la classification . . . 18
7.2 Résultats de classification pixellaire ciel/non ciel . . . 19
7.3 Repositionnement des satellites et détection de la ligne d'horizon . . . 27

8 Conclusion **32**

Table des figures **33**

Table des tableaux **34**

Bibliographie **37**

1 Introduction

Le projet CAPLOC vise l'amélioration de la précision de localisation d'un véhicule en milieu urbain dense dans un contexte de réduction globale des coûts : l'idée est d'utiliser la perception vidéo ainsi que des outils de traitement des images pour étudier l'environnement de propagation situé autour de l'antenne GNSS d'un récepteur faible coût. Ce projet de recherche amont s'inscrit dans le cadre d'un brevet [1] déposé par l'Ifsttar qui constitue un apport innovant pour la localisation précise. Deux briques technologiques liées à l'analyse d'images sont abordées dans ce projet.

La première (relative à la tâche 1) vise à classer en temps réel l'état de réception de chacun des satellites reçus (soit directement, soit après réflexion) ; sur la base d'un système mono-caméra, les pixels de l'image sont dans une première phase, étiquetés en deux classes, « ciel » et « non-ciel », permettant ainsi, de déterminer si le satellite, dont la position est re-projetée dans l'image, est localisé dans une région de ciel ou une région de non-ciel. Dans ce dernier cas, les satellites peuvent être considérés comme reçus après réflexion. L'algorithmie associée pourra choisir d'exclure ou de pondérer l'impact de chacun de ces satellites dans le calcul de position de localisation. Dans une deuxième phase, une classification plus fine sera réalisée afin de pouvoir étiqueter les pixels de l'image en trois classes, « ciel », « végétation » et « bâtiment », permettant ainsi de i) pondérer différemment l'impact de chacun des satellites, ii) caractériser par des attributs pertinents certaines textures rencontrées afin de pouvoir reproduire la texture des plans du modèle 3D. Cette première brique est composée de trois sous-tâches.

1. Étalonnage et calibrage du système d'acquisition ;
2. Détection de la ligne d'horizon et évaluation du pourcentage de ciel visible par analyse d'images couleur ;
3. Extraction et caractérisation de la texture ;

La deuxième brique (relative à la tâche 2) technologique vise à corriger l'erreur induite par les réflexions des signaux GNSS (effet de multi-trajets) en modélisant en 3D l'environnement situé autour de l'antenne de réception GNSS. Dans ce contexte applicatif, un des objectifs à plus long terme est d'obtenir cette modélisation 3D en temps réel. Pour ce faire, un compromis entre la rapidité et la précision des résultats obtenus est à définir.

Ce livrable est dédié à la sous-tâche T1.2 dont l'objectif est de détecter la ligne d'horizon et d'évaluer le pourcentage de ciel visible. La sous-tâche 1.1 relative à la rectification des images fisheye acquises a été reportée au vu des résultats très satisfaisants de la stratégie de détection de ciel visible appliquée directement sur les images acquises. Dans un premier temps, nous allons situer le contexte du projet CAPLOC et réaliser un état de l'art des travaux similaires de détection de ciel dans des images d'environnements urbains. Ensuite, nous présenterons d'une part, le système expérimental mis en œuvre nous ayant permis d'acquérir une nouvelle base d'images et d'autre part, l'approche globale proposée. Enfin, nous présenterons les premiers résultats de détection de ciel et de la ligne d'horizon avant de conclure et de présenter nos perspectives de travail pour la suite de ce projet.

2 Contexte et État de l'art

Comme indiqué dans l'introduction de ce livrable, la finalité de cette sous-tâche consiste à détecter le pourcentage de ciel visible dans l'image ainsi que la ligne d'horizon. Cela sous-entend que la stratégie de traitement d'images proposée devra permettre (après repositionnement des satellites dans l'image) d'indiquer si un satellite se situe dans une région de « ciel de l'image (i.e. les signaux sont émis depuis le satellite jusqu'au récepteur via des trajets directs) ou dans une région de « non-ciel » (i.e. les signaux émis depuis le satellite sont soit réfléchis, soit bloqués). Cette information fournie, nous pourrons recalculer une position de localisation en excluant ou pondérant l'impact de chacun des satellites (cf. tâche 3 pour plus de détails sur l'utilisation des résultats de cette tâche).

Durant ces dernières années, nous avons observé une forte croissance des travaux utilisant la vision par ordinateur pour des applications liées aux systèmes de localisation. Ces travaux peuvent être différenciés en fonction du type de caméras utilisées ou en fonction de la façon dont l'image est exploitée dans les systèmes de localisation. Sur le premier point, nous pouvons distinguer les méthodes utilisant des images perspectives [2, 3, 4, 5, 6] et d'autres méthodes utilisant des images omni directionnelles permettant ainsi d'avoir un champ de vision plus large [7, 8, 9, 10, 11]. Si l'on considère la façon dont l'image est exploitée dans les systèmes de localisation, nous pouvons citer des méthodes basées sur des algorithmes d'appariement qui consistent à trouver, à partir d'une base de données d'images géoréférencées (par un GPS), l'image la plus proche de l'image requête [2, 3, 5, 6]. Une autre solution consiste à utiliser des images (principalement omni directionnelles) souvent couplées à des modèles 3D pour détecter la ligne d'horizon séparant le ciel des autres objets constituants l'environnement [9, 11].

Dans le cadre du projet CAPLOC, nous sommes plus sensibilisés à la deuxième catégorie de méthodes qui utilisent l'image pour la détection de la ligne d'horizon. Nous décrirons donc brièvement le principe de quelques méthodes existantes afin de mieux situer l'approche proposée dans ce livrable.

Koch et Teller proposent une méthode de localisation qui utilise un modèle 3D connu de la scène étudiée (des scènes d'intérieur ont été utilisées) et une caméra grand angle. Leur technique consiste à apparier les lignes existantes dans le modèle 3D avec les lignes correspondantes dans la séquence d'images omni-directionnelles [8]. La mise en correspondance est effectuée entre les contours présents dans l'image du monde réel et les segments issus du modèle 3D. L'hypothèse de ce système est de considérer que l'environnement présente un nombre important de lignes droites. La solution proposée est constituée de deux phases : la première concerne la phase d'initialisation de la pose de la caméra et requiert 1 minute pour un volume de recherche de trois mètres. Cela s'effectue par une première mise en correspondance entre les pixels-contours dans l'image et les segments dans le modèle 3D. Quant à la deuxième, elle relève de la phase de maintenance. Cette solution a été testée sur de longues séquences d'images dont la taille de chacune d'elles était de 512x384 pixels et le temps de calcul optimisé atteignait 10 secondes avec une pose verticale de l'image. Cette solution ne semble pas adaptée à notre application au vu de la contrainte temps-réel requise pour la sous-tâche 1.2. Meguro et al. proposent une méthode qui vise l'amélioration de la précision du positionnement [9]. Les auteurs utilisent une caméra infrarouge en visant l'élimination des satellites NLOS, considérés comme un facteur dégradant de la performance du système de positionnement. En effet, grâce à la caméra infrarouge, le ciel apparaît particulièrement sombre, ce qui facilite la détection de la frontière entre le ciel (sombre) et les bâtiments (blancs). Les inconvénients majeurs de cette approche résident tout d'abord dans le coût élevé de ce système rendant impossible son déploiement à grande échelle. De plus,

l'utilisation de ce type particulier de caméra demande une méthode de projection précise. Un étalonnage spécifique avec des sources ponctuelles thermiques disposées à l'intérieur est requis, ce qui rend une fois encore l'exploitation de cette solution compliquée. Enfin, selon les auteurs, la détection de la ligne d'horizon est principalement basée sur une projection azimutale, mais le résultat obtenu ne reflète pas la ligne d'horizon réelle et l'approche peut potentiellement mener à des erreurs lorsque la caméra n'est pas orientée verticalement. RAMALINGAM et al. propose une méthode qui vise la détection de la ligne d'horizon en utilisant une caméra omni directionnelle visible au lieu d'une caméra infrarouge ainsi qu'un modèle 3D de l'environnement [11]. La technique proposée consiste dans un premier temps à calibrer la caméra à objectif fisheye afin d'obtenir ses paramètres. Une détection de contours est ensuite réalisée afin d'extraire une première ligne d'horizon dite synthétique. Ceci est réalisé en utilisant l'image fisheye synthétique (obtenue à l'aide des paramètres de calibrage trouvés précédemment) et le modèle 3D. D'autre part, une classification de l'image réelle (acquise à l'aide de la caméra fisheye) est effectuée via la résolution d'un problème de minimisation d'énergie afin de détecter la deuxième ligne d'horizon dite réelle. La troisième et dernière phase consiste à mettre en correspondance les deux résultats de ligne d'horizon (synthétique et réelle) obtenus. Pour en terminer avec cet état de l'art, nous rappelons d'anciens travaux réalisés dans le cadre du projet région intitulé ViLoc (faisant intervenir l'Ifsttar-LEOST et l'UTBM/IRTES-SET) et qui consistaient à caractériser l'environnement de réception des signaux GNSS sans tenir compte de la contrainte temps-réel [12, 13, 14]. Une méthode de détection de ciel dans les images a été proposée et s'appuyait sur une segmentation de l'image par combinaison adaptative des informations de couleur et de texture couplée à une classification pixellaire adaptée. Les résultats étaient très satisfaisants en termes de qualité de détection des régions de « ciel » et de « non-ciel », mais avait l'inconvénient majeur de nécessiter un temps de traitement élevé et donc inapproprié pour les besoins du projet CAPLOC.

3 Protocole d'expérimentation

Dans le projet CAPLOC, nous utilisons un système complet d'acquisition d'images installé sur un véhicule de laboratoire qui permet de réaliser des acquisitions de données en mobilité et de se confronter aux difficultés de l'embarqué (contrainte temps-réel, surexposition de l'image, changement brusque de luminosité, etc). Dans le cadre des travaux réalisés dans cette sous-tâche, nous utilisons une caméra monoCCD couleur avec un objectif fisheye installée sur le toit du véhicule de laboratoire. Plusieurs récepteurs GNSS sont également utilisés : un récepteur bas-coût et un récepteur Septentrio afin de récupérer l'information de position et les données brutes, ainsi qu'un récepteur GNSS-RTK pour disposer d'une trajectoire de référence. La figure 1 illustre le système d'acquisition installé sur le véhicule de l'UTBM/IRTES-SET et sur le camion de laboratoire de l'Ifsttar-LEOST.



(a) Véhicule de l'IRTES-SET



(b) Véhicule de l'Ifsttar-LEOST

FIGURE 1 – Les véhicules de laboratoire de l'UTBM/IRTES-SET et de l'Ifsttar-LEOST

Dans la suite de ce rapport, nous testons la stratégie proposée sur deux bases de données images. La première, présenté sur la figure 2, a été acquise dans le cadre du projet LOCOPROL (en décembre 2004) et nous permettra de procéder à un première série de tests permettant d'identifier les deux meilleurs algorithmes de simplification/classification.

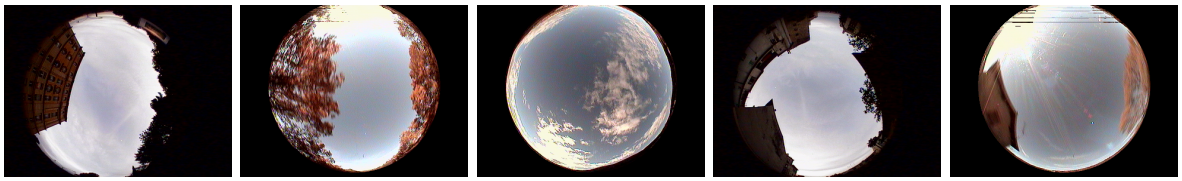


FIGURE 2 – Quelques images de la base du projet LOCOPROL

La seconde base nouvellement acquise dans le cadre du projet CAPLOC, nous permettra d'optimiser les paramètres de l'approche proposée et de prouver la robustesse du système. En effet, cette base a été réalisée en juin 2010 à Belfort : elle nous permet de disposer de données hétérogènes et de complexité variée (canyon urbain, abondance de végétation, surexposition, changements brusques de luminosité, etc) et se trouve, de ce fait, particulièrement adaptée pour la validation des objectifs de la tâche 1. La figure 3 illustre un trajet effectué à Belfort avec les images acquises en mobilité sur lesquelles les algorithmes présentés dans la section des résultats expérimentaux ont été optimisés et validés.

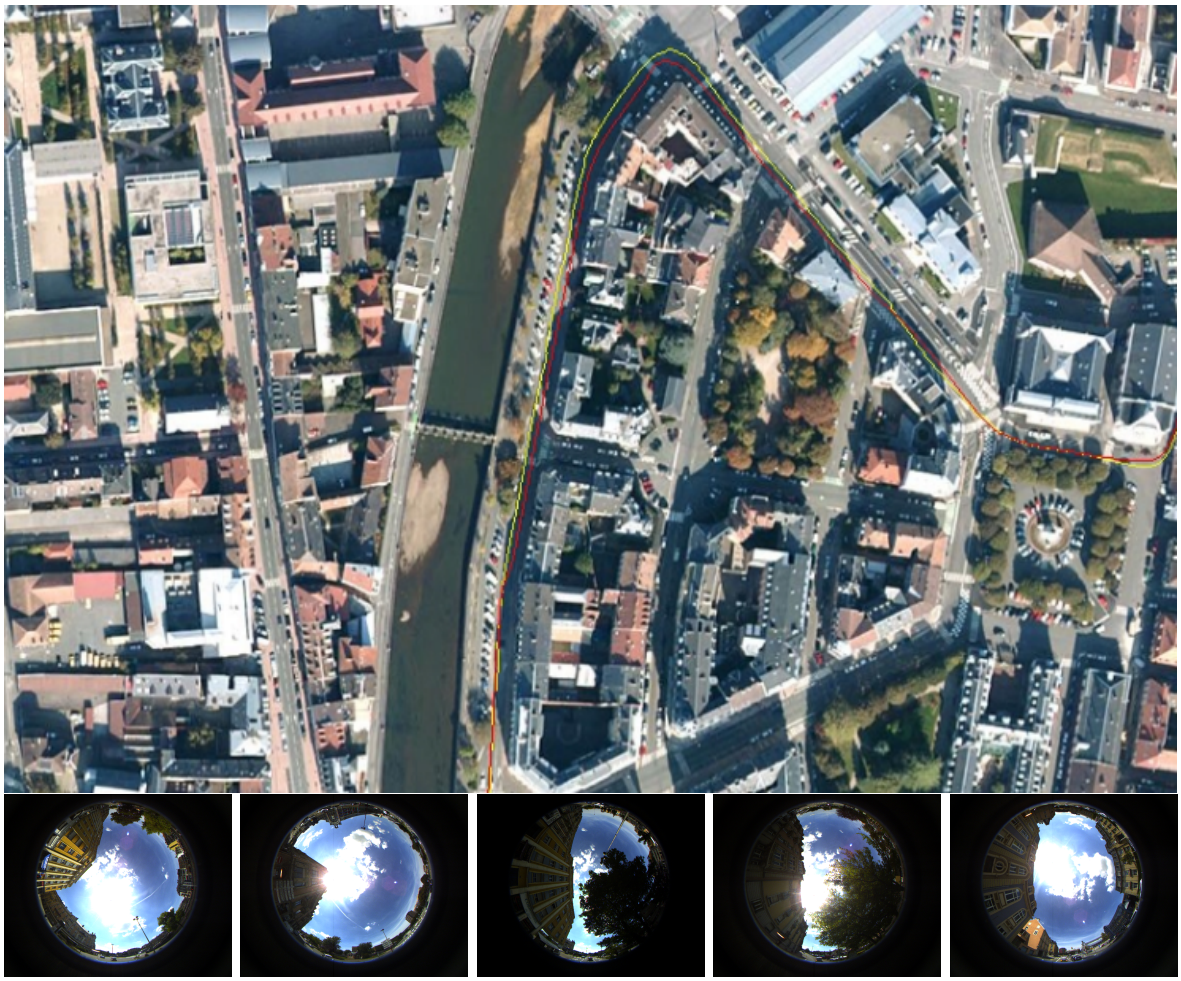


FIGURE 3 – Exemple de parcours et quelques images de la base du projet CAPLOC

4 Stratégie proposée

Dans cette section, nous présentons le synopsis global d'une stratégie permettant de détecter les zones de « ciel » et de « non-ciel » d'une image fisheye. Ceci permettra, après repositionnement des satellites dans l'image, d'exclure ou pondérer l'impact de chacun d'entre eux dans le calcul de position de localisation. Ce synopsis illustré sur la figure 4 est composé d'une brique d'acquisition de données, de cinq briques de traitement d'images, d'une brique d'évaluation et d'une dernière brique de calcul de position de localisation. La première brique concerne donc la phase d'acquisition des données de localisation (à l'aide des récepteurs GNSS embarqués) et images (à l'aide de la caméra couleur à objectif fisheye installée sur le toit du véhicule). Pour la partie traitement d'images, nous commençons par une phase de simplification de l'image afin de réduire les effets de luminosité, ombres, etc. Une classification pixellaire est ensuite réalisée afin d'affecter à chaque pixel de l'image, une classe d'appartenance parmi les classes « ciel » ou « non-ciel ». Une étape de post-traitement morphologique est ensuite effectuée pour optimiser la phase de classification « ciel »/« non-ciel ». A ce moment de la stratégie, une évaluation des résultats est effectuée par rapport à une vérité terrain (c-a-d par rapport à une classification faite à la main par un expert visant à identifier les régions de « ciel » des régions de « non-ciel » de l'image). Ceci permettra d'évaluer le pourcentage de pixels correctement classés par notre approche par rapport à une classification de référence. Il est donc important de signaler que les algorithmes ont été testés sur la totalité des images de la base et évalués par rapport à une vérité

terrain. L'étape suivante consiste à repositionner les satellites dans l'image afin d'identifier ceux se situant dans une zone de « ciel » (satellites Line-Of-Sight (LOS)) ou de « non-ciel » (satellites Non-Line-Of-Sight (NLOS)). Cette dernière étape de traitement d'images effectuée, il devient possible de calculer une nouvelle position de localisation en excluant/pondérant les différents satellites suivant leur classification dans l'image.

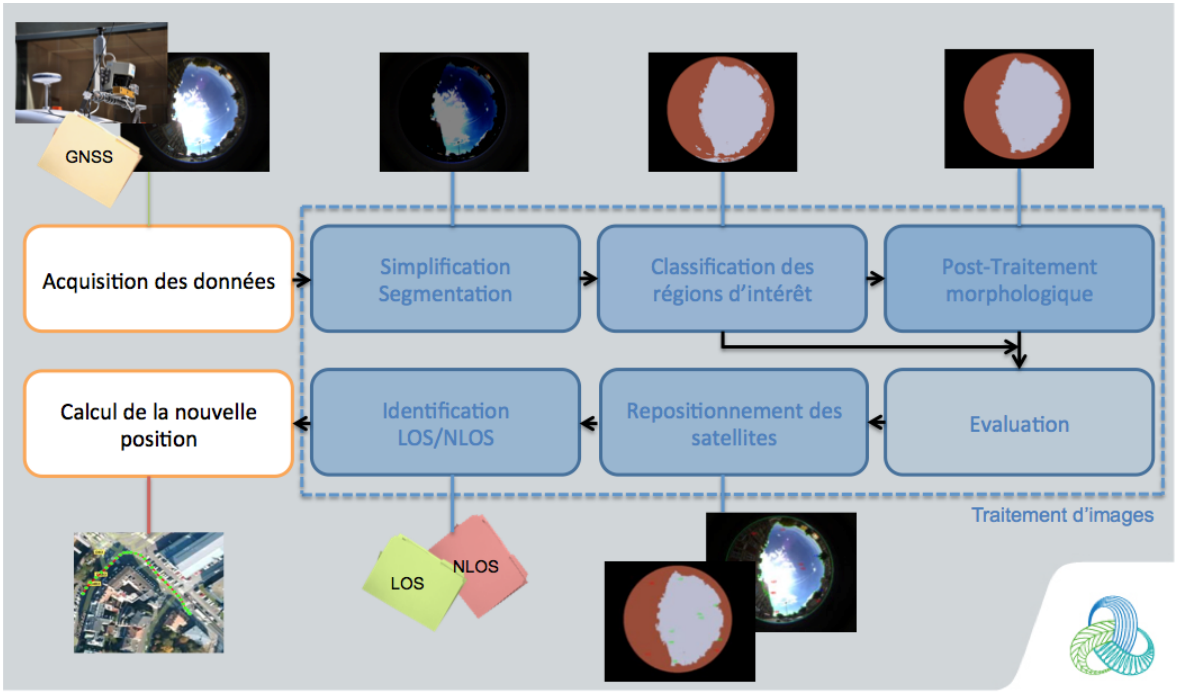


FIGURE 4 – Synopsis de la stratégie de détection de ciel proposée

5 Simplification de l'image

La simplification de l'image est une étape importante de notre stratégie de détection de ciel. Elle permet d'atténuer les effets des changements brusques de luminosité, ombres, etc venant perturber la classification des pixels de l'image. Une importance toute particulière est donnée au temps de traitement de l'approche globale proposée. Par conséquent, nous avons sélectionné un ensemble d'algorithmes de simplification (lissage) de la littérature connus pour leur rapidité et leurs bons résultats. Nous décrivons donc rapidement le principe de chacun d'entre eux et illustrons leur comportement sur une image de la base d'images du projet LOCOPROL. Une analyse plus complète sera réalisée dans la section 7 afin de définir, à partir de tests réalisés sur les deux bases de tests, le meilleur algorithme de simplification.

5.1 Lissage par filtre médian

Un lissage par filtre médian est une opération de lissage non linéaire qui consiste à parcourir l'image en remplaçant la valeur du pixel courant par la valeur médiane de ses voisins définis dans un voisinage de taille W. Le filtre médian est caractérisé par sa capacité à lisser tout bruit impulsionnel ou exponentiel. Notons que plus la taille du voisinage est grande, plus le lissage est fort et donc plus les détails de l'image sont supprimés. La figure 5 illustre sur une image de notre application, l'effet du filtre médian en fonction de la taille du voisinage.



(a) Image initiale (b) $W = 2$ (c) $W = 4$ (d) $W = 6$ (e) $W = 8$

FIGURE 5 – Résultats d'un lissage effectué par un filtre médian avec différentes tailles de voisinage.

5.2 Lissage par filtre gaussien

Un lissage par filtre gaussien est un lissage par convolution caractérisé par le paramètre σ et l'écart type de la fonction gaussienne utilisée et donnée par l'équation 1. Ce paramètre prend ses valeurs dans l'intervalle $[0, \frac{TailleImage}{6}]$. Cette opération de lissage est similaire dans son principe au filtre moyenneur à la différence près d'avoir une moyenne pondérée telle que les pixels proches du centre de l'image disposent d'un poids plus élevé.

$$g(i,j) = \frac{1}{\pi\sigma^2} \exp(\frac{i^2+j^2}{2\sigma^2}) \tag{1}$$

La figure 6 illustre sur une image, l'effet du filtre gaussien en fonction du paramètre σ. Notons que plus la valeur de σ est élevée (i.e. plus la « cloche » de la gaussienne est large), plus l'image est filtrée donc floue.



(a) Image initiale (b) $\sigma = 2$ (c) $\sigma = 4$ (d) $\sigma = 6$ (e) $\sigma = 8$

FIGURE 6 – Résultats d'un lissage effectué par un filtre gaussien avec différentes valeurs de σ.

5.3 Lissage par filtre moyenneur

Un filtre moyenneur correspond à un filtre linéaire spatial isotrope basé sur le fait que la valeur d'un pixel est relativement similaire à son voisinage. Le principe de ce filtre consiste à appliquer un masque de convolution de taille W afin de remplacer la valeur de chaque pixel par la moyenne pondérée de ses voisins (dans ce masque). L'équation 2 donne un exemple du masque de convolution h de taille $W = 3$.

$$h = \frac{1}{9} \times \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \tag{2}$$

La figure 7 illustre sur une image, l'effet du filtre moyenneur en fonction du paramètre W. Nous pouvons noter que le lissage préserve toutes les zones de l'image où le niveau de gris est constant (zone homogène) mais que l'image devient de plus en plus floue avec l'augmentation de la taille du masque de convolution. En effet, sur l'image de la figure 7 (i.e. $W = 8$), les contours des objets deviennent très flous contrairement à ceux du filtre médian, pour une même taille de masque de convolution (qui lisse les zones homogènes en préservant les contours). De plus, ce

filtre à l'inconvénient d'engendrer des phénomènes de fausses couleurs, contrairement au filtre médian.



(a) Image initiale (b) $W = 2$ (c) $W = 4$ (d) $W = 6$ (e) $W = 8$

FIGURE 7 – Résultats d'un lissage effectué par un filtre moyenneur avec différentes valeurs de W.

5.4 Simplification par Reconstruction Géodésique par Dilatation

Pour décrire précisément l'opération de reconstruction géodésique par dilatation (RGD), il est nécessaire de rappeler quelques définitions basiques de morphologie mathématique. Rappelons que ces définitions sont présentées avec beaucoup plus de détails dans les travaux [15, 16].

Définition 1 : Un maximum régional noté M est un plateau tels que les niveaux de gris des pixels voisins sont inférieurs à la valeur $I(M)$.

Définition 2 : Un plateau en un pixel p d'une image I, noté $Plt_p(I)$ est une composante connectée contenant p et ayant un niveau de gris constant égal à $I(p)$.

Notons $Max_{reg}(I)$ l'ensemble des maximums régionaux de l'image I. En effet, M peut ne pas être un seul pixel mais une région de l'image. Par conséquent, un maximum régional $M \subseteq \mathbf{IR}^2$. Formellement, M est un maximum régional si et seulement si il existe un voisinage $\mathcal{V}_M$ qui satisfait l'équation 4.

$$\forall p \in \mathcal{V}_M \backslash \{M\} \qquad I(p) < I(M) \tag{3}$$

La reconstruction géodésique par dilatation est une technique des plus utilisées pour l'extraction des maximums régionaux [17]. En effet, cette technique fait partie des méthodes de filtrage par lissage de l'image utilisant les opérateurs de la morphologie mathématique (dilatation, érosion, ouverture, fermeture, filtre alterné séquentiel, etc). La RGD utilise deux images en entrée : une image *masque* I et une image *marqueur* I', telles qu'elles vérifient la condition $\forall p,\ I'(p) < I(p)$. Le principe de la RGD visant à extraire les maximums régionaux de l'image consiste à appliquer une suite finie de dilatation géodésique jusqu'à idempotence.

Définition 3 : L'idempotence est une description de stabilité du résultat tel que le résultat obtenu à l'itération $(n+1)$ est identique au résultat obtenu à l'itération n.

Dans le cas usuel, la formule de la RGD est donnée par l'équation 4 relatée dans [18] :

$$I_{RGD} = \bigvee_{(n \geq 1)} \delta_I^{(n)}(I') \tag{4}$$

$$\delta_I^{(n)}(I') = \delta_I^{(1)} \circ \delta_I^{(1)} \circ \cdots \circ \delta_I^{(1)}(I') \tag{5}$$

$$\delta_I^{(1)}(I') = (I' \oplus B) \wedge I \tag{6}$$

avec

$$\forall p \in I, \qquad I'(p) < I(p) \tag{7}$$

où $\oplus$ correspond à une dilatation morphologique avec l'élément structurant B et $\delta_I^{(1)}(I')$ est le premier rang de dilatation géodésique. $\delta_I^{(n)}(I')$ correspond au résultat de composition (notée par $\circ$) des dilatations géodésiques jusqu'à stabilité (idempotence donnée par n^{eme} rang).

À partir de l'équation 4 et en prenant $\forall p \in I,\ I'(p) = I(p) - 1$, nous pouvons obtenir les maximums locaux $Max_{reg}(I)$ par l'équation 8 :

$$Max_{reg}(I) = \{p \in I / (I - I_{RGD})(p) > 0\} \tag{8}$$

Dans les images naturelles, les maximums locaux ont généralement une différence de niveaux de gris supérieure à 1. Soit H la hauteur d'un maximum régional (pic), l'image *marqueur* I' doit être obtenue en soustrayant une valeur égale à H. Autrement dit, $\forall p \in I,\ I'(p) = I(p) - H$. Nous pouvons facilement noter que la RGD a permis de supprimer les pics locaux de l'image en permettant d'avoir des zones plus homogènes en niveaux de gris. La figure 8 illustre sur une image l'effet de la RGD suivant différentes valeurs de H. Notons que l'effet de la simplification par RGD varie en fonction du paramètre de contraste H et influencera donc, nous le verrons, sur le taux de bonne classification.



(a) Image initiale (b) H=50 (c) H=100 (d) H=150 (e) H=250

FIGURE 8 – Resultats d'une simplification par RGD avec différentes valeurs de H.

5.5 Lissage par filtre de Nagao

Le lissage de NAGAO est basé sur le fractionnement du voisinage du pixel courant en différents domaines. En effet, pour chaque pixel p, un voisinage de taille 5×5 est considéré et 9 domaines distincts sont par la suite définis (cf. figure 9). Pour chaque domaine D_p, la moyenne μ_p et la variance v_p des niveaux de gris sont calculées. Le domaine le plus homogène et ayant la variance la plus faible est finalement retenu. La valeur du pixel d'intérêt (le point central de la fenêtre 5×5) est remplacé par la valeur moyenne des niveaux de gris du domaine sélectionné. L'avantage du filtre de NAGAO revient à réduire le bruit tout en renforçant les contrastes par diminution de la largeur de bande de la zone de transition. La figure 10 illustre sur une image, l'effet du filtre de NAGAO.

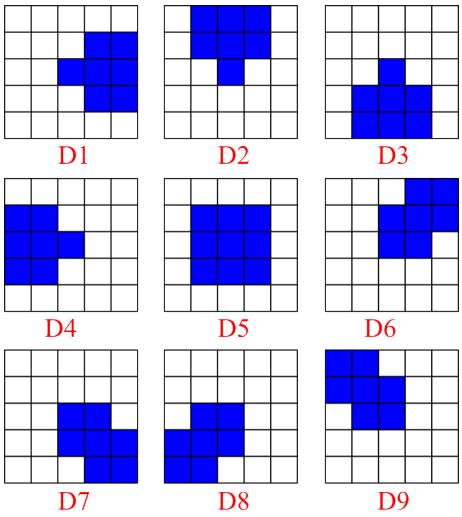
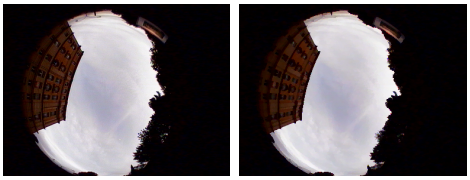


FIGURE 9 – Les 9 domaines de NAGAO.



(a) Image initiale (b) Image simplifiée

FIGURE 10 – Résultat d'une image simplifiée par un filtre de Nagao.

Plusieurs méthodes de simplification d'images ayant été décrites en détails dans cette section, nous allons maintenant nous attacher à rappeler la définition de quelques classifieurs (non)supervisés afin de pouvoir, in fine, classer les régions de (« ciel » et de « non-ciel ») de l'image fisheye. Nous tenons à informer l'auteur qu'il retrouvera tous les tests effectués pour définir la meilleure méthode de simplification et de classification dans la section 7 de ce rapport.

6 Classification pixellaire

Disposant d'une image simplifiée, nous avons maintenant un choix à effectuer entre, d'une part, segmenter l'image en régions d'intérêt pour effectuer ensuite une classification de ces régions en deux classes (« ciel » et « non-ciel ») et, d'autre part, procéder directement à une classification au niveau du pixel. La première solution est très utilisée mais présente l'inconvénient par rapport à la deuxième, d'ajouter une étape supplémentaire et donc d'augmenter le temps de traitement de l'approche globale. L'objectif de la sous-tâche 1.2 n'étant que de distinguer les régions « ciel » des régions « non-ciel », la deuxième solution revenant à effectuer une classification pixellaire en deux classes nous semble suffisante. A contrario, l'extraction et la caractérisation de la texture demandée dans la sous-tâche 1.3 nécessitera d'utiliser cette étape préalable de segmentation.

La problématique de la classification se pose dans différents domaines comme l'extraction de données [19, 20], la compression de données et la quantification vectorielle [21, 22], la reconnaissance et la classification de formes [23, 24], etc. Les méthodes de classification sont généralement classées en différentes familles :

- Les approches basées sur la décomposition/fusion (ou split/merge en anglais) [25],
- Les approches aléatoires [26],
- Les méthodes basées sur les réseaux de neurones [27],
- Les méthodes dédiées aux bases de données de grandes dimensions [28], etc.

Dans la littérature, il existe aussi une catégorisation plus globale des algorithmes de classification à savoir :

- Les approches de classification non-supervisée,
- Les approches de classification supervisée,

Dans ce rapport, nous adopterons cette dernière catégorisation en présentant les algorithmes les plus connus et utilisés de la littérature. Pour plus d'informations sur les différents algorithmes de classification, le lecteur pourra se reporter aux travaux référencés ci-après : [29, 30, 31, 32, 33, 34].

6.1 Méthodes de classification non-supervisée

Dans cette section, nous rappelons brièvement le principe de quatre algorithmes de classification non supervisés : KMLocal (avec ces différentes versions), les c-moyennes floues (ou FuzzyCmeans), Fisher et enfin Statistical Region Merging (SRM).

6.1.1 KMlocal

Cet algorithme reprend les différentes implémentations possibles de l'algorithme de classification des k-moyennes [30]. En effet, ces implémentations ont été découvertes afin de réduire le temps dépensé dans la recherche du plus proche centre de gravité (dans l'algorithme classique k-moyennes). Cet algorithme regroupe donc quatre versions améliorées : Lloyd's, Swap, Hybrid et EZ-Hybrid que nous détaillons ci-après. Mais avant cela, nous commençons par rappeler la définition de base de l'algorithme des k-moyennes. La classification par l'algorithme des k-moyennes a été proposée par MaCQueen [35] et consiste à regrouper un ensemble de points initial P formé par N points, de dimension d, en K sous-ensembles (que nous appellerons par la suite « classes »).

$$P = \{p_n\}_{n=1...N} \tag{9}$$

tel que :

$$p_n \in \Re^d, \forall n = 1 \dots N \tag{10}$$

Chaque classe C_k possède un centre w_k qui correspond concrètement au centre de gravité de chaque sous-ensemble. L'algorithme des k-moyennes consiste à minimiser la distance quadratique moyenne entre chaque élément p_n appartenant à un sous-ensemble k et son centre w_k afin d'associer cet élément à la classe dont le centre de gravité est le plus proche. Il s'avère que la recherche du centre le plus proche est l'étape la plus complexe au vu de la difficulté de trouver un minimum global. Cette complexité est résolue par une autre version de l'algorithme des k-moyennes connue sous le nom de Lloyd's et qui vise à chercher les minimums locaux de l'équation donnée par (11). Cette version a été introduite par [36] et est utilisée par différents auteurs [37, 29].

$$argmin_{1 \leq k \leq K} \|p_n - w_k\|^2 \tag{11}$$

Dans ce qui suit, nous détaillons les quatre versions de l'algorithme KMlocal :

- **Lloyd's** est souvent confondu avec l'algorithme des k-moyennes. Il se base sur un processus itératif dont le but est de trouver les K centres donnant la meilleure classification possible. Il commence par sélectionner K centres initiaux choisis aléatoirement parmi les N points. Il répète ensuite la recherche des voisins les plus proches à chaque centre w_k, ce qui permet de former les sous-ensembles initiaux. L'étape suivante consiste à chercher les nouveaux centres de gravité. Ces deux étapes seront répétées jusqu'à atteindre un nombre maximal d'itérations fixé a priori ou bien validé par une condition de convergence comme par exemple la stabilité des centres. Il s'avère que la qualité de la classification résultante dépend du premier choix intuitif des centres de gravité et du nombre maximal d'itérations.

- **Swap** est un algorithme qui intervient au niveau de la phase de changement des centres. En effet, cet algorithme sélectionne initialement un sous ensemble de K centres initiaux parmi les centres candidats. Il vise, ensuite, à échanger un par un les centres en comparant le résultat de classification en fonction d'un critère mesurant la distorsion. Soient Ω l'ensemble des centres candidats et Ψ un sous-ensemble de Ω. Swap remplace un centre $\psi \in \Psi$ par un nouveau candidat $\psi' \in \Omega - \Psi$ afin d'obtenir un nouveau sous-ensemble de centres $\Psi' = \Psi - \{\psi\} \cup \{\psi'\}$. Nous remplaçons Ψ par Ψ' si la nouvelle solution réduit le critère de distorsion. L'algorithme converge quand le critère de distorsion est inchangé.

- **Hybrid** est un algorithme qui combine les deux algorithmes Lloyd's et Swap avec des appariements variés.

- **EZ-Hybrid** est une version hybride moins complexe que la version hybride totale. EZ-Hybrid applique une première fois la version Swap suivie d'un ensemble d'itérations de l'algorithme Lloyd's. La version Lloyd's est utilisée jusqu'à avoir la RDL (Relative Distorsion Loss) donnée par l'équation (12) inférieure à un seuil donné.

$$RDL = \frac{DistorsionInitiale - DistorsionCourante}{DistorsionInitiale} \tag{12}$$

6.1.2 Fuzzy Cmeans

L'algorithme FuzzyCmeans (ou C-moyennes floues en français) a été développé par [38] puis amélioré par [39]. Il s'approche dans son principe de l'algorithme des k-moyennes et vise à classer un ensemble de N points en C classes. Il minimise un critère noté J_m qui est la somme quadratique pondérée (par des cœfficients d'appartenance μ_{nk}) d'un point p_n à une classe c_k. Ce critère quadratique est donné par l'équation (13) où $m \geq 1$ désigne le degré de flou.

$$J_m = \sum_{n=1}^{N} \sum_{k=1}^{C} \mu_{nk}^m \|p_n - c_k\|^2 \tag{13}$$

Les cœfficients d'appartenance μ_{nk} forment la matrice d'appartenance notée U et donnée par l'équation (14). Chaque terme de cette matrice d'appartenance est défini par l'équation (15) et possède une valeur comprise dans l'intervalle $[0, 1]$.

$$U_{N \times C} = \{\mu_{nk}^m\}_{\substack{n = 1 \ldots N \\ k = 1 \ldots C}} \tag{14}$$

$$\mu_{nk}^m = \frac{1}{\sum_{j=1}^{C} (\frac{\|p_n - c_k\|}{\|p_n - c_j\|})^{\frac{2}{m-1}}} \tag{15}$$

avec c_k donné par l'équation (16)

$$c_k = \frac{\sum_{n=1}^{N} \mu_{nk}^m \cdot p_n}{\sum_{n=1}^{N} \mu_{nk}^m} \tag{16}$$

L'algorithme des C-moyennes floues débute par une initialisation de la matrice d'appartenance $U^{(0)}$. La deuxième étape consiste à calculer à l'aide de l'équation (16) les centres c_k. Dans la troisième étape, la nouvelle matrice d'appartenance $U^{(j+1)}$ est calculée à partir de $U^{(j)}$ en utilisant l'équation (15). La condition d'arrêt est définie par l'inégalité $\|U^{(j+1)} - U^{(j)}\| < \varepsilon$.

6.1.3 Fisher

L'algorithme de Fisher fait partie des algorithmes de classification non-supervisée et à la particularité de ne s'appliquer que sur des images en niveaux de gris [40, 41]. L'idée consiste à diviser l'histogramme de l'image en niveaux de gris en K classes en minimisant la somme des variances des classes. En effet, la méthode de Fisher consiste à localiser des seuils représentant les séparateurs de la distribution des données. Soient s_k le seuil associé à la classe k et m_k la moyenne des intensités des pixels appartenant à la classe k. La répartition des pixels en K classes revient à localiser les seuils $\{s_k\}_{k=1...K}$ comme des séparateurs de distributions en utilisant un critère qui minimise la somme des inerties associées aux différentes classes.

$$Histog(s_k) = min_{p\in]m_k,m_{k+1}[}(Histog(p)) \tag{17}$$

A chaque classe k, on associe une taille notée T_k (18) et une variance notée V_k (19)

$$T_k = \sum_{p\in D_k} \{Histog(p)\} \tag{18}$$

$$V_k = \frac{1}{T_k} \sum_{p\in D_k} (p - m_k)^2 . Histog(p) \tag{19}$$

$D_k =]s_{k-1}, s_k[$ est la gamme des niveaux de gris de la classe k. Le critère de minimisation est donné par l'équation (20) :

$$W = \sum_{k=1}^{K} T_k . V_k \tag{20}$$

où m_k représente la moyenne pondérée donnée par l'équation (21)

$$m_k = \frac{1}{T_k} . \sum_{p\in D_k} p . Histog(p) \tag{21}$$

6.1.4 Statistical Region Merging (SRM)

SRM est une approche de segmentation/classification basée sur une nouvelle disposition des pixels dans l'image. En effet, ils existent une multitude d'approches basées sur la fusion de régions afin de segmenter/classer l'image [42] ; SRM a été introduit par F.Nielson et R.Nock dans [31] et a été amélioré dans les travaux [32, 43]. Cet algorithme se base sur la définition d'un prédicat de fusion des régions et d'un certain ordre de fusion. L'algorithme commence par définir les couples de pixels adjacents qu'on note (p, p'). Soit S_I l'ensemble qui regroupe ces couples de pixels et I l'image à classifier. Une fonction $f(p, p')$ est définie afin d'ordonner S_I dans un ordre croissant. L'ensemble S_I (maintenant ordonné), sera parcouru une seule fois en vérifiant le prédicat de fusion. En effet, pour chaque couple $(p, p') \in S_I$ de label $L(p) \neq L(p')$ différent, le prédicat

de fusion courant $P(L(p), L(p'))$ est vérifié. La fusion de région n'est donc effectuée que si et seulement si $P(L(p), L(p'))$ est vrai (cf. équation (22)).

$$P(L(p), L(p')) = \begin{cases} vrai & \text{si} |\bar{L'} - \bar{L}| \leq \sqrt{b^2(L) + b^2(L')} \\ faux & \text{sinon} \end{cases} \tag{22}$$

où $\bar{L}$ correspond à la valeur moyenne du niveau de gris de la région L et $b(L)$ donné par l'équation (23)

$$b(L) = g\sqrt{(\frac{1}{(2Q|L|)}) \ln(|\Re_l|/\delta)} \tag{23}$$

où $\Re_l$ est l'ensemble des régions ayant l pixels, δ est une constante, Q est le paramètre de quantification de complexité statistique de l'image classée, et $\{1, 2, \ldots, g\}$ est l'ensemble des valeurs de niveau de gris. Dans la pratique $g = 256$ [32].

6.2 Méthodes de classification supervisée

Dans cette section, nous détaillons trois algorithmes de classification supervisés communément utilisés à savoir : Bayes, les K-plus proches voisins (Kppv) et les Séparateurs à Vaste Marges (SVM). Ces classifieurs dits supervisés nécessitent donc la création d'une base d'apprentissage en plus de la base de tests.

6.2.1 K-Plus proches voisins (Kppv)

L'algorithme de classification des K-plus proches voisins (Kppv) appartient à l'ensemble des méthodes de classification par apprentissage. Cet algorithme commence par chercher les K plus proches voisins en utilisant une mesure de distance (la distance euclidienne est très souvent utilisée). Ensuite, nous calculons l'occurrence de chaque classe en se basant sur les résultats d'apprentissage. La dernière étape consiste à attribuer le point courant p à la classe c_i la plus fréquente. Une première formalisation du problème a été présentée dans [44] sous la forme de l'équation (24) rappelée ci-dessous.

$$f(p) = c_i \quad avec \quad OCC(c_i) > OCC(c_j) \quad \forall i \neq j \in \{1, 2, \ldots, C\} \tag{24}$$

où OCC est l'opérateur définissant le nombre d'occurrences de c_i parmi les labels des K plus proches voisins de p. Une deuxième formalisation de l'algorithme des Kppv se base non plus sur l'occurrence mais sur l'estimation d'une densité de probabilité donnée par l'équation (25). Cette équation traduit la densité de probabilité d'avoir la classe c_i au point p. x_i est le nombre de points ayant le label de la classe c_i parmi les K plus proches voisins, n_i est le cardinal de c_i et $V(p)$ est la plus petite boule contenant les K plus proches voisins de p.

$$r_i = \frac{x_i(p)}{n_i \times V(p)} \tag{25}$$

6.2.2 Bayes

L'algorithme de classification de Bayes est un algorithme supervisé paramétrique. Il est basé sur le modèle Bayésien qui repose sur un concept stochastique. La théorie Bayésienne a été introduite par Thomas Bayes en 1764 [45] et consiste, dans la pratique, à modéliser les classes selon les lois gaussiennes. Comme cela est rappelé dans [46], pour un pixel donné x, la classe optimale contenant ce pixel est donnée par la maximisation du critère défini par l'équation (26) :

$$f(x, C_i) = -\frac{1}{2}(x - \mu_i)^T \sum_i^{-1}(x - \mu_i) - \frac{1}{2}\log|\sum_i| - \log p_i + \frac{k}{2}\log 2\pi \quad (26)$$

où k représente le nombre de classes, μ_i correspond à la moyenne des éléments de la classe C_i, $\sum_i$ est la matrice de variance-covariance et p_i est la probabilité à priori de la classe C_i.

6.2.3 Les Supports Vecteurs Machines

Les « Séparateurs à Vaste Marges » (SVM), connu sous le nom de « Supports Vecteurs Machine » en anglais, est un classifieur supervisé qui fut introduit par Vapnik [47] et qui est très utilisé pour la classification de données de grandes dimensions. Tout comme les deux classifieurs précédents, les SVM font partie de la famille des classifieurs supervisés, et requièrent donc une phase d'apprentissage permettant d'apprendre les paramètres du modèle de classification approprié pour l'identification des objets d'intérêts. Les SVM reposent sur l'utilisation d'une fonction noyau (kernel) permettant la séparation optimale (vaste marge) des données [48]. Il existe dans la littérature plusieurs fonctions noyaux et définition du critère de similarité [28, 49]. A titre d'exemple, Lafarge et al. présentent dans leurs travaux [28] un noyau adaptatif tenant compte des informations de texture et radiométriques dans le but de reconnaître les zones de feu dans les images SPOT5 de la région PACA. Néanmoins, la fonction noyau gaussien reste la plus utilisée et est donc rappelée ci-dessous (27).

$$k(x_i, x_j) = exp(\frac{-\|x_i - x_j\|^2}{2\sigma^2}) \quad (27)$$

7 Premiers résultats sur la classification des régions de ciel/non-ciel et sur la détection de la ligne d'horizon

Maintenant que nous avons présenté différentes méthodes de simplification d'images et quelques uns des classifieurs (non-supervisés et supervisés) les plus couramment utilisés, nous allons pouvoir tester notre approche dans son ensemble et définir les meilleurs paramètres des algorithmes. Comme indiqué sur la figure 4, nous pouvons évaluer en sortie de l'étape de post-traitement morphologique ou de classification pixellaire (pour que le choix des paramètres ne soit pas influencé par cette étape de post-traitement), la qualité de la détection des zones de « ciel » et de « non-ciel » de l'image. Pour ce faire, nous avons procédé à une série de combinaisons « méthode de simplification utilisée / classifieur utilisé » afin de définir les paramètres optimaux et le meilleur couple à utiliser. Dans un premier temps, les tests ont été effectués sur la base d'images du projet LOCOPROL afin d'identifier les deux meilleurs méthodes de simplification. Dans un deuxième temps, nous avons ré-itéré une série de tests mais cette fois sur la base d'images du projet CAPLOC (disposant d'un grand nombre d'images hétérogènes) afin de définir les deux meilleurs couples « méthode de simplification utilisée / classifieur utilisé ». Notons que les résultats ont été évalués avec une méthode d'évaluation adaptée à notre application et utilisant une vérité-terrain.

7.1 Méthode d'évaluation de la qualité de la classification

Les résultats de classification présentés dans ce rapport sont effectués sur quelques images des projets LOCOPROL (8 images) et CAPLOC (149 images). La méthode d'évaluation utilisée est composée de cinq mesures décrites ci-dessous :

1. le pourcentage de pixels faisant partie de la région « ciel » dans l'image classée et dans l'image de référence ;

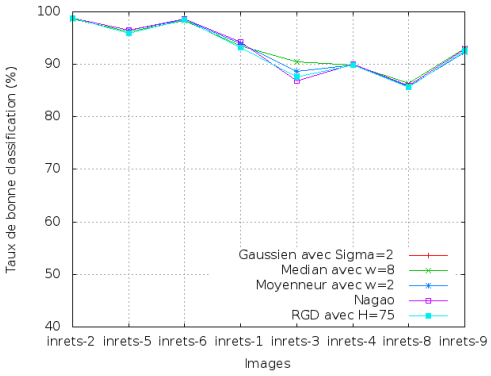
2. le pourcentage de pixels classés comme « ciel » dans l'image classée et ne faisant pas partie de la région « ciel » dans l'image de référence ;
3. le pourcentage de pixels classés comme « non-ciel » dans les deux images ;
4. le pourcentage de pixels classés comme « non-ciel » dans l'image classée et qui ne font pas partie de la région « ciel » dans l'image de référence ;
5. la somme de la première et de la troisième mesure. Cette dernière mesure traduit aussi le pourcentage de pixels correctement classés (appelée « taux de bonne classification » dans la suite de ce rapport).

7.2 Résultats de classification pixellaire ciel/non ciel

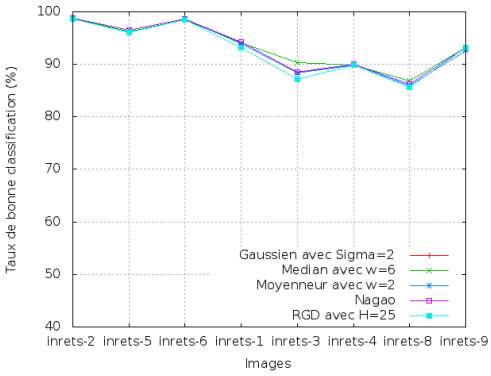
Comme indiqué précédemment, nous allons dans cette section, présenter les résultats de classification permettant d'une part et à partir des images de la base du projet LOCOPROL, d'identifier les deux meilleurs méthodes de simplification d'images (cf. figures 11, 12, 13 et tableau 1), et d'autre part, de définir le meilleur couple « méthode de simplification/classifieur » à partir des images de la base du projet CAPLOC (cf. figures 14, 15, 16 et tableau 2). Il est important de noter que le choix du couple finalement retenu tiendra aussi compte du temps de traitement de l'approche globale (cf. tableau 3).

Les figures 11, 12 et le tableau 1 illustrent les « taux de bonne classification » (comme défini dans la section précédente) obtenus avec différentes combinaisons possibles et faisant intervenir différentes méthodes de simplification (filtres médian, gaussien, moyenneur, RGD, nagao) et différents classifieurs (KMlocal, FuzzyCmeans, Fisher, SRM, Bayes, Kppv, SVM). Nous informons le lecteur que nous avons fait le choix de ne présenter sur ces figures que les différences notables. Par exemple, deux versions du KMlocal sont illustrées ici, les deux autres donnant des résultats quasiment identiques. De la même manière, nous montrons les résultats du classifieur FuzzyCmeans avec $m = 2$ seulement, les autres valeurs de m n'offrant pas de résultats significativement différents. Malgré tout, nous tenons à signaler que tous les résultats figurent dans le tableau 1.

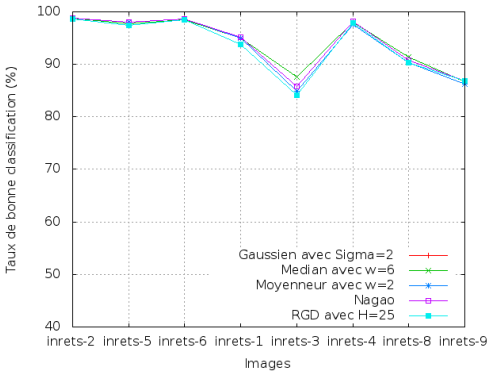
Pour ce qui est des classifieurs non-supervisés (cf. figure 11), nous pouvons conclure que les résultats obtenus avec KMlocal et FuzzyCmeans sont très proches mais que le filtre médian se départage pour une image. Pour SRM, les résultats se départagent pour deux images et mettent donc en avant le filtre de Nagao. Pour Fisher, les résultats sont sensiblement identiques, mais le filtre médian offre, là encore, un meilleur taux pour une image donnée. Il est aussi intéressant de noter que l'utilisation de Fisher sur la composante bleue de l'espace RGB fournit les meilleurs résultats. En ce qui concerne les classifieurs supervisés (cf. figure 12), nous pouvons constater que les résultats de classification sont plus hétérogènes. Néanmoins, la méthode de simplification qui offre dans la grande majorité des cas, le meilleur « taux de bonne classification », quel que soit le classifieur utilisé est la RGD. Nous constatons aussi l' importance du choix des paramètres en illustrant que les résultats de classification pour le classifieur des Kppv sont sensiblement différents, en terme de pourcentage (parfois près de 6%) pour un $nb_{Kppv} = 4$ ou pour un $nb_{Kppv} = 6$.



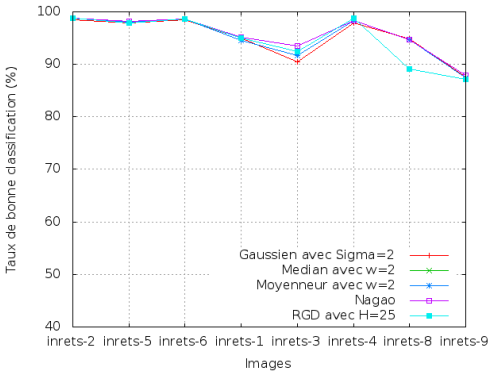
(a) KMlocal (Swap)



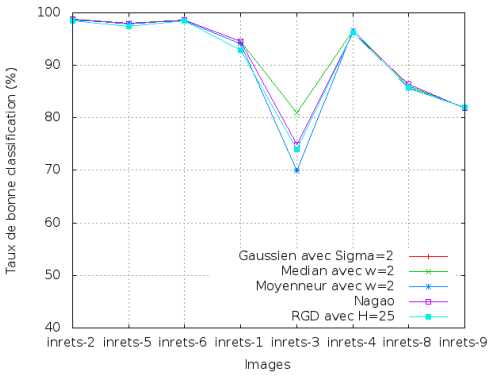
(b) KMlocal (EZHybrid)



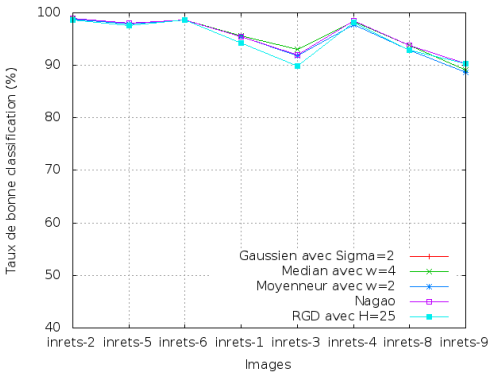
(c) FuzzyCmeans (m=2)



(d) SRM

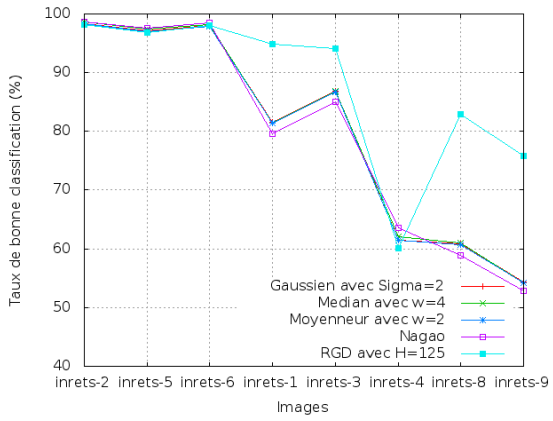


(e) Fisher (composante rouge)

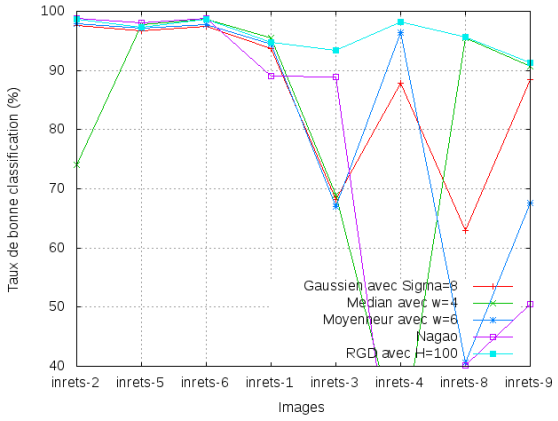


(f) Fisher (composante bleue)

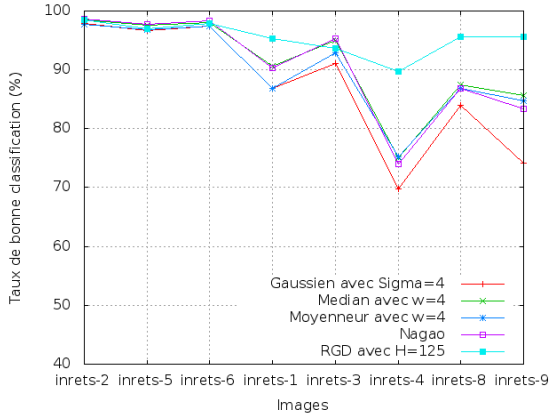
FIGURE 11 – Résultats de la classification ciel/non-ciel avec différents algorithmes non-supervisés en fonction de la méthode de simplification utilisée (sur la base LOCOPROL).



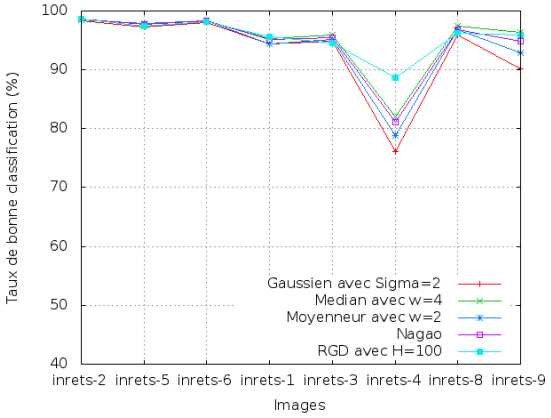
(a) Bayes



(b) SVM



(c) Kppv (nb Kppv=4)



(d) Kppv (nb Kppv=6)

FIGURE 12 – Résultats de la classification ciel/non-ciel avec différents algorithmes supervisés en fonction de la méthode de simplification utilisée (sur la base LOCOPROL).

Nous pouvons alors conclure en s'appuyant sur les résultats du tableau 1 qu'une simplification effectuée par un filtre médian suivi d'une classification faite par l'algorithme des KMlocal, FuzzyCmeans ou encore Fisher donnent de très bons résultats. Le meilleur taux de bonne classification obtenu étant de 95,7% avec l'algorithme de Fisher sur la composante bleue. Il en est de même pour une simplification faite par une Reconstruction Géodésique par Dilatation (RGD) suivie d'une classification effectuée par l'algorithme des Kppv, Bayes ou SVM. Le meilleur taux de bonne classification obtenu étant cette fois de 96% avec les SVM. Pour une meilleure visualisation des résultats, le lecteur pourra se reporter à la figure 13 illustrant pour chaque image, la classification ciel (en bleu) / non-ciel (en marron) obtenue avec la meilleure méthode de simplification pour chaque classifieur présenté. En conclusion, nous retenons le filtre médian et la RGD comme les deux méthodes de simplification à utiliser pour la deuxième série de tests effectuée sur la base d'images du projet CAPLOC.

		Filtres									
		Gaussien		Median		Moyenneur		RGD		Nagao	
		Performance	σ	Performance	W	Performance	W	Performance	H	Performance	
Classification non-supervisée											
Fisher	Rouge	90.45	2	91.96	2	90.45	2	90.67	25	91.16	
	Vert	94.43	2	94.9	4	94.43	2	94.1	50	94.65	
	Bleue	95.24	2	95.7	4	95.24	2	95.04	25	94.65	
	Moyenne RGB	93.64	2	94.18	6	93.64	2	93.3	25	93.92	
KMlocal	EZHybrid	93.06	2	93.44	6	93.06	2	92.75	50	93.27	
	Hybrid	93.09	2	93.46	6	93.09	2	92.78	25	93.27	
	Loyds	93.05	2	93.44	6	93.05	2	92.79	25	93.27	
	Swap	93.05	2	93.31	8	93.04	2	92.75	75	92.98	
Fuzzy Cmeans	m=1	93.74	2	94.29	6	93.74	2	93.46	25	94.04	
	m=1.25	93.74	2	94.27	6	93.74	2	93.46	25	94.05	
	m=1.5	93.73	2	94.26	6	93.73	2	93.46	25	94.05	
	m=1.75	93.72	2	94.25	6	93.72	2	93.45	25	94.05	
	m=2	93.70	2	94.23	6	93.70	2	93.45	25	94.04	
SRM		95.09	2	95.26	2	95.26	2	94.75	25	95.66	
Classification supervisée											
Bayes		79.8	2	79.98	8	79.71	2	87.58	125	79.38	
Kppv	nbKppv = 4	87.23	4	90.28	4	89.86	4	95.4	125	90.56	
	nbKppv = 6	93.13	2	95.22	4	93.79	2	95.63	100	94.76	
SVM		86.6	8	80.89	4	82.34	6	96.0	100	72.03	

TABLE 1 – Résultats de la classification ciel/non-ciel avec différents algorithmes, paramètres et méthodes de simplification, sur la base LOCO-PROL.

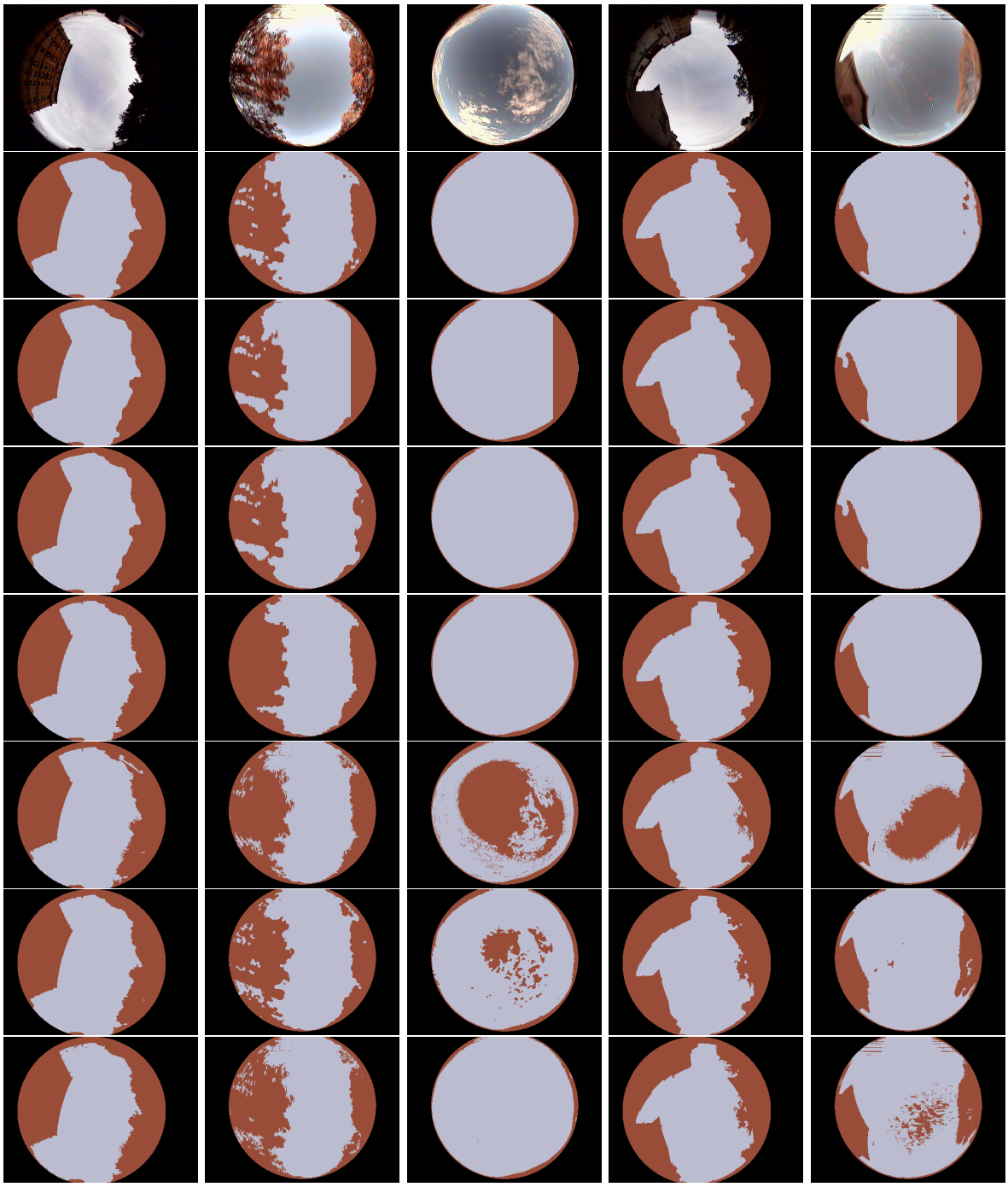
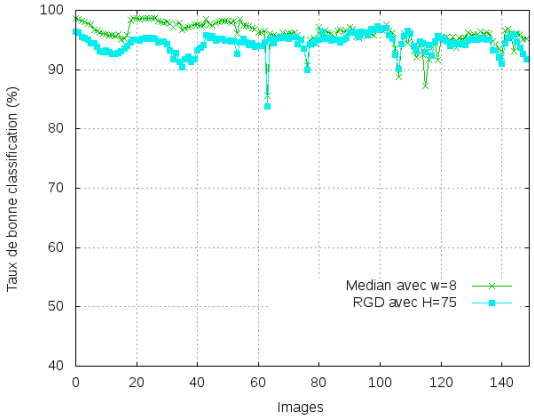


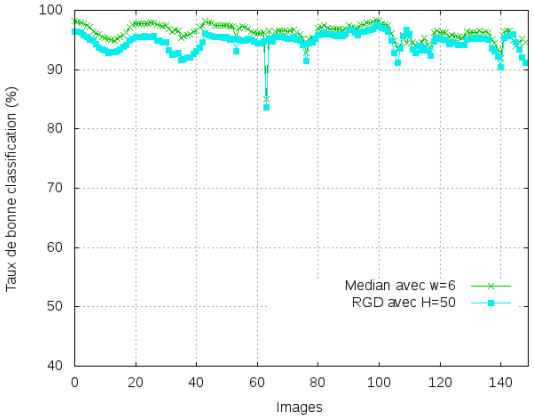
FIGURE 13 – Résultats de classification ciel/non-ciel sur 5 images de la base LOCOPROL (de haut en bas : Fisher (bleue), KMlocal (hybrid), FuzzyCmeans (m=1), SRM, Bayes, Kppv (nb Kppv=6), SVM).

Cette première série de tests ayant permis d'identifier les deux meilleurs méthodes de simplification, nous avons ensuite chercher à définir le meilleur couple « méthode de simplification/classifieur » à partir des images de la base du projet CAPLOC. Les figures 14 et 15 illustrent les taux de bonne classification obtenus en fonction du classifieur utilisé. Pour les classifieurs non-supervisés, nous pouvons remarquer que les résultats sont sensiblement identiques hormis pour Fisher (sur la composante rouge) et SRM qui montrent quant à eux une très nette amélioration avec l'utilisation de la RGD (jusqu'à 24%). Pour les classifieurs supervisés, les résultats sont eux

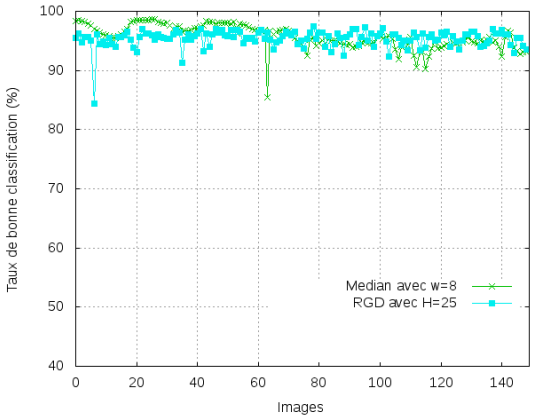
aussi très proches (hormis pour les SVM), et montrent que la RGD offre dans tous les cas des résultats supérieurs à ceux obtenus avec un filtre médian.



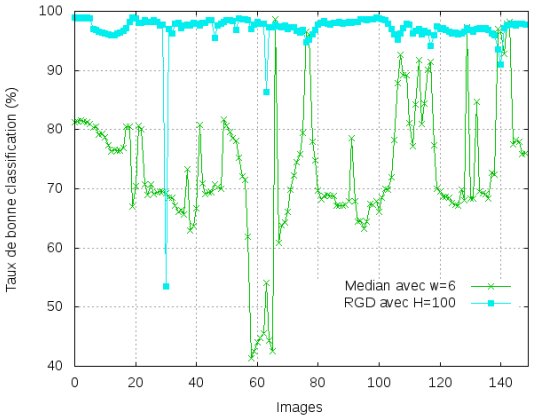
(a) KMlocal (Swap)



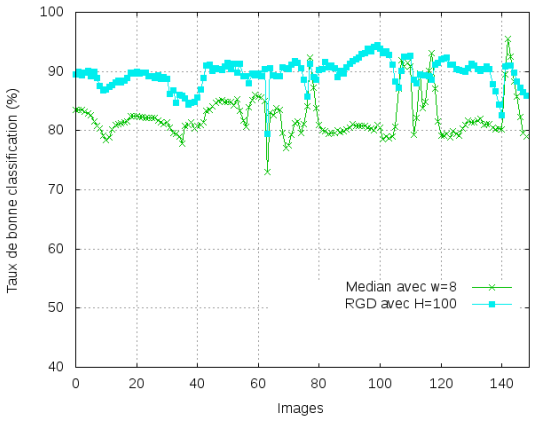
(b) KMlocal (EZHybrid)



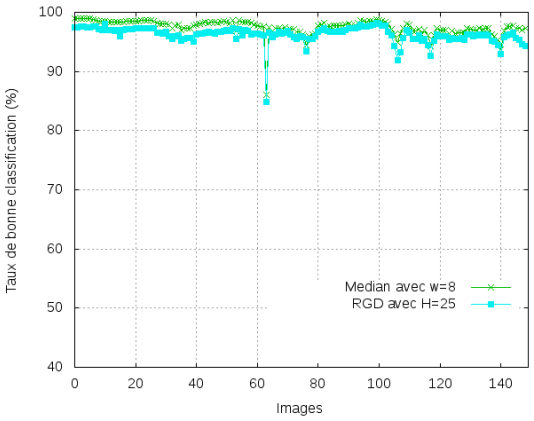
(c) FuzzyCmeans (m=2)



(d) SRM

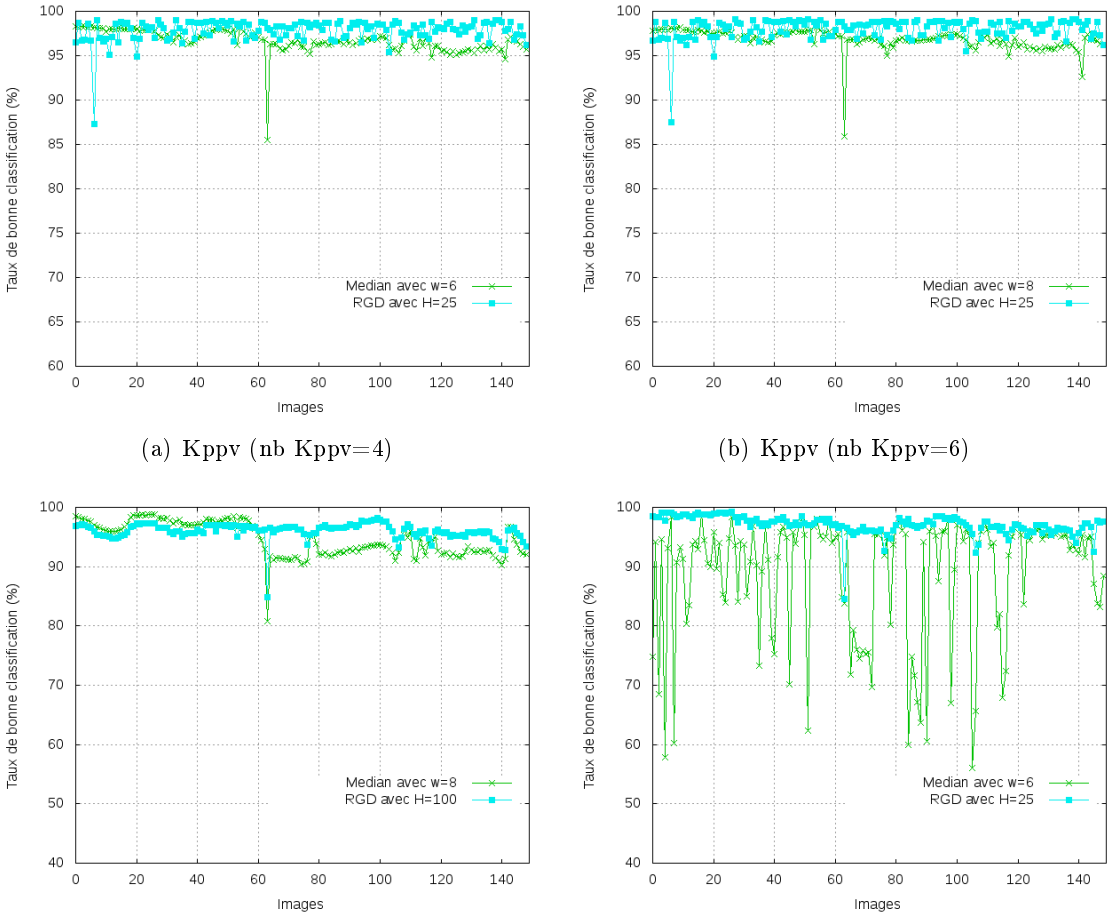


(e) Fisher (composante rouge)



(f) Fisher (composante bleue)

FIGURE 14 – Résultats de la classification ciel/non-ciel avec différents algorithmes non-supervisés en fonction de la méthode de simplification utilisée (sur la base CAPLOC).



(a) Kppv (nb Kppv=4)

(b) Kppv (nb Kppv=6)

(c) Bayes

(d) SVM

FIGURE 15 – Résultats de la classification ciel/non-ciel avec différents algorithmes supervisés en fonction de la méthode de simplification utilisée (sur la base CAPLOC).

Nous pouvons alors conclure en s'appuyant sur les résultats du tableau 2 que dans un cadre non-supervisé, une classification réalisée par l'algorithme de Fisher (sur la composante bleue de l'espace RGB) donne les meilleurs résultats (i.e. 97,53%) si elle est couplée à une simplification par un filtre médian et arrive en deuxième position (derrière SRM) si elle est associée à une simplification par RGD (i.e. 96,40%). Dans un cadre supervisé, l'utilisation d'une RGD offre toujours les meilleurs résultats, mais son utilisation couplée à l'algorithme des Kppv (avec un $nb_{Kppv} = 6$) permet d'obtenir le meilleur taux de bonne classification à savoir 97.99%. Là encore, pour une meilleur visualisation, la figure 16 illustrant pour chaque image, la classification ciel (en bleu) / non-ciel (en marron) obtenue avec la meilleure méthode de simplification pour chaque classifieur présenté. Le tableau 3 illustre quant à lui, le temps de traitement de l'approche en fonction du couple « méthode de simplification/classifieur » utilisé. Nous remarquons que le temps de traitement de l'approche proposée est peu affecté par la méthode de simplification utilisée, mais peut être jusqu'à 400 fois supérieur selon le classifieur employé (par exemple entre Fisher et Kppv). Devant considérer la contrainte temps-réel dans la sous-tâche 1.2, nous retiendrons finalement l'algorithme de Fisher (et la composante bleue) comme meilleur classifieur. Une première préférence est donnée au filtre médian par rapport à la RGD mais reste à confirmer sur les images (beaucoup plus nombreuses) de la prochaine campagne d'acquisition du projet CAPLOC.

		Filtres			
		Median		RGD	
		Performance	W	Performance	H
Classification non-supervisée					
Fisher	Rouge	82.27	8	89.60	100
	Vert	91.72	8	93.49	50
	Bleue	97.53	8	96.40	25
	Moyenne RGB	92.76	8	93.55	50
KMlocal	EZHybrid	96.39	8	95.20	25
	Hybrid	96.39	8	95.20	25
	Loyds	96.39	8	95.20	25
	Swap	96.06	8	95.44	25
Fuzzy Cmeans	m=1	96.44	8	95.25	25
	m=1.25	96.35	8	95.28	25
	m=1.5	96.16	8	95.36	25
	m=1.75	95.93	8	95.38	25
	m=2	95.73	8	95.38	25
SRM		73.18	6	97.02	100
Classification supervisée					
Bayes		94.61	8	96.02	100
Kppv	nbKppv = 4	96.68	6	97.83	25
	nbKppv = 6	96.83	8	97.99	25
SVM		88.81	6	97.07	25

TABLE 2 – Résultats de la classification ciel/non-ciel avec différents algorithmes, paramètres et méthodes de simplification, sur la base CAPLOC.

-	Fisher	KMlocal	FuzzyCmeans	SRM	Bayes	Kppv	SVM
Median (en s)	0.035	0.41	3.30	0.35	0.34	15.3	5.0
RGD (en s)	0.037	0.45	3.34	0.47	0.47	15.4	5.04

TABLE 3 – Temps de traitement de l'approche en fonction du couple classifieur/simplification utilisé

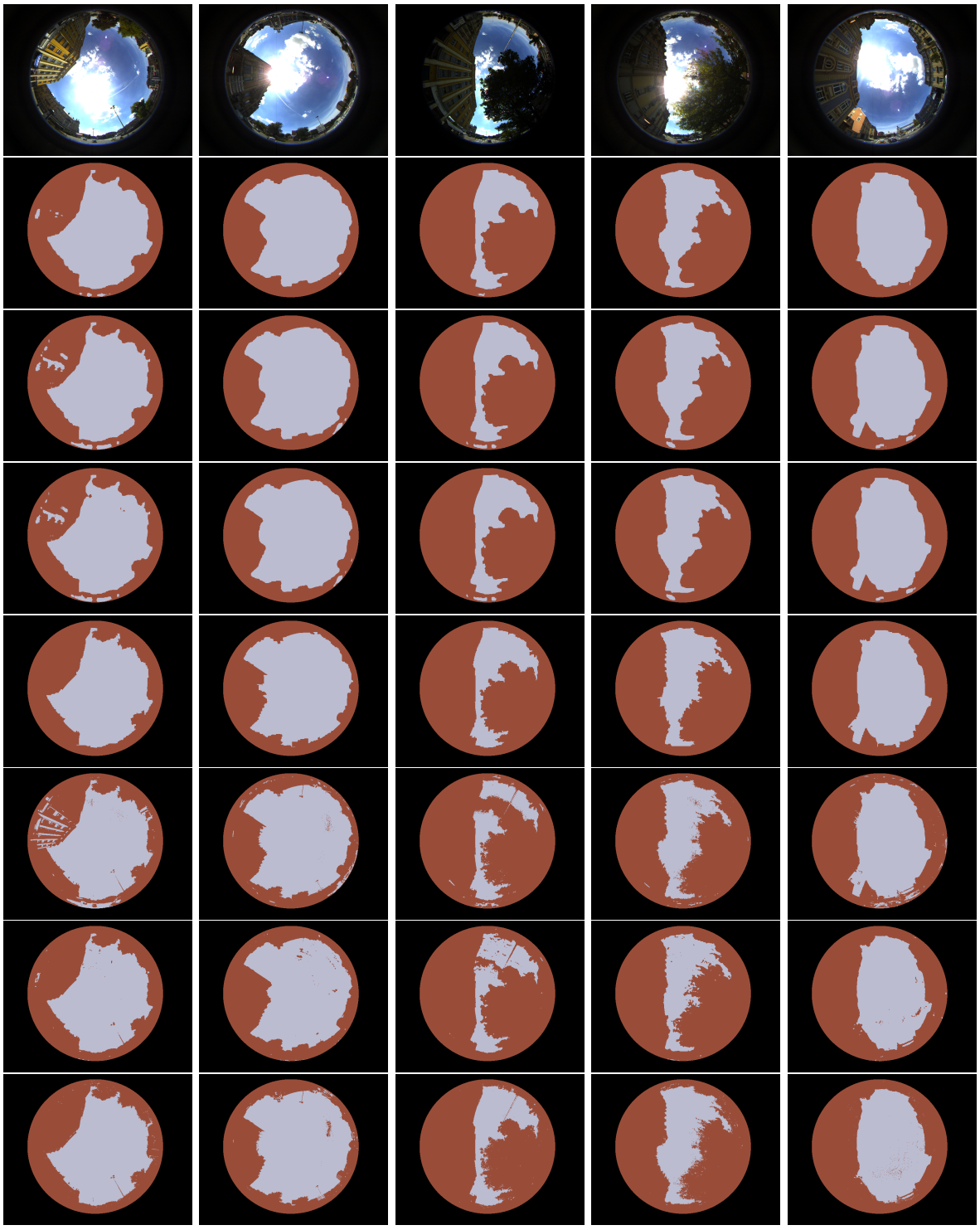


FIGURE 16 – Résultats de classification ciel/non-ciel sur 5 images de la base CAPLOC (de haut en bas : Fisher (bleue), KMlocal (hybrid), FuzzyCmeans (m=1), SRM, Bayes, Kppv (nb Kppv=6), SVM).

7.3 Repositionnement des satellites et détection de la ligne d'horizon

Une fois l'étape de classification pixellaire effectuée, nous cherchons à déterminer le nombre de satellites dont les signaux sont reçus en trajets directs (respectivement réfléchis/bloqués). Pour

ce faire, nous re-projetons les satellites dans l'image classée, et comptons le nombre de satellites se trouvant dans une région de « ciel » (respectivement dans une région de « non-ciel »). Pour cette raison, nous détaillons ci-dessous, la méthodologie permettant de replacer les satellites dans l'image. Celle-ci se décompose en trois sous-étapes. i) l'acquisition des données ; ii) le calibrage et la mise en forme des données ; iii) l'utilisation du logiciel STK.

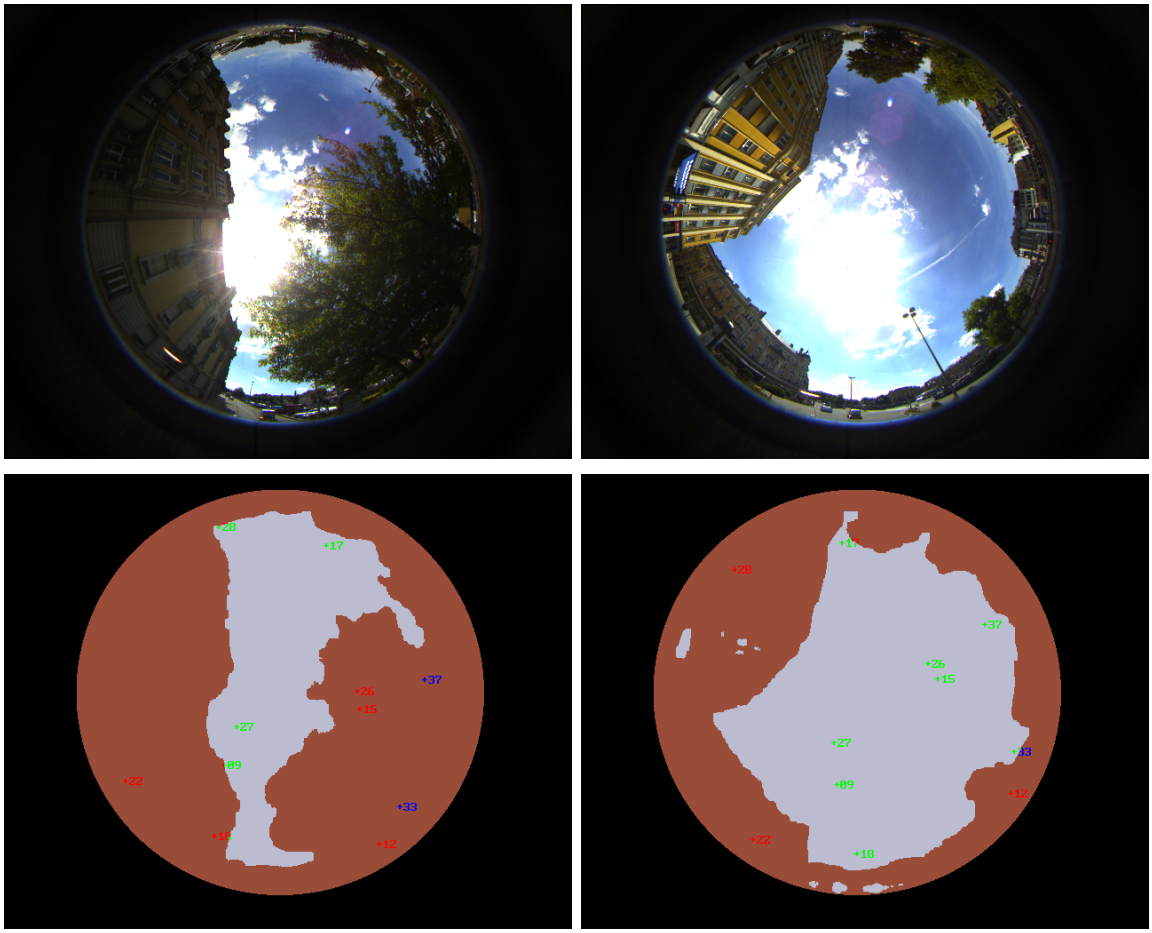
1. L'acquisition des données : Le dispositif d'acquisition d'images est constitué d'une caméra mono-CCD couleur équipée d'un objectif fisheye (grand angle) fixée sur le toit d'un véhicule de laboratoire et orientée vers le ciel. La fréquence d'acquisition des images est cadencée par un odomètre autorisant un échantillonnage régulier de l'environnement. En parallèle, un enregistrement des trames NMEA issues d'un récepteur Septentrio est effectué.

2. Le calibrage et la mise en forme des données : L'utilisation d'un objectif grand angle a l'avantage de disposer d'un angle de vision important mais présente aussi l'inconvénient d'avoir des phénomènes de distorsion. Cette aberration liée aux propriétés optiques des lentilles de l'objectif fisheye se caractérise par une déformation de l'image. Un calibrage, basé sur une mise en correspondance des points du monde réel avec les pixels de l'image est nécessaire pour corriger cette déformation. Il est, en effet, indispensable de positionner les satellites dans chaque image avec une précision de l'ordre du pixel. Deux pistes ont été suivies pour atteindre nos objectifs. La première s'appuie sur une méthode mathématique intégrant le modèle géométrique d'une caméra et ses paramètres intrinsèques. Cette méthode, basée sur l'algorithme de Zhai implémenté dans OpenCV a montré ses limites pour un objectif offrant un champ de vison aussi élevé. La seconde méthode est en partie expérimentale. Le dispositif mis en place en laboratoire, basé sur un système de mires, nous a permis de répondre à nos attentes avec la précision escomptée. Les trames GPGGA extraites des données NMEA nous informent sur la position du véhicule toutes les secondes. Les informations temporelles associées à chaque système d'acquisition nous permettent de recaler les données de localisation et les images.

3. L'utilisation du logiciel STK : Pour une trajectoire connue à une date donnée, le logiciel STK va générer la position d'une ou de plusieurs constellations de satellites le long de ce trajet. Dans le cadre de notre étude, le fichier de sortie obtenu contient les informations d'évolution de la position de chaque satellite des constellations GPS et EGNOS. Ces informations sont replacées dans chaque image.

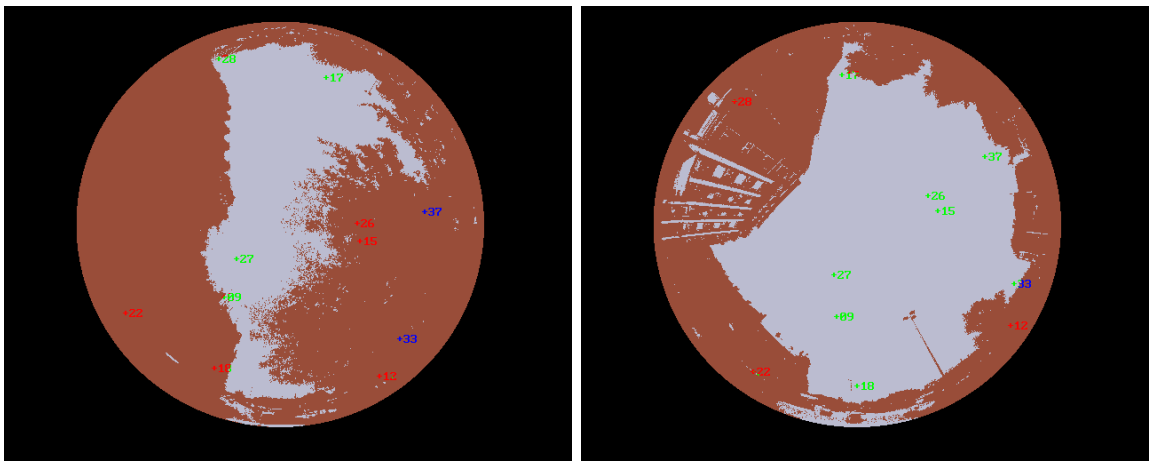
Maintenant que nous avons défini comment replacer les satellites dans l'image fisheye, il est relativement facile de repositionner les satellites dans l'image classée, comme l'illustre la figure 17. Les satellites se situant dans une région de « ciel » sont représentés par une couleur verte, tandis que ceux qui se situent dans une région de « non-ciel » sont représentés par une couleur rouge (pour la constellation GPS) ou bleue (pour la constellation EGNOS). Nous pouvons ensuite aisément identifier les satellites avec des trajets directs (en vert) et ceux dont les signaux sont bloqués ou réfléchis (en rouge ou bleu). Le tableau 4 illustre ce point et montre clairement l'importance du choix du couple « méthode de simplification/classifieur » puisque nous constatons que suivant la méthode utilisée, les satellites identifiés comme LOS (Line-Of-Sight) et NLOS (Non-Line-Of-Sight) ne sont pas les mêmes. Là encore, nous justifions l'intérêt de la prochaine campagne d'acquisition pour valider la stratégie sur une très grande base de données. Au vu des traitements réalisés, nous pouvons également fournir le pourcentage de ciel visible dans l'image, qui constitue un des objectifs de la sous-tâche 1.2. A titre d'illustration, cette information est donnée pour les images de la figure 17. Notons que ces diverses informations seront utilisées en entrée de la tâche 3 pour choisir d'exclure ou de pondérer l'impact de chacun de ces satellites

dans le calcul de position de localisation.



(a) Pourcentage de ciel = 21.31%

(b) Pourcentage de ciel = 49.14%



(c) Pourcentage de ciel = 26.58%

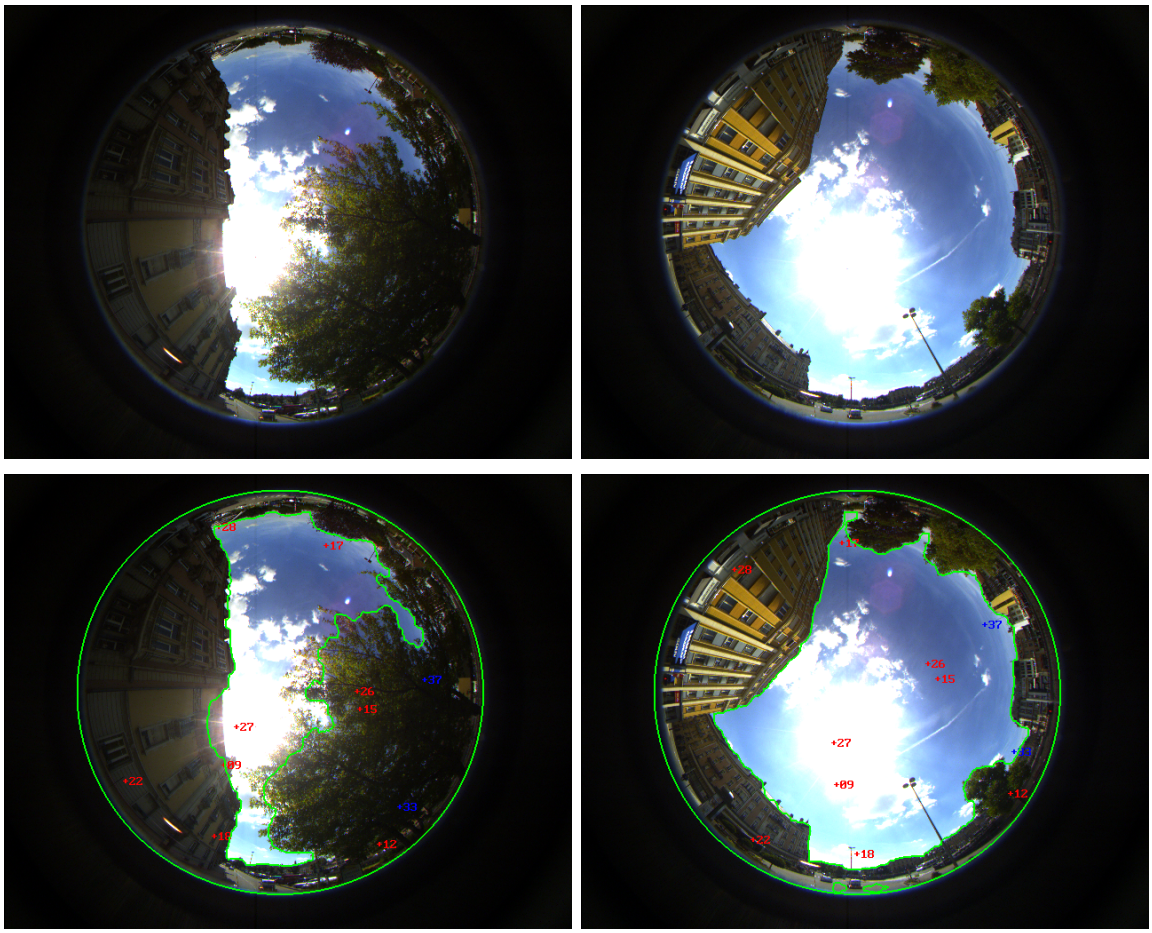
(d) Pourcentage de ciel = 54.15%

FIGURE 17 – Détection du ciel avec repositionnement des satellites et évaluation du pourcentage de ciel visible dans l'image (de haut en bas : image initiale, avec une classification par Fisher après filtrage médian, avec une classification par Fisher après simplification par RGD).

	Images				
	Image 1	Image 2	Image 3	Image 4	Image 5
	Satellites LOS				
Median	09, 15, 17 18, 26, 27 33, 37	09, 15, 17 26, 27, 33 37	09, 17, 18 27, 37	17, 27	09, 15 26, 27
	8/11	7/11	5/11	2/11	4/11
RGD	09, 15, 17 18, 26, 27 33, 37	15, 17 26, 27, 33 37	09, 17, 18, 27, 37	17, 27	09, 15, 17 26, 27,37
	8/11	6/11	5/11	2/11	6/11
	Satellites NLOS				
Median	12, 22, 28	12, 18, 22, 28	12, 15 22, 26, 28 33	09, 12, 15 18, 22, 26 28, 33, 37	12, 17, 18 22, 28, 33 37
	3/11	4/11	6/11	9/11	7/11
RGD	12, 22, 28	12, 18, 22, 28	09, 12, 18 22, 28	09, 12, 15 18, 22, 26 28, 33, 37	12, 18 22, 28, 33
	3/11	4/11	5/11	9/11	5/11

TABLE 4 – Identification, pour chacune des images, des satellites LOS et NLOS selon la méthode de simplification/classification utilisée.

Afin de répondre au deuxième objectif de la sous-tâche 1.2, nous détectons la ligne d'horizon sur chaque image de la figure 18. La ligne d'horizon affichée (en couleur verte sur les images initiales) correspond en traitement d'images à la frontière entre les régions de « ciel » et de « non-ciel » de l'image.



(a) Pourcentage de ciel = 21.31%

(b) Pourcentage de ciel = 49.14%

(c) Pourcentage de ciel = 26.58%

(d) Pourcentage de ciel = 54.15%

FIGURE 18 – Détection de la ligne d'horizon avec évaluation du pourcentage de ciel visible dans l'image (de haut en bas : image initiale, avec une classification par Fisher après filtrage médian, avec une classification par Fisher après simplification par RGD).

8 Conclusion

L'objectif de la tâche 1 consiste à développer un système automatique de traitement et d'analyse d'images permettant de caractériser l'environnement de réception des signaux GNSS. Pour ce faire, nous utilisons un véhicule de laboratoire instrumenté permettant d'acquérir, pour un parcours donné, des informations de position (grâce à un récepteur GNSS) et des données images (à partir d'une caméra couleur munie d'un objectif fisheye disposée sur le toit du véhicule). De par le matériel d'acquisition utilisé, nous avions initialement projeté de réaliser une première étape de rectification des images afin de corriger les fortes distorsions dues à l'objectif grand angle utilisé (sous-tâche 1.1). Une étude réalisée dans le cadre de travaux antérieurs a permis de montrer que cette opération ajoute un coût calculatoire qui impacte le temps de traitement des images. En conséquence, cette sous-tâche a été reportée et sera annulée si les travaux futurs de la sous-tâche 1.3 s'avèrent satisfaisants sans considérer cette étape.

La sous-tâche 1.2 consistant à détecter la ligne d'horizon et évaluer le pourcentage de ciel visible présent dans l'image a donc été avancée. Nous avons donc, dans cette sous-tâche, proposé une stratégie basée sur une simplification de l'image couplée à une classification pixellaire adaptée. Différentes méthodes de simplification d'images issues de la littérature (filtre médian, filtre gaussien, filtre moyenneur, filtre de Nagao, reconstruction géodésique par dilatation) ont été testées, tout comme un certain nombre de classifieurs non-supervisés et supervisés reconnus pour leurs bons résultats (KMlocal, FuzzyCmeans, Fisher, SRM, Bayes, Kppv, SVM). Les premiers tests ont été effectués sur quelques images de la base du projet LOCOPROL, afin d'identifier les deux meilleurs méthodes de simplification. Nous sommes arrivés à la conclusion que le filtre médian et la reconstruction géodésique par dilatation (RGD) offraient les meilleurs résultats. Nous avons ensuite réalisé une deuxième série de tests sur des images (très hétérogènes et de complexité variée) de la base du projet CAPLOC afin de définir le meilleur couple « méthode de simplification/classifieur utilisé ». Nous avons conclu que l'utilisation d'un filtre médian ou d'une RGD couplée à l'algorithme de Fisher (sur la composante bleue) ou à l'algorithme des Kppv (avec un nombre de $K_{ppv} = 6$) offrent les meilleurs taux de bonne classification. En effet, nous obtenons un taux de 97,53% pour une combinaison « Médian/Fisher » et de 97.99% pour une combinaison « RGD/Kppv ». La prise en compte du temps de traitement de l'approche proposée nous a fait choisir le couple Médian/Fisher (0.035s au lieu de 15.4s soit un rapport de 1/400). Cependant, nous souhaitons confirmer ce choix sur un nombre important d'images issues de la prochaine campagne d'acquisition du projet CAPLOC par rapport au couple « RGD/Fisher » très bien placé lui aussi (avec un taux de 96.4%). Ces traitements nous ont donc permis de fournir le pourcentage de ciel dans l'image, de détecter la ligne d'horizon et enfin d'identifier précisément les satellites dont les signaux sont reçus via des trajets directs (resp. réfléchis voire bloqués). Les résultats issus de cette sous-tâche ont été récemment utilisés dans la tâche 3 afin de montrer (dans certaines conditions) l'apport de l'analyse d'images.

Nos efforts se concentrent actuellement sur une caractérisation plus fine de l'environnement de réception des signaux satellitaires (tâche 1.3). En effet, nous cherchons à détecter en plus du ciel, la végétation et les bâtiments afin de pouvoir intégrer ces informations dans le calcul d'une nouvelle position. Pour ce faire, nous cherchons d'une part à détecter de manière encore plus précise les régions de « ciel » de l'image sans tenir compte de la contrainte temps-réel et d'autre part, de détecter les régions de « bâtiments » en utilisant une stratégie basée sur une détection de segments.

Table des figures

1 Les véhicules de laboratoire de l'UTBM/IRTES-SET et de l'Ifsttar-LEOST . . . 7
2 Quelques images de la base du projet LOCOPROL . . . 7
3 Exemple de parcours et quelques images de la base du projet CAPLOC . . . 8
4 Synopsis de la stratégie de détection de ciel proposée . . . 9
5 Résultats d'un lissage effectué par un filtre médian avec différentes tailles de voisinage. . . . 10
6 Résultats d'un lissage effectué par un filtre gaussien avec différentes valeurs de σ. 10
7 Résultats d'un lissage effectué par un filtre moyenneur avec différentes valeurs de W. . . . 11
8 Resultats d'une simplification par RGD avec différentes valeurs de H. . . . 12
9 Les 9 domaines de Nagao. . . . 13
10 Résultat d'une image simplifiée par un filtre de Nagao. . . . 13
11 Résultats de la classification ciel/non-ciel avec différents algorithmes non-supervisés en fonction de la méthode de simplification utilisée (sur la base LOCOPROL). . . 20
12 Résultats de la classification ciel/non-ciel avec différents algorithmes supervisés en fonction de la méthode de simplification utilisée (sur la base LOCOPROL). . . 21
13 Résultats de classification ciel/non-ciel sur 5 images de la base LOCOPROL (de haut en bas : Fisher (bleue), KMlocal (hybrid), FuzzyCmeans (m=1), SRM, Bayes, Kppv (nb Kppv=6), SVM). . . . 23
14 Résultats de la classification ciel/non-ciel avec différents algorithmes non-supervisés en fonction de la méthode de simplification utilisée (sur la base CAPLOC). . . . 24
15 Résultats de la classification ciel/non-ciel avec différents algorithmes supervisés en fonction de la méthode de simplification utilisée (sur la base CAPLOC). . . . 25
16 Résultats de classification ciel/non-ciel sur 5 images de la base CAPLOC (de haut en bas : Fisher (bleue), KMlocal (hybrid), FuzzyCmeans (m=1), SRM, Bayes, Kppv (nb Kppv=6), SVM). . . . 27
17 Détection du ciel avec repositionnement des satellites et évaluation du pourcentage de ciel visible dans l'image (de haut en bas : image initiale, avec une classification par Fisher après filtrage médian, avec une classification par Fisher après simplification par RGD). . . . 29
18 Détection de la ligne d'horizon avec évaluation du pourcentage de ciel visible dans l'image (de haut en bas : image initiale, avec une classification par Fisher après filtrage médian, avec une classification par Fisher après simplification par RGD). 31

Liste des tableaux

1 Résultats de la classification ciel/non-ciel avec différents algorithmes, paramètres et méthodes de simplification, sur la base LOCOPROL. 22
2 Résultats de la classification ciel/non-ciel avec différents algorithmes, paramètres et méthodes de simplification, sur la base CAPLOC. 26
3 Temps de traitement de l'approche en fonction du couple classifieur/simplification utilisé 26
4 Identification, pour chacune des images, des satellites LOS et NLOS selon la méthode de simplification/classification utilisée. 30

Références

[1] J. Marais, S. Ambellouis, A. Flancquart et M. Berbineau : Dispositif de localisation - Demande de brevet français déposée le 29/09/2006 n ° 06 54018. N ° de publication : WO/2008/040896, date de publication internationale : 10 avril 2008. 4

[2] D. Robertson et R. Cipolla : An image-based system for urban navigation. *British Machine Vision Conference (BMVC)*, 2004. 5

[3] W. Zhang et J. Kosecka : Image based localization in urban environments. *3D Data Processing Visualization and Transmission (3DPVT)*, 2006. 5

[4] E. Royer, M. Lhuillier et M. Dhome : Monocular vision for mobile robot localization. *IJCV*, 2007. 5

[5] N. Jacobs, S. Satkin, N. Roman, R. Speyer et R. Pless : Geolocating static cameras. *International Conference on Computer Vision (ICCV)*, 2007. 5

[6] J. Hays et A-A. Efros : Im2gps : estimating geographic images from single images. *Computer Vision and Pattern Recognition (CVPR)*, 2008. 5

[7] F. Stein et G. Medioni : Map-based localization using the panoramic horizon. *In International Conference on Robotics and Automation (ICRA)*, 1995. 5

[8] O. Koch et S. Teller : Wide-area egomotion estimation from known 3D structure. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007. 5

[9] J-I Meguro, Taishi Murata, J-I Takiguchi, Yoshiharu Amano et Takumi Hashizume : Gps multipath mitigation for urban area using omnidirectional infrared camera. *IEEE Trans. On Intelligent Transportation Systems*, 10(1):22–30, mars 2009. 5

[10] J. Tardif, Y. Pavlidis et K. Daniilidis : Monocular visual odometry in urban environments using an omnidirectional camera. *IROS*, 2008. 5

[11] S. Ramalingam, S. Bouaziz, P. Sturm et M. Brand : Skyline2gps : Localization in urban canyons using omni-skylines. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010. 5, 6

[12] C. Meurie, Y. Ruichek, A. Cohen et J. Marais : An hybrid an adaptive segmentation method using color and textural information. *Image Processing : Machine Vision Applications III*, janvier 2010. 6

[13] A. Cohen, C. Meurie, Y. Ruichek, J. Marais et A. Flanquart : Quantification of gnss signals accuracy : an image segmentation method for estimating the percentage of sky. *IEEE Internayinal confrenece on Vehicular Electronics and Safety (ICVES)*, pages 40–45, 2009. 6

[14] A. Cohen, C. Meurie, Y. Ruichek et J. Marais : Characterization of the reception environment of gnss signals using a texture and color based adaptive segmentation technique. *IEEE Intelligent Vehicles Symposium*, pages 275–280, June 2010. 6

[15] J. Serra : Image analysis and mathematical morphology. *Theorical advances*, 2, 1988. 11

[16] M. Shmitt et J. Mattioli : *Morphologie mathématique*. Masson, 1994. 11

[17] Z-B. Sbeh et L-D. Cohen : A new approach of geodesic reconstruction for drusen segmentation in eye funds images. *IEEE Transactions on medical imaging*, 20:1321–1333, 2001. 11

[18] L. Vincent : Morphological gray scale reconstruction in image analysis : Applications and efficient algorithms. *IEEE Transactions on image processing*, 2:176–201, 1993. 11

[19] N. Bolshakova et F. Azuaje : Cluster validation techniques for genome expression data. *Signal processing*, 83:825–833, 2003. 13

[20] K. Muata et O. Bryson : Towards supporting expert of clustering results using a data mining process model. *Information Sciences*, 180:414–431, 2010. 13

[21] X. Yanga, H. Rena et B. Lib : Embedded zerotree wavelets coding based on adaptive fuzzy clustering for image compression. *Image and Vision Computing*, 26:812–819, 2008. 13

[22] E. Lughofer : Extensions of vectorquantization for incremental clustering. *Pattern Recognition*, 41:995–1011, 2008. 13

[23] M. Kyperountasa, A. Tefasa et I. Pitasa : Dynamic training using multistage clustering for face recognition. *Pattern Recognition*, 41:894–905, 2008. 13

[24] J-T-H Lo : A cortex-like learning machine for temporal hierarchical patternclustering, detection, and recognition. *Neurocomputing*, 78:89–103, 2012. 13

[25] R. Nock et F. Nielsen : Statistical region merging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26:1452–1458, 2004. 14

[26] F. Moosmann, E. Nowak et F. Jurie : Randomized clustering forests for image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30:1632–1646, 2008. 14

[27] L. Gupta, U-G. Mangai et S. Das : Integrating region and edge information for texture segmentation using a modified constraint satisfaction neuralnetwork. *Image and Vision Computing*, 26:1106–1117, 2008. 14

[28] F. Lafarge, X. Descombes et J. Zerubia : Noyaux texturaux pour les problèmes de classification par svm en télédétection. Rapport technique 5370, INRIA, Novembre 2004. 14, 18

[29] T. Kanungo, D-M. Mount, N. Netanyahu, C. Piatko, R. Silverman et A-Y. Wu : A local search approximation algorithm for k-means clustering. *Computational Geometry : Theory and Applications*, 28:89–112, 2003. 14

[30] D-M. Mount : Kmlocal : A testbed for k-means clustering algorithms. Rapport technique, Department of Computer Science and Institute for Advanced Computer Studies University of Maryland, 2005. 14

[31] F. Nielsen et R. Nock : On region merging : the statistical soundness of fast sorting, with applications. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2:19–26, 2003. 14, 16

[32] R. Nock et F. Nielsen : Statistical region merging. *IEEE transactions on pattern analysis and machine intelligence*, 26(11):1452–1458, November 2004. 14, 16, 17

[33] S. Lefèvre : A new approach for unsupervised classification in image segmentation. *EGC*, pages 127–138, 2009. 14

[34] S-K. Mathankera, P-R. Wecklera, T-J. Bowsera, N. Wanga et N-O. Manessb : Adaboost classifiers for pecan defect classification. *Computers and Electronics in Agriculture*, 77:60–68, 2011. 14

[35] J. MacQueen : Some methods for classification and analysis of multivariate observations. *In Proceedings of the Fifth Berkley Symposium on Mathematical Satistics and Probability*, 1:281–296, 1967. 14

[36] S-P. LLoyd : Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28:129–137, 1982. 14

[37] S-Z. Selim et M-A. Ismail : K-means-type algorithms : A generalized convergence theorem and characterization of local optimality. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:81–87, 1984. 14

[38] J-C. Dunn : A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*, 3:32–52, 1973. 15

[39] J-C. Bezdek : *Pattern Recognition with Fuzzy Objective Function Algoritms.* New York Plenum Press, 1981. 15

[40] R. Fisher : The use of multiple measurements in taxonomics problems. *Anals of Eugenics*, 7:179–188, 1936. 16

[41] W. Fisher : On grouping for maximum homogenity. *JASA*, 53:789–798, 1958. 16

[42] S-C. Zhu et A. Yuille : Region competition : unifiying snakes, region growing, and bayes/mdl for multiband image segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18:884–900, 1996. 16

[43] R. Nock et F. Nielson : Semi-supervised statistical region refinementfor color image segmentation. *Pattern Recognition*, 38:835–846, 2005. 16

[44] V. Guigue, A. Rakotomamonjy et S. Canu : SVM et k-ppv pour la reconnaissance d'émotions. *19ème Colloque sur le traitement du signal et des images*, pages 770–773. Groupe dÉtudes du Traitement du Signal et des Images GRETSI, 2003. 17

[45] S. Bazeille, P. Hemar et B. Mougeland F. NG FOK : Rapport dímages numériques. Rapport technique, Université de la Rochelle, 2004. 17

[46] C. Meurie : *Segmentation of color images by pixels classification and hierarchy of partitions.* Thèse de doctorat, University of Caen Basse-Normandie, Caen, France, octobre 2005. 17

[47] V. Vapnik : *The nature of statistical learning theory.* Springer, 1996. 18

[48] Mohamadally Hasan et Fomani Boris : SVM : Machines à vecteurs de support ou séparateurs à vastes marges. Rapport technique, Versailles St Quentin, France, 16 janvier 2006. 18

[49] J-O. Moussafir : Introduction aux SVM. 20 octobre 2005. 18



Groupe opérationnel n°2 du Predit 4

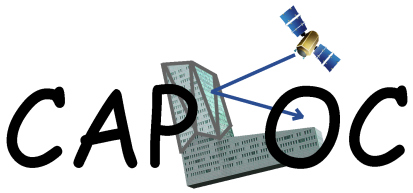
« Gestion du trafic et des mobilités »



CAPLOC

Combinaison de l'**A**nalyse d'images
et de la connaissance de la **P**ropagation
des signaux pour la **LOC**alisation

Convention 10-MT-CVS-001



Livrable de la tâche 1.3

Segmentation et classification fine de l'environnement de réception des signaux GNSS

18 janvier 2015

Cyril MEURIE, Christophe CONIGLIO, Amaury FLANCQUART
IFSTTAR-LEOST
Dhouha ATTIA, Yassine RUICHEK
UTBM-SeT





Identification

Acronyme du projet	CAPLOC
Titre du projet	Combinaison de l'Analyse d'images et de la connaissance de la Propagation des signaux pour la LOCalisation
Coordinatrice du projet	Juliette MARAIS (IFSTTAR-LEOST)
Date de début de projet	19 octobre 2010
Date de fin de projet	18 janvier 2015
Partenaires	IFSTTAR-LEOST, UTBM-SeT
Type de document	Livrable
Date contractuelle de livraison	18 janvier 2015
Accessibilité	La diffusion de ce document est limitée au programme PREDIT

Gestion des versions

Version	**Date**	**Auteurs**	**Modifications**
1.0	24/11/2014	Cyril MEURIE	Rédaction
1.1	13/01/2015	Christophe CONIGLIO	Corrections
1.2	14/01/2015	Amaury FLANCQUART	Corrections
1.3	14/01/2015	Yassine RUICHEK	Corrections
2.0	16/01/2015	Cyril MEURIE	Approbation

Table des matières

1 Introduction **4**

2 Une nouvelle méthode de segmentation couleur/texture pour la détection ciel/végétation/bâtiments **6**
2.1 L'analyse en composantes principales 6
2.2 La combinaison couleur/texture par comparaison de valeurs propres 9
2.3 La Ligne de Partage des Eaux 13

3 Stratégie de détection de bâtiments par analyse de segments **15**
3.1 Détection de segments 15
3.1.1 Transformée de Hough 15
3.1.2 Dseg : Détection directe de segments dans une image 16
3.1.3 LSD : a Line Segment Detector 18
3.2 Segmentation de l'image 19
3.2.1 Création d'un pré-maillage 19
3.2.2 Simplification du maillage par élagage 20
3.2.3 Suppression des segments indésirables 21
3.2.4 Création d'une carte de régions 22
3.3 Classification des régions en bâtiment/non-bâtiment 23
3.3.1 Carte de densité de segments 23
3.3.2 Classification bâtiment VS non-bâtiment 24

4 Conclusion **26**

Table des figures **27**

Bibliographie **29**

1 Introduction

Dans le projet CAPLOC, l'objectif de la tâche 1 est de développer un système automatique de traitement et d'analyse d'images permettant de caractériser l'environnement de réception des signaux satellitaires et de fournir un ensemble de primitives pertinentes. Pour ce faire, nous utilisons un véhicule de laboratoire instrumenté permettant d'acquérir, pour un parcours donné, des informations de position (grâce à un GPS) et des données images (à partir d'une caméra couleur munie d'un objectif fisheye disposée sur le toit du véhicule). Le travail réalisé dans cette première tâche est donc décomposé en trois sous-tâches rappelées ci-après :

1. Étalonnage et calibrage du système d'acquisition ;
2. Détection de la ligne d'horizon et évaluation du pourcentage de ciel visible par analyse d'images couleur ;
3. Extraction et caractérisation de la texture ;

Dans la sous-tâche 1.1 (cf. livrable L1.2 du 28/10/2013), nous avions initialement projeté de réaliser une première étape de rectification des images acquises en mobilité afin de corriger les fortes distorsions dues à l'objectif grand angle utilisé. Cependant, comme indiqué, dans le livrable L1.1 du 18/10/2012, une étude réalisée dans le cadre de travaux antérieurs a permis de montrer que cette opération ajoute un coût calculatoire qui impacte le temps de traitement des images et nous éloigne ainsi de la contrainte temps réel imposée dans la sous-tâche 1.2. Les très bons résultats de détection de la ligne d'horizon et de pourcentage de ciel visible de la sous-tâche 1.2 (cf. livrable L1.1 du 18 octobre 2012) ainsi que les premiers résultats d'extraction de descripteurs couleur/texture nous ont amené à conclure que cette opération de rectification n'est finalement pas nécessaire au vue des attentes finales de la tâche 1. Néanmoins, afin de montrer l'avancement de cette tâche, nous avons décidé, non pas de supprimer le livrable correspondant, mais de remplacer son contenu par les travaux réalisés sur l'extraction de descripteurs couleur/texture pour la caractérisation de l'environnement de réception des signaux satellitaires. En effet, les travaux finaux de la tâche 1 ont pour objectif de caractériser l'environnement de réception des signaux satellitaires de manière plus fine (que la classification ciel/non-ciel effectuée dans la tâche 1.2) de sorte à pouvoir i) pondérer l'impact de chacun des satellites dans le calcul de position de localisation (selon qu'ils se trouvent dans une classe « ciel », « végétation », « bâtiment », etc) ; ii) caractériser par des attributs pertinents certaines textures rencontrées afin de pouvoir reproduire la texture des plans du modèle 3D (tâche 2). Nous avons donc, dans le le livrable de cette première sous-tâche, présenté un état de l'art exhaustif des descripteurs de couleur et de texture jugés pertinents dans le cadre de notre application, et les avons catégorisés parmi les attributs géométriques, les attributs basés sur la modélisation spatiale des textures, les attributs spatio-fréquentiels, les attributs statistiques et ceux issus de la morphologie mathématique. Cette sélection d'attributs couleur et texture (décrite dans la thèse de Dhouha ATTIA, doctorante financée par la région Franche-Comté et mis à disposition pour la tâche 1 du projet CAPLOC [1]) est utilisée dans la définition d'un gradient structurel combinant les informations couleur/texture les plus pertinentes qui sera utilisé en entrée de l'algorithme de segmentation par croissance de régions. L'image acquise en mobilité et ainsi segmentée (partitionnée en régions homogènes à l'aide de ces informations couleur/texture) facilitera sans aucun doute la classification de l'environnement de réception des signaux satellitaires en trois classes : « ciel », « végétation », « bâtiment ».

Dans la sous-tâche 1.2 (cf. livrable L1.1 du 18/10/2012), nous avons cherché à détecter la ligne d'horizon et évaluer le pourcentage de ciel visible présent dans l'image. Pour ce faire, nous avons proposé une stratégie basée sur une simplification de l'image couplée à une classification pixellaire

adaptée. Différentes méthodes de simplification d'images issues de la littérature (filtre médian, filtre gaussien, filtre moyenneur, filtre de nagao, reconstruction géodésique par dilatation) ont été testées, tout comme un certain nombre de classifieurs (non)supervisés reconnus pour leur bons résultats (KMlocal, FuzzyCmeans, Fisher, SRM, Bayes, Kppv, SVM). Les premiers tests ont été effectués sur quelques images de la base du projet LOCOPROL, afin d'identifier les deux meilleures méthodes de simplification. Nous sommes arrivés à la conclusion que le filtre médian et la reconstruction géodésique par dilatation (RGD) offraient les meilleurs résultats. Nous avons ensuite réalisé une deuxième série de tests sur des images (très hétérogènes et de complexité variée) de la base du projet CAPLOC afin de définir le meilleur couple « méthode de simplification/classifieur utilisé ». Nous avons conclu que l'utilisation d'un filtre médian ou d'une RGD couplée à l'algorithme de Fisher (sur la composante bleue) ou à l'algorithme des Kppv (avec un nombre de $kppv = 6$) offre les meilleurs taux de bonne classification. En effet, nous obtenons un taux de 97,53% pour une combinaison « Médian/Fisher » et de 97.99% pour une combinaison « RGD/Kppv ». La prise en compte du temps de traitement de l'approche proposée nous a fait choisir le couple Médian/Fisher (0.035s au lieu de 15.4s soit un rapport de 1/400). Ces traitements nous ont donc permis d'identifier précisément les régions de « ciel » dans l'image (respectivement de « non-ciel »), de détecter la ligne d'horizon et de pouvoir après repositionnement des satellites dans l'image, identifier ceux dont les signaux sont reçus via des trajets directs (resp. réfléchis voire bloqués). Les résultats issus de cette sous-tâche ont été utilisés dans la tâche 3 (cf. livrable 3.1) afin de montrer (dans certaines conditions) l'apport de l'analyse d'images pour l'exclusion de certains satellites dans le calcul d'une nouvelle position (se voulant plus précise).

Dans la sous-tâche 1.3 (cf. ce dernier livrable du 18/01/2015), nous caractérisons l'environnement de réception des signaux satellitaires de manière plus fine (que la classification ciel/non-ciel effectuée dans la sous-tâche 1.2) de sorte à pouvoir i) pondérer l'impact de chacun des satellites dans le calcul de position de localisation, selon s'ils se trouvent dans une classe « ciel », « végétation », « bâtiment » (tâche 3) ; ii) caractériser par des attributs pertinents certaines textures rencontrées afin de pouvoir reproduire la texture des plans du modèle 3D (tâche 2). Pour ce faire, nous avions initialement pensé qu'une segmentation de l'image utilisant conjointement les informations de texture et de couleur faite au préalable, n'aurait pu qu'améliorer la classification du contenu de l'image en différentes classes (« ciel », « végétation », « bâtiment »). Mais cette stratégie qui fait appel à l'utilisation de descripteurs de couleur et de texture voit son cout calculatoire largement augmenté et donc incompatible avec la contrainte temps réel visée à moyen/long terme pour les objectifs de la tâche 1. Nous avons donc fait le choix dans la deuxième partie de se rapport de nous orienter vers une stratégie de détection de bâtiment basée en grande partie sur une analyse de segments (décrite en détails dans le rapport de stage de projet de fin d'étude d'ingénieur de Christophe Coniglio [2]), qui couplée à celle de la détection de ciel faite dans la sous-tâche 1.2, nous permettra par simple soustraction d'identifier les zones de végétation et donc de disposer d'une caractérisation fine (ciel, végétation, bâtiment) de l'environnement de réception des signaux satellitaires.

2 Une nouvelle méthode de segmentation couleur/texture pour la détection ciel/végétation/bâtiments

Dans le précédent livrable, nous avons présenté un certain nombre de descripteurs couleur et texture, en stipulant que ces derniers pourraient être utilisés dans la définition d'un gradient structurel lui-même utilisé dans une méthode de segmentation par croissance de régions. Nous allons donc dans cette section, présenter cette nouvelle méthode de combinaison couleur/texture non paramétrique basée sur l'analyse des composantes principales et permettant d'obtenir un gradient structurel en niveau de gris qui sera ensuite utilisé par l'algorithme de LPE en vue de segmenter l'image. Le lecteur desireux de plus de détails sur cette stratégie pourra consulter la thèse de Dhouha ATTIA, doctorante financée par la région Franche-Comté et mis à disposition pour la tâche 1 du projet CAPLOC [1]. Mais, rappelons tout d'abord le principe de l'analyse en composantes principales (ACP).

2.1 L'analyse en composantes principales

L'analyse en composantes principales (ACP) est une technique qui exploite l'aspect géométrique et les représentations graphiques d'un ensemble de données que nous appelons observations afin d'étudier sa variabilité et sa dispersion dans l'espace dans lequel il est représenté. Soit P l'ensemble de données comportant m variables quantitatives ayant n unités (notées souvent individus) défini par l'équation 1.

$$P = \left[\begin{array}{cccccc} \begin{pmatrix} p_{11} \\ p_{21} \\ \vdots \\ p_{i1} \\ \vdots \\ p_{n1} \end{pmatrix} & \begin{pmatrix} p_{12} \\ p_{22} \\ \vdots \\ p_{i2} \\ \vdots \\ p_{n2} \end{pmatrix} & \cdots & \begin{pmatrix} p_{1j} \\ p_{2j} \\ \vdots \\ p_{ij} \\ \vdots \\ p_{nj} \end{pmatrix} & \begin{pmatrix} p_{1m} \\ p_{2m} \\ \vdots \\ p_{im} \\ \vdots \\ p_{nm} \end{pmatrix} \end{array} \right] \rightarrow \left[\begin{array}{c} (\vec{p}_{1.})^t \\ (\vec{p}_{2.})^t \\ \vdots \\ (\vec{p}_{i.})^t \\ \vdots \\ (\vec{p}_{n.})^t \end{array} \right] \quad (1)$$

$$\downarrow$$

$$\left[\begin{array}{ccccc} \vec{p}_{.1} & \vec{p}_{.2} & \cdots & \vec{p}_{.j} & \cdots & \vec{p}_{.m} \end{array} \right]$$

En partant de cette écriture matricielle, nous pouvons réécrire l'ensemble de données P sous deux formes vectorielles (cf. équation 1) : le vecteur vertical est composé d'un ensemble de points tel que chaque point est noté $\vec{p}_{i.}$ (de dimension m). Le vecteur horizontal constitue la deuxième forme vectorielle et se compose d'un ensemble de points tel que chacun est noté $\vec{p}_{.j}$ (de dimension n). Notons que t désigne la fonction transposé. Les points $\vec{p}_{.i}$ et $\vec{p}_{.j}$ peuvent être donnés par les équations suivantes :

$$\vec{p}_{.j} = \begin{pmatrix} p_{1j} \\ p_{2j} \\ \vdots \\ p_{nj} \end{pmatrix}_{1 \leq j \leq m} \quad \text{et} \quad \vec{p}_{i.} = \begin{pmatrix} p_{i1} \\ p_{i2} \\ \vdots \\ p_{im} \end{pmatrix}_{1 \leq i \leq n} \quad (2)$$

où i et j indiquent respectivement l'individu et la variable. Les représentations géométriques des unités permettent de déterminer les unités qui se ressemblent entre elles. Les représentations des variables permettent quant à elles d'étudier les structures de liaisons linéaires et de déterminer les variables qui sont corrélées [3, 4]. Dans ce rapport, nous avons recours à l'ACP grâce à son utilité d'extraire les axes principaux qui conservent le maximum d'information d'un ensemble de données étudiées [4].

L'ACP est une technique basée sur l'algèbre linéaire, qui consiste dans une première étape àă-calculer la matrice de covariance afin de déterminer les axes principaux de l'ensemble de données étudiées. Ceci permet de décrire la dispersion des données par rapport à leurs axes principaux et de conclure sur les informations les plus pertinentes. Nous commençons donc par calculer la matrice de covariance d'un ensemble de données P.

Dans ce qui suit, nous rappelons les définitions de la matrice de covariance $\mathcal{COV}_P$ de l'ensemble de données P. En effet, calculer la matrice de covariance revient à calculer l'espérance $\mathcal{E}$ du produit de l'ensemble de points P centré sur leurs moyennes et du même vecteur résidu transposé comme présenté par l'équation 3.

$$\mathcal{COV}_P \equiv \mathcal{VAR}(P) \equiv \mathcal{E}((P - \mathcal{E}(P))(P - \mathcal{E}(P))^t) \tag{3}$$

En utilisant l'équation 1, nous pouvons redéfinir la matrice de covariance de $\mathcal{COV}_P$ comme illustré l'équation 4.

$$\mathcal{COV}_P \equiv \mathcal{VAR} \begin{bmatrix} (\vec{p}_{1.})^t \\ (\vec{p}_{2.})^t \\ \vdots \\ (\vec{p}_{i.})^t \\ \vdots \\ (\vec{p}_{n.})^t \end{bmatrix} \tag{4}$$

La matrice de covariance est formée par les variances/covariances de deux points $\vec{p}_{.j}$ et $\vec{p}_{.j'}$ [4]. Par conséquent, elle est définie par l'écriture matricielle donnée par l'équation 5.

$$\mathcal{COV}_P = \begin{pmatrix} var(\vec{p}_{1.}) & cov((\vec{p}_{1.})^t, \vec{p}_{2.}) & \cdots & cov((\vec{p}_{1.})^t, \vec{p}_{n.}) \\ cov((\vec{p}_{2.})^t, \vec{p}_{1.}) & \ddots & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ cov((\vec{p}_{n.})^t, \vec{p}_{1.}) & \cdots & \cdots & var(\vec{p}_{n.}) \end{pmatrix} \tag{5}$$

avec $\forall 1 \leq i \leq n$

$$cov((\vec{p}_{i.})^t, \vec{p}_{i.}) = var(\vec{p}_{i.}) \tag{6}$$

$$cov((\vec{p}_{i.})^t, \vec{p}_{i'.}) = \frac{1}{m} \sum_{j=1}^{m} (p_{ij} - \bar{p}_{i.})(p_{i'j} - \bar{p}_{i'.}) \tag{7}$$

$$\bar{p}_{i.} = \frac{1}{m} \sum_{j=1}^{m} p_{ij} \tag{8}$$

La matrice de covariance étant calculée, il s'agit ensuite d'extraire les valeurs propres et leurs vecteurs propres associés correspondants aux axes principaux de l'ensemble de données étudiées permettant ainsi d'avoir l'allure (la représentation géométrique) de ces données dans leur espace de représentation. Les valeurs propres correspondent à la variance de l'ensemble de données (qui peut être assimilé àăun nuage de points de dimension m) suivant les directions données par les axes principaux trouvés.

L'ACP est une technique souvent utilisée pour décolérer les données étudiées et réduire leur taille initiale. Dans le domaine de la vision par ordinateur, l'ACP a été utilisée dans de nombreuses applications et notamment dans la classification des images [5, 6, 7], la compression de

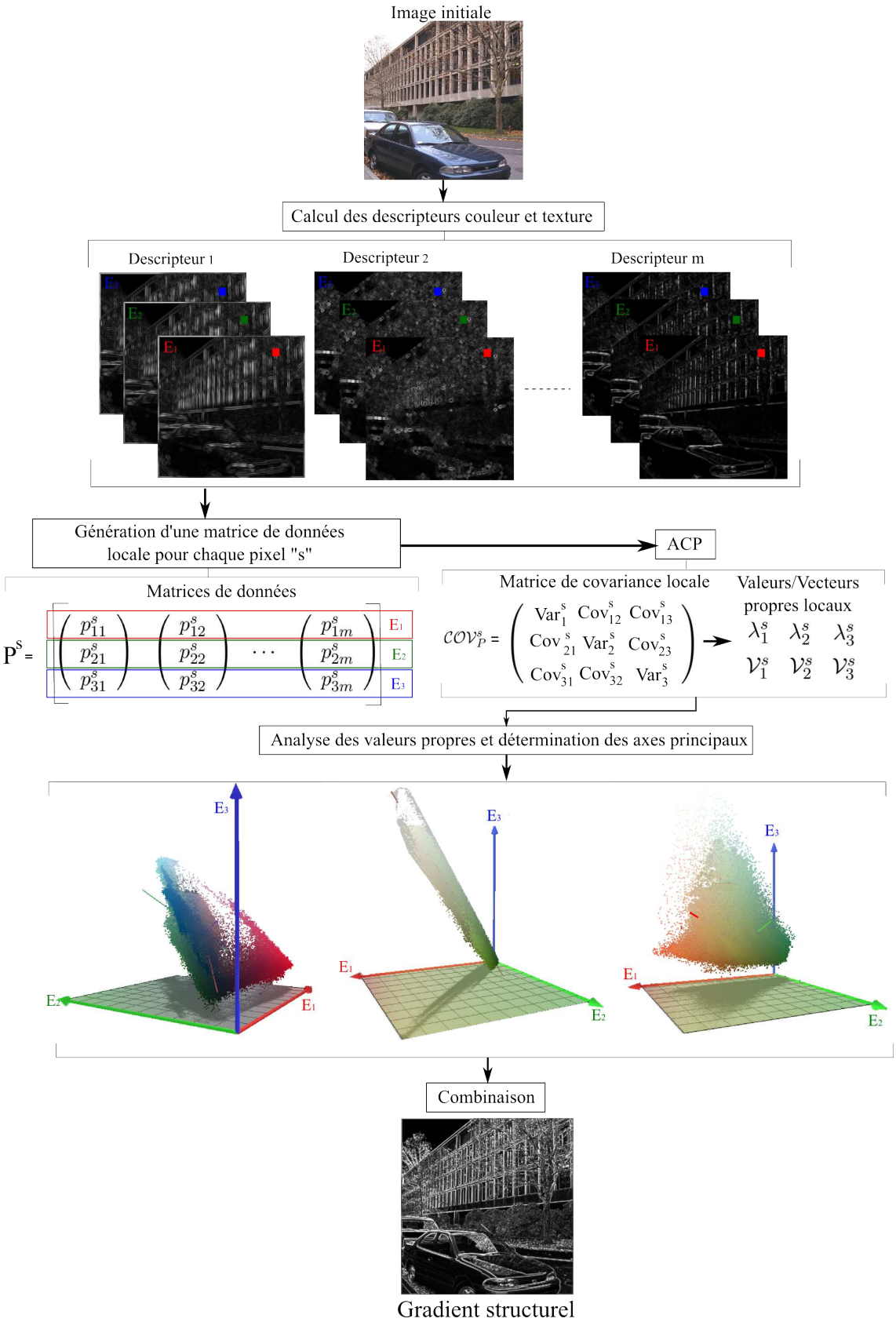


FIGURE 1 – Synopsis global de la combinaison couleur-texture basée sur l'ACP.

l'information [8, 9, 10], la reconnaissance des objets [11, 12, 13], etc. Dans ce rapport, nous utilisons l'ACP dans le but de proposer une nouvelle méthode de combinaison des informations de couleur et de texture présentes dans l'image. Plus précisément et comme précédemment mentionné, l'ACP permet d'extraire les axes qui conservent le maximum d'information. Nous utilisons donc cette technique afin d'identifier les informations (couleur et/ou texture) les plus pertinentes à combiner et ainsi obtenir un nouveau gradient structurel hybride permettant d'améliorer les résultats de segmentation par LPE. Dans ce qui suit, nous détaillons la stratégie globale de cette combinaison couleur/texture basée sur l'ACP, en présentant le calcul de la matrice de covariance et la méthode de combinaison des descripteurs calculés. Pour une meilleure compréhension, nous détaillons également (pour le cas particulier $n = 3$) le processus de calcul de la matrice de covariance sur les descripteurs couleur/texture de l'image à analyser ainsi que l'extraction des axes principaux des données obtenues (cf. figure 1).

2.2 La combinaison couleur/texture par comparaison de valeurs propres

Commençons par définir l'ensemble de données sur lequel nous allons appliquer la combinaison couleur/texture proposée. Nous avons initialement calculé un ensemble de descripteurs couleur/texture de l'image à analyser. m désigne le nombre de descripteurs utilisés (les variables en considérant la terminologie de l'ACP). I désigne l'image initiale et s un pixel donné de I. Nous associons à s un ensemble de données P^s (un nuage de points 3D dans notre cas puisque le nombre d'unités n est égal àă 3 pour chaque variable). En un pixel s de l'image, l'ensemble de données est généré (localement) àă partir des valeurs des m descripteurs de couleur et de texture. Les descripteurs utilisés sont des gradients couleur et des gradients texture calculés àă partir de l'image initiale I dans un espace couleur donné $E_1E_2E_3$. $\vec{g}^s_{.j}$ correspond donc au $j^{\text{ème}}$ gradient défini dans cet espace couleur avec $1 \leq j \leq m$. Par analogie avec la notation de l'ACP, nous pouvons définir le nuage de points 3D selon les deux écritures matricielle (cf. équation 9) et vectorielle (cf. équation 10) suivantes :

$$P^s = \left[\begin{pmatrix} g^s_{11} \\ g^s_{21} \\ g^s_{31} \end{pmatrix} \begin{pmatrix} g^s_{12} \\ g^s_{22} \\ g^s_{32} \end{pmatrix} \cdots \begin{pmatrix} g^s_{1m} \\ g^s_{2m} \\ g^s_{3m} \end{pmatrix} \right] \tag{9}$$

$$P^s = [\vec{g^s}_{.1}, \vec{g^s}_{.2}, \ldots, \vec{g^s}_{.j}, \ldots, \vec{g^s}_{.m}] \tag{10}$$

La deuxième étape consiste à appliquer l'ACP (cf. figure 1) et à calculer la matrice de covariance du nuage de points P^s. La matrice de covariance de P^s notée $\mathcal{COV}^s_P$ est une matrice de taille 3×3 et elle est donnée par l'équation 11 :

$$\mathcal{COV}^s_P = \begin{pmatrix} var(\vec{g^s}_{1.}) & cov((\vec{g^s}_{1.})^t, \vec{g^s}_{2.}) & cov((\vec{g^s}_{1.})^t, \vec{g^s}_{3.}) \\ cov((\vec{g^s}_{2.})^t, \vec{g^s}_{1.}) & cov((\vec{g^s}_{2.})^t, \vec{g^s}_{2.}) & cov((\vec{g^s}_{2.})^t, \vec{g^s}_{3.}) \\ cov((\vec{g^s}_{3.})^t, \vec{g^s}_{1.}) & cov((\vec{g^s}_{3.})^t, \vec{g^s}_{2.}) & var(\vec{g^s}_{3.}) \end{pmatrix} \tag{11}$$

avec $\forall 1 \leq i \leq 3$

$$cov((\vec{g^s}_{i.})^t, \vec{g^s}_{i.}) = var(\vec{g^s}_{i.}) \tag{12}$$

$$cov((\vec{g^s}_{i.})^t, \vec{g^s}_{i'.}) = \frac{1}{m} \sum_{j=1}^{m} (g^s_{ij} - \bar{g^s}_{i.})(p_{i'j} - \bar{g^s}_{i'.}) \tag{13}$$

$$\bar{g^s}_{i.} = \frac{1}{m} \sum_{j=1}^{m} g^s_{ij} \tag{14}$$

Selon les règles d'algèbre linéaire, il existe deux matrices notées L^s et D^s telles que :

$$(L^s)^{-1}\mathcal{COV}_P^s L^s = D^s \tag{15}$$

Dans ce cas particulier que nous étudions dans ce rapport, la matrice de covariance obtenue est une matrice de taille 3×3. Il en résulte trois valeurs propres et trois vecteurs propres associés. Nous désignons par λ_1^s, λ_2^s et λ_3^s, les valeurs propres de $\mathcal{COV}^s$ telles que $\lambda_3^s \leq \lambda_2^s \leq \lambda_1^s$ et $\overrightarrow{\mathcal{V}}_1^s$, $\overrightarrow{\mathcal{V}}_2^s$, $\overrightarrow{\mathcal{V}}_3^s$ les vecteurs propres associés. Les matrices D^s et L^s sont alors définies par les deux équations données ci-après :

$$D^s = \begin{pmatrix} \lambda_1^s & 0 & 0 \\ 0 & \lambda_2^s & 0 \\ 0 & 0 & \lambda_3^s \end{pmatrix} \tag{16}$$

$$L^s = [\overrightarrow{\mathcal{V}}_1^s \quad \overrightarrow{\mathcal{V}}_2^s \quad \overrightarrow{\mathcal{V}}_3^s] \tag{17}$$

Pour déterminer les axes principaux qui offrent le maximum d'informations et décider ensuite quelles sont les informations à combiner, nous procédons à une comparaison des valeurs propres. Soit $\hat{G}$ le gradient structurel obtenu et $\hat{G}(s)$ sa valeur estimée en s. Le processus de comparaison distingue trois cas décrits ci-après :

Cas 1 $\lambda_1^s \gg \lambda_2^s$ (à titre informatif $\frac{\lambda_1^s}{\lambda_2^s} \geq 10^2$) : Il existe un seul axe principal dont la direction est celle du vecteur propre $\overrightarrow{\mathcal{V}}_1^s$, autour duquel le nuage de points 3D est concentré (cf. figure 2(a)). Dans ce cas, nous choisissons la seule composante principale qui conserve le maximum d'informations et dont la direction est donnée par le vecteur propre $\overrightarrow{\mathcal{V}}_1^s$. La valeur attribuée au pixel est donnée par la formulation mathématique suivante :

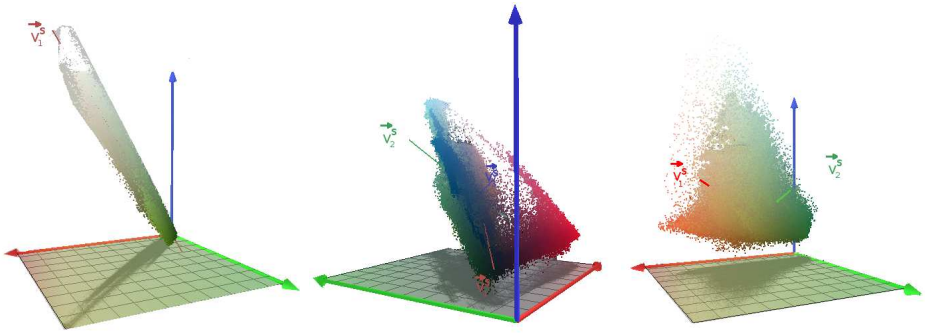
$$\hat{G}(s) = argmax_{j\in\{1,\ldots,m\}}\{g_{1j}\}$$

Cas 2 : $\lambda_3^s \ll \lambda_2^s$ (à titre informatif $\frac{\lambda_2^s}{\lambda_3^s} \geq 10$) : Dans ce cas, λ_1^s et λ_2^s ont le même ordre de grandeur. Le nuage de points 3D s'apparente à un plan formé par les deux vecteurs propres $\overrightarrow{\mathcal{V}}_1^s$ et $\overrightarrow{\mathcal{V}}_2^s$ (associés aux valeurs propres λ_2^s et λ_2^s) (cf. figure 2(b)). Le vecteur propre $\overrightarrow{\mathcal{V}}_3^s$ constitue la normale du plan formé par les vecteurs propres $\overrightarrow{\mathcal{V}}_1^s$ et $\overrightarrow{\mathcal{V}}_2^s$. Par conséquent, les composantes principales qui conservent le maximum d'information sont celles qui forment le plan $(\overrightarrow{\mathcal{V}}_1^s, \overrightarrow{\mathcal{V}}_2^s)$. La valeur attribuée au pixel est alors donnée par la formulation mathématique suivante :

$$\hat{G}(s) = argmax_{j\in\{1,\ldots,m\}}\{\frac{1}{2}(g_{1j}^s + g_{2j}^s)\}$$

Cas 3 : $\lambda_3^s \simeq \lambda_2^s \simeq \lambda_3^s$: Dans ce dernier cas, le nuage de points 3D est dispersé dans toutes les directions (cf. figure 2(c)) et aucune information n'est alors privilégiée. Nous combinons donc toutes les composantes principales équitablement et la valeur attribuée au pixel est donnée par la formulation mathématique suivante :

$$\hat{G}(s) = argmax_{j\in\{1,\ldots,m\}}\{\frac{1}{3}(g_{1j}^s + g_{2j}^s + g_{3j}^s)\}$$



(a) NP 3D allongé (b) NP 3D planaire (c) NP 3D volumineux

FIGURE 2 – Exemple de trois dispersions géométriques possibles d'un nuage de points (NP) 3D.

L'algorithme 1 donné ci-après reprend les différentes étapes du calcul du gradient structurel basé sur la combinaison couleur/texture non paramétrique proposée.

Maintenant que nous avons détaillé les différents descripteurs de couleur et de texture utilisés dans le processus de combinaison ainsi que notre méthode de combinaison couleur/texture non paramétrique basée sur l'analyse des composantes principales et permettant d'obtenir un gradient structurel en niveau de gris, nous allons détailler la technique de segmentation par Ligne de Partage des Eaux faisant appel au gradient structurel calculé.

Entrées: I une image couleur.
Calculer les différents descripteurs (m) sur l'image I en considérant l'espace couleur $E_1E_2E_3$.
Répéter

1. Générer l'ensemble des données

$$P^s = [\vec{g^s}_{.1}, \vec{g^s}_{.2}, \ldots, \vec{g^s}_{.j}, \ldots, \vec{g^s}_{.m}]$$

2. Calculer la matrice locale

$$\mathcal{COV}_P^s = \begin{pmatrix} var(\vec{g^s}_{1.}) & cov((\vec{g^s}_{1.})^t, \vec{g^s}_{2.}) & cov((\vec{g^s}_{1.})^t, \vec{g^s}_{3.}) \\ cov((\vec{g^s}_{2.})^t, \vec{g^s}_{1.}) & cov((\vec{g^s}_{2.})^t, \vec{g^s}_{2.}) & cov((\vec{g^s}_{2.})^t, \vec{g^s}_{3.}) \\ cov((\vec{g^s}_{3.})^t, \vec{g^s}_{1.}) & cov((\vec{g^s}_{3.})^t, \vec{g^s}_{2.}) & var(\vec{g^s}_{3.}) \end{pmatrix}$$

3. Calculer les matrices D^s et L^s.

$$D^s = \begin{pmatrix} \lambda_1^s & 0 & 0 \\ 0 & \lambda_2^s & 0 \\ 0 & 0 & \lambda_3^s \end{pmatrix} \quad \text{et} \quad L^s = [\vec{\mathcal{V}}_1^s \quad \vec{\mathcal{V}}_2^s \quad \vec{\mathcal{V}}_3^s]$$

4. Extraire les valeurs propres λ_1^s, λ_2^s et λ_3^s et leurs vecteurs propres associés $\vec{\mathcal{V}}_1^s$, $\vec{\mathcal{V}}_2^s$ et $\vec{\mathcal{V}}_3^s$.
5. **Suivant** *la valeur de* $\{\lambda_i^s\}_{1\leq i\leq 3}$ **faire**
   - **Cas où** $\lambda_1^s \gg \lambda_2^s$
     $$\hat{G}(s) = argmax_{j\in\{1,\ldots,m\}}\{g_{3j}^s\}$$
   - **fin**
   - **Cas où** $\lambda_3^s \ll \lambda_2^s$
     $$\hat{G}(s) = argmax_{j\in\{1,\ldots,m\}}\{g_{1j}^s + g_{2j}^s\}$$
   - **fin**
   - **Cas où** $\lambda_1^s \simeq \lambda_2^s \simeq \lambda_3^s$
     $$\hat{G}(s) = argmax_{j\in\{1,\ldots,m\}}\{\tfrac{1}{3}(g_{1j}^s + g_{2j}^s + g_{3j}^s)\}$$
   - **fin**

fin
Jusqu'à *pour chaque pixel* s *de* I;
Sorties: Gradient structurel $\hat{G}$
Algorithm 1: Méthode de combinaison couleur/texture non paramétrique proposée

2.3 La Ligne de Partage des Eaux

La technique de Ligne de Partage des Eaux (LPE) issue de la morphologie mathématique est l'une des méthodes de segmentation morphologique les plus communément utilisées dans la littérature. Cette technique consiste à traiter l'image à segmenter comme un relief (surface topographique) tels que les niveaux de gris de l'image constituent son altitude. En effet, la méthode LPE est à la base une notion interprétée par analogie géographique comme le complémentaire des bassins versants [14]. En effet, Angelini et al. définissent un bassin versant comme étant la zone associée à un minimum régional telle une goutte d'eau tombant dans cette zone et suivant la ligne de plus grande pente s'arrêtera dans ce minimum [14]. L'illustration de ce phénomène est donnée par la figure 3.

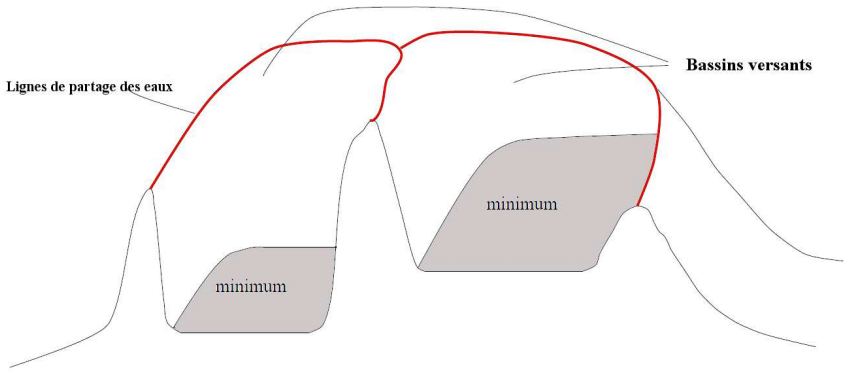


Figure 3 – Bassins versants et ligne de partage des eaux [15].

Le paradigme d'immersion introduit par Vincent et Soille est l'un des algorithmes les plus connu pour la construction de LPE. Ce paradigme consiste à remplir les bassins versants (obtenus par les minimums régionaux) pour arriver à leurs limites (qui constituent la LPE). Des travaux récents permettent de donner des définitions plus rigoureuses de la LPE [16].

L'avantage de la méthode LPE en tant que méthode de segmentation est qu'elle permet de fournir des régions bornées par des contours fermés. Dans le cas usuel, les minimums régionaux qui définissent les bassins versants sont souvent déterminés par le gradient de l'image (les contours). En appliquant la LPE sur le gradient, les bassins versants devraient théoriquement correspondre à des régions homogènes de cette image. Après avoir déterminé les contours de ces bassins (qui forment les minima locaux dans l'image), nous obtenons une carte de régions. La méthode LPE la plus répandue utilise en entrée deux images : une image de potentiel (gradient) et une image de germes [15, 17].

Dans la pratique, cette transformation produit une importante sur-segmentation due au bruit ou aux irrégularités locales dans l'image de potentiel. En effet, ces irrégularités influencent le nombre de germes (généralement le nombre de minima locaux) qui devient souvent très important et engendre la sur-segmentation en question. Plusieurs alternatives ont été proposées dans la littérature pour remédier àăcet inconvénient. Une première alternative consiste à utiliser une méthode de Ligne de Partage des Eaux hiérarchique non paramétrique, connue également sous le nom d'algorithme de Waterfall [18]. Pour cette raison, différents gradients ont été définis [19, 20]. L'inconvénient majeure de cette solution réside dans le temps de traitement qui s'avère important et nécessite la définition d'un niveau optimal de la hiérarchie (autrement dit celui qui offre la meilleure segmentation). Une deuxième alternative consiste à agir sur l'image de germes. C'est la raison pour laquelle, nous avons proposé [21] dans de précédent travaux, une solution

permettant de réduire le nombre de minima locaux dans l'image en fonction de deux paramètres fixés expérimentalement.

La figure 4 illustre sur deux images du projet CAPLOC, les résultats de segmentation obtenus par LPE avec la combinaison couleur/texture par ACP proposée. Nous pouvons en déduire que la méthode de segmentation proposée préserve les différentes zones d'intérêt de l'image et que la qualité des contours est très satisfaisante. A titre d'information, la qualité de segmentation obtenue avec l'approche proposée, à permis d'obtenir un taux de bonne classification ciel/non-ciel de 97,92%. Une fois l'image segmentée par cette stratégie, il reste à classer les différentes régions en deux classes : bâtiment/non-bâtiment (ou plus : ciel, végétation, bâtiment). Ceci peut être réalisé à l'aide de différents classifieurs non-supervisés ou supervisés, comme ceux utilisés dans la sous-tache 1.2 pour la détection du ciel/non-ciel. La seule différence ici, est que nous ne considérons plus le pixel mais la région. Néanmoins, devant le temps calculatoire important nécessaire pour parvenir à cette segmentation de l'image (et donc incompatible avec la contrainte temps réel visée à moyen/long terme pour les objectifs de la tâche 1), nous ne procéderons pas à cette étape de classification, mais faisons le choix de nous orienter, dans la section suivante, vers une stratégie plus rapide qui consiste à détecter directement les zones de bâtiments dans l'image en se basant sur une détection de segments.

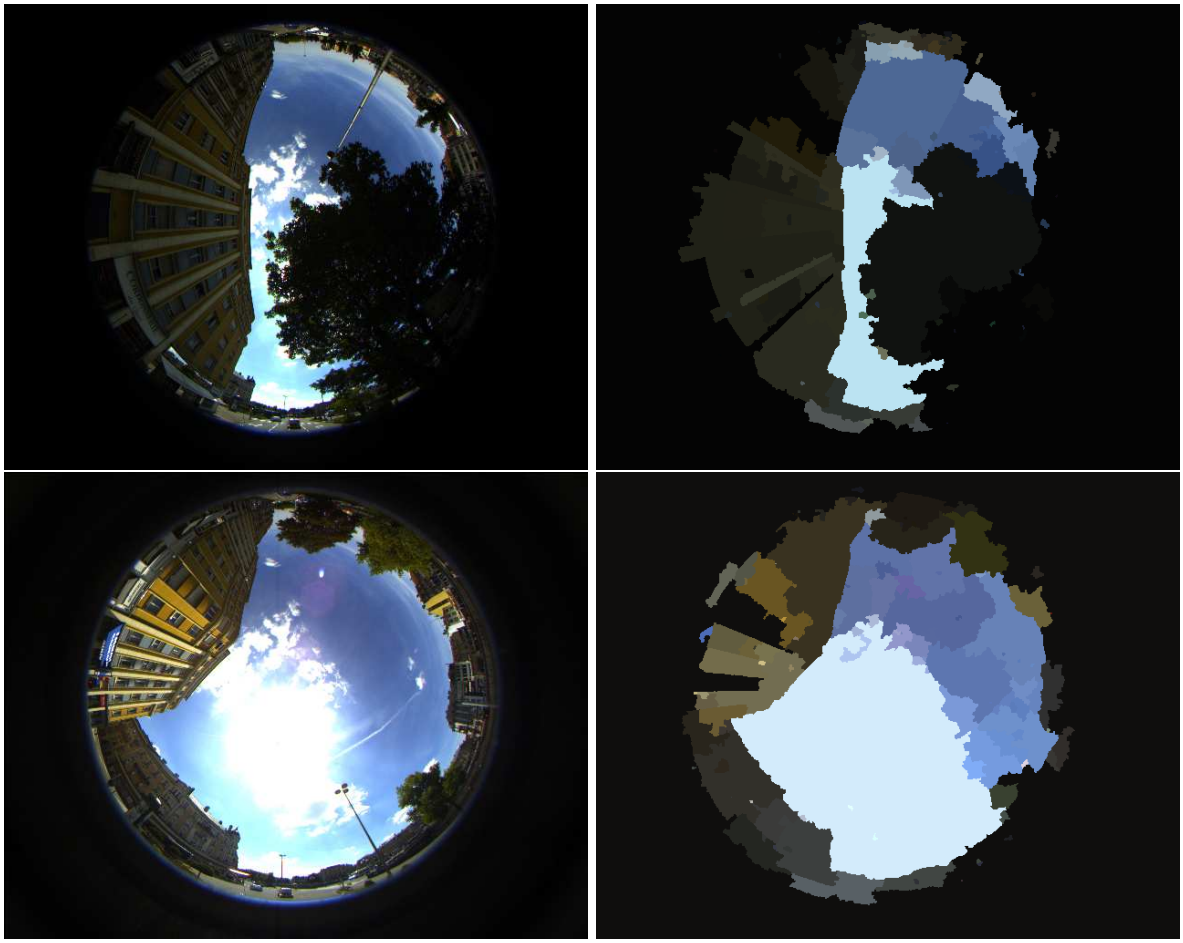


FIGURE 4 – Résultats de segmentation avec la combinaison couleur/texture proposée (droite), sur deux images de la base du projet CAPLOC (gauche).

3 Stratégie de détection de bâtiments par analyse de segments

La stratégie de détection de bâtiments que nous détaillons ci-après est décrite en détails dans le rapport de stage de projet de fin d'étude d'ingénieur de Christophe Coniglio [2]. Elle s'articule autour d'une détection de segments sur laquelle nous établissons deux hypothèses. La première est qu'un bâtiment, par sa constitution, contient un nombre important de segments. La deuxième est que le ratio entre le nombre de segments d'un bâtiment et le nombre de segments total de l'image est proche de 1. Nous avons donc statistiquement une probabilité importante de trouver un bâtiment dans une zone à forte concentration de segments. La stratégie de détection de bâtiments que nous avons développée est composée de cinq étapes décrites en détails ci-après : 1/ Détection des segments dans l'image ; 2/ Création d'un maillage simplifié ; 3/ Conversion en une carte de régions ; 4/ Création d'une carte de densité de segments ; 5/ Classification des régions en deux classes : bâtiments/non-bâtiment.

3.1 Détection de segments

L'étape de détection de segments est l'étape principale de la stratégie globale de détection de bâtiments, puisque le résultat final dépend en grande partie de la qualité de cette détection de segments. Comme introduit ci-dessus, nous faisons l'hypothèse qu'il existe une forte densité de segments dans une zone de bâtiments et que la proportion de segments détectés dans une zone de bâtiments est nettement plus importante que dans les autres zones réunies. De nombreuses méthodes de détection de segments existent dans la littérature, mais nombre d'entre elles voient leur paramètres optimisés pour une application visée. Dans cette optique, nous avons réalisé une analyse de trois méthodes très utilisées (Transformée de Hough, Dseg, LSD) en fonction d'une de nos contraintes applicatives, à savoir le capteur optique utilisé : une caméra couleur avec un objectif fish-eye (offrant un champ de vision de plus de 180 degrés mais avec des distorsions importantes).

3.1.1 Transformée de Hough

La transformée de Hough est une technique de reconnaissance des formes, qui permet de détecter des formes et notamment des lignes dans une image. Elle fait appel à une image binaire de contours généralement obtenue à partir de mesures de gradients locaux sur l'image initiale. Dans le cas d'une détection de ligne, chaque point de contours (x,y) va être converti dans le domaine de Hough et donc représenté par une courbe (sinusoïde). Si les sinusoïdes associées à deux points se coupent, l'endroit où elles se coupent dans l'espace de Hough correspond aux paramètres d'une droite qui relie ces deux points. Pour déterminer l'image de contours, nous avons choisi de calculer deux gradients simples à savoir : Canny et Sobel (en ne considérant que l'amplitude du gradient).

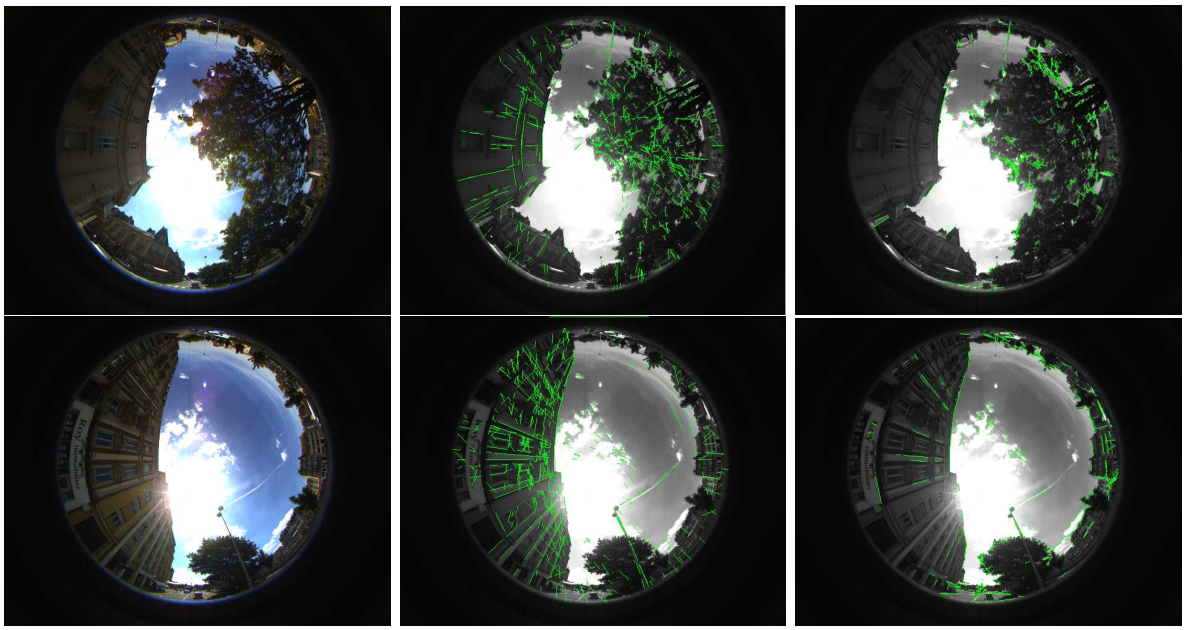


FIGURE 5 – Détection de segments par transformée de Hough avec un gradient de Canny (milieu) et de Sobel (droite) sur deux images acquises (gauche).

La figure 5 illustre le résultat de la détection de segments par transformée de Hough avec un gradient de Canny et de Sobel sur deux images représentatives de notre base. Nous pouvons constater que le gradient de Canny détecte quasiment autant de segments dans les différentes zones de l'image comparé à celui de Sobel qui lui favorise les zones de végétation. En effet, en moyenne, nous détectons 45% de segments dans les zones de bâtiments avec le gradient de Canny (VS 10% avec le gradient de Sobel) et 55% de segments dans les autres zones de l'image (VS 90% avec le gradient de Sobel). Notons que ces résultats ont été obtenus avec une image couleur convertie en niveau de gris, mais que l'utilisation d'une seule des trois composantes (R,V,B) n'améliore pas les résultats. En conclusion, cela n'est pas en phase avec notre hypothèse de départ qui est d'obtenir davantage de segments dans les zones de bâtiments que dans les autres zones de l'image. Le problème provient probablement de l'image binaire obtenue avec le gradient de Canny ou de Sobel fourni en entrée de la transformée de Hough. En effet le gradient favorise les optimums locaux qui se situent aux niveaux des contours mais aussi dans les zones bruitées comme la végétation, raison pour laquelle nous trouvons énormément de segments dans ces zones.

3.1.2 Dseg : Détection directe de segments dans une image

La méthode Dseg (qui signifie "Détection directe de segments dans une image") est une méthode de détection de segments qui se veut peu gourmande en ressources et donc rapide. Elle repose sur une détection de points d'intérêts (calculés sur une image de gradient) considérés comme points de départ d'un segment avec comme direction, la phase du gradient. Nous déterminons ensuite tous les pixels appartenants au segment considéré. Une étape de post-traitement visant à fusionner les segments proches conclut le processus. La figure 6 illustre le résultat de la détection de segments réalisée par la méthode Dseg avec un gradient de Canny et de Sobel.

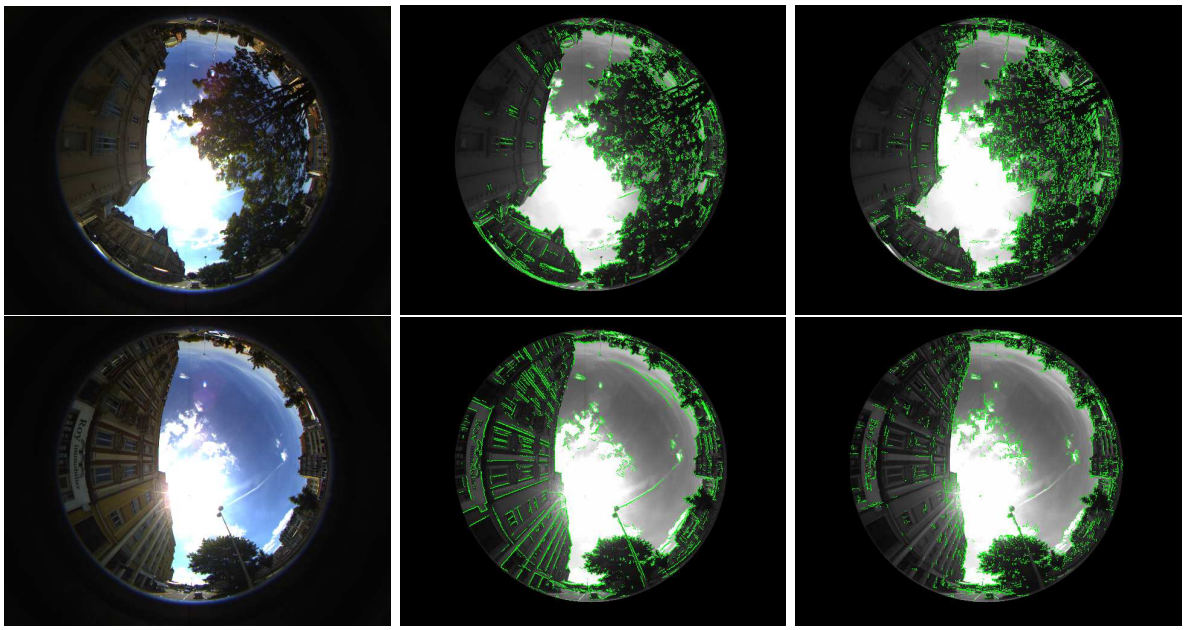


FIGURE 6 – Détection de segments par l'algorithme Dseg avec un gradient de Canny (milieu) et de Sobel (droite) sur deux images acquises (gauche).

D'après cette figure, nous pouvons constater que le gradient de Canny détecte davantage de segments dans les zones de bâtiments comparé au gradient de Sobel. Notons que, comme pour la transformée de Hough, de nombreux segments sont détectés en dehors des zones de bâtiments, principalement dans les zones de végétation. Les segments provenant préalablement de points d'intérêt, il est certain que plus nous aurons de points d'intérêt dans les zones de végétation, plus nous aurons de segments détectés dans ces zones. Revenons alors sur la définition des contraintes pour qu'un pixel de l'image soit considéré comme un point d'intérêt.

$$G_{i,j} - G_{i+cos(\phi_{i,j}),j+sin(\phi_{i,j})} > \Gamma_{gradmax} \tag{18}$$

$$G_{i,j} - G_{i-cos(\phi_{i,j}),j-sin(\phi_{i,j})} > \Gamma_{gradmax} \tag{19}$$

$$G_{i,j} > G_{i-cos(\phi_{i,j}),j+sin(\phi_{i,j})} \tag{20}$$

$$G_{i,j} > G_{i+cos(\phi_{i,j}),j-sin(\phi_{i,j})} \tag{21}$$

où $G_{i,j}$ correspond à l'amplitude du gradient au pixel (i,j), $\phi_{i,j}$ est la phase du gradient au pixel (i,j), et $\Gamma_{gradmax}$ une constante. Ces contraintes prennent uniquement en compte le gradient des pixels voisins proches. Un point d'intérêt est en réalité un optimum local, ce qui implique de le retrouver particulièrement dans les zones de végétation de nos images.

A l'état brut, l'algorithme DSeg ne semble pas être adapté au type d'images que nous rencontrons dans notre application. Néanmoins, les points d'intérêt issus des zones de végétation sont généralement arrêtés rapidement par le filtre de Kalman et offrent des segments de petites tailles. Une solution consiste alors à filtrer les segments les plus petits de sorte à conserver principalement les segments pouvant appartenir à des zones de bâtiments. Nous constatons que les résultats sont améliorés si nous supprimons les segments d'une longueur inférieure à 6 pixels. En effet, comme l'illustre la figure 11, le nombre de segments supprimés en dehors des zones de bâtiments est largement supérieur aux nombre de segments supprimés dans ces zones.

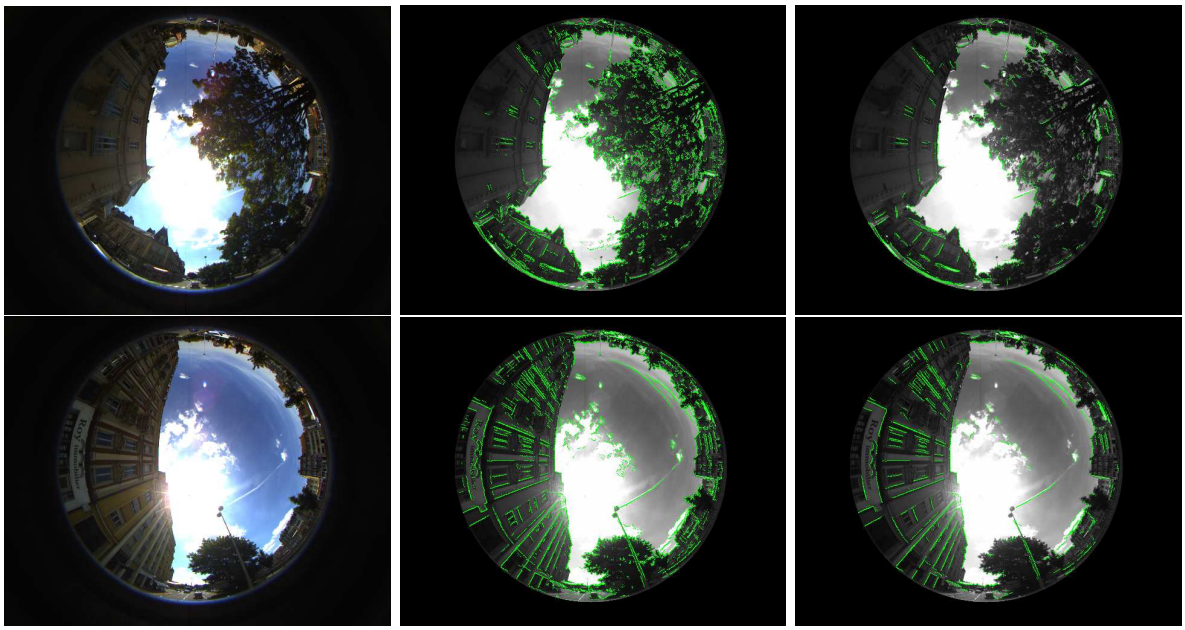


FIGURE 7 – Illustration de la suppression des segments de faible longueur (droite) suite à une détection de segments par l'algorithme Dseg (milieu) sur deux images acquises (gauche).

Nous pouvons aussi noter que la composante couleur sur laquelle la détection est réalisée a son importance. En effet, suite à une série de tests, nous pouvons conclure que les résultats de détection de segments obtenus avec les composantes verte et rouge sont assez satisfaisants et proches, mais médiocres avec la composante bleue. Le nombre de segments détectés diminue fortement et le ratio de segments détectés dans les zones de bâtiments par rapport aux autres zones est moins élevé lorsque la composante bleue est utilisée. En moyenne, sur les composantes rouge et verte, nous détectons 60% de segments dans les zones de bâtiments avec le gradient de Canny (VS 25% avec le gradient de Sobel) et 40% de segments dans les autres zones de l'image (VS 75% avec le gradient de Sobel). En conclusion, les résultats de détection de segments obtenus avec l'algorithme Dseg (couplée à une suppression des segments de petites tailles) sont meilleurs que ceux obtenus avec la transformée de Hough, mais ne nous satisfont pas complètement, d'autant plus que la méthode semble inefficace dans les zones sombres, ce qui n'est pas sans poser problème lorsque les conditions de luminosité ne sont pas idéales. Nous allons donc dans la section suivante tester un dernier algorithme appelé LSD.

3.1.3 LSD : a Line Segment Detector

La méthode LSD (qui signifie "a Line Segment Detector") est une technique de détection de segments qui se concentre sur l'orientation du gradient et une sélection stricte des segments détectés. La première étape de l'algorithme consiste à regrouper des pixels ayant une orientation de gradient proche. Chaque groupe de pixels supposé constitué un segment est validé comme tel après une série de tests capables de détecter des groupes de pixels correspondant à du bruit pour ensuite les éliminer de la sélection.

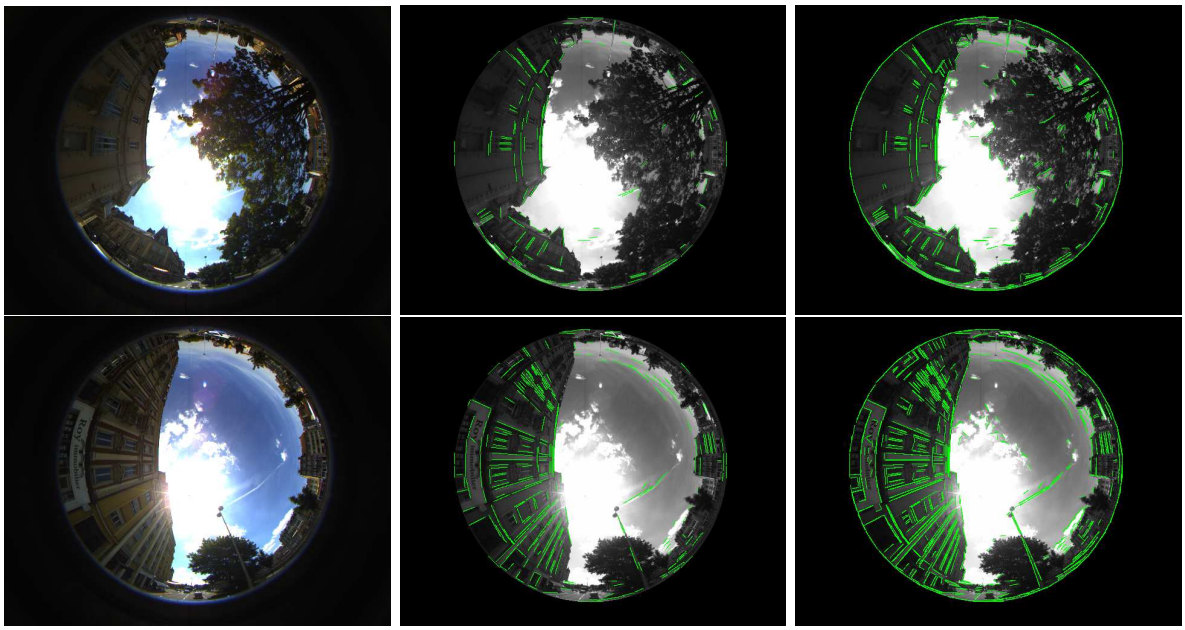


FIGURE 8 – Détection de segments par l'algorithme LSD (milieu) avec utilisation d'un redimensionnement de l'image (droite) sur deux images acquises (gauche).

Les résultats de détection obtenus, avec cet algorithme, dans le cadre de notre application sont prometteurs puisque nous avons un grand nombre de segments détectés dans les zones de bâtiments et peu dans les autres zones de l'image (végétation, etc). La série de tests réalisés met en avant que l'utilisation du gradient de Sobel en comparaison avec le gradient proposé dans LSD améliore significativement les résultats déjà très satisfaisants. Comme cela est suggéré dans la littérature, nous constatons également que les résultats de détection de segments par cet algorithme sont encore une fois améliorés lorsque un/plusieurs redimensionnements de l'image sont utilisés. En effet, comme l'illustre la figure 8, le fait d'effectuer une détection de segments sur une image redimensionnée, fusionnée à celle réalisée sur l'image initiale, permet d'augmenter le nombre total de segments détectés, ce qui renforce l'une de nos hypothèse de départ qui est d'obtenir davantage de segments dans les zones de bâtiments que dans les autres zones de l'image. Comme pour l'algorithme Dseg, la composante couleur sur laquelle la détection est réalisée a son importance, puisque les résultats de détection de segments obtenus avec les composantes verte et rouge sont assez satisfaisants et proches, mais médiocres avec la composante bleue. En effet, le nombre de segments détectés à partir de cette composante est divisé par trois. Notons aussi que le gradient de Sobel fournit de meilleurs résultats que le gradient de Canny. En moyenne, nous détectons 85% de segments dans les zones de bâtiments avec le gradient de Sobel (VS 80% avec le gradient de Canny) et 15% de segments dans les autres zones de l'image (VS 20% avec le gradient de Canny). Notons aussi que le redimensionnement de l'image permet d'augmenter avec ce gradient, le nombre de segments détectés de 400 à 1300 en moyenne. Ces résultats qui confortent nos hypothèses de départ, sont donc très satisfaisants et permettent de conclure que c'est cette dernière approche qui sera conservée pour la détection de segments dans le cadre de notre application et donc dans la suite de ce rapport.

3.2 Segmentation de l'image

3.2.1 Création d'un pré-maillage

La segmentation consiste à partitioner une image en régions homogènes c'est-à-dire en un ensemble de points de l'image ayant des propriétés communes. L'étape de segmentation (qui in

fine regroupera les pixels pour former des régions) est primordiale pour notre application puisqu'elle influencera directement la qualité/précision de notre classification finale. Si nous nous concentrons sur les résultats de détection de segments ci-dessus, nous constatons que la majorité des contours des bâtiments sont détectés comme des segments (ce qui est dû à la forme géométrique rectiligne de ces derniers). Nous avons donc une forte densité de segments détectés dans les bâtiments mais aussi sur leurs contours. Il est donc utile de relier différents segments afin de créer des formes avec des contours fermés. Celles-ci seront ensuite étiquetées pour créer une carte de régions coïncidant avec des zones de bâtiments.

La connexion de segments s'apparente à la création d'un maillage. Les extrémités de segments comprisent dans un voisinage donné sont reliées par un nouveau segment créé. La distance maximale entre ces deux extrémités est calculé par rapport à la longueur du segment testé, de sorte à favoriser les plus grands segments. Ces grands segments étant généralement détectés sur les contours des bâtiments, les privilégier augmente les chances de fermer correctement ces contours (ce qui s'avère d'autant plus difficile lorsque la détection est de qualité moindre). Le résultat du maillage relève une concentration importante de régions de petites tailles dans les zones de bâtiments (dû à une forte densité de segments ne pouvant être reliés) contrairement aux autres zones de l'image (végétation) qui laissent apparaître peu de régions mais avec des surfaces plus importantes. La figure 9 illustre le résultat de différents maillages sur une détection de segments réalisée par l'approche retenue et détaillée dans la section précédente.



FIGURE 9 – Création d'un maillage léger (milieu) et d'une maillage fort (droite) à partir d'une image de détection de segments réalisée via de l'algorithme LSD utilisant différents redimensionnements de l'image (gauche).

3.2.2 Simplification du maillage par élagage

La création du pré-maillage à partir d'un nombre important de pixels segments peut provoquer des amas de segments qui peuvent aller jusqu'à dégrader la qualité de la segmentation de l'image en créant des frontières épaisses (supérieures à 3 pixels) entre les régions. Pour pallier ce problème, nous proposons de corriger (élaguer) le maillage réalisé à l'étape précédente en convertissant les pixels identifiés comme segments et reliés entre eux dans un voisinage en 4-connexité en pixels neutres. Cette règle a pour effet de diminuer l'épaisseur des frontières des régions à 2 pixels maximum et de créer une multitude de petites régions de surface égale à 1 pixel dans les zones à forte concentration de segments et donc généralement associées à des bâtiments. Ceci correspond à l'objectif de segmentation souhaité qui est d'obtenir une sur-segmentation des zones de bâtiments. La figure 10 illustre l'élagage d'un maillage réalisée à partir d'une détection de segments via l'algorithme LSD utilisant plusieurs redimensionnements de l'image.

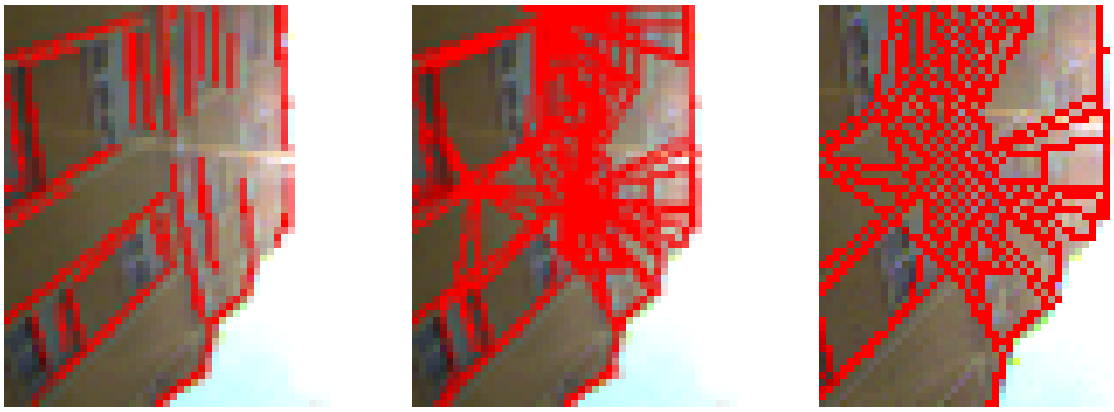


FIGURE 10 – Illustration de la correction (droite) d'un maillage (milieu) réalisé à partir d'une détection de segments (gauche).

3.2.3 Suppression des segments indésirables

La stratégie employée pour maximiser la détection de segments dans les zones de bâtiments et minimiser la détection dans les autres zones de l'image donnent des résultats satisfaisants. Néanmoins, il est impossible de supprimer complètement les segments détectés en dehors des zones de bâtiments. Les segments détectés dans les zones de végétation ne posent pas vraiment de problème mais ceux détectés dans les zones de ciel (et principalement à cause des nuages) sont plus problématiques, raison pour laquelle, nous présentons ci-après deux stratégies pour les supprimer. En effet, les segments détectés dans les zones de ciel induisent des erreurs de classification car l'opération de création de maillage aurait tendance à relier les segments provenant des bâtiments et ceux provenant des zones de ciel proches des bâtiments. Cela a pour conséquence de créer une sur-segmentation du ciel et provoquer des erreurs sur la détermination des frontières (entre les zones de ciel et de bâtiments).

La première solution que nous proposons pour différencier les segments contenus dans les zones de ciel et dans les zones de bâtiments consiste à les filtrer par rapport à leur orientation. En effet, nous savons que nous retrouvons un grand nombre de rectangle dans les bâtiments (les fenêtres par exemple), c'est pourquoi, nous choisissons de conserver uniquement les segments horizontaux et verticaux. Les caractéristiques de l'objectif fisheye utilisé nécessite d'opérer des calculs différents pour chaque type de segments. Les segments verticaux (les arrêtes entre deux façades de bâtiments par exemple) sont en direction du centre de l'objectif. Pour calculer si un segment est vertical, nous calculons l'angle formé entre le dit segment et un segment parfaitement vertical que nous définissons, puis regardons si l'angle formé est inférieur à 5 degrés. A l'inverse, les segments horizontaux se retrouvent perpendiculaires au segment passant par le centre. Nous effectuons alors le même calcul mais gardons cette fois-ci les segments avec une différence d'angle supérieure à 85 degrés. Cette première stratégie ne faisant appel à aucun connaissance à priori sur l'image, ni à une détection du ciel dans l'image, est intéressante car elle permet d'améliorer les résultats de classification, mais reste peu robuste en raison de la forte probabilité qu'un segment situé dans une zone de ciel soit horizontal ou vertical.

La deuxième solution que nous proposons tient compte de la détection du ciel réalisée dans nos travaux précédents. Nous pouvons alors soit l'utiliser comme un masque et ne considérer pour la détection de segments que les pixels non-ciel, soit supprimer les segments ayant une de leur extrémités présente dans une zone de ciel.

La figure 11 illustre le résultat d'une détection de segments sans étape de suppression de segments indésirables (et donc localisés dans les zones de ciel), avec une étape de suppression de segments horizontaux/verticaux, et avec une étape de suppression de segments basée sur une détection préalable du ciel. Nous pouvons conclure que la dernière stratégie proposée permet d'obtenir de très bons résultats mais nécessite une détection de ciel efficace et robuste.



FIGURE 11 – Illustration d'une détection de segments sans étape de suppression de segments indésirables et donc localisés dans les zones de ciel (gauche), avec une étape de suppression de segments horizontaux/verticaux (milieu), et avec une étape de suppression de segments basée sur une détection préalable du ciel (droite).

3.2.4 Création d'une carte de régions

Une fois cette étape de suppression de segments indésirables effectuée, nous disposons d'une segmentation de l'image utilisable (réalisée globalement à partir d'une détection de segments couplée à la création d'un maillage). Il reste néanmoins à étiqueter les différentes zones détectées en régions disposant chacune d'un numéro d'identification qui leur est propre. L'image en entrée de ce processus est une image binaire (le pixel est neutre ou fait partie d'un segment). La première étape de la procédure d'étiquetage consiste à regrouper tous les pixels neutres d'une même région en leur donnant le même numéro d'identification. Tous les pixels neutres d'une même région obtenus par croissance de régions se voient donc attribuer la même étiquette que le pixel initial. Il reste donc à définir l'appartenance des pixels segments à une région.Par définition, il n'est pas possible de connaître l'appartenance d'un pixel à une région puisque les pixels segments ne disposent pas de cette information. Prenons l'exemple où deux régions sont séparées par une frontière (de largeur supérieure à 1 pixel), il n'est pas possible d'identifier à quelle région appartient cette frontière. Deux choix sont possibles : le premier est d'affecter le pixel considéré à une nouvelle nouvelle région ; le second est d'affecter (selon une règle donnée) le pixel considéré à une région déjà existante. Etant donné que nous avons déjà réalisé un élagage suite à la création du maillage dans la partie détection de segments et que cet élagage a pour effet de créer des régions d'une surface d'un pixel dans les zones à forte densité de segments, nous optons pour le deuxième cas qui affecte (selon une règle donnée) le pixel considéré à une région existante.

Bien que l'élagage est pour effet de réduire les frontières, le choix de l'attribution d'un pixel à une région reste essentiel pour optimiser la classification finale. Nous proposons alors deux méthodes que nous détaillons ci-après : La première méthode consiste à ne pas prendre en compte les caractéristiques des bâtiments ni même le matériel d'acquisition utilisé (une caméra couleur à objectif fisheye) et d'attribuer la même importance à chaque région. Le principe est simple et consiste alors à parcourir chaque région puis de la faire croitre d'un pixel jusqu'à avoir complètement traité tous les pixels segments. La deuxième méthode plus évoluée repose sur l'hypothèse que les bâtiments reposent sur des fondations (reliées au sol) et donc de favoriser, dans le processus d'agrégation d'un pixel à une région, les régions qui se trouvent les plus proches du sol, de

sorte à augmenter la précision des frontières situées entre le ciel et les bâtiments. En pratique, cela consiste à parcourir l'image du sol vers le ciel et d'affecter les pixels segments aux régions rencontrées précédemment et ainsi favoriser les régions les plus proches du sol. Il en va sans dire que nous devons prendre en compte la déformation de l'image (due à l'objectif fisheye) lors du parcours de celle-ci. D'une manière simplifiée, cela revient à parcourir chaque rayon d'un cercle en favorisant les pixels les plus éloignés de son centre. Les différents tests réalisés permettent d'attribuer un léger avantage à la première méthode.

La figure 12 illustre la carte de régions associée à une détection de segments réalisée sur une image acquise en mobilité à l'aide de la première méthode proposée.

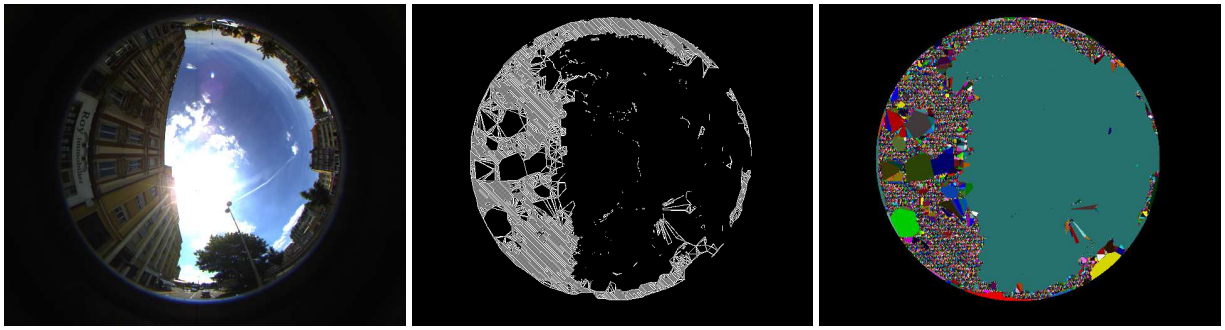


FIGURE 12 – Illustration d'une carte de régions (droite) associée à une détection de segments (milieu) réalisée sur une image acquise en mobilité (gauche).

3.3 Classification des régions en bâtiment/non-bâtiment

3.3.1 Carte de densité de segments

La classification finale que nous allons présenter dans la section suivante requiert une analyse de la densité de segments autour de chaque pixel qui est relativement long. Pour ne pas compliquer l'opérateur de classification, nous avons choisi d'exporter cette opération que nous présentons ci-dessous.

Calculer le nombre de pixel de segment pour chaque pixel et pour une distance donnée s'avère fastidieux. Nous recherchons alors uniquement les pixels segments, puis incrémentons par pas de 1, tous ceux qui se situent dans un voisinage donné. Pour ce faire, une distance euclidienne est utilisée. Il est assez difficile de se faire une idée de la distance optimale pour chaque pixel segment. Néanmoins, les tests que nous avons réalisés nous permettent de conclure qu'une valeur de 42 pixels donnent les meilleurs résultats, soit environ 1/3 de la hauteur moyenne des bâtiments. La figure 13 illustre l'influence de la valeur de distance choisie sur le calcul d'une carte de densité de segments. Il est important de noter que la carte de densité a été normalisée (plus la densité est élevée, plus les pixels tendent vers le blanc).

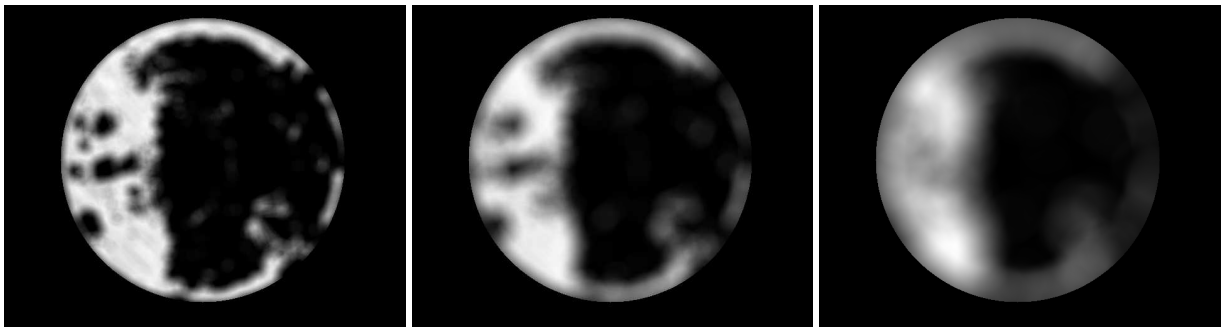


FIGURE 13 – Création d'une carte de de densité de segments avec des valeurs de distance de 11 (gauche), 22 (milieu) et 42 pixels (droite).

3.3.2 Classification bâtiment VS non-bâtiment

L'étape de classification constitue l'étape finale de notre processus de détection de bâtiments. Il s'agit de définir, pour chaque pixel, à partir de la carte de régions et de la carte de densité de segments toutes deux calculées précédemment, sa classe d'appartenance parmi les deux classes possibles que sont : bâtiment et non-bâtiment. L'utilisation de la carte de densité de segments permet d'associer à chaque région, une densité moyenne de segments (en divisant la somme de densité par le nombre de pixels présents dans la région traitée). Pour optimiser la prise de décision sur la classe d'appartenance associée à chaque région de l'image, le processus de classification contient quatre étapes que nous détaillons ci-dessous.

En premier lieu, nous faisons l'hypothèse que nous avons toujours la présence de bâtiments dans l'image puisque l'application visée concerne l'amélioration de la précision de localisation d'un mobile en milieu urbain dense. Il est néanmoins possible que selon leur architecture, le nombre de segments détectés sur les bâtiments varie, c'est pourquoi, nous analysons la moyenne de densité de segments des différentes régions. Nous rappelons notre hypothèse de départ qui a pu être vérifiée précédemment et qui consiste à dire qu'il y a davantage de segments dans les zones de bâtiments que dans les autres zones de l'image (comme les zones de végétation par exemple). On peut alors considérer que les régions dont la densité de segments moyenne est significativement plus élevée que la densité de segments moyenne de l'image correspondent à des zones de bâtiments. Cette première étape de classification comporte logiquement de nombreuses erreurs, principalement dues aux regroupements de petites régions, qui de part leurs caractéristiques, ont une forte probabilité d'être classées comme bâtiment.
La deuxième étape du processus de classification consiste alors à supprimer certaines erreurs en isolant les petites régions. Cela revient à réorganiser l'étiquetage de la carte de régions, en regroupant toutes les régions connexes de même classe (pixel classé comme bâtiment ou comme non-bâtiment). Ceci a pour effet de diminuer fortement la moyenne de densité de segments des régions dites parasites. A l'inverse, les régions classées comme bâtiments voient leur moyenne augmentée grâce à l'agrégation des petites zones situées dans les bâtiments (obtenues suite à l'étape d'élagage du maillage) et dont la moyenne de densité de segments est très élevée.
Dans la troisième étape, nous répétons la sélection réalisée dans la première étape en supprimant les nouvelles régions dont la moyenne de densité de segments est trop faible, ce qui a pour effet de supprimer les petites régions isolées. Enfin, nous supprimons les régions parasites dont la surface est inférieure à 24 pixels.
La classification obtenue jusqu'alors donne des résultas intéressants en présence de bâtiments mais n'est pas en mesure de détecter l'absence de bâtiment. L'étape finale de notre processus de classification consiste à lever l'incertitude sur la présence de bâtiment ou non. Pour ce faire,

nous comparons la moyenne de densité de segments de chaque région supposée être considérée comme bâtiment, à une constante dont la valeur a été déterminée suite à de nombreux tests expérimentaux. Si la moyenne en question et inférieure à cette constante, la zone est alors considérée comme non-bâtiment.

La figure 14 illustre le résultat final d'une détection de bâtiment sur un exemple de deux images acquises en mobilité.

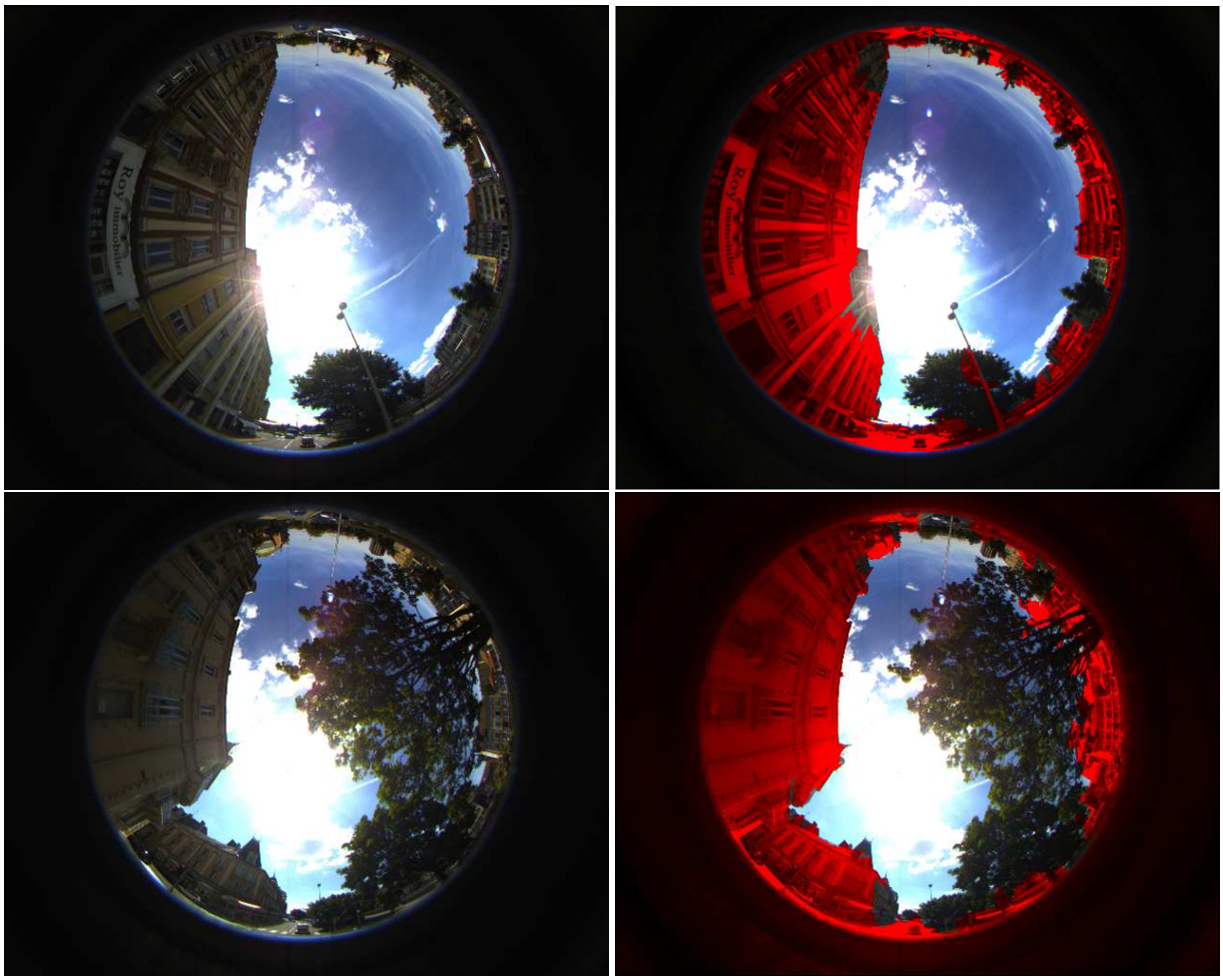


FIGURE 14 – Illustration de la détection de bâtiments (droite) sur deux images acquises en mobilité (gauche).

4 Conclusion

Ce rapport nommé L1.3 et dédié à la sous-tâche 1.3 avait pour objectif final de caractériser l'environnement de réception des signaux GNSS de manière plus fine que la détection ciel/non-ciel (réalisée dans la sous tâche 1.2), c'est-à-dire, pouvoir fournir en sortie de cette tâche 1, une détection du ciel/végétation/bâtiment sur des images couleur fisheye acquises en mobilité. Ceci pourra permettre à moyen terme de i) pondérer l'impact de chacun des satellites dans le calcul de position de localisation, selon s'ils se trouvent dans une classe « ciel », « végétation », « bâtiment », (tâche 3) ; ii) caractériser par des attributs pertinents certaines textures rencontrées afin de pouvoir reproduire la texture des plans du modèle 3D (tâche 2).

Dans cette sous-tâche, nous avons donc en premier lieu, présenté une méthode de segmentation d'images couleur basée sur une croissance de région (réalisée par l'algorithme de Ligne de Partage des Eaux) et utilisant un gradient structurel obtenu à partir d'une nouvelle méthode de combinaison couleur/texture adaptative et non paramétrique. Cette méthode est décrite en détails dans la thèse de Dhouha ATTIA, doctorante financée par la région Franche-Comté et mis à disposition pour la tâche 1 du projet CAPLOC. Les résultats de segmentation utilisés en entrée de l'étape de classification sont relativement satisfaisants en terme de qualité mais requiert un cout calculatoire important et donc incompatible avec la contrainte temps réel visée à moyen/long terme pour la totalité des objectifs de la tâche 1. Ceci étant, cette piste pourra être exploitée à long terme pour reproduire la texture des différentes surfaces à plaquer sur les plans du modèle 3D, dans la mesure où nous pourrons envisager d'implémenter ces algorithmes sur des architectures parallèles (type GPGPU) et ainsi garantir une rapidité de traitement.

Au vue de cette première conclusion, nous avons, dans un second temps, proposé une stratégie de détection de bâtiments basée majoritairement sur une analyse de segments. Cette stratégie est décrite en détails dans le rapport de stage de projet de fin d'étude d'ingénieur de Christophe Coniglio et se décomposent en cinq grandes étapes : 1/ une détection de segments réalisée à l'aide de l'algorithme LSD utilisant plusieurs redimensionnements de l'image ; 2/ la création d'un maillage de sorte à relier différents segments et créer des formes avec des contours fermés, son élagage ainsi que la suppression des segments indésirables situés dans les zones de ciel ; 3/ la conversion en une carte de régions ; 4/ la création d'une carte de densité de segments ; 5/ une étape de classification visant à attribuer, à chaque région, une classe d'appartenance parmi les deux classes possibles que sont : bâtiment et non-bâtiment. Cette stratégie de détection de bâtiments offrent de très bon résultats comme nous avons pu l'illustrer dans la dernière section de ce livrable. Toutefois, il est important de préciser que l'optimisation des paramètres reste bien évidement à ré-effectuer dans le cas où il y aurait de profonds changements dans les caractéristiques des images acquises en mobilité. Enfin, nous conclurons les travaux de ce livrable en précisant que cette stratégie de détection de bâtiments couplée à celle de détection de ciel (réalisée dans la sous-tâche 1.2), nous permet par soustraction d'identifier les zones de végétation présentes dans l'image et donc de disposer en sortie de la première tâche du projet CAPLOC, d'une caractérisation fine (ciel, végétation, bâtiment) de l'environnement de réception des signaux satellitaires.

Table des figures

1 Synopsis global de la combinaison couleur-texture basée sur l'ACP. 8
2 Exemple de trois dispersions géométriques possibles d'un nuage de points (NP) 3D. 11
3 Bassins versants et ligne de partage des eaux [15]. 13
4 Résultats de segmentation avec la combinaison couleur/texture proposée (droite), sur deux images de la base du projet CAPLOC (gauche). 14
5 Détection de segments par transformée de Hough avec un gradient de Canny (milieu) et de Sobel (droite) sur deux images acquises (gauche). 16
6 Détection de segments par l'algorithme Dseg avec un gradient de Canny (milieu) et de Sobel (droite) sur deux images acquises (gauche). 17
7 Illustration de la suppression des segments de faible longueur (droite) suite à une détection de segments par l'algorithme Dseg (milieu) sur deux images acquises (gauche). 18
8 Détection de segments par l'algorithme LSD (milieu) avec utilisation d'un redimensionnement de l'image (droite) sur deux images acquises (gauche). 19
9 Création d'un maillage léger (milieu) et d'une maillage fort (droite) à partir d'une image de détection de segments réalisée via de l'algorithme LSD utilisant différents redimensionnements de l'image (gauche). 20
10 Illustration de la correction (droite) d'un maillage (milieu) réalisé à partir d'une détection de segments (gauche). 21
11 Illustration d'une détection de segments sans étape de suppression de segments indésirables et donc localisés dans les zones de ciel (gauche), avec une étape de suppression de segments horizontaux/verticaux (milieu), et avec une étape de suppression de segments basée sur une détection préalable du ciel (droite). 22
12 Illustration d'une carte de régions (droite) associée à une détection de segments (milieu) réalisée sur une image acquise en mobilité (gauche). 23
13 Création d'une carte de de densité de segments avec des valeurs de distance de 11 (gauche), 22 (milieu) et 42 pixels (droite). 24
14 Illustration de la détection de bâtiments (droite) sur deux images acquises en mobilité (gauche). 25

Références

[1] D. Attia : *Segmentation d'images par combinaison adaptative couleur/texture et classification de pixels. Application a la caractérisation de lénvironnement de réception des signaux GNSS*. Thèse de doctorat, Université de Technologie de Belfort-Montbéliard, Octobre 2013. 4, 6

[2] C. Coniglio : Caratérisation d'environnement urbain par analyse de segments et de textures. Mémoire de D.E.A., Université de Technologie de Belfort-Montbéliard, Septembre 2012. 5, 15

[3] C. Duby et S. Robin : Analyse en composantes principales. Rapport technique, Institut National Agronomique Paris - Grignon, 2006. Département O.M.I.P, Institut National Agronomique Paris - Grignon. 6

[4] J. Gergaud : Unité fondamentale : Algèbre linéaire : une application l'analyse en composantes principales. Rapport technique, INP ENSAT, 2006. 6, 7

[5] A. A. Maciejewski et R. G. Roberts : An example of principal componenet analysis applied to correlated images. *IEEE Proceedins of the 33rd Southeastern Symposium*, 2001. 7

[6] C. Fredembach, M. Shroder et S. Susstrunk : Eigneregions for image classification. *IEEE Tansaction on Pattern Analysis and Machine Intelligence*, 26:1645–1649, 2004. 7

[7] Wei Zeng et Yan Zhang : A novel improvement to pca for image classification. *International conference on Computer science and service system (CSSS)*, pages 1964–1967, 2011. 7

[8] Pun-Mo Ho, Tien-Tsin Wong et Chi-Sing Leung : Compressing the illumination-adjustable images with principal componenet analysis. *IEEE Transaction on Circuits and Systems for Video Technology*, 15:355–364, 2005. 9

[9] M. Mudrova et A. Prochazka : Principal component analysis in image processing. Departement of Computing and Control Engineering, Institute of Chemical Technology, Prague. 9

[10] Q. Du, W. Zhu, H. Yang et J-E. Fowler : Segmented principal component analysis for parallel compression of hyperspectral imagery. *IEEE Geoscience and Remote Sensing Letters*, 6:713–717, 2009. 9

[11] L. Siroyich et M. Kirby : Low-dimensional procedure for the characterization of human faces. *Journal of Optical Society of America*, 4:519–524, 1987. 9

[12] M. Turk et A. Pentland : Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3:71–86, 1991. 9

[13] Alexander Ilin et Tapani Raiko : Practical approaches to principal componenet analysis in the presence of missing values. *Journal of Machine Learning Research 11*, 2010. 9

[14] E. Angelini, I. Bloch, Y. Gousseau, S. Ladjal, B. Pesquet-Popescu, M. Sigelle et F. Tupin : Morphologie mathématique. *Méthodes Avancées de Traitement dâĂŹImages SI 343 - MATIM*, chapitre 6, pages 95–126. Polycopié du cours SI343 - MATIM, 2007. 13

[15] L. Vincent et P. Soille : Watersheds in digital spaces : an efficient algorithm based on immersions simulations. *IEEE Trans. On Pattern Analysis and Machine Intelligence (PAMI)*, 13(16):583–598, 1991. 13, 27

[16] M. Schmitt et J. Mattioli : *Morphologie mathématique*. Paris, 1994. 13

[17] L. Shafarenko, M. Petrou et J. Kittler : Automatic watershed segmentation of randomly textured color images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 6(11):1530âĂŞ–1544, November 1997. 13

[18] S. Beucher : Watershed, hierarchical segmentation and waterfall algorithm. *Mathematical morphology and its applications to image and signal processing*, pages 69–76, 1994. 13

[19] Jesus Angulo et Jean Serra : Color segmentation by ordered mergings. *International Conference on Image Processing (ICIP)*, volume 2, pages 125–128, 2003. 13

[20] O. Lezoray, C. Meurie et A. Elmoataz : Graph-based ordering scheme for color image filtering. *International Journal of Image and Graphics*, 8(3):473–493, juillet 2008. 13

[21] Andrea Cohen, Dhouha Attia, Cyril Meurie et Yassine Ruichek : Une méthode de segmentation hybride par combinaison adaptative des informations texture et couleur. *MAnifestation des JEunes Chercheurs en Sciences et Technologies de l'Information et de la Communication (MajeSTIC10)*, 2010. 13



Groupe opérationnel n°2 du Predit 4

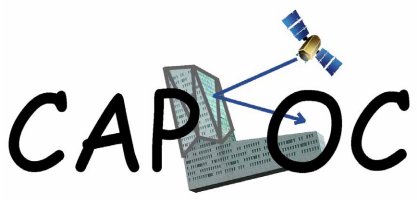
« Gestion du trafic et des mobilités »



CAPLOC

Combinaison de l'**A**nalyse d'images
et de la connaissance de la **P**ropagation
des signaux pour la **LOC**alisation

Convention 10-MT-CVS-001



Livrable de la tâche 2.1

Calcul du nuage de points 3D

18 octobre 2012

Sébastien AMBELLOUIS, Sébastien LEFEBVRE, Amaury FLANCQUART
Ifsttar/LEOST

Julien MOREAU, Yassine RUICHEK
UTBM/IRTES-SeT





Identification

Acronyme du projet	CAPLOC
Titre du projet	Combinaison de l'Analyse d'images et de la connaissance de la Propagation des signaux pour la LOCalisation
Coordinatrice du projet	Juliette MARAIS (Ifsttar-LEOST)
Date de début de projet	19 octobre 2010
Date de fin de projet	18 octobre 2014
Partenaires	Ifsttar/LEOST, UTBM/IRTES-SeT
Type de document	Livrable
Date contractuelle de livraison	18 octobre 2012
Accessibilité	La diffusion de ce document est limitée au programme PREDIT

Gestion des versions

Version	**Date**	**Auteurs**	**Modifications**
1.0	08/10/2012	Sébastien LEFEBVRE	Rédaction
1.1	12/10/2012	Julien MOREAU, Sébastien AMBELLOUIS	Rédaction
1.2	16/10/2012	Cyril MEURIE	Corrections
1.3	16/10/2012	Juliette MARAIS	Corrections
1.4	17/10/2012	Yassine RUICHEK	Corrections
VF	18/10/2012	Sébastien AMBELLOUIS	Approbation de la version finale

Table des matières

1 Introduction **4**

2 La reconstruction 3D d'environnements urbains **5**
2.1 Contexte du projet 5
2.2 Les systèmes actuels de reconstruction 3D d'environnements urbains 5
2.3 Les capteurs panoramiques et omnidirectionnels 7
2.4 Premiers résultats en stéréoscopie classique 10
2.5 Nouvelle approche proposée 12

3 Stéréovision avec des caméras à objectifs fisheye : état de l'art **14**
3.1 Principe de la géométrie épipolaire 14
3.2 Calibrage des caméras fisheye et rectification épipolaire 15
3.3 La mise en correspondance stéréoscopique d'images fisheye 17

4 Système CAPLOC et perception 3D **20**
4.1 Mise en œuvre matérielle du système 20
4.2 Géométrie épipolaire 21
4.3 Calcul du nuage de points 3D 25

5 Conclusion et perspectives de travail **33**

Table des figures **35**

Liste des tableaux **36**

Références bibliographiques **39**

1 Introduction

Le projet CAPLOC vise l'amélioration de la précision de localisation d'un véhicule en milieu urbain dense dans un contexte de réduction globale des coûts : l'idée est d'utiliser la perception vidéo ainsi que des outils de traitement d'images pour étudier l'environnement de propagation situé autour de l'antenne GNSS d'un récepteur faible coût. Ce projet de recherche amont s'inscrit dans le cadre d'un brevet [1] déposé par l'Ifsttar qui constitue un apport innovant pour la localisation précise. Deux briques technologiques liées à l'analyse d'images sont abordées dans ce projet.

La première vise à classifier en temps-réel l'état de réception de chacun des satellites reçus (soit directement soit après réflexion) sur la base d'un système mono-caméra. Les pixels de l'image sont segmentés et étiquetés en deux classes, « ciel » et « non-ciel », permettant ainsi de déterminer si le satellite, dont la position est re-projetée dans l'image, est localisé dans une région de « ciel » ou une région de « non-ciel ». Dans ce dernier cas, les satellites peuvent être considérés comme reçus après réflexion et l'algorithmie associée pourra choisir d'exclure ou de pondérer l'impact de chacun de ces satellites dans le calcul de position de localisation.

La deuxième brique technologique vise à corriger l'erreur induite par les réflexions des signaux GNSS (effet de multi-trajets) en modélisant en 3D l'environnement situé autour de l'antenne de réception GNSS. Dans ce contexte applicatif, un des objectifs à plus long terme est d'obtenir cette modélisation 3D en temps-réel. Pour ce faire, un compromis entre la rapidité et la précision des résultats obtenus est à définir.

Cette brique est décomposée en trois sous-tâches :

- T2.1 : Calcul du nuage de points 3D ;
- T2.2 : Construction du maillage 3D ;
- T2.3 : Extraction et caractérisation de la texture à partir du modèle.

Ce livrable est dédié à la tâche T2.1 dont l'objectif est de calculer le nuage de points 3D de la scène observée. Dans un premier temps, nous allons situer le contexte du projet CAPLOC et réaliser un état de l'art des travaux similaires en reconstruction 3D d'environnements urbains ainsi que sur les systèmes panoramiques et omnidirectionnels ; des premiers résultats sont présentés afin d'introduire notre approche globale dédiée à répondre aux objectifs du projet. Ensuite nous nous focaliserons sur les systèmes de caméras munies d'objectifs fisheye avec un état de l'art ainsi que l'introduction des notions de géométrie épipolaire, de calibrage et de rectification. Une troisième section présentera l'étape de mise en correspondance de pixels homologues et les premiers résultats obtenus pour le calcul de cartes des disparités et du nuage de points 3D et ce, avant de conclure et de présenter nos perspectives de travail pour la suite de ce projet.

2 La reconstruction 3D d'environnements urbains

2.1 Contexte du projet

La reconstruction 3D d'environnements urbains a connu un essor important cette dernière décennie ; celle-ci est de plus en plus automatisée notamment à partir de techniques basées sur l'imagerie satellitaire [2], l'imagerie aérienne [3] ou encore les lasers/scanners aéroportés [4]. Ces techniques ne permettent cependant pas d'obtenir des détails précis sur la géométrie des façades. Pour ce faire, les techniques d'acquisition de données depuis le sol ont été développées afin de combler cette lacune.

Dans la littérature, de nombreuses méthodes exploitant la vision artificielle existent et permettent de se localiser dans son environnement par l'extraction d'amers visuels [5, 6]. Dans le projet CAPLOC, l'objectif n'est pas la localisation mais bien l'utilisation des informations extraites de l'image pour qualifier et améliorer la précision de localisation GNSS : CAPLOC propose d'améliorer cette précision de localisation en tenant compte de la structure de l'environnement traversé par le véhicule, structure influant directement sur le canal de propagation des signaux GNSS.

La tâche 2 du projet CAPLOC vise à définir un procédé de création de modèles 3D de l'environnement facile à mettre en œuvre et compatible avec les exigences opérationnelles des opérateurs de transport ; il permettra d'appréhender l'environnement autour de l'antenne de réception GNSS avec une mobilisation du véhicule la plus courte possible. Le modèle 3D défini, associé à un moteur de lancer de rayons [7, 8] permettra de calculer puis de corriger l'erreur de pseudodistances causée par la présence de réflexions du signal sur les bâtiments présents dans la scène et ainsi d'améliorer la précision de localisation du véhicule.

2.2 Les systèmes actuels de reconstruction 3D d'environnements urbains

L'acquisition de données terrestres à l'échelle d'un quartier ou d'une ville était encore souvent délaissée, jusque récemment, pour des acquisitions aériennes beaucoup moins gourmandes en termes de volume de données ; seuls les bâtiments ou monuments remarquables pouvaient être visés par des acquisitions vues du sol. Ces acquisitions sont constituées d'images ou de relevés laser, utilisés séparément ou conjointement.

De par l'évolution des systèmes de stockage de données et l'amélioration de la vitesse des traitements, les données terrestres sont de plus en plus utilisées. Le « tout-automatique » est visé afin de pouvoir gagner du temps et appréhender des environnements à très grande échelle. La cartographie mobile permet ainsi d'effectuer très rapidement, grâce à une plateforme mobile dédiée, l'acquisition d'images et/ou de points 3D laser de quartiers ou de villes. Force est de constater que la plupart des systèmes proposés pour la reconstruction 3D d'environnements urbains utilisent des laser-scanners permettant d'obtenir directement une précision et une densité importantes du nuage de points 3D acquis ; les données images servent souvent à texturer le modèle 3D obtenu grâce aux points laser.

Les systèmes de modélisation 3D actuellement développés visent des applications de cartographie et numérisent des quartiers entiers de ville. Dans la suite de cette partie, nous présentons les principaux travaux visant à la reconstruction 3D d'environnements urbains à partir de données vues du sol.

Le laboratoire MATIS (laboratoire de Méthodes d'Analyses et de Traitement d'Images pour la Stéréo-restitution) de l'IGN (Institut national de l'information géographique et forestière) fait figure de référence sur le plan national mais aussi sur le plan mondial sur les aspects liés à la cartographie et à la numérisation 3D de villes. Les techniques développées se basent, pour la plupart, sur des images vues du ciel des zones urbaines à modéliser. De plus en plus, les

informations vues du sol sont prises en compte dans leurs traitements notamment pour associer les textures au modèle 3D calculé grâce aux données aériennes et affiner la géométrie des façades. Dans [9], l'auteur génère un modèle 3D de ville à partir d'images aériennes pour la géométrie générale des bâtiments et à partir de données au sol (images et laser) pour la modélisation des façades ; l'intérêt du laser est grand puisqu'il permet de prédire la visibilité image et de combler les parties cachées au fur et à mesure de l'avancée du véhicule porteur.

Le système Stéréopolis de l'IGN, présenté dans la figure 1, est un outil d'acquisition de données images et laser très abouti : il est composé d'une vingtaine de capteurs dont deux antennes GNSS, une centrale inertielle haut de gamme, un odomètre, un laser 3D à 64 plans (1 million de points/sec), deux lasers 2D (plans verticaux avançant avec le véhicule), 16 caméras Full HD sur 12 bits (la synchronisation de caméras ayant nécessité un travail important). Dix des seize caméras sont disposées sur le mât central et permettent d'acquérir une image complète de la scène en très haute résolution : sur les images obtenues, un pixel représente quelques mm^2 de la façade du bâtiment. Dans les faits, les modèles établis ont une précision métrique voire décimétrique selon les systèmes d'acquisition et les techniques de reconstruction employées. Le recalage à partir des images aériennes, sur la base de marquages au sol de gabarit connu, permet le géoréférencement centimétrique du modèle 3D vu du sol. Les traitements post-acquisition sont très coûteux en temps de calculs et sont effectués hors-ligne. Dans un avenir proche, la troisième version de la plateforme Stéréopolis aura pour objectif la numérisation complète de Paris intra-muros, ce qui représente 500 milliards de points laser à traiter : la rapidité des traitements à effectuer reste un enjeu important pour la reconstruction 3D automatisée à partir d'un tel jeu de données. Ceci est d'autant plus vrai que les actions de recherche du laboratoire MATIS visent à extraire des niveaux de détails de plus en plus fins, que ce soit au niveau des toitures (cheminées, chiens assis,...), des façades (fenêtres, volets, balcons,...) et plus largement de tous les éléments de la scène urbaine (panneaux, marquages au sol, mobilier urbain,...).



FIGURE 1 – Véhicule instrumenté de l'IGN, Stéréopolis II.

Le calibrage des imageurs du système Stéréopolis II permet de calculer de façon fine les positions et les orientations de chaque caméra par rapport à la plateforme. Deux méthodes sont utilisées. La première dite « off-line » est réalisée sur un réseau de cibles topométriques précis implantées à demeure sur la façade d'un bâtiment à l'IGN. Elle permet une estimation précise du calibrage à condition d'être à proximité de ce site dédié. Une deuxième méthode dite « on-line » permet une estimation de la calibration sur les éléments présents dans la scène observée.

La plupart des travaux de reconstruction 3D de scènes urbaines se basent sur l'acquisition de points 3D laser ; cependant, quelques travaux de la littérature permettent la constitution de ces modèles 3D par la seule utilisation de l'image, que ce soit à partir de caméras classiques ou de

systèmes catadioptriques plus élaborés.

Dans [10], les auteurs présentent une approche complète de reconstruction 3D d'un environnement urbain sur la base d'un système multi-capteur composé d'un récepteur GNSS, d'une centrale inertielle et de huit caméras (4 par côté du véhicule parmi lesquelles trois sont dans un même plan horizontal offrant un champ de vue d'environ 120 °). Les traitements sont réalisés sur GPU ce qui permet d'atteindre des performances temps-réel proches des 30 Hz. Dans le cadre du programme DARPA Urbanscape, ces travaux utilisent uniquement les données image pour acquérir la géométrie de la scène : l'utilisation de capteurs passifs sont plus économiques en termes d'encombrement, de poids et de coûts. Le calcul de pose des caméras est réalisé grâce à une étape de fusion des données issues du récepteur GNSS et de la centrale inertielle ainsi qu'à un suivi de points d'intérêts dans les images par filtrage de Kalman. Les cartes de profondeur sont calculées à partir d'une technique de balayage par plans [11], retenue pour sa simplicité de mise en œuvre et son potentiel de parallélisation. Dans un premier temps, les acquisitions image sont réalisées puis sont déplacées vers un cluster de calculs où chaque nœud réalise une tâche particulière afin d'atteindre des performances temps-réel : ces tâches vont de l'acquisition des données images/inertielles/satellitaires à l'élaboration du modèle 3D texturé. Les résultats de reconstruction 3D de façades sont présentés sur la figure 2.



FIGURE 2 – Résultats de reconstruction 3D obtenus dans le cadre du projet *Urbanscape*.

Dans [12], les auteurs proposent une reconstruction 3D automatique de façades de bâtiments à partir d'un ensemble de plusieurs vues géoréférencées et en recouvrement. L'idée est de déterminer un nuage de points 3D par des techniques de corrélation multi-images en multi-résolution. Sur la base des scores multi-corrélation, une étape d'optimisation globale de surface permet l'obtention d'un nuage de points 3D moins dense mais bien plus robuste et précis. Ensuite, le modèle 3D obtenu après triangulation est texturé. Cette méthode « tout-automatique » ne nécessite aucune étape d'initialisation que ce soit par un plan cadastral ou par l'acquisition de points laser.

Pour une application de reconstruction 3D d'environnements urbains, les capteurs panoramiques et omnidirectionnels sont pressentis pour leur capacité à observer de très larges champs de vision.

2.3 Les capteurs panoramiques et omnidirectionnels

Une caméra conventionnelle a un champ de vision limité. Or, de nombreuses applications ont besoin d'une vue globale de l'environnement situé autour d'un mobile par exemple pour pouvoir se localiser, se déplacer (applications de robotique mobile ou cas des systèmes de « transports

intelligents ») ou comme dans CAPLOC estimer la structure 3D de l'environnement urbain autour du mobile. Dans cette section, nous proposons un état de l'art des capteurs panoramiques et omnidirectionnels décrits dans la littérature depuis une quinzaine d'années. Il répertorie notamment les systèmes catadioptriques (systèmes utilisant un ou plusieurs miroirs) et des systèmes utilisant des caméras à objectif fisheye.

Dans [13, 14, 15, 16], Mouaddib, Gonzalez-Barbosa et Ragot présentent différentes façons d'accéder à la vision omnidirectionnelle ou panoramique :

- ◇ par la rotation d'un capteur classique [17]. Cette approche pose des problèmes de positionnement, de synchronisation des prises de vues et de temps d'exposition pour l'acquisition des images. Il est nécessaire d'effectuer plusieurs acquisitions avec un dispositif tournant pour obtenir l'image omnidirectionnelle avec une caméra classique. Ce procédé n'est pas applicable pour des scènes dynamiques. Dans [17], les auteurs utilisent une paire de caméras alignées horizontalement en rotation selon un axe centré pour appliquer les principes de la stéréovision (en géométrie épipolaire simple).

- ◇ en multipliant les caméras classiques pour former une ceinture ou un réseau de caméras. Un tel système comporte des angles morts et des parties de l'environnement proche non acquises (zones entre deux caméras avant le recouvrement de leurs champs de vision). La concaténation des différentes vues en une seule vue cohérente, par coïncidence des centres de vues, n'est pas immédiate autant dans la fabrication du système que dans le processus d'assemblage des images. Dans [16], plusieurs réseaux de caméras existants sont cités, les plus complexes étant ceux de la série « ASTRO *Sensor Series* » développés par la société japonaise ViewPLUS, comme présenté sur la figure 3.



FIGURE 3 – Exemple de capteur composé d'un réseau de caméras, ASTRO Sensor (Jupiter) [16].

- ◇ par l'association d'une caméra classique et d'un miroir de révolution, formant ainsi un capteur catadioptrique [15]. Le miroir, réfléchissant les rayons lumineux provenant de toutes les directions, forme, une fois projetés sur le capteur, une image omnidirectionnelle. C'est le type de capteur actuellement le plus étudié, en particulier dans des applications de navigation en robotique mobile. La figure 4 présente le capteur catadioptrique utilisé dans [15]. Cependant, comme relaté dans [18], ce type de capteur ne convient pas à certaines applications car la partie centrale de l'image (l'azimut) est inexploitable, on y voit le reflet de la caméra elle-même. On distingue cinq types principaux de capteurs catadioptriques répartis en fonction de la forme du miroir utilisé :
  - miroir sphérique : angle de vue le plus large, mais résolution faible dans la périphérie de

l'image ;
- miroir conique : bonne résolution dans la périphérie de l'image, les droites parallèles à l'axe optique se projettent de façon radiale dans l'image, mais l'astigmatisme est grand ;
- miroir hyperbolique : l'astigmatisme est faible mais la conception impose que le point focal de l'hyperboloïde soit placé au centre de la caméra ;
- miroir parabolique : avec une lentille télécentrique alignée sur l'axe du miroir, seuls les rayons incidents qui arrivent sur le miroir et qui traversent le foyer du paraboloïde sont transmis, l'astigmatisme est faible ;
- miroir elliptique ou miroir hyperbolique au foyer desquels est placé le centre optique d'une caméra perspective.



FIGURE 4 – Capteur catadioptrique avec un miroir hyperbolique utilisé par [15].

Certains systèmes catadioptriques sont à point de vue unique (ou centraux) mais ils sont difficiles à construire et à entretenir. Beaucoup d'entre eux sont en réalité non centraux. Il présentent une zone de point de vue qui est en fait une courbe ou un ensemble de points. Là encore, le type du miroir associé à la caméra perspective est un élément de différenciation. Ainsi, peut être utilisé : un miroir conique, un miroir sphérique ou une matrice de micro-miroirs par laquelle est observée la scène (chaque miroir pouvant être orienté et positionné par une commande).

◇ par l'utilisation de plusieurs dispositifs optiques basés sur la réflexion à l'aide de miroirs plans [19]. Le capteur omnidirectionnel est composé de plusieurs couples miroir/caméra disposés en pyramide pour lesquels les différents points de vue doivent être ensuite juxtaposés. Dans [19], les auteurs présentent le dispositif OmniCam développé pour obtenir des images panoramiques de 180 ° pour la télévision. Certaines versions couvrent un champ de 360 ° et assurent une perception 3D en associant une paire de caméras par miroir, comme illustré sur la figure 5.



FIGURE 5 – Le système OmniCam 3D développé au Fraunhofer HHI.

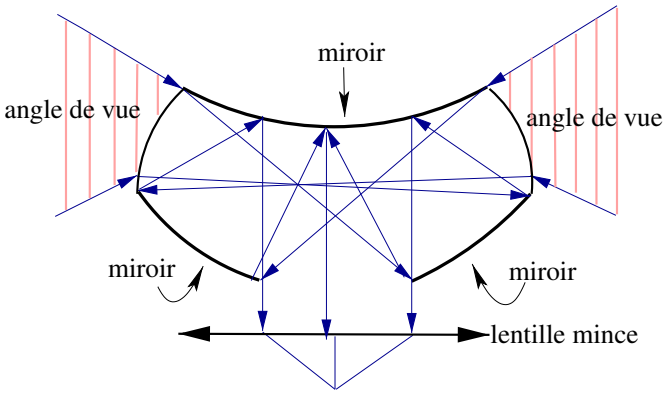


FIGURE 6 – Schéma du PAL, les rayons arrivant tout autour de l'objectif sont réfléchis pour atteindre en parallèle une lentille mince convergente puis le capteur [15].

◇ par l'utilisation d'un objectif spécial permettant d'obtenir une image panoramique sans distorsion, le PAL (*Panoramic Annular Lens*) inventé par Majoros Sándor [20] : sa modélisation est complexe et le champ de vue vertical est relativement faible (cf figure 6).

◇ par l'utilisation d'un objectif ultra grand angle, de focale très courte, appelé objectif hypergone ou le plus souvent objectif « fisheye ». Ces objectifs sont simples d'utilisation puisqu'ils s'adaptent comme tous les autres objectifs sur les caméras y compris celles de haute résolution. Toutefois, cet objectif introduit de fortes distorsions dans l'image, en particulier sur la périphérie où la résolution devient faible. En théorie, les objectifs fisheye ne sont pas des dispositifs à point de vue unique [21, 22, 23, 24]. Les rayons se croisent en un petit disque dans l'espace plutôt qu'en un point unique. Dans [25], l'auteur précise que l'utilisation d'objectifs fisheye est sans doute la manière la plus courante pour obtenir des images omnidirectionnelles mais elle souffre de ne pas avoir un centre de projection unique. Ainsi, l'analyse des objets de la scène proche de la caméra devient extrêmement complexe. Lorsque les objets sont relativement éloignés de l'objectif, on peut considérer que l'objectif fisheye dispose d'un centre de projection unique. Le mode de projection d'une caméra à objectif fisheye peut être vu comme la projection d'un ensemble de points 3D situés sur une demi-sphère.

2.4 Premiers résultats en stéréoscopie classique

Rapidement après le début du projet CAPLOC, nous avons réalisé une campagne d'acquisition avec le matériel pressenti lors de la rédaction de la réponse à l'appel à projet « Gestion du traffic et des mobilités » : la première version du système multi-vues envisagée pour la modélisation 3D de l'environnement était composée d'une ceinture de systèmes stéréoscopiques XB3 [26] commercialisés par la société Point Grey, la résolution des capteurs étant suffisamment élevée. L'entraxe entre les deux capteurs est de 24 cm et les focales des objectifs de 3.8 mm, conférant une ouverture horizontale de 65 ° environ et une ouverture verticale de 51 ° environ.

Cette première mise en œuvre du système d'acquisition 3D a été réalisée sur le véhicule expérimental du laboratoire de l'IFSTTAR-LEOST. Le système stéréoscopique était réduit à un seul système XB3 visant vers la droite du véhicule ; la ligne de base du stéréoscope est placée parallèlement au sens de déplacement du véhicule et vise à 45 ° vers le ciel. Les images issues de ce stéréoscope sont enregistrées et datées grâce au logiciel RT-Maps. Une première campagne d'acquisition a été menée dans un quartier de Villeneuve d'Ascq (Nord) très similaire au centre

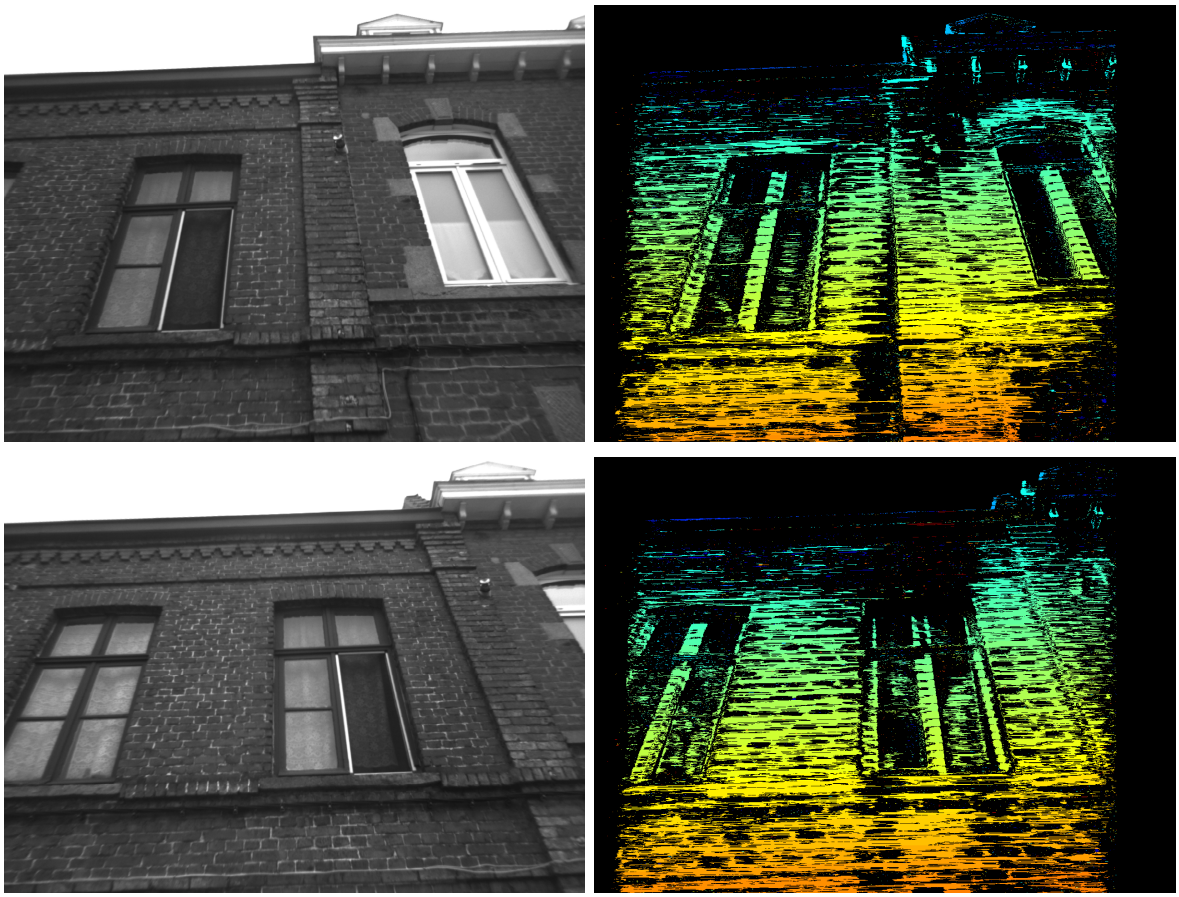
Lillois, futur lieu de nos campagnes d'acquisition. La figure 7 montre le style des bâtiments caractérisant ce quartier de ville. Très vite, nous nous sommes rendus compte que ce système était muni d'un champ de vue très réduit ne permettant d'acquérir qu'une vue très partielle de chacune des façades présentes dans la scène, sauf à démultiplier considérablement le nombre de systèmes XB3 dans les deux directions, horizontale et verticale. En effet, le système XB3 est muni d'objectifs de longueur focale de 3.8 mm non interchangeables. De par les objectifs de ce projet, l'idée de devoir mettre en œuvre un système composé d'un grand nombre de caméras, fournissant un volume de données très important, semble rédhibitoire !



FIGURE 7 – Exemples d'images acquises avec le stéréoscope XB3 à Villeneuve d'Ascq.

Malgré ce constat, nous avons procédé à l'étape de mise en correspondance par des techniques basées sur la corrélation sur la base de ces premières acquisitions. Ces méthodes locales, aussi appelées méthodes par corrélation, permettent de mesurer la ressemblance entre deux ensembles de pixels (dans les images gauche et droite) par des mesures de corrélation. Elles calculent des cartes éparses des disparités, si les primitives employées sont des points caractéristiques de l'image, mais aussi de cartes denses. Ces dernières sont obtenues en effectuant l'analyse des niveaux de gris (ou couleur) des pixels contenus dans une zone d'agrégation centrée ou non sur chaque pixel à apparier. L'objectif est de mesurer la similarité (ou la dissimilarité) des pixels contenus dans deux zones de forme identique extraites des images gauche et droite. Selon les caractéristiques de la mesure de similarité utilisée, une ressemblance forte entre deux zones situées autour de deux pixels caractérisera un couple potentiel de pixels homologues. Sur la base de la contrainte épipolaire, les coordonnées de deux pixels homologues ne se différencient que de par leurs abscisses, cette différence définissant la valeur de disparité à effecter au pixel dans l'image de référence. La figure 8 présente des exemples des cartes de disparités estimées avec la méthode proposée dans [27]. Les cartes des disparités obtenues apparaissent comme quasi-denses et assez précises. Malgré les caractéristiques de structures répétitives (briques/joints de ciment), la méthode s'en sort très bien : une erreur d'appariement en présence de structures répétitives serait due au fait de comparer deux joints de ciment différents dans les deux images, et de déterminer un score de similarité important. On peut constater que les zones de l'image correspondant à des fenêtres sont très éparses en termes de disparités estimées.

Dans ce contexte d'acquisition, cette méthode de mise en correspondance montre des résultats très intéressants puisque les cartes des disparités obtenues sont assez précises et quasi-denses. Dans la suite de ce document, nous jugerons de son efficacité dans le cadre de la nouvelle approche



(a) (b)

FIGURE 8 – (a) Image gauche et (b) carte des disparités calculées par [27] pour les couples 936 et 940 de la séquence.

proposée pour notre application de reconstruction 3D d'environnements urbains.

2.5 Nouvelle approche proposée

Dans le projet CAPLOC, nous cherchons à modéliser la structure 3D de bâtiments à l'aide de techniques de mise en correspondance stéréoscopique permettant de retrouver l'information de profondeur à partir d'images 2D. Notons que selon l'application visée et le matériel mis en œuvre, les techniques employées sont différentes. Notre système embarqué de reconstruction 3D doit être simple et rapide à mettre en œuvre. Compte tenu du volume de données à traiter, nous nous orientons vers des méthodes de reconstruction 3D rapides et génériques à l'instar des travaux de thèse de L. Pénard [28] réalisés à l'IGN. L'idée est de calculer un modèle 3D moins précis que ceux calculés par le MATIS, mais suffisamment précis pour nous permettre de corriger si possible en temps-réel les pseudodistances (relative aux rayons réfléchis). Cette approche permet de limiter le nombre de capteurs associés au récepteur GNSS et à apporter au système une information sur les événements locaux qui dégradent les performances de localisation. Dans un premier temps, nous nous intéressons aux obstacles les plus volumineux de la scène (immeubles par exemple), les plus petits créant des phénomènes locaux très furtifs pour un récepteur mobile ne présentant pas de difficultés majeures : l'environnement urbain étant quelque chose de très complexe à modéliser, notre premier objectif sera de proposer un modèle simple définissant une façade par un plan.

Il nous faut développer un système multi-vues capable de reconstruire en 3D l'environnement autour de l'antenne GNSS. Un premier objectif est d'obtenir un nuage de points 3D suffisamment dense et précis pour appréhender l'étape suivante de maillage 3D dans les meilleures conditions. Afin de limiter le nombre de caméras, nous avons opté pour un système de caméras munies d'objectifs fisheye pour la reconstruction 3D d'environnements urbains : de par leurs champs de vision très larges, les systèmes de vision omnidirectionnels à base de caméras fisheye connaissent une attention accrue ces dernières années pour des applications de vision par ordinateur. Deux caméras de grande résolution munies d'objectifs grand angle seront utilisées en configuration stéréoscopique classique à l'instar de la vision humaine, ces caméras étant placées à la verticale du mobile et visant le ciel. Limité à deux caméras, ce système permettra un champ de vision et une reconstruction à 360 ° de l'environnement autour de l'antenne GNSS placée sur le toit du véhicule comme présenté sur la figure 9 ; celle-ci présente deux images fisheye issues d'une même caméra collectées lors d'une séance d'acquisition réalisées à Belfort (90).

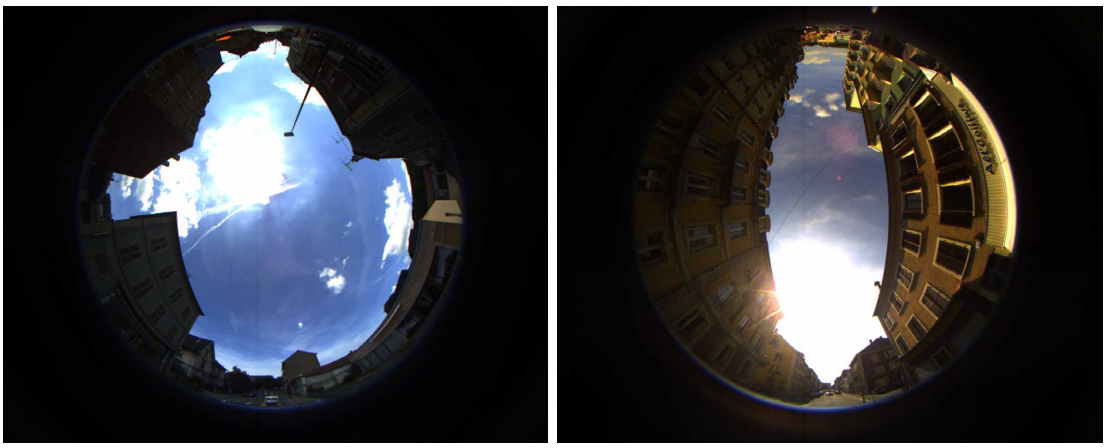


FIGURE 9 – Exemples d'images issues d'une caméra fisheye pointant orthogonalement vers le ciel.

Dans la prochaine partie de ce livrable, nous nous intéressons aux principes de la stéréovision à partir de caméras munies d'objectifs fisheye comme la géométrie épipolaire, le calibrage, la rectification et la mise en correspondance stéréoscopique.

3 Stéréovision avec des caméras à objectifs fisheye : état de l'art

Pour obtenir un nuage de points 3D par des techniques liées au traitement d'images, il existe deux principales approches. La première exploite des images prises successivement au cours du mouvement (Structure From Motion) pour laquelle il faut connaître très précisément le changement de pose opéré entre deux prises de vues successives. La deuxième exploite des couples d'images prises au même instant mais à des points de vue légèrement différents. Dans CAPLOC, nous avons adopté la seconde approche. Cette partie est consacrée à l'état de l'art des travaux en matière de stéréovision fisheye. Les deux premières sections exposent les principes généraux de la géométrie épipolaire, les procédures de calibrage des systèmes stéréoscopiques fiseye et la rectification des images qu'ils fournissent. Nous clôturons cette partie par une présentation de quelques travaux sur la mise en correspondance stéréoscopique d'images fisheye dans les domaines d'application telles que la reconstruction de scènes urbaines et l'assistance au conducteur.

3.1 Principe de la géométrie épipolaire

Afin d'assurer la reconstruction 3D d'une scène, il faut disposer de correspondances entre les éléments contenus dans les deux images de la scène [29]. Pour y parvenir, une analyse stéréoscopique requière deux étapes préliminaires. La première est l'étape de calibrage dont l'objectif est d'estimer les paramètres intrinsèques et extrinsèques (i.e. la position relative des deux caméras) du système stéréoscopique. La seconde est l'étape de rectification dont l'objectif est de produire deux images telles qu'elles seraient fournies par deux caméras dont les capteurs seraient coplanaires et alignés ligne à ligne. Cette propriété a la particularité de simplifier la phase de mise en correspondance au détriment d'une information intra-image modifiée.

Le calibrage des objectifs fisheye nécessite un modèle mathématique pour décrire la projection des points 3D sur le plan image 2D. Cette transformation 3D/2D est décomposée en un modèle de projection (figure 10) et un modèle de distorsion. Le modèle de projection décrit la projection idéale sans erreur et le modèle de distorsion décrit les altérations du modèle de projection causées par les distorsions dues à l'objectif. Les modèles de projection fisheye sont généralement des projections d'une sphère sur un plan. Les modèles de projection présentent une symétrie radiale autour de l'axe optique. Tous les modèles font référence à un point de base, étant l'intersection entre l'axe optique et le plan image. La distance d'un point dans l'image au point de base dépend uniquement de l'angle entre le rayon lumineux depuis un point de la scène à la caméra et l'axe optique. Les distorsions appliquées sur le modèle de projection sont définies dans le modèle de distorsion, de la même manière qu'avec une caméra perspective (les coordonnées dans l'image se trouvent légèrement décalées par les distorsions). L'ensemble de tous ces paramètres définit la caméra.

Plusieurs modèles de projection 3D/2D idéaux sont proposés dans la littérature [22] :

- projection perspective, normale ou non-fisheye (*perspective projection*)
  $r = f\tan(\theta)$
- projection stéréographique ou angle fidèle (*stereographic* ou *conform projection*)
  $r = 2f\tan(\frac{\theta}{2})$
- projection équidistante ou linéairement divisée (*equidistant* ou *linear scaled projection*)
  $r = f\theta$
- projection équisolide ou surface fidèle (*equisolid angle* ou *equal area projection*)
  $r = 2f\sin(\frac{\theta}{2})$
- projection orthographique (*orthographic* ou *sine law projection*)
  $r = f\sin(\theta)$

Ici f correspond à la longueur focale de l'objectif et θ à l'angle formé par la ligne de projection et l'axe de l'objectif. Dans [24], les auteurs expriment les équations de manière différente, où f

est remplacé par un élément $\delta = \frac{f}{w}$ avec w la largeur d'un pixel sur le capteur.

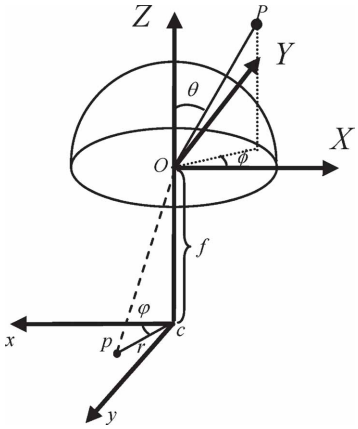


FIGURE 10 – Modèle de projection décrivant la relation entre l'objectif fisheye de la caméra et le plan image [23].

Rectifier un couple d'images fisheye consiste à réaliser une transformation géométrique sur chacune d'entre elles de telle sorte que chaque point de la scène soit projeté sur la même ligne dans les deux images. Le but est de réduire l'espace de recherche à une dimension le long des lignes dans les images lors de l'étape de mise en correspondance stéréoscopique. Cette transformation facilite non seulement l'implémentation d'algorithmes de mise en correspondance mais accélère aussi le temps de calcul. Dans le cas de la stéréovision fisheye, le modèle de rectification est défini comme une fonction non linéaire qui transforme une coordonnée de la caméra en une coordonnée image d'une caméra virtuelle. Le principe de la géométrie épipolaire avec les caméras fisheye est assez proche de celui utilisé dans le cas des caméras perspectives classiques. Compte-tenu de la déformation présente dans les images, la différence réside dans le fait que les lignes épipolaires se manifestent sous la forme de coniques à la surface de la sphère de projection. La transformation a pour objectif de faire en sorte que les épipoles coïncident avec les pôles de la sphère de projection. Ainsi, les courbes épipolaires correspondent exactement aux cercles de longitude de la sphère. Dans ce cas de figure, l'appariement se présente comme un problème 1D le long de ces cercles.

3.2 Calibrage des caméras fisheye et rectification épipolaire

Dans [30], les auteurs présentent une procédure de calibrage spécifique pour les caméras à objectif fisheye intégrant une correction des distorsions de l'objectif. Les auteurs proposent une méthode pour calibrer chacun des paramètres suivants :

- la longueur focale réelle ;
- la dimension d'un pixel sur le plan image (le capteur) ;
- le centre optique ;
- les coefficients de distorsion.

L'auteur estime la position du centre optique en utilisant la forme d'une frange lumineuse grâce à l'émission d'un rayon laser de faible puissance à travers un écran blanc perforé en direction du centre de l'objectif. En réduisant la frange à un point, il obtient la direction du centre de l'objectif. Les orientations du rayon laser et de l'objectif sont ajustées jusqu'à ce que le rayon laser se réfléchisse sur lui-même. Le but étant d'aligner le rayon laser avec l'axe optique de l'objectif. Un rayon lumineux est acquis dans l'image et ce point indique la position du centre optique sur l'image plane. L'auteur estime la longueur focale de l'objectif à partir de l'analyse des coordonnées dans l'image de quelques points voisins d'une mire de calibrage dont il connait

exactement les coordonnées relatives. Cette analyse est réalisée dans une très proche région du centre optique dont la position est estimée comme indiqué précédemment et en y considérant la distorsion comme négligeable. Les distorsions radiales et tangentielles sont modélisées par deux polynômes d'ordre 5 dont les coefficients sont calculés en minimisant une fonctionnelle.

Les méthodes plus récentes estiment les paramètres du modèle de projection et du modèle de distorsion en minimisant une fonctionnelle dépendant des coordonnées 3D des points d'une mire de calibrage et des coordonnées de ces mêmes points après projection par le modèle dont les paramètres sont à estimer. Très souvent les distorsions tangentielles sont négligées et seules les distorsions radiales sont considérées. Elles sont souvent représentées par un polynôme de degré n, la valeur de n étant un compromis entre la complexité d'obtention du modèle et sa précision d'attache aux données. Des méthodes similaires sont utilisées pour l'estimation des paramètres extrinsèques.

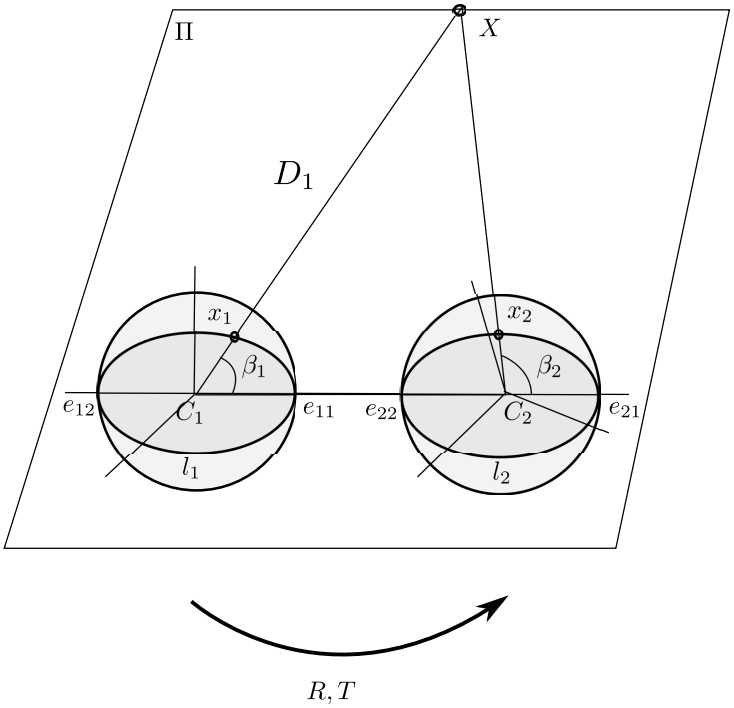


FIGURE 11 – Géométrie épipolaire pour des images sphériques [31].

Dans [21], les auteurs travaillent sur le calibrage géométrique et la rectification épipolaire pour des bancs stéréoscopiques munis d'objectifs fisheye. Ce papier propose la re-projection des images fisheye en images classiques (latitude-longitude) notamment pour les modèles de projection équidistant et stéréographique. La méthode est applicable à des champs de vue supérieurs à 180 ° mais les résultats présentés sur le nuage de points calculé manquent de précision.

Pour le calibrage de caméras omnidirectionnelles, il existe des bibliothèques sous le logiciel Matlab comme présenté par D. Scaramuzza dans [32] ou par C. Mei dans [33]. Ces boîtes à outils permettent de calibrer différents types de caméras omnidirectionnelles et ce, sans avoir connaissance des paramètres de la caméra ou de la forme du miroir ; cela se fait à partir de plusieurs images de mires de calibrage ayant des positions et des orientations différentes. Les résultats obtenus expérimentalement confirment que ces méthodes peuvent être utilisées pour le calibrage d'une large gamme de capteurs omnidirectionnels.

Dans une grande partie des solutions proposées, la rectification épipolaire mène à un alignement des courbes épipolaires (projection stéréographique). Cette solution induit de fortes distorsions dans les images corrigées et une modification assez importante du contenu des images. Ceci peut être un lourd problème et contre-productif lorsque l'on utilise des techniques de mise en correspondance basées sur le voisinage des points. Dans [34], les auteurs proposent une rectification épipolaire qui transforme les courbes épipolaires en cercles. Ils démontrent le bon fonctionnement et la qualité de leur méthode par des expérimentations avec des images fisheye. Ils comparent leur méthode basée sur la projection stéréographique à la méthode de rectification généralement utilisée. Les résultats sont similaires lorsque les vues omnidirectionnelles sont prises avec un décalage sur le côté. En revanche, pour d'autres configurations (essais réalisés avec des vues prises en avançant, et des vues prises avec une rotation du capteur de 90 ° et de 45 ° autour d'un cercle de 2 mètres de diamètre), la méthode de rectification stéréographique permet de moins détériorer les images originales car les distorsions sont nettement moins prononcées en particulier aux épipoles.

3.3 La mise en correspondance stéréoscopique d'images fisheye

Parmi les premiers travaux de reconstruction de scènes urbaines, on peut citer [35] : les auteurs font partie des précurseurs de la stéréovision fisheye dans ce cadre applicatif. Les premiers résultats sont obtenus sur des images de synthèse de scènes urbaines très simples (quatre immeubles) ; les coins les plus proches du centre des images fisheye sont localisés dans toutes les directions afin de créer deux listes de coins respectivement pour les images gauche et droite. Ensuite, un algorithme lié à la programmation dynamique permet la mise en correspondance de ces points avant leur reprojection dans l'espace tridimensionnel.

Dans [36], les auteurs utilisent de manière originale un couple de caméras fisheye en n'exploitant que la partie centrale des images stéréoscopiques (régions les moins déformées). Ils appliquent ainsi une méthode de stéréovision classique sur des portions d'images n'ayant subi que peu de déformations. Ils ne disposent pas des informations de profondeur pour l'ensemble du champ de vision mais utilisent les autres parties des images fisheye pour calculer le flot optique et assurer la navigation très précise de leur mobile, ici un drone.

Dans [37], l'auteur propose l'utilisation d'un banc stéréoscopique muni de deux caméras à objectifs grand-angle. Les images sphériques sont reprojetées en images perspectives classiques : cette transformation permet d'appliquer, telles quelles, les méthodes de mise en correspondance stéréoscopique classiques de la littérature. La figure 12 montre comment les lignes épipolaires sphériques sont transformées en lignes épipolaires rectilignes et parallèles aux lignes de l'image : les lignes épipolaires sphériques sont des arcs de cercles passant par les épipoles e_1 et e_2, ces deux points étant considérés comme les deux pôles pour un ré-échantillonnage « latitude-longitude ». Les résultats sont obtenus en temps-réel et proposent des cartes des disparités semi-denses assez précises.

Dans [38], les auteurs cherchent à détecter et à suivre un objet en mouvement par une méthode de mesure 3D dans l'espace vu par deux caméras omnidirectionnelles. Ils corrigent les images fisheye en calculant une représentation inverse à l'effet fisheye. Pour cela, ils utilisent le modèle équidistant. Les coordonnées 3D de chaque point dans les images corrigées sont déterminées. Ainsi, le système détecte la position et le volume de l'objet dans l'espace réel et le suit.

Dans [39], Gehrig et al. étudient de manière approfondie la stéréovision fisheye. Ils proposent un système de stéréovision fisheye pour la détection d'objets dans des applications d'assistance au conducteur et de sécurité pour les véhicules. Ils déterminent le champ de vision nécessaire à leur application, qui correspond à un angle de champ horizontal de 150 ° et vertical de 25 ° devant

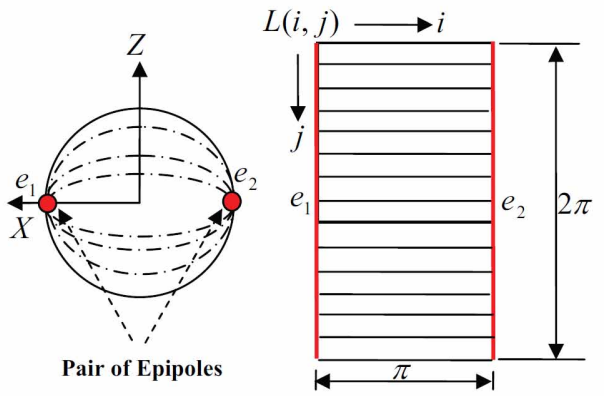


FIGURE 12 – Ré-échantilonnage latitude-longitude proposé dans [37].

le véhicule ; ils n'utilisent donc qu'une bande horizontale au centre des images fisheye obtenues. Après une phase de calibrage des caméras fisheye, la méthode opère une rectification cylindrique des images panoramiques extraites des images omnidirectionnelles pour ensuite appliquer une mise en correspondance classique et pouvoir détecter des objets à partir des informations 3D calculées. Pour le calibrage des objectifs fisheye, les auteurs se réfèrent aux données du constructeur indiquant que l'objectif répond à une fonction de représentation équidistante (ou linéairement divisée). Une rectification cylindrique est utilisée au lieu d'une rectification planaire afin de moins altérer l'image et mieux préserver les informations originales. Dans cette configuration, les lignes épipolaires sont finalement alignées horizontalement afin d'y appliquer des algorithmes classiques de mise en correspondance stéréoscopique. Des méthodes de mise en correspondance locales et globales ont été testées. Les méthodes locales ne rencontrent que peu de difficultés lorsqu'elles sont appliquées à des images cylindriques à condition que les lignes épipolaires soient bien alignées. Cependant, elles sont incapables de calculer les disparités des objets proches du stéréoscope, car dans ce cas, le voisinage d'un même point de la scène apparait différemment dans l'image droite et dans l'image gauche. Les coordonnées 3D des points peuvent être ensuite facilement calculées par triangulation. Malgré cette difficulté, les cartes des disparités sont exploitables et permettent d'obtenir une bonne reconstruction de l'environnement.

Dans [24], les auteurs proposent une méthode pour mesurer un environnement 3D à l'aide d'une caméra fisheye qu'ils déplacent, pour acquérir une séquence d'images fisheye et travailler en particulier avec les deux images des extrêmes. Ce travail est à placer parmi les méthodes de type « Depth from Motion » mais possède des caractéristiques communes avec les méthodes utilisant un couple de caméras. Les modèles de projection et de distorsion de la caméra fisheye sont calculés par un polynôme du cinquième ordre. Les auteurs n'appliquent pas de méthodes d'appariement qui conduisent à des cartes denses des disparités. Des points SIFT sont extraits dans les images et mis en correspondance entre deux images successives (temporellement). Ces points d'intérêt sont suivis tout au long de la séquence. La méthode RANSAC est utilisée pour éliminer les points mal suivis qui perturbent le calcul de la matrice essentielle (et qui permet de retrouver la structure 3D de la scène au travers des matrices de translation et de rotation). Les positions 3D des points correspondant aux couples de points d'intérêt sont calculées pour ensuite conduire à un maillage 3D grâce à la méthode de triangulation de Delaunay. Une texture est plaquée sur chaque surface du maillage triangulaire généré pour construire un modèle avec des informations de couleur et de texture.

Dans [40], Herrera et al. présentent une méthode de calcul de cartes des disparités à partir

d'images hémisphériques obtenues avec des caméras munies d'objectifs fisheye en environnement forestier. La première étape consiste en une segmentation destinée à distinguer les textures d'intérêt qui seront appariées (les troncs des arbres) de celles à ignorer (l'herbe, les feuilles des arbres et le ciel). Dans un deuxième temps, une méthode de mise en correspondance stéréoscopique est appliquée. Elle suit la contrainte épipolaire pour guider le processus d'appariement. Elle tient compte des contraintes de similarité et d'unicité à travers une stratégie locale aboutissant à une carte des disparités initiale. La similarité ainsi estimée est une combinaison par logique floue pondérée de la mesure des six attributs suivants : un coefficient de corrélation, la variance comme mesure de la texture, la couleur de chaque pixel, la mesure du gradient, l'angle de direction du gradient et le Laplacien. Les auteurs ne rectifient pas les images et travaillent directement avec les images issues de la projection de l'objectif fisheye sur le plan correspondant au capteur de la caméra. Ils travaillent en coordonnées cartésiennes dans l'image et calculent la disparité comme la différence entre les angles des points appariés donnés par les coordonnées polaires correspondantes de chacun d'eux. Cette carte est ensuite lissée par l'application d'une méthode globale basée sur les réseaux de neurones de Hopfield. L'optimisation globale supprime les erreurs d'appariement et lisse les disparités le long des bords des troncs.

Dans [41], l'auteur propose une méthode pour mesurer les niveaux (ou reliefs) du sol pour un robot humanoïde à partir de la stéréovision fisheye, le système d'acquisition visant devant lui. Les images acquises sont partiellement rectifiées de sorte que les images du sol soient projetées sur un plan plat et incluant les pieds à leur position actuelle. La correction transforme également l'image de telle sorte que les lignes épipolaires correspondent aux lignes de l'image. Selon l'auteur, cette rectification rend l'appariement stéréoscopique plus aisé et fiable (car on applique l'une des nombreuses méthodes connues pour le modèle sténopé) mais permet aussi de calculer directement les niveaux du sol à partir de la carte des disparités. Pour la mise en correspondance stéréoscopique, il utilise un algorithme local avec une mesure de corrélation SSD (Sum of Squared Differences). Comme la méthode décrite ne fait pas de conversion de coordonnées, l'erreur apparaissant lorsque l'on convertit les mesures basées sur le système de coordonnées du capteur stéréo en coordonnées du monde réel, n'existe pas. Les résultats expérimentaux montrent que les mesures des niveaux du sol sont relativement stables entre les différentes poses du système stéréoscopique. Dans ses essais, l'auteur utilise des marqueurs pour disposer de points connus pour le calibrage des capteurs.

Dans la suite de ce document, nous présentons le système d'acquisition dimensionné pour le projet CAPLOC ainsi que la brique de perception 3D nécessaire à l'obtention du nuage de points 3D.

4 Système CAPLOC et perception 3D

Dans cette section, nous présentons les avancées techniques obtenues dans le cadre de la tâche 2. Nous commençons par la description du système stéréoscopique retenu et sa mise en œuvre. Les choix effectués quant aux éléments du système ont été guidés par la large analyse bibliographique qui a été menée et détaillée dans les deux sections précédentes. Nous poursuivons en présentant la géométrie épipolaire du système fisheye CAPLOC. Un ensemble de formulations mathématiques est donné afin de mettre en œuvre les algorithmes de mise en correspondance épipolaire et de produire les nuages de points 3D. Nous terminons ce chapitre en présentant quelques cartes de disparités obtenues et en visualisant en 3D les nuages de points correspondants.

4.1 Mise en œuvre matérielle du système

L'objectif de cette plateforme d'acquisition est de pouvoir enregistrer des séquences d'images stéréoscopiques de grande résolution capables d'appréhender l'environnement autour de l'antenne GNSS et de le reconstruire en trois dimensions. Pour cela, l'approche stéréoscopique a été retenue sur la base de couples d'images fisheye. Chaque couple devra être daté et géo-référencé. Nous proposons d'utiliser un récepteur GNSS RTK afin d'obtenir une trajectoire dite de référence, même si celle-ci n'est pas optimale : en effet, celle-ci pourrait être encore améliorée par couplage du récepteur GNSS avec une centrale inertielle et/ou un système d'odométrie et l'algorithmie dédiée.

Cette référence pourra être utilisée pour quantifier l'apport de nos méthodes de traitement d'images pour l'amélioration de données de localisation issues de récepteurs GNSS bas-coût ; ces derniers feront partie intégrante de la plateforme d'acquisition finale.

La figure 13 présente un schéma descriptif de la plateforme d'acquisition. Deux caméras munies d'objectifs fisheye sont placées en position stéréoscopique classique : la ligne de base du stéréoscope est dans le sens longitudinal du véhicule expérimental et les caméras sont à la verticale visant vers le ciel. Chaque caméra est montée sur une platine micrométrique permettant son orientation précise selon les trois axes de liberté. L'entraxe du stéréoscope séparant les deux caméras pourra ainsi être modifié aisément. Les flux vidéos sont enregistrés et datés par un module dédié, l'horloge du module étant recalée toutes les secondes grâce au signal PPS (Pulse-Per-Second) généré par le récepteur GNSS.

Concernant les choix techniques, nous avons opté pour des caméras JAI AM800 CL monochromes, progressives et haute résolution : à pleine résolution, nos images seront de 3296 x 2472 pixels à la fréquence d'acquisition maximale de 17 images/seconde. Ces caméras présentent un très bon compromis entre la résolution des images et la fréquence d'acquisition ; dans notre contexte mobile, à une vitesse constante de 60 km/h, un couple d'images est acquis tous les mètres environ.

Les objectifs Sigma HSM Circular Fisheye de longueur focale 4.5 mm avec une ouverture égale à 2.8 ont été retenus. Leur association avec les caméras JAI semble optimale en ce qui concerne le type de monture ainsi que la proportion du cercle fisheye (l'image utile) occupée sur l'image totale. La projection de ce modèle d'objectif est de type « angle équisolide ». L'ouverture du diaphragme à 2.8 permet une cadence d'acquisition élevée : pour des ouvertures inférieures, le temps d'exposition serait plus important de manière à intégrer plus de lumière, ce qui diminuerait la vitesse d'acquisition.

Le module d'acquisition vidéo retenu est le DVR Express Core de la société IO Industries. Celui-ci permet l'enregistrement temps-réel des deux flux vidéo via une interface CameraLink ayant de bonnes propriétés en termes de synchronisation et de débit. Ce module d'acquisition permet l'acquisition en pleine résolution des flux issus des deux caméras à une cadence de 17 Hz et ce, sans compression des images. Le module permet le recalage de son horloge interne par

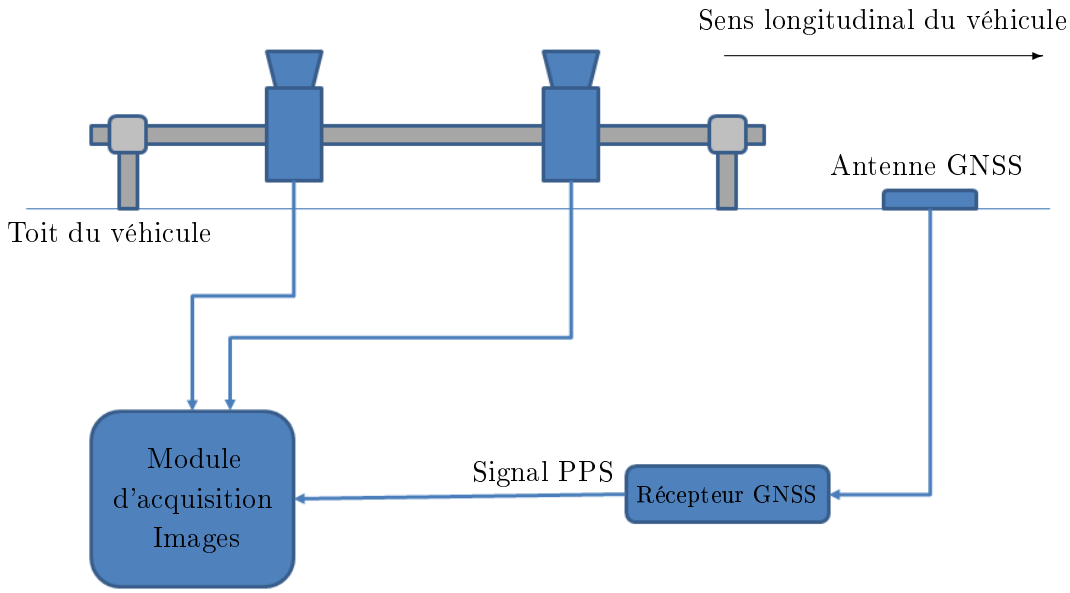


FIGURE 13 – Schéma de la plateforme expérimentale du projet CAPLOC.

le biais d'un signal externe (toutes les secondes à partir du signal PPS du récepteur GNSS). Compte tenu de la cadence d'acquisition et de la résolution importante des images, les vitesses d'écriture nécessaires sont très importantes : afin de profiter pleinement de nos données images dont le flux vidéo est d'environ 132 Mo/s, nous avons opté pour un système d'écriture sur disques SSD (Solid-State Drive) configurés en RAID 0.

Nous avons opté pour un récepteur GNSS RTK (récepteur haut de gamme), le Proflex 500 de la société Ashtech : celui-ci est composé d'une base fixe et d'un récepteur mobile. Les positions brutes de localisation sont reçues à une fréquence de 20 Hz avec des corrections envoyées par HF. En configuration RTK, la précision atteinte après le post-traitement des données peut être centimétrique ce qui nous permettra de considérer ces informations de localisation comme notre référence. La portée du kit HF pour l'envoi et la réception des corrections est de 30 kilomètres avec les antennes cierges, nous conférant une certaine souplesse dans nos déplacements. Une fois configurés, le récepteur GNSS RTK ainsi que le module d'acquisition images sont complètement autonomes et ne nécessitent plus la présence d'un ordinateur ni d'un opérateur.

Une procédure de géo-référencement de chaque capteur (caméras et antennes GNSS) positionné sur le toit du véhicule expérimental sera mis en œuvre afin de pouvoir utiliser l'ensemble des données acquises dans un seul et même référentiel.

Le système est désormais opérationnel : il nous reste à finaliser la partie mécanique pour la fixation des différentes entités sur le toit du véhicule expérimental de l'IFSTTAR-LEOST. Les premières campagnes expérimentales devraient débuter au premier trimestre 2013.

4.2 Géométrie épipolaire

Les méthodes de calibrage classiques de modèle de caméra Pinhole ne peuvent pas être directement appliquées sur notre banc stéréoscopique fisheye. Comme indiqué précédemment dans ce rapport, la transformation fisheye peut s'opérer selon plusieurs types de projection. Le tableau 1 rappelle ces types de projection :

Modèle de projection fisheye	$\rho = \text{proj}_f(\theta)$
Perspective	$\rho = f\tan(\theta)$
Linear scaled (equidistant)	$\rho = f \cdot \theta$
Orthographic	$\rho = f\sin(\theta)$
Equal area (equisolid angle)	$\rho = 2f\sin(\theta/2)$
Stereographic (conform)	$\rho = 2f\tan(\theta/2)$

TABLE 1 – Description des différents modèles de projection

Dans un premier temps, nous avons étudié le modèle de projection équidistant : celui-ci correspond au modèle utilisé pour la génération des images de synthèse utilisées dans ce livrable. Ce modèle est défini par :

$$\begin{bmatrix} \rho_i \\ \varphi_i \end{bmatrix} = \begin{bmatrix} f & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \theta \\ \varphi \end{bmatrix} \tag{1}$$

avec (ρ_i, φ_i) les coordonnées du point projeté dans l'image fisheye, (θ, φ) les coordonnées polaires du point de la scène dans le système de coordonnées sphériques de la caméra fisheye et f la longueur de la focale de l'objectif fisheye [42].

Le centre de l'image correspond au point zénithal et la distance à ce centre, de chaque point dans l'image, est directement proportionnelle à son angle à partir du zénith, ceci étant défini par l'équation :

$$\rho = c.\theta \tag{2}$$

où ρ est la distance du point au centre de l'image, θ est l'angle du faisceau incident et c une constante déterminée par la focale et la taille du capteur. Supposons un rayon 3D incident caractérisé par les deux angles θ et φ comme illustré dans la figure 14. Le modèle de projection dit équidistant projette le faisceau 3D incident aux coordonnées (x, y) de l'image telles que :

$$\begin{cases} x = \rho.cos(\varphi) \\ y = \rho.sin(\varphi) \end{cases} \tag{3}$$

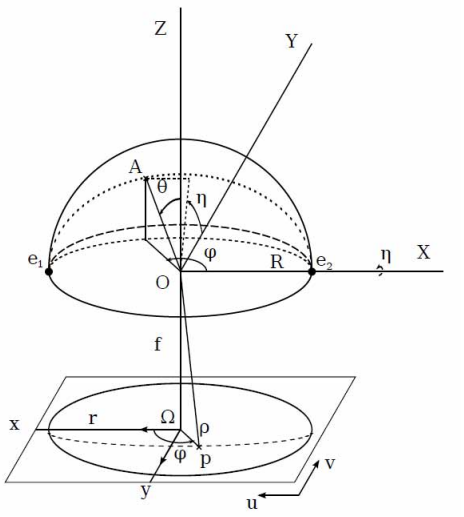


FIGURE 14 – Modèle de projection équidistant

Après la phase de calibrage et de rectification, le système stéréoscopique fisheye se trouve virtuellement dans la configuration décrite dans la figure 15. Sur cette illustration, nous constatons que les quatre épipoles sont alignés et que les capteurs de chaque caméra sont coplanaires.

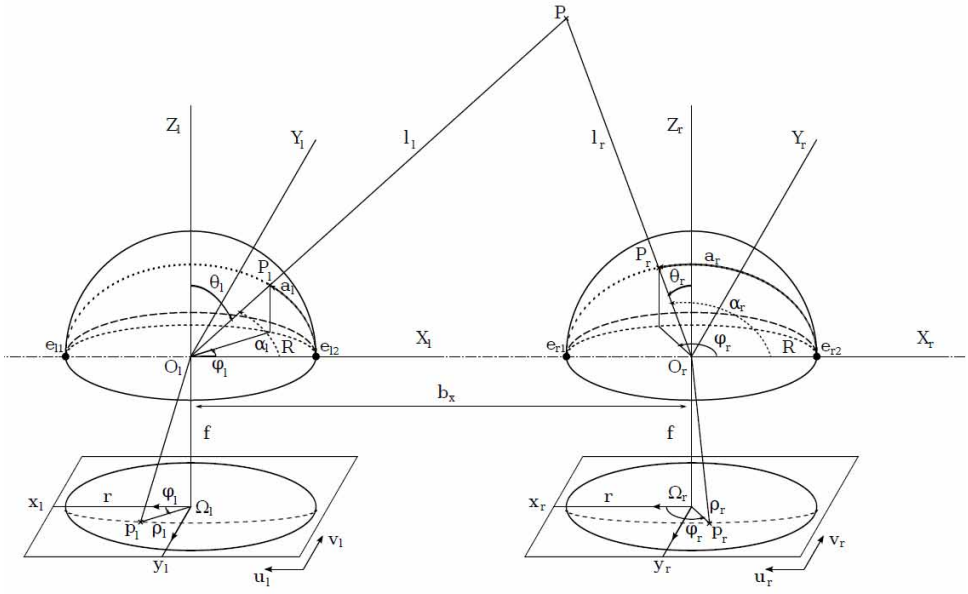


FIGURE 15 – Système stéréoscopique fisheye calibré. Le point P de la scène intersecte les hémisphères gauche et droite respectivement en P_l et P_r, et se projette aux points p_l et p_r dans les deux plans image.

Soit p_i un pixel de la première image (dans notre cas, i peut être l pour le capteur de gauche ou r pour le capteur de droite) dont les coordonnées sont (x_i, y_i) dans le repère centré au centre optique du capteur. Dans le repère polaire, les coordonnées de p_i sont données par les relations suivantes :

$$p_i \begin{pmatrix} \rho_i = \sqrt{x_i^2 + y_i^2} \\ \varphi_i = \arctan(\frac{y_i}{x_i}) \end{pmatrix}_{\text{repère polaire}}$$

La projection P_i de p_i sur l'hémisphère de la lentille est obtenue grâce à l'un des modèles de projection inverse décrit précédemment :

$$P_i \begin{pmatrix} R \\ \theta_i = \text{proj}_f^{-1}(\rho_i) \\ \varphi_i \end{pmatrix}_{\text{repère sphérique}}$$

La conversion en coordonnées cartésiennes 3D donne :

$$P_i \begin{pmatrix} X_i = R\sin(\theta_i)\cos(\varphi_i) \\ Y_i = R\sin(\theta_i)\sin(\varphi_i) \\ Z_i = R\cos(\theta_i) \end{pmatrix}_{\text{repère cartésien}}$$

La courbe épipolaire 3D est décrite par l'arc de cercle passant par P_i et les épipoles e_{i1} et e_{i2} et dont le centre est le centre optique de l'image. Son équation est :

$$\begin{pmatrix} R\cos(\gamma) \\ R\cos(\eta_i)\sin(\gamma) \\ R\sin(\eta_i)\sin(\gamma) \end{pmatrix}_{\text{repère cartésien}}$$

où R est le rayon de l'hémisphère, η est l'angle de rotation autour de l'axe x, γ est variable dans l'intervalle $[0; \pi]$ et

$$\cos(\eta_i) = \frac{\sin(\theta_i)\sin(\varphi_i)}{\sqrt{1 - \sin(\theta_i)^2\cos(\varphi_i)^2}} \tag{4}$$

$$\text{et } \sin(\eta_i) = \frac{\cos(\theta_i)}{\sqrt{1 - \sin(\theta_i)^2\cos(\varphi_i)^2}} \tag{5}$$

Dans le système de coordonnées sphériques, l'équation est donnée par :

$$\begin{pmatrix} R \\ \Theta = \arccos(\sin(\eta_i)\sin(\gamma)) \\ \Phi = \pm\arccos(\frac{\cos(\gamma)}{\sqrt{1-(\sin(\eta_i)\sin(\gamma))^2}}) \end{pmatrix}_{\text{repère sphérique}}$$

Enfin, l'équation des courbes épipolaires projetées sur le plan de la caméra est donnée par l'expression suivante :

$$\begin{pmatrix} \text{proj}_f(\Theta) \\ \Phi \end{pmatrix}_{\text{repère polaire}} = \begin{pmatrix} x_i = \text{proj}_f(\Theta).\cos(\Phi) \\ y_i = \text{proj}_f(\Theta).\sin(\Phi) \end{pmatrix}_{\text{repère cartésien centré dans le plan image}}$$

Notons que dans le cas calibré illustré dans la figure 15, η_i est constant quelle que soit la valeur de i. Dans la suite, nous notons $\eta_i = \eta$.

Chaque épipolaire étant définie par son équation, l'obtention de nuages de points 3D est assurée grâce à l'application de techniques d'appariement le long de chacune d'entre elles. Nous proposons deux solutions pour calculer les coordonnées 3D des points de chaque nuage.

La première solution exploite la reprojection de chaque couple apparié sur les deux hémisphères.

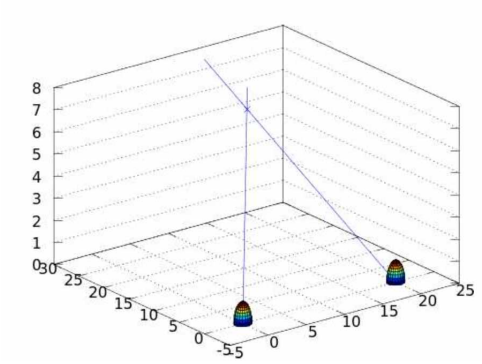


FIGURE 16 – Les deux hémisphères colorées représentent les deux objectifs fisheye. Les deux droites issues des centres optiques et passant par les projections du point de la scène sur les deux hémisphères se coupent en ce même point.

Comme le montre la figure 16, les droites passant par chaque centre optique et par chaque reprojection des points appariés sur les deux hémisphères se coupent au point original de la scène. Les coordonnées de P, dans la caméra gauche, dans le repère cartésien (O_l, x_l, y_l, z_l), s'expriment en fonction des coordonnées de ses projections $P_l(R, \theta_l, \varphi_l)$ et $P_r(R, \theta_r, \varphi_r)$ (voir figure 15). Dans le repère (O_l, x_l, y_l, z_l), les coordonnées des centres optiques sont :

$$O_l \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}_{\text{repère gauche}} \quad \text{et } O_r \begin{pmatrix} b_x \\ 0 \\ 0 \end{pmatrix}_{\text{repère gauche}}$$

où b_x est l'entraxe du système stéréoscopique.

O_lP and O_rP sont données par :

$$O_lP \begin{pmatrix} t\sin(\theta_l)\cos(\varphi_l) \\ t\sin(\theta_l)\sin(\varphi_l) \\ t\cos(\theta_l) \end{pmatrix}_{\text{repère gauche}} \quad \text{avec } t \in [0;+\infty[$$

$$O_rP \begin{pmatrix} b_x + t'\sin(\theta_r)\cos(\varphi_r) \\ t'\sin(\theta_r)\sin(\varphi_r) \\ t'\cos(\theta_r) \end{pmatrix}_{\text{repère gauche}} \quad \text{avec } t' \in [0;+\infty[$$

P étant l'intersection de O_lP et O_rP, ses coordonnées sont obtenues en remplaçant t ou t' par :

$$t = \frac{b_x}{\sin(\theta_l)\cos(\varphi_l) - \cos(\theta_l)\tan(\theta_r)\cos(\varphi_r)}$$

et

$$t' = \frac{b_x}{\cos(\theta_r)\tan(\theta_l)\cos(\varphi_l) - \sin(\theta_r)\cos(\varphi_r)}$$

La seconde solution propose de calculer ce que Li appelle la disparité sphérique [18]. Elle est définie par la différence entre α_r et α_l. Elle est reliée à la distance entre le point P et le centre des deux hémisphères. Soient l_l and l_r les distances de P au centre de la caméra gauche et de la caméra droite. Comme illustré sur la figure 15, en appliquant la loi des sinus, nous obtenons :

$$l_l = \frac{b_x\sin(\alpha_r)}{\sin(\alpha_r - \alpha_l)} \text{ et } l_r = \frac{b_x\sin(\alpha_l)}{\sin(\alpha_r - \alpha_l)}$$

Les valeurs de α_i sont fixées avec la relation : $\alpha_i = \arccos(\sin(\theta_i)\cos(\varphi_i))$. Finalement, les coordonnées de P dans le repère de la caméra gauche sont données par :

$$P \begin{pmatrix} l_l\cos(\alpha_l) \\ l_l\sin(\alpha_l)\cos(\eta) \\ l_l\sin(\alpha_l)\sin(\eta) \end{pmatrix}_{\text{repère gauche}}$$

La section suivante décrit les méthodes que nous avons retenues et les résultats obtenus sur des couples d'images fisheye que nous avons synthétisées.

4.3 Calcul du nuage de points 3D

De nombreuses méthodes de mise en correspondance stéréoscopique ont été proposées dans la littérature [43] mais pour nos travaux, nous souhaitons mettre en œuvre une méthode simple et rapide capable de traiter dans un temps réduit un volume important d'images. Nous avons choisi de tester plusieurs méthodes calculant des cartes de profondeur semi-denses voire denses soit à partir des images fisheye soit à partir des images fisheye rectifiées, puis d'en extraire des nuages de points en vue de leur triangulation. Nos choix se sont portés sur des techniques compatibles avec notre contrainte temps-réel.

Dans [27], une méthode alternative de mise en correspondance stéréoscopique est présentée sur la base de l'utilisation de fenêtres de corrélation 1D multiples pour le calcul des cartes des disparités quasi-denses et des cartes des confiances associées. Reposant sur la contrainte épipolaire utilisée classiquement en stéréovision, nous considérons que l'information nécessaire pour effectuer un appariement efficace est localisée sur les lignes courantes des images gauche et droite.

L'approche 1D peut être décomposée en quatre étapes :

1. Les scores de similarité sont calculés pour chaque pixel avec des fenêtres de similarité de largeurs différentes et pour lesquelles le pixel à apparier est disposé à différentes positions dans ces fenêtres ;
2. Un indice de confiance est calculé pour chaque score de similarité ;
3. Parmi toutes ces configurations de fenêtres, une valeur de disparité (la plus vraisemblable) est affectée à la plupart des pixels de l'image ;
4. L'invalidation d'estimations incorrectes de disparités est effectuée par la méthode « Left-Right Consistency » (plus de détails dans [44]).

Pour chacune de ces étapes, différents outils ou théories peuvent être utilisés. Dans des travaux précédents [45], une étude de cas a été présentée pour laquelle les techniques les plus basiques ont été retenues : l'idée est de dire que si les résultats sont bons avec des techniques basiques, ils le seront d'autant plus avec des techniques plus évoluées. Dans ce travail, la méthode calcule des scores de similarité à partir de fenêtres de corrélation de taille 3×1 à 21×1.

Dans [46], les auteurs proposent une méthode de mise en correspondance stéréoscopique ayant fait ses preuves ; celle-ci est fondée sur l'utilisation de fenêtres support pour lesquelles le poids des pixels dans la mesure de dissimilarité est régi par la similarité colorimétrique ou encore la distance géométrique par rapport au pixel de référence. Cette méthode est ainsi robuste au choix de la taille de la fenêtre support au détriment du temps de calcul. Cependant, cette méthode par corrélation pourra être efficacement implantée sur une architecture de type GPU.

Dans [47], les auteurs proposent de trouver les appariements grâce à une technique de programmation dynamique globale. La procédure décrite dans [47] est implantée sur des couples d'images rectifiées et opère le long des courbes épipolaires. La programmation dynamique est une méthode à base de graphes. Un graphe est construit pour chaque couple de courbes épipolaires. La valeur de chaque nœud du graphe est calculée à partir de la différence du niveau de gris pixel à pixel. La plupart des travaux existants ([48, 18]) transforment les images sphériques pour produire des images rectifiées dans lesquelles des lignes épipolaires sont horizontales ; ceci permet d'appliquer les algorithmes d'appariement classiquement utilisés dans le cas de caméras perspectives. Pour cette méthode de mise en correspondance stéréoscopique, nous avons choisi de ne pas suivre la même approche afin de réduire l'effet de la rectification sur le contenu des images dû à l'étape d'interpolation du processus de rectification. Nous proposons ainsi d'analyser les courbes épipolaires en faisant varier γ dans la gamme $[0, \pi]$. Le nombre de graphes analysés est égal à la résolution verticale des images fisheye.

La plateforme d'acquisition dédiée au projet n'étant pas encore complètement opérationnelle, nous avons testé nos méthodes sur des images de synthèse. Un logiciel de l'IFSTTAR-LIVIC permet la génération et le re-jeu d'ensembles de données telles que pourraient les acquérir un ensemble de capteurs embarqués sur un véhicule réel : ceci permet de tester et d'évaluer nos algorithmes de perception pour l'aide à la conduite. L'un des points forts de ce logiciel est la génération de vérité-terrain pour l'évaluation objective et quantitative des algorithmes élaborés. Pour notre projet, un banc stéréoscopique de caméras munies d'objectifs fisheye a été mis en œuvre dans ce logiciel pour la génération d'un scénario de conduite en environnement urbain. La ligne de base de ce banc stéréoscopique est orientée dans le sens longitudinal du véhicule porteur et les caméras sont orientées verticalement vers le ciel. Les images générées, présentées sur la figure 17 nous permettront de tester les méthodes retenues pour la génération du nuage de points 3D.



FIGURE 17 – (a) Image gauche, (b) image droite et (c) carte des profondeurs pour le couple d'images de synthèse #51 généré par le logiciel du LIVIC.

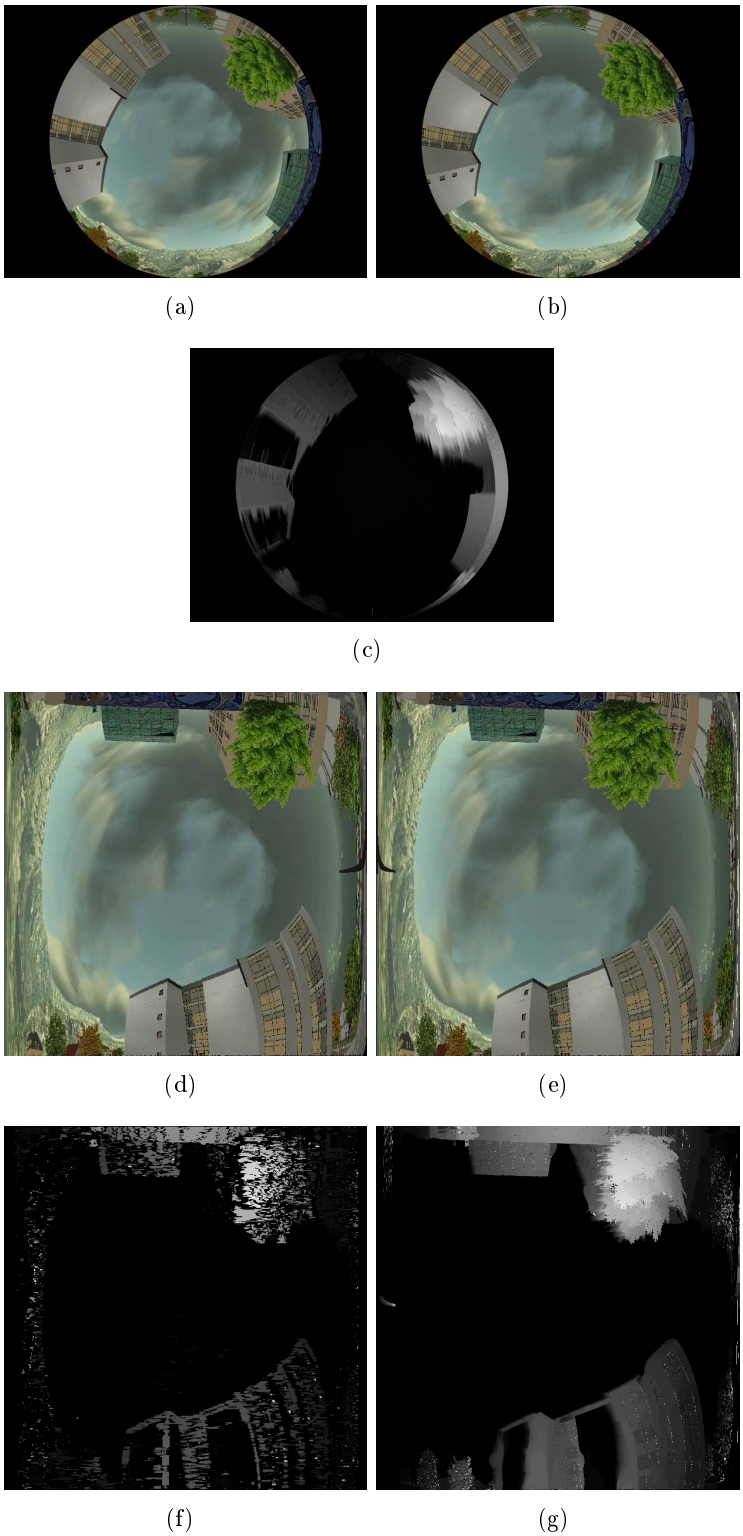
Nous avons appliqué chacune des trois méthodes précédentes sur les couples d'images de la séquence dont nous présentons un échantillon dans la figure 17 i.e. le couple d'images 51 de la séquence. Cette figure présente les images fisheye provenant des deux caméras du système. Ces deux caméras sont placées dans la scène virtuelle comme elles le seraient dans le monde réel. Les caméras virtuelles se déplacent le long d'une trajectoire dans une rue d'une scène urbaine aux dimensions réelles. La séquence est acquise à la cadence vidéo des capteurs du commerce. Le logiciel utilisé comporte un modèle dynamique du véhicule qui permet de synthétiser l'effet du roulis et du tangage du véhicule permettant ainsi de s'approcher des conditions réelles d'utilisation. Ce logiciel produit également une carte de profondeur de référence s'apparentant à une vérité terrain. Cette carte est particulièrement importante puisqu'elle nous permettra de procéder à l'évaluation objective de la précision du nuage des points 3D estimés. Cette évaluation ne fait pas partie de ce livrable mais sera présentée dans le prochain livrable.

Les figures 18, 19 et 20 illustrent les résultats obtenus sur trois couples d'images de la séquence en appliquant les trois méthodes précédemment décrites. Sur ces images, les valeurs des disparités sont codées selon une échelle de gris : la disparité et le niveau de gris augmentent dans le même sens. La méthode par programmation dynamique a été appliquée sans rectification des couples d'images stéréoscopiques. Les équations des courbes épipolaires le long desquelles la mise en correspondance a été réalisée, ont été directement intégrées à l'algorithme. Ce choix a été réalisé afin d'éviter la modification du contenu des images lors de la procédure de rectification. Les deux autres méthodes ([27] et [46]) ont été appliquées sur des images rectifiées selon la méthode décrite dans [18]. Chaque couple d'images rectifiées est présenté sur les images (d) et (e) des figures 18, 19 et 20.

Dans les trois cas, les cartes des disparités sont semi-denses. Dans [27], les auteurs ont montré que la méthode *Fuzzy 1D* fournit des cartes des disparités précises pour des densités de pixels affectés assez faibles. Ils précisent qu'il est possible d'augmenter la densité de la carte en réduisant le seuil de sélection appliqué aux indices de confiance calculés. Toutefois, la densité augmentant, la précision globale des cartes des disparités diminue. Nous avons constaté les mêmes comportements avec les images fisheye CAPLOC. Pour les images présentées dans ce rapport, le seuil a été fixé à 0.8. Dans bon nombre de cas de figure, ce seuil s'est avéré être le meilleur compromis entre densité et précision. Cette valeur devra être confortée lors des étapes d'évaluation. Pour conclure, nous constatons une très bonne estimation de la disparité aux pixels de discontinuité correspondant par exemple aux frontières entre deux régions de la scène situées à des distances différentes. Cette propriété est très intéressante, compte tenu du fait que l'objectif final est de produire un modèle 3D dans lequel tous les bâtiments sont localisés précisément.
La méthode *Adaptative Support-Weight* produit des estimations différentes selon le choix fait sur la taille de la fenêtre support servant au calcul des similarités et de deux autres paramètres notés γ_c et γ_p dans l'article de Yoon *et al.*. Le premier paramètre permet de quantifier la proximité

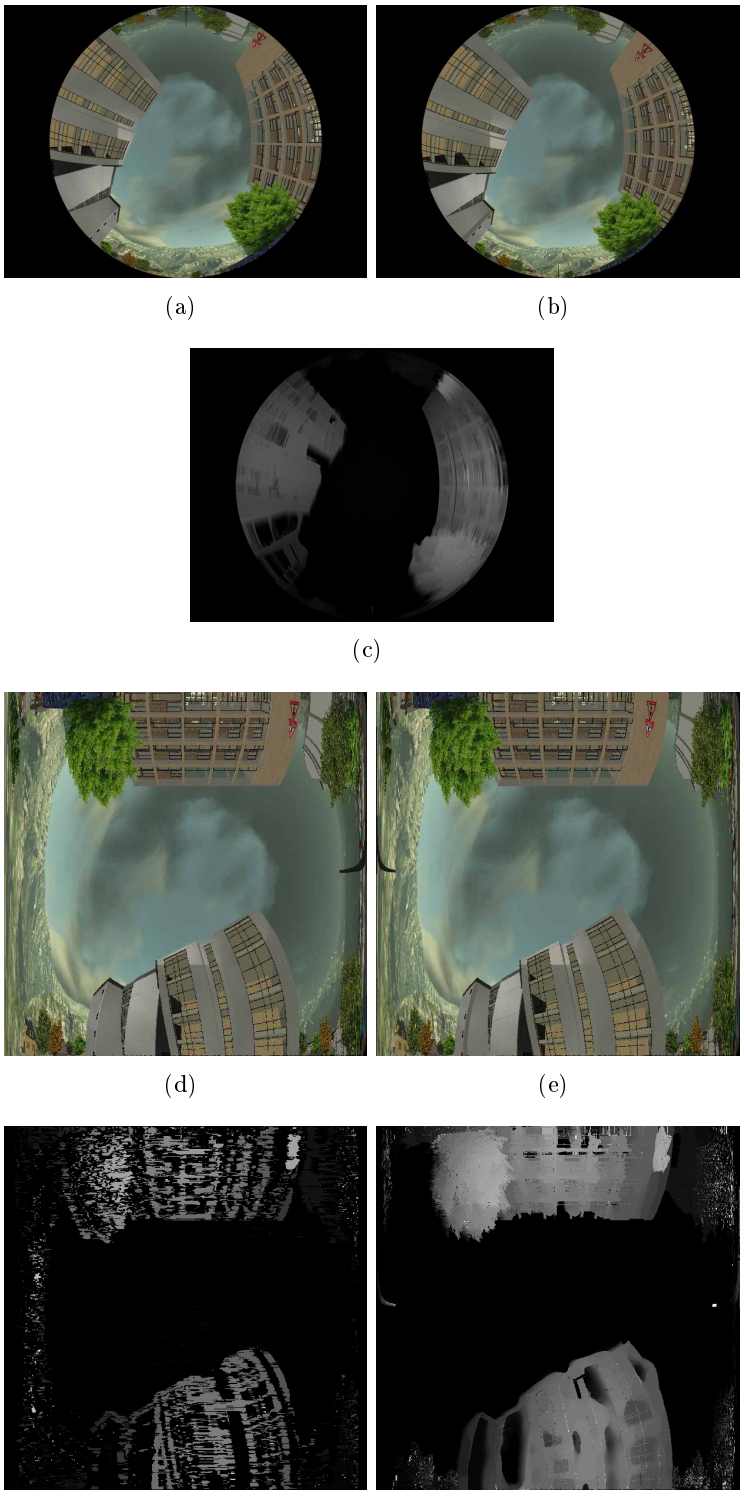
de deux pixels selon l'information colorimétrique qu'ils portent. Le second paramètre est fonction de la taille de la fenêtre support et est utilisé dans l'évaluation de la proximité spatiale des pixels. La méthode proposée par Yoon montre une stabilité des résultats lorsque γ_p augmente. Les cartes des disparités présentées sur les sous-figures 18(g), 19(g) et 20(g) ont été obtenues avec une valeur de γ_p telle qu'elle corresponde à une fenêtre support carrée de dimensions 25×25. Le paramètre γ_c prend une valeur comprise entre 4 et 7 afin de n'engendrer que de légères variations de la précision. Il apparait clairement que les cartes des disparités sont plus denses qu'avec la méthode précédente. Toutefois, les erreurs sont plus importantes aux discontinuités, ce qui réduit la capacité de cette méthode à localiser précisément les arêtes des bâtiments.
La méthode *Programmation Dynamique* ne requiert aucun paramètrage particulier. S'agissant d'une méthode d'appariement par graphe, seule la profondeur d'exploration est limitée afin de ne pas atteindre des temps de calculs trop importants. Il s'agit d'une méthode globale qui apparie tous les pixels de l'image de référence. Toutefois, lorsque le graphe propose plusieurs solutions pour un pixel donné, ce pixel est considéré comme non affecté. La méthode produit tout de même une carte des disparités plus dense que la méthode *Adaptative Support-Weight*. Elle offre de bonnes performances aux discontinuités et permet de localiser efficacement les bâtiments. Néanmoins, des améliorations peuvent encore être apportées pour gérer les phénomènes d'effet de bord comme illustré sur la figure 20(c).

Un nuage de points 3D est calculé à partir de chaque carte des disparités estimées en appliquant le jeu d'équations présenté dans la sous-section précédente. Une interface en OpenGL a été développée afin de visualiser et de naviguer dans les nuages de points 3D calculés. Afin d'opérer un rendu réaliste, le logiciel plaque la texture des images sur le nuage de points repositionnés dans l'espace 3D. Les images de la figure 21 illustrent la qualité des résultats obtenus à partir de la méthode d'appariement par programmation dynamique. Nous constatons l'effet de bord sur la première image de la figure 21. Les deux autres représentations montrent, d'une part, que la précision de la méthode est suffisante pour bien replacer, dans l'espace, les panneaux de signalisation verticale représentés pourtant par peu de pixels dans les images. D'autre part, les contours des bâtiments sont précisément localisés et la géométrie urbaine semble efficacement sauvegardée.



(a) (b) (c) (d) (e) (f) (g)

FIGURE 18 – Résultats de mise en correspondance pour le couple #51 : (a) Image gauche, (b) Image droite, (c) Carte des disparités estimées par *Programmation Dynamique*, (d) et (e) respectivement images gauche et droite rectifiées, (f) Carte des disparités estimées par la méthode *Fuzzy 1D* [27] et (g) Carte des disparités estimées par la méthode *Adaptative Support-Weight* [46].



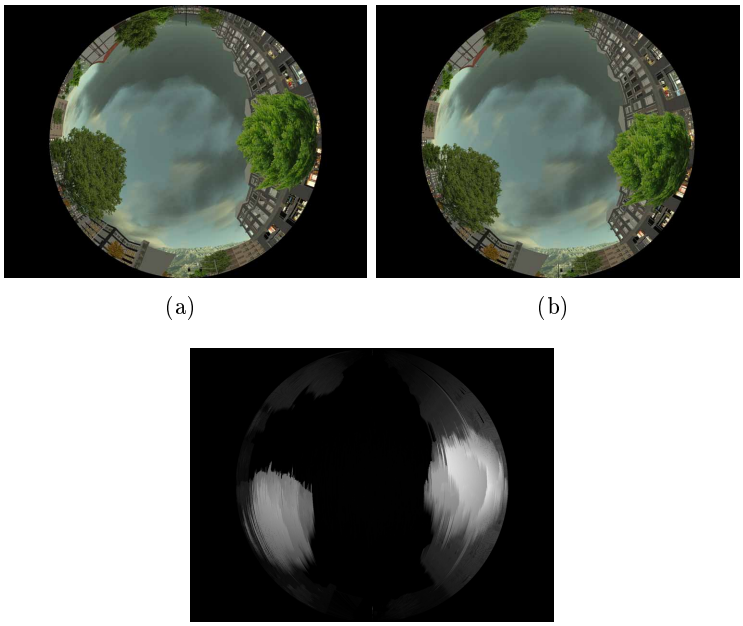
(a) (b)

(c)

(d) (e)

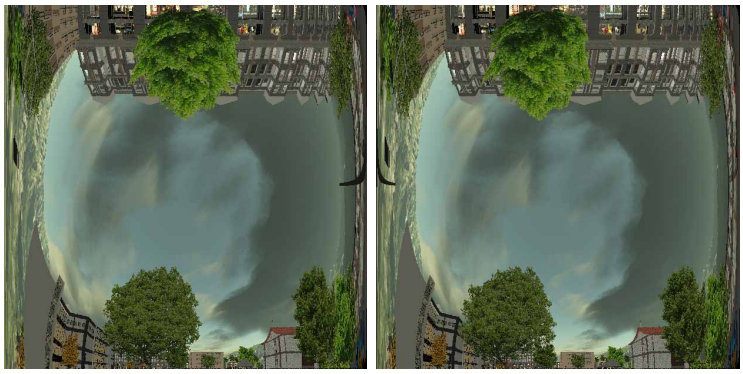
(f) (g)

FIGURE 19 – Résultats de mise en correspondance pour le couple #100 : (a) Image gauche, (b) Image droite, (c) Carte des disparités estimées par *Programmation Dynamique*, (d) et (e) respectivement images gauche et droite rectifiées, (f) Carte des disparités estimées par la méthode *Fuzzy 1D* [27] et (g) Carte des disparités estimées par la méthode *Adaptative Support-Weight* [46].

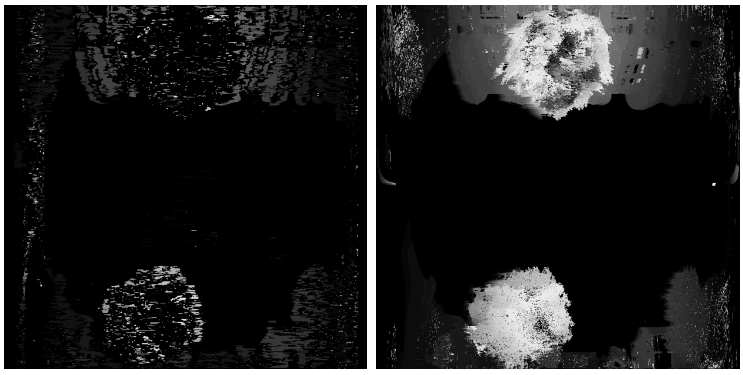


(a) (b)

(c)



(d) (e)



(f) (g)

FIGURE 20 – Résultats de mise en correspondance pour le couple #300 : (a) Image gauche, (b) Image droite, (c) Carte des disparités estimées par *Programmation Dynamique*, (d) et (e) respectivement images gauche et droite rectifiées, (f) Carte des disparités estimées par la méthode *Fuzzy 1D* [27] et (g) Carte des disparités estimées par la méthode *Adaptative Support-Weight* [46].

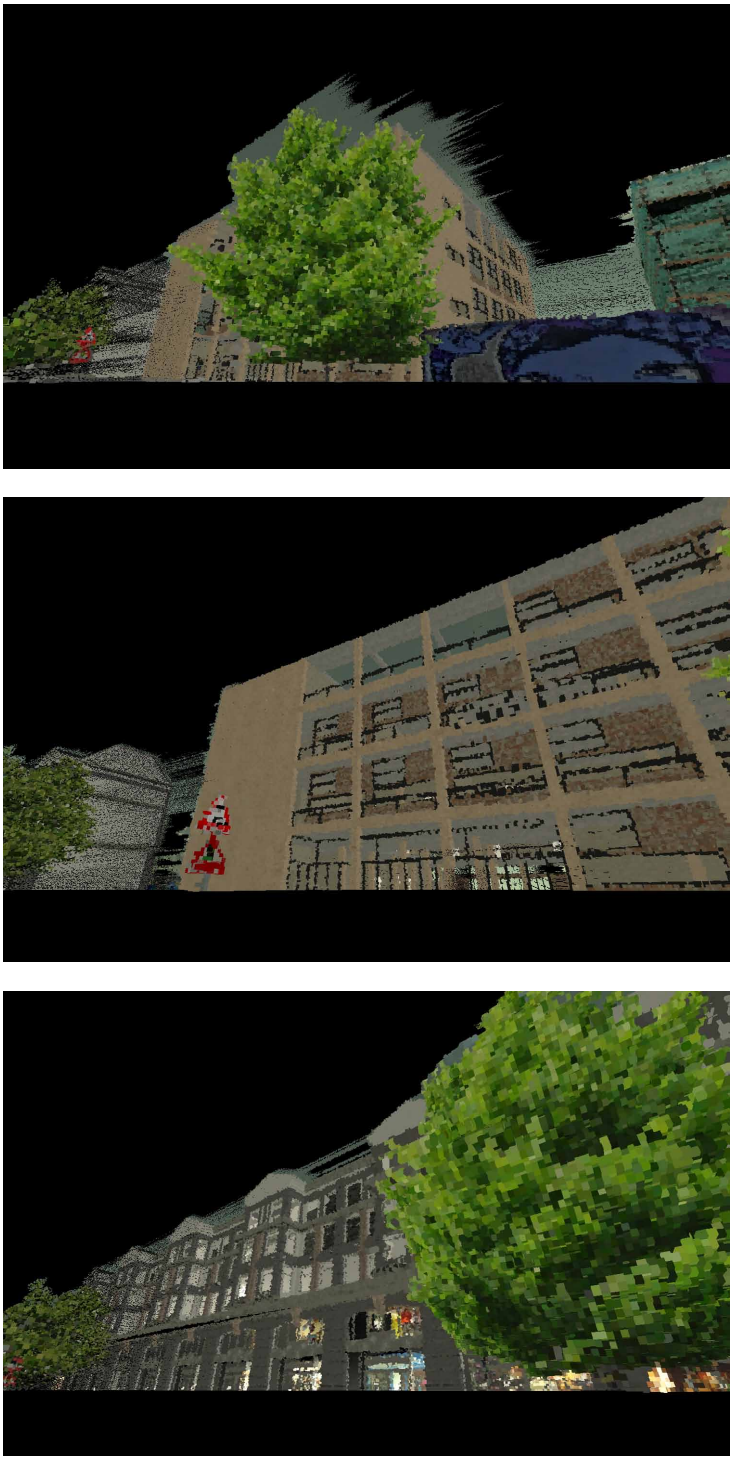


FIGURE 21 – Les nuages de points 3D générés grâce à l'interface OpenGL pour les couples #51, #100 et #300 (de haut en bas).

5 Conclusion et perspectives de travail

Ce rapport constitue le livrable 1 de la tâche 2 du projet CAPLOC. Cette tâche est relative à la reconstruction 3D d'un environnement urbain. Sur la période concernée, les équipes se sont focalisées sur les aspects capteur et perception 3D.

Dans le but d'appréhender l'environnement situé à 360 ° autour d'un mobile, nous avions opté, lors du montage du projet, pour l'utilisation d'un réseau de stéréoscopes composés de caméras répondant au modèle de projection perspective. Ce choix nous imposait de travailler avec un nombre de capteurs très élevé. Ce système multi-caméra nécessitait de faire appel à des techniques de synchronisation très complexes et rendait la tâche de reconstruction impossible à mener face aux contraintes temps-réel que nous souhaitions respecter sauf à utiliser des systèmes de traitement fortement parallèles et donc fortement coûteux. Nous avons donc pris l'option de réduire le nombre de caméras du système. Afin de consolider notre ré-orientation, nous avons organisé une première campagne d'acquisition avec le matériel pressenti lors de la réponse à l'appel à projet. Nous en avons présenté quelques résultats dans la partie 2.4 de ce livrable.

Nous avons mené une large étude bibliographique des systèmes et des méthodes de reconstruction 3D d'environnements urbains et des systèmes assurant une surveillance à 360 ° (catadioptriques et omnidirectionnels). Cette étude nous a conduit à proposer l'utilisation de caméras munies d'objectifs fisheye. Comme l'illustrent nos premiers résultats, l'apparition sur le marché de caméras de très haute résolution nous permet de percevoir un niveau de texture suffisant sur les objets à des distances raisonnables du capteur. Ainsi, la mise en correspondance stéréoscopique fournit des résultats concluants.

Au cours de cette période, nous avons abouti à la définition et à la mise en place de la nouvelle plateforme de perception et d'acquisition composée de caméras hautes résolutions munies d'objectifs fisheye. La majorité des travaux exploitant ce type de capteur visent des applications de détection d'obstacles pour des problématiques de sécurité routière. Dans ces travaux, les axes optiques des caméras sont horizontaux et les entraxes sont comparables à ceux des stéréoscopes classiques. Dans notre application, le système est positionné de telle sorte que les axes optiques soient verticaux. Dans cette configuration, l'entraxe entre les deux caméras doit être assez important pour assurer une estimation de la structure 3D de la scène assez précise qui se manifeste par une différence suffisante entre les deux images : il s'agit d'une caractéristique qui ne nous permet pas d'appliquer aussi simplement les techniques de calibrage décrites dans la littérature et éprouvées. La phase de calibrage est une tâche délicate qui n'est pas encore achevée.

Pour ne pas ralentir les développements en matière de mise en correspondance et de construction de nuages de points 3D, nous avons donc travaillé avec des séquences d'images synthétisées de scènes urbaines. Ce choix nous a permis de nous soustraire à la phase de calibrage et de fournir les premières cartes des disparités et des premiers nuages de points 3D. Plusieurs algorithmes d'appariement ont été implantés. Le logiciel de synthèse calcule une vérité-terrain que nous exploitons actuellement pour mener une évaluation objective des techniques de mise en correspondance retenues. Cette étude n'étant pas encore terminée, elle n'apparait pas dans ce livrable. Toutefois, les résultats préliminaires font apparaitre des conclusions assez positives quant au choix du système et des algorithmes d'appariement. Afin d'illustrer de manière qualitative les nuages de points 3D obtenus, une interface en OpenGL a été développée pour la visualisation et la navigation dans ces nuages de points.

Les perspectives pour la seconde partie de la tâche 2 sont nombreuses. Dans un premier temps, il faudra terminer l'action en cours sur la mise en œuvre d'une procédure de calibrage du système stéréoscopique adaptée à la configuration particulière de notre application i.e. grand entraxe et axes optiques verticaux. Actuellement, un nuage de points 3D est calculé pour un couple

d'images acquis à un instant donné. Il faudra parvenir au recalage temporel des nuages de points 3D produits à partir de plusieurs couples consécutifs ; l'objectif étant de créer un unique nuage de points 3D et dans lequel les outliers ont été supprimés. La méthode ICP « Iterative Closest Points » est pressentie pour recaler les nuages et obtenir un nuage de points 3D cohérent. Pour conclure sur les aspects algorithmiques, nous produirons un maillage 3D à partir de l'analyse du nuage des points 3D obtenu. Là encore, nous envisageons d'appliquer une triangulation du nuage de points 3D par la méthode de Delaunay, méthode relativement simple, rapide et efficace.

Nous validerons cette chaîne logicielle en statique et tout particulièrement le modèle 3D obtenu en l'intégrant au module de calcul de la position de localisation GNSS. Dans la tâche 3, seront testées les techniques de calcul par exclusion et par correction des pseudodistances. Ce dernier test permettra, nous l'espérons, de quantifier l'impact de la précision de notre modèle 3D sur le gain apporté à l'amélioration de la précision de localisation grâce au logiciel de lancer de rayons SE-N@V.

Table des figures

1 Véhicule instrumenté de l'IGN, Stéréopolis II. . . . 6
2 Résultats de reconstruction 3D obtenus dans le cadre du projet *Urbanscape*. . . . 7
3 Exemple de capteur composé d'un réseau de caméras, ASTRO Sensor (Jupiter) [16]. . . . 8
4 Capteur catadioptrique avec un miroir hyperbolique utilisé par [15]. . . . 9
5 Le système OmniCam 3D développé au Fraunhofer HHI. . . . 9
6 Schéma du PAL, les rayons arrivant tout autour de l'objectif sont réfléchis pour atteindre en parallèle une lentille mince convergente puis le capteur [15]. . . . 10
7 Exemples d'images acquises avec le stéréoscope XB3 à Villeneuve d'Ascq. . . . 11
8 (a) Image gauche et (b) carte des disparités calculées par [27] pour les couples 936 et 940 de la séquence. . . . 12
9 Exemples d'images issues d'une caméra fisheye pointant orthogonalement vers le ciel. . . . 13
10 Modèle de projection décrivant la relation entre l'objectif fisheye de la caméra et le plan image [23]. . . . 15
11 Géométrie épipolaire pour des images sphériques [31]. . . . 16
12 Ré-échantilonnage latitude-longitude proposé dans [37]. . . . 18
13 Schéma de la plateforme expérimentale du projet CAPLOC. . . . 21
14 Modèle de projection équidistant 22
15 Système stéréoscopique fisheye calibré. Le point P de la scène intersecte les hémisphères gauche et droite respectivement en P_l et P_r, et se projette aux points p_l et p_r dans les deux plans image. . . . 23
16 Les deux hémisphères colorées représentent les deux objectifs fisheye. Les deux droites issues des centres optiques et passant par les projections du point de la scène sur les deux hémisphères se coupent en ce même point. . . . 24
17 (a) Image gauche, (b) image droite et (c) carte des profondeurs pour le couple d'images de synthèse #51 généré par le logiciel du LIVIC. . . . 27
18 Résultats de mise en correspondance pour le couple #51 : (a) Image gauche, (b) Image droite, (c) Carte des disparités estimées par *Programmation Dynamique*, (d) et (e) respectivement images gauche et droite rectifiées, (f) Carte des disparités estimées par la méthode *Fuzzy 1D* [27] et (g) Carte des disparités estimées par la méthode *Adaptative Support-Weight* [46]. . . . 29
19 Résultats de mise en correspondance pour le couple #100 : (a) Image gauche, (b) Image droite, (c) Carte des disparités estimées par *Programmation Dynamique*, (d) et (e) respectivement images gauche et droite rectifiées, (f) Carte des disparités estimées par la méthode *Fuzzy 1D* [27] et (g) Carte des disparités estimées par la méthode *Adaptative Support-Weight* [46]. . . . 30
20 Résultats de mise en correspondance pour le couple #300 : (a) Image gauche, (b) Image droite, (c) Carte des disparités estimées par *Programmation Dynamique*, (d) et (e) respectivement images gauche et droite rectifiées, (f) Carte des disparités estimées par la méthode *Fuzzy 1D* [27] et (g) Carte des disparités estimées par la méthode *Adaptative Support-Weight* [46]. . . . 31
21 Les nuages de points 3D générés grâce à l'interface OpenGL pour les couples #51, #100 et #300 (de haut en bas). . . . 32

Liste des tableaux

1 Description des différents modèles de projection 22

Références bibliographiques

[1] J. Marais, S. Ambellouis, A. Flancquart et M. Berbineau : Dispositif de localisation - Demande de brevet français déposée le 29/09/2006 n ° 06 54018. N ° de publication : WO/2008/040896, date de publication internationale : 10 avril 2008. 4

[2] F. Lafarge, X. Descombes, J. Zerubia et M. Pierrot-Deseilligny : An automatic 3D city model : a bayesian approach using satellite images. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Toulouse, France, mai 2006. 5

[3] F. Taillandier : Automatic building reconstruction from cadastral maps and aerial images. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume 36, Vienne, Austria, 2005. 5

[4] F. Bretar : *Couplage de Données Laser Aéroporté et Photogrammétriques pour l'Analyse de Scènes Tridimensionnelles.* Thèse de doctorat, Ecole Nationale Supérieure des Télécommunications de Paris, juin 2006. 5

[5] C. Capelle : *Localisation de véhicules et détection d'obstacles, apport d'un modèle virtuel 3D urbain.* Thèse de doctorat, Université des Sciences et Technologies de Lille, décembre 2008. 5

[6] H. Zhang : BoRF : Loop-closure detection with scale invariant visual features. *IEEE International Conference on Robotics and Automation*, pages 3125–3130, 2011. 5

[7] B. Godefroy et P. Sayerse : Amélioration du positionnement par satellites dans un environnement fortement urbanisé, grâce à la prédiction de la propagation des signaux GNSS. *Revue Navigation*, 54(213):17–35, 2006. 5

[8] Y. Suh et R. Shibasaki : Evaluation of satellite-based navigation services in complex urban environments using a three-dimensional GIS. *IEICE Transactions*, pages 1816–1825, 2007. 5

[9] K. Hammoudi : *Contributions to the 3*D *city modeling : 3*D *polyhedral building model reconstruction from aerial images and 3*D *facade modeling from terrestrial 3*D *point cloud and images.* Thèse de doctorat, Universite Paris-Est, 2011. 6

[10] M. Pollefeys, D. Nistér, J. M. Frahm, A. Akbarzadeh, P. Mordohai, B. Clipp, C. Engels, D. Gallup, S. J. Kim, P. Merrell, C. Salmi, S. Sinha, B. Talton, L. Wang, Q. Yang, H. Stewénius, R. Yang, G. Welch et H. Towles : Detailed real-time urban 3D reconstruction from video. *International Journal of Computer Vision*, 78(2-3):143–167, juillet 2008. 7

[11] D. Gallup, J.-M. Frahm, P. Mordohai, Q. Yang et M. Pollefeys : Real-time plane-sweeping stereo with multiple sweeping directions. *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 1–8, Minneapolis, MN, USA, juin 2007. 7

[12] L. Pénard, N. Paparoditis et M. Pierrot-Deseilligny : Reconstruction 3D automatique de façades de bâtiments en multi-vues. RFIA *(Reconnaissance des Formes et Intelligence Artificielle)*, Tours, France, janvier 2006. 7

[13] E. M. Mouaddib : La vision omnidirectionnelle. Guidel, éditeur : *Journées nationales de la recherche en robotique*, 2005. 8

[14] E. M. Mouaddib : Introduction à la vision panoramique catadioptrique. *Traitement du Signal*, 22:409–417, 2005. 8

[15] J.-J. Gonzalez-Barbosa : *Vision panoramique pour la robotique mobile : stéréovision et localisation par indexation d'images.* Thèse de doctorat, Ecole doctorale en informatique et télécomunications, Laboratoire d'Analyse et d'Architecture des Systèmes, Université Toulouse III, 2004. 8, 9, 10, 35

[16] N. Ragot : *Conception d'un capteur de stéréovision omnidirectionnelle : architecture, étalonnage et applications à la reconstruction de scènes 3D.* Thèse de doctorat, Ecole Doctorale Sciences Physiques, Mathématiques et de l'Information pour l'Ingénieur, Université de Rouen, Institut de Recherche en Systèmes Electroniques Embarqués, 2009. 8, 35

[17] T.-T. Lin, Y.-K. Hsiung, G.-L. Hong, H.-K. Chang et F.-M. Lu : Development of a virtual reality GIS using stereo vision. *Computers and Electronics in Agriculture*, 63:38–48, 2008. 8

[18] S. Li : Binocular spherical stereo. *IEEE Transactions On Iintelligent Transportation Systems*, 9:589–600, 2008. 8, 25, 26, 27

[19] C. Weissig, O. Schreer, P. Eisert et P. Kauff : The ultimate immersive experience : Panoramic 3D video acquisition. *Lecture Notes in Computer Science*, 7131:671–681, 2012. 9

[20] L. Paracampo et F. Zoltán : The pal lens. 10

[21] S. Abraham et W. Förstner : Fish-eye-stereo calibration and epipolar rectification. *ISPRS Journal of Photogrammetry and Remote Sensing*, (59):278–288, 2005. 10, 16

[22] S. Ramalingam : *Generic Imaging Models : Calibration and* 3D *Reconstruction Algorithms.* Thèse de doctorat, Institut National Polytechnique de Grenoble, Laboratoire GRAVIR-IMAG, Department of Computer Science, University of California, Santa Cruz, 2006. 10, 14

[23] S. Li : Monitoring around a vehicle by a spherical image sensor. *IEEE Transactions on Intelligent Transportation Systems*, 7(4):541–550, 2006. 10, 15, 35

[24] K. Terabayashi, H. Mitsumoto, T. Morita, Y. Aragaki, N. Shimomura et K. Umeda : Measurement of three-dimensional environment with a fish-eye camera based on structure from motion - error analysis. *Journal of Robotics and Mechatronics*, 21:680–688, 2009. 10, 14, 18

[25] Z. Arican : *Analysis of Multiview Omnidirectional Images in a Spherical Framework.* Thèse de doctorat, Ecole Polytechnique Fédérale de Lausanne, 2010. 10

[26] `http://www.ptgrey.com`. 10

[27] S. Lefebvre, S. Ambellouis et F. Cabestaing : A 1D approach to correlation-based stereo matching. *Image and Vision Computing*, 29:580–593, 2011. 11, 12, 25, 27, 29, 30, 31, 35

[28] L. Pénard : *Reconstruction de façades de bâtiments par appariement dense d'images terrestres géoréférencées haute résolution.* Thèse de doctorat, Université Paris Descartes, France, mars 2009. 12

[29] R. Horaud et O. Monga : *Vision par ordinateur : outils fondamentaux.* Editions Hermès, second édition, 1995. 14

[30] S. Shah et J. K. Aggarwal : Intrinsic parameter calibration procedure for (high-distortion) fish-eye lens camera with distortion model and accuracy estimation. *Pattern Recognition*, 29:1775–1788, 1996. 15

[31] Z. Arican et P. Frossard : Dense depth estimation from omnidirectional images. *Journal of Computer Vision and Image Understanding*, 2010. 16, 35

[32] D. Scaramuzza et R. Siegwart : *Book Vision Systems : Applications*, chapitre A Practical Toolbox for Calibrating Omnidirectional Cameras. 2007. 16

[33] C. Mei et P. Rives : Single view point omnidirectional camera calibration from planar grids. *IEEE International Conference on Robotics and Automation*, avril 2007. 16

[34] J. Heller et T. Pajdla : Stereographic rectification of omnidirectional stereo pairs. *Computer Vision and Pattern Recognition*, pages 1414–1421, 2009. 17

[35] H. Zhao et J. K. Aggarwal : 3D reconstruction of an urban scene from synthetic fish-eye images. *Proceedings of the 4th IEEE Southwest Symposium on Image Analysis and Interpretation*, pages 219–223, Washington, DC, USA, 2000. IEEE Computer Society. 17

[36] S. Hrabar, G. S. Sukhatme, P. Corke, K. Usher et J. Roberts : Combined optic-flow and stereo-based navigation of urban canyons for a UAV. *IEEE International Conference on Intelligent Robots and Systems*, 2005. 17

[37] S. Li : Real-time spherical stereo. *18th IEEE International Conference on Pattern Recognition (ICPR 2006)*, pages 1046–1049, août 2006. 17, 18, 35

[38] T. Nishimoto et J. Yamaguchi : Three dimensional measurement using fisheye stereo vision. *SICE Annual Conference*, pages 2008–2012, 2007. 17

[39] S. K. Gehrig, C. Rabe et L. Krüger : 6D vision goes fisheye for intersection assistance. *Canadian Conference on Computer and Robot Vision*, pages 34–41, 2008. 17

[40] P. J. Herrera, G. Pajares, M. Guijarro, J. J. Ruz et J. M. De La Cruz : A stereovision matching strategy for images captured with fish-eye lenses in forest environments. *Sensors*, 11:1756–1783, 2011. 18

[41] N. Kita : Direct floor height measurement for biped walking robot by fisheye stereo. *IEEE-RAS International Conference on Humanoid Robots*, pages 187–192, 2011. 19

[42] K. Miyamoto : Fish eye lens. *Journal of the Optical Society of America*, 54(8):1060–1061, 1964. 22

[43] D. Scharstein et R. Szeliski : A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1):7–42, avril 2002. `http://vision.middlebury.edu/stereo/`. 25

[44] P. Fua : A parallel stereo algorithm that produces dense depth maps and preserves image features. Rapport de recherche RR-1369, Institut National de Recherche en Informatique et en Automatique, INRIA, janvier 1991. 26

[45] S. Lefebvre : *Approche monodimensionnelle de la mise en correcpondance stéréoscopique par corrélation - Application à la détection d'obstacles routiers*. Thèse de doctorat, Université des Sciences et Technologies de Lille, 2008. 26

[46] K.-J. Yoon et I. S. Kweon : Adaptive support-weight approach for correspondence search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28:650–656, 2006. 26, 27, 29, 30, 31, 35

[47] S. Forstmann, Y. Kanou, J. Ohya, S. Thuering et A. Schmitt : Real-time stereo by using dynamic programming. *Computer Vision and Pattern Recognition Workshop*, 2004. 26

[48] S. Shah et J. K. Aggarwal : Mobile robot navigation and scene modeling using stereo fish-eye lens system. *Machine Vision and Applications*, 10:159–173, 1997. 26



Groupe opérationnel n°2 du Predit 4

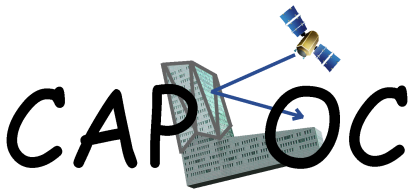
« Gestion du trafic et des mobilités »



CAPLOC

Combinaison de l'**A**nalyse d'images
et de la connaissance de la **P**ropagation
des signaux pour la **LOC**alisation

Convention 10-MT-CVS-001



Livrable de la tâche 2.2

Calcul du nuage de points 3D et extraction d'informations

18 janvier 2015

Sébastien AMBELLOUIS, Amaury FLANCQUART
Ifsttar/LEOST

Julien MOREAU, Yassine RUICHEK
UTBM/IRTES-SeT





Identification

Acronyme du projet	CAPLOC
Titre du projet	Combinaison de l'Analyse d'images et de la connaissance de la Propagation des signaux pour la LOCalisation
Coordinatrice du projet	Juliette MARAIS (Ifsttar-LEOST)
Date de début de projet	19 octobre 2010
Date de fin de projet	18 janvier 2015
Partenaires	Ifsttar/LEOST, UTBM/IRTES-SeT
Type de document	Livrable
Date contractuelle de livraison	18 janvier 2015
Accessibilité	La diffusion de ce document est limitée au programme PREDIT

Gestion des versions

Version	**Date**	**Auteurs**	**Modifications**
1.0	08/12/2014	Julien MOREAU	Rédaction
1.1	14/01/2015	Sébastien AMBELLOUIS,	Rédaction
1.2	16/01/2015	Juliette MARAIS,	Correction
1.3	16/01/2015	Sébastien AMBELLOUIS,	Correction
VF	18/01/2015	Sébastien AMBELLOUIS	Approbation de la version finale

Table des matières

1 Introduction **4**

2 Description des architectures **6**

3 Procédure de calibrage **7**
3.1 Rappels théoriques . . . 8
3.1.1 Paramètres intrinsèques . . . 8
3.1.2 Transformation caméra/image . . . 10
3.1.3 Expression matricielle des paramètres intrinsèques . . . 11
3.1.4 Paramètres extrinsèques . . . 12
3.1.5 Modèle sphérique avec calibrage de la toolbox de Scaramuzza . . . 12
3.1.6 Propriétés de stéréovision pour des capteurs calibrés de manière forte . . . 13
3.2 Description de la méthode . . . 14
3.2.1 Modèle de projection fisheye utilisé . . . 14
3.2.2 Étape de précalibrage . . . 15
3.2.3 Extraction des points d'intérêt . . . 16
3.2.4 Mise à l'échelle unitaire . . . 17
3.2.5 RANSAC (*RANdom SAmple Consensus*) . . . 17
3.2.6 Algorithme des 9 points : Estimation des paramètres et de la matrice fondamentale à partir de 9 correspondances . . . 18
3.3 Evaluation des détecteurs, des descripteurs et des algorithmes d'appariement . . . 20

4 Production du nuage de points **31**
4.1 Géométrie épipolaire fisheye . . . 31
4.1.1 Description et obtention des courbes épipolaires fisheye . . . 32
4.1.2 Obtention des cercles épipolaires conjugués . . . 33
4.1.3 Calcul de la position des points dans l'espace . . . 33
4.2 Mise en correspondance par programmation dynamique . . . 36
4.2.1 Méthode d'optimisation par programmation dynamique . . . 36
4.2.2 Indexation des courbes épipolaires du modèle sphérique . . . 37
4.2.3 Résultats . . . 39
4.3 Post-traitement des nuages de points locaux . . . 39
4.3.1 Extraction de l'empreinte des bâtiments et de leur hauteur . . . 39
4.3.2 Fusion des nuages de points locaux . . . 40
4.3.3 Rayons GNSS dans le nuage de points 3D . . . 42

5 Conclusion et perspectives de travail **50**

Table des figures **52**

Liste des tableaux **53**

Références bibliographiques **55**

1 Introduction

Ce livrable fait la synthèse des derniers résultats de la tâche 2 obtenus durant les derniers mois du projet CAPLOC. Cette tâche porte sur l'estimation de la structure 3D de l'environnement urbain par traitement d'une séquence d'images multi-vues fisheye.

Dans le projet CAPLOC, l'objectif est d'utiliser les informations extraites de l'image pour qualifier et améliorer la précision de localisation GNSS : CAPLOC propose d'améliorer cette précision de localisation en tenant compte de la structure de l'environnement traversé par le véhicule, structure influant directement sur le canal de propagation des signaux GNSS.

La tâche 2 du projet CAPLOC vise à définir un procédé d'extraction de la position et de la structure tri-dimensionnelle des bâtiments afin de calculer puis corriger l'erreur des pseudo-distances asssociée à aux signaux ayant subi une ou plusieurs réflexions.

Nous rappelons que nous avions abandonné l'idée d'utiliser un réseau de stéréoscopes composés de caméras répondant au modèle de projection perspective. Ce choix nous imposait de travailler avec un nombre de capteurs très élevé qui nécessitait de faire appel à des techniques de synchronisation très complexes et rendait la tâche de reconstruction impossible à mener face aux contraintes temps-réel que nous ambitionnions de respecter.

Le système retenu est donc composé de caméras à objectif fisheye dont les axes optiques sont verticales et orientée vers le ciel. Ces caméras sont capables d'opérer une surveillance à 360 ° autour du mobile porteur. Associée à des caméras de très haute résolution, une telle association permet d'améliorer la résolution des surfaces projetées sur les capteurs et ainsi de sauvegarder de manière satisfaisante la qualité des textures tout particulièrement sur les objets situés à des distances raisonnables du mobile.

Deux architectures de perception ont été définies et utilisées. La première est semblable à un système stéréoscopique composé de deux caméras munies d'un objectif fisheye dont les deux axes optiques sont parallèles et orientés verticalement vers le ciel. Les traitements proposés mettent en correspondance le contenu d'un couple d'images acquises simultanément. Les images de la caméra centrale sont utilisées pour réaliser la segmentation entre les éléments bâtiments, de végétation et du ciel. La structure 3D des bâtiments est estimée à partir du traitement des images fournies par les deux autres caméras ; l'entraxe entre ces deux caméras est de 2m et permet de calculer des estimation assez précise compte tenu des dimensions du canyon urbain.

La deuxième architecture exploite une seule caméra fisheye couleur dont l'axe optique est vertical et orienté vers le ciel. Elle est moins encombrante que l'architecture précédente et permet (1) d'estimer la struture 3D de l'environnement et (2) de renseigner directement (sans nécessité de calibrage) la nature des éléments de la scène urbaine à partir des résultats de la segmentation et de la classification ciel/bâtiment/végétation (outil développé dans la tâche 1). Les propriétés 3D des surfaces observées sont estimées à partir de la mise en correspondance du contenu d'images acquises à des instants successifs au cours du déplacement du mobile (depth from motion).

Ce livrable fait suite au livrable 2.1 qui proposait (1) un état de l'art des méthodes sur la reconstruction 3D d'un environnement à partir des images de capteurs panoramiques et omnidirectionnels et qui décrivait (2) les résultats préliminaires en matière de calibrage et de mise en correspondance dans le cadre d'un système fisheye.

Le livrable 2.2 décrit les travaux correspondants aux perspectives définis dans le livrable 2.1 :

- la mise en oeuvre d'une procédure de calibrage compatible quelle que soit l'architecture de perception utilisée ;
- la mise en correspondance du contenu de deux images pour calculer un nuage de points 3D ;
- l'extraction des bâtiments présents dans la scène ;
- la fusion de plusieurs nuages de points consécutifs ;
- le repositionnement des satellites dans le nuage de points.

La première section de ce rapport sera consacrée à une succinte description du sysème mis en place. Elle sera suivie par deux sections sections correspondant respectivement aux deux premiers item et aux trois derniers. Nous terminerons par une section de conclusions et de perpectives. Des résulats seront présentés au fur et à mesure de l'exposé.

2 Description des architectures

Nous commençons par la description du système stéréoscopique retenu et sa mise en oeuvre. Les choix effectués quant aux éléments du système ont été guidés par la large analyse bibliographique qui a été menée et détaillée dans le livrable 2.1. Le système mis en place permet de travailler avec deux architectures pour lesquelles la méthodologie menant au nuage de points 3D est différente. L'une (Arch 1) est fondée sur la mise en correspondance du contenu de deux images acquises au même instant mais à deux positions différentes (stéréovision classique). Une phase de calibrage est réalisée à l'initialisation du système. Elle permet d'estimer la géométrie épipolaire pour faciliter la mise en correspondance stéréoscopique et pour retrouver le facteur d'échelle de la reconstruction. L'autre architecture (Arch 2) est réduite à une seule caméra et exploite le déplacement du mobile pour produire deux vues à deux positions différentes. Elle constitue une architecture multi-vues non rigide pour laquelle il est requis d'estimer le déplacement du mobile entre chaque vue pour estimer la géométrie épipolaire.

La figure 1 illustre le dispositif caméras installé sur le toit du camion laboratoire. Les trois caméras fisheye sont synchronisées et reliées à un système d'acquisition haut débit géoréférencé. Grâce à ce dispositif, il est possible de se placer dans l'une ou l'autre de l'architecture décrite précédemment simplement en exploitant une seule ou plusieurs caméras. Ce dispositif nous a permis de produire des bases de séquences d'images que nous avons utilisées pour le développement et l'évaluation des différents algorithmes proposés. La caméra centrale (caméra 2) du dispositif est utilisée pour réaliser la tâche de classification bâtiment/ciel/végétation des éléments de la scène (tâche 1). Les deux autres caméras (caméra 1 et 3) sont distantes d'un entraxe de $2m$ et sont utilisées pour la reconstruction 3D de la scène. Lorsque le problème Caploc est placé sous la seconde architecture, seule la caméra centrale est utilisée simplifiant ainsi le système.



FIGURE 1 – Dispositif Caploc.

La figure 2 représente un couple d'images fournies par le dispositif embarqué. Il s'agit ici d'un couple stéréo obtenu à partir des caméras 1 et 3.

La tâche 2 comporte une action dont l'objectf est d'identifier la nature de l'objet auquel appartient chaque point 3D reconstruit i.e. si l'objet est un bâtiment, un élément de la végétation ou du ciel. Cette tâche de classification étant réalisée dans la tâche 1, dans le cas de l'architecture mono-caméra, la correspondance est assez simple car la nature de l'objet est déterminée dans l'espace image de la caméra centrale et les points 3D de la scène sont calculés à partir des

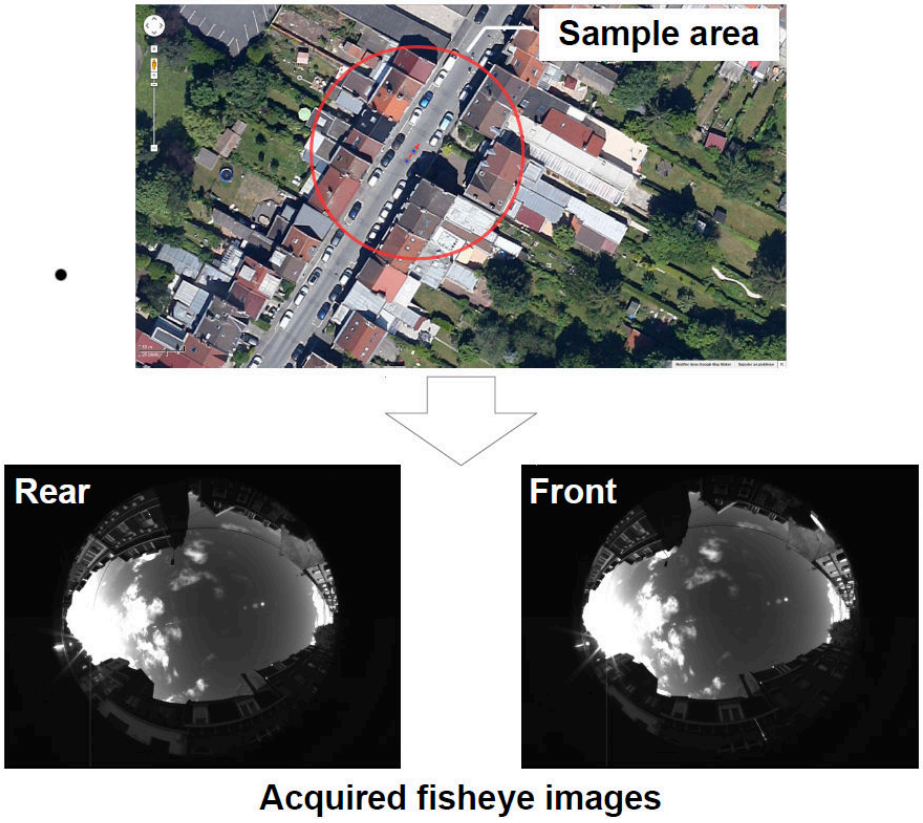


FIGURE 2 – Couple d'images fisheye fournies par le dispositif.

coordonnées de tous les pixels de cette même image. Lorsque nous utilisons trois caméras, il est nécessaire de connaitre la matrice de passage entre le repère des caméras 1-3 et le repère de la caméra 2 pour assurer la correspondance. Le calibrage est donc une étape particulièrement importante dont la qualité influe directement sur la précision du nuage de points 3D et du modèle 3D calculés.

3 Procédure de calibrage

Le processus de mise en correspondance du contenu de plusieurs images nécessite la détermination des paramètres intrinsèques et extrinsèques de chaque caméra ayant servie à produire chaque image considérée. Ces paramètres sont estimés lors d'une phase de calibrage. Dans le cas de l'architecture à 3 caméras, ces paramètres sont déterminés une fois à l'installation du système puis ré-estimeés ponctuellement au cours du déplacement du mobile afin de compenser l'effet des vibrations sur les parties mécaniques du système. Ce récalibrage ponctuel permet de garantir un niveau de précision suffisant et n'impacte globalement que les paramètres intrinsèques du système. Dans le cas de l'architecture mono-caméra, le calibrage est requis tout au long du trajet puisque la transformation entre les repères caméra prises à des position successives n'est pas fixe. Seule les paramètres intrinsèques de la caméra sont constantes. Par ailleurs, il faut noter que, pour cette architecture, les acquisitions sont réalisées tous les $2m$ en triggant la caméra. Cette spécificité permet d'estimer le facteur d'échelle à appliquer sur le modèle afin d'avoir une échelle 1 :1.

Les paramètres intrinsèques décrivent les paramètres internes à la caméra, c'est-à-dire la projection en une image et la relation entre le repère métrique centré sur le centre optique de la caméra (repère caméra 3D) et les coordonnées image en pixel (repère image 2D). Ils représentent

les paramètres du modèle géométrique de la caméra. Les paramètres extrinsèques décrivent le lien entre un repère du monde réel et un repère centré sur la caméra. Ils définissent la transformation appliquée pour passer d'un repère à l'autre. Elle est composée d'une translation et d'une rotation. De nombreuses solutions ont été proposées dans la littérature pour des systèmes vidéo classique c'est à dire des systèmes utilisant des caméras dont la projection est pespective (modèle du sténopé). Avec l'apparition des systèmes omnidirectionnels, ces solutions algorithmiques ont dû être adaptés. Le système CAPLOC est basé sur des caméras munies d'objectifs fisheye dont la projection est sphérique. Dans [1], Mičušìk formalise mathématiquement ce mode de projection ainsi que le problème du calibrage (notamment les paramètres intrinsèques). Il propose des résultats pour une projection de type "équidistante". Dans Caploc, nous utilisons des objectifs pour lesquels le type équidistant n'est pas adapté. Nous avons développé un processus de calibrage pour une projection sphérique de type "équisolide".

Cette section propose de faire un parallèle entre les caméras dont l'objectif répond à un modèle de projection perspective et à un modèle de projection sphérique afin de faciliter le lecteur habitué des projections plus classiques. Le contenu concernant les caméras respectant le modèle perspective est inspiré du livre de Horaud et Monga [2]. Le contenu concernant les caméras omnidirectionnelles est un rassemblement d'informations issues des documents de Mičušìk et Scaramuzza.

3.1 Rappels théoriques

3.1.1 Paramètres intrinsèques

Les paramètres intrinsèques décrivent les paramètres internes à la caméra, c'est-à-dire la projection en une image et la relation entre le repère métrique centré sur le centre optique de la caméra (repère caméra 3D) et les coordonnées image en pixel (repère image 2D). Ils représentent les paramètres du modèle géométrique de la caméra, qui est caractérisé par :

- une projection qui transforme un point de l'espace (3D) en un point image (2D) dans le repère caméra, représentée par la matrice **P**,
- et une transformation du repère caméra (métrique) au repère image (pixel), représentée par la matrice **K**.

3.1.1.1 La projection perspective d'une caméra

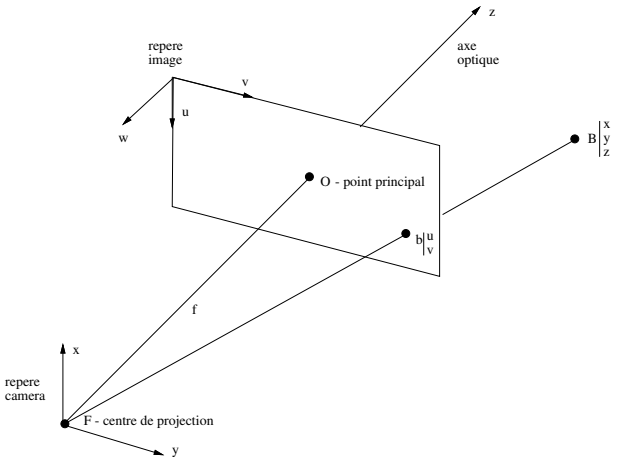


FIGURE 3 – Modèle de projection perspective d'une caméra. Issu de [2].

La projection perspective est décrite par la figure 3. On travaille dans le repère de coordonnées

attaché à la caméra. Le point $B \begin{pmatrix} x \\ y \\ z \end{pmatrix}$ de l'espace se projette en $b \begin{pmatrix} x' = x\frac{f}{z} \\ y' = y\frac{f}{z} \\ z' = f \end{pmatrix}$ dans le plan image, parallèle au plan (x, y), à une distance f le long de l'axe z, où f est la longueur focale de l'objectif.
Cette transformation peut s'écrire sous forme matricielle, en coordonnées homogènes :

$$\mathbf{P} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{1}{f} & 0 \end{pmatrix} \tag{1}$$

3.1.1.2 La projection omnidirectionnelle (modèle sphérique)

Le modèle décrit suppose les caméras omnidirectionnelles, dioptriques fisheye et catadioptriques, comme étant centrales. Présenter un point de vue central signifie que les rayons optiques se croisent en un seul point et ne forment pas de *locus* (lieu géométrique), une zone de croisement des rayons optiques. On considère également que le capteur omnidirectionnel présente une parfaite symétrie radiale autour de son axe optique, et que le plan image est parfaitement parallèle à l'objectif et au miroir.

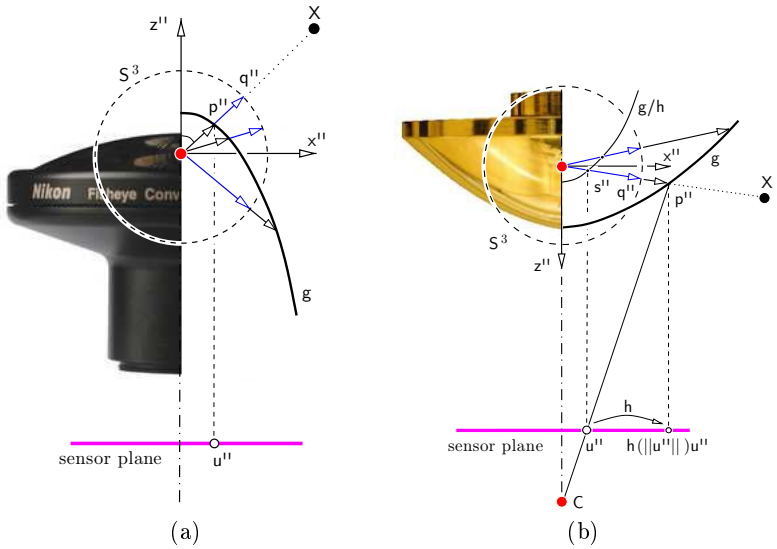


FIGURE 4 – Modèle de projection sphérique pour un objectif fisheye (a) et pour un capteur catadioptrique (b). Issu de [1].

La projection sphérique est décrite figure 4. On travaille dans le repère de coordonnées attaché à la caméra, centré sur la sphère unitaire. Notons que dans le cas du capteur catadioptrique, la fonction g représente la forme du miroir utilisé, et une deuxième fonction h est utilisée pour représenter la projection de la caméra (perspective ou orthographique selon les cas). Ainsi, la projection sur le plan image à partir de la sphère unitaire s'obtient grâce à $\frac{g}{h}$. Dans le cas fisheye, seule la fonction g est utilisée, et elle représente à elle seule la projection fisheye utilisée et ses distorsions propres.
Pour généraliser, on choisit de reprendre la notation de Scaramuzza, et d'appeler la fonction de projection $f(\rho)$ avec $\rho = \sqrt{x'^2 + y'^2}$ dans les deux cas catadioptrique ($\frac{g}{h}$) et fisheye (g). Le

point $X \begin{pmatrix} x \\ y \\ z \end{pmatrix}$ de l'espace se projette en un vecteur $q'' \begin{pmatrix} x_n = \frac{x}{\|\overrightarrow{OX}\|} \\ y_n = \frac{y}{\|\overrightarrow{OX}\|} \\ z_n = \frac{z}{\|\overrightarrow{OX}\|} \end{pmatrix}$ sur la sphère unitaire, puis en un vecteur $p'' \begin{pmatrix} x'' = x_n\|p''\| \\ y'' = y_n\|p''\| \\ z'' = f(\rho) = z_n\|p''\| \end{pmatrix}$, avec $\|p''\| = \frac{f(\rho)}{z_n}$ par rapport à la fonction de projection $f(\rho)$ utilisée, et enfin en un point $u'' \begin{pmatrix} x' = x'' \\ y' = y'' \\ z' = k \end{pmatrix}$ sur le plan du capteur avec k un réel quelconque.
On cherche donc $\mathbf{P}$ en coordonnées homogènes telle que :

$$
\begin{aligned}
u'' &= \mathbf{P}X \\
u'' &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & k \\ 0 & 0 & 0 & 1 \end{pmatrix} \underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{1}{f(\rho)} & 0 \end{pmatrix} \underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \|\overrightarrow{OX}\| \end{pmatrix}}_{q''} X}_{p''}
\end{aligned}
\tag{2}
$$

On obtient :

$$
\mathbf{P} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & k\frac{1}{f(\rho)} & 0 \\ 0 & 0 & \frac{1}{f(\rho)} & 0 \end{pmatrix} \tag{3}
$$

3.1.2 Transformation caméra/image

3.1.2.1 Caméra pinhole

La transformation caméra/image est une transformation affine représentant un changement d'échelle, une rotation et une translation. On peut l'écrire sous la forme d'une matrice 3×4 en coordonnées homogènes pour représenter le passage d'un système en 3 dimensions (repère caméra dans l'espace) à un système à 2 dimensions (repère image, sans axe z). On obtient ainsi $b\binom{u}{v}$ dans le repère image, à partir de $b \begin{pmatrix} x' \\ y' \\ z' \end{pmatrix}$ dans le repère caméra. Cette transformation linéaire s'écrit en coordonnées homogènes :

$$
\mathbf{K} = \begin{pmatrix} -k_u & 0 & 0 & u_0 \\ 0 & k_v & 0 & v_0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{4}
$$

La composante z' de b n'a aucune importance sur le résultat du passage au repère 2D. Les valeurs u_0 et v_0 indiquent les coordonnées de la position du centre de la caméra (point principal) dans l'image, en pixels (dans le repère image).

3.1.2.2 Caméra omnidirectionnelle

Scaramuzza, avec sa méthode de calibrage, présente la transformation affine caméra/image sous

la forme suivante, mais toujours en 2 dimensions (reprise avec la notation de Horaud et Monga) :

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} c & d \\ e & 1 \end{pmatrix} \begin{pmatrix} x' \\ y' \end{pmatrix} + \begin{pmatrix} u_0 \\ v_0 \end{pmatrix} \tag{5}$$

Les valeurs c, d et e sont les paramètres issus du calibrage de Scaramuzza, et décrivent la transformation de l'image en forme d'ellipse en un cercle parfait, en plus du changement d'échelle et de la rotation. En mettant la transformation décrite par Scaramuzza sous la forme d'une matrice en coordonnées homogènes passant de 3 (repère caméra) à 2 dimensions (repère image), on aboutit à :

$$\mathbf{K} = \begin{pmatrix} c & d & 0 & u_0 \\ e & 1 & 0 & v_0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{6}$$

3.1.3 Expression matricielle des paramètres intrinsèques

Le modèle géométrique de la caméra permet de passer des coordonnées du point $B \begin{pmatrix} x \\ y \\ z \end{pmatrix}$ dans le repère caméra aux coordonnées du point projeté $b\binom{u}{v}$ dans le repère image. Celui-ci s'obtient en appliquant la projection de la caméra (perspective ou fisheye ou catadioptrique), suivie d'une transformation affine, c'est-à-dire en multipliant les matrices $\mathbf{K}$ et $\mathbf{P}$. On note $\mathbf{I_C}$ la matrice qui exprime la transformation complète.

3.1.3.1 Cas du modèle sténopé

Le produit $\mathbf{KP}$ est :

$$\mathbf{KP} = \begin{pmatrix} -k_u & 0 & \frac{u_0}{f} & 0 \\ 0 & k_v & \frac{v_0}{f} & 0 \\ 0 & 0 & \frac{1}{f} & 0 \end{pmatrix} \tag{7}$$

En appliquant un facteur d'échelle égal à f, on obtient la matrice des paramètres intrinsèques :

$$\mathbf{I_C} = \begin{pmatrix} \alpha_u & 0 & u_0 & 0 \\ 0 & \alpha_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{8}$$

avec $\alpha_u = -k_u f$ et $\alpha_v = k_v f$. On peut décomposer $\mathbf{I_C}$ en une transformation affine caméra/image et en une transformation projective :

$$\mathbf{I_C} = \begin{pmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{9}$$

Soit $\mathbf{C}$ la transformation affine caméra/image :

$$\mathbf{C} = \begin{pmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{pmatrix} \tag{10}$$

3.1.3.2 Cas du modèle sphérique

Le produit **KP** est :

$$\mathbf{KP} = \begin{pmatrix} c & d & \frac{u_0}{f(\rho)} & 0 \\ e & 1 & \frac{v_0}{f(\rho)} & 0 \\ 0 & 0 & \frac{1}{f(\rho)} & 0 \end{pmatrix} \tag{11}$$

En appliquant un facteur d'échelle égal à $f(\rho)$, on obtient la matrice des paramètres intrinsèques :

$$\mathbf{I_C} = \begin{pmatrix} c.f(\rho) & d.f(\rho) & u_0 & 0 \\ e.f(\rho) & f(\rho) & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{12}$$

Soit **C** la transformation affine caméra/image :

$$\mathbf{C} = \begin{pmatrix} c.f(\rho) & d.f(\rho) & u_0 \\ e.f(\rho) & f(\rho) & v_0 \\ 0 & 0 & 1 \end{pmatrix} \tag{13}$$

3.1.4 Paramètres extrinsèques

Les paramètres extrinsèques décrivent le lien entre le repère de la mire et le repère centré sur la caméra. Il s'agit en fait de la transformation mire/caméra, composée d'une translation et d'une rotation.

3.1.4.1 Propriétés valables quelque soit le modèle de caméra utilisé

La transformation mire/caméra est une transformation rigide, qui peut s'écrire sous la forme d'une matrice en coordonnées homogènes :

$$\mathbf{A} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{pmatrix} \tag{14}$$

La matrice inverse est donnée par :

$$\mathbf{A^{-1}} = \begin{pmatrix} \mathbf{R}^t & -\mathbf{R}^t\mathbf{t} \\ \mathbf{0} & 1 \end{pmatrix} \tag{15}$$

3.1.5 Modèle sphérique avec calibrage de la toolbox de Scaramuzza

Pour écrire l'expression de **A** à partir des informations que l'on obtient en utilisant la toolbox de Scaramuzza OCamCalib, Scaramuzza fournit les paramètres extrinsèques du calibrage avec les données **RRfin**. C'est une matrice $3\times3\times M$, où M correspond au nombre d'images utilisées pour le calibrage (on connaît les paramètres extrinsèques 3×3 pour chaque position de la mire). La $m^{\text{ème}}$ matrice 3×3 peut s'écrire :

$$\mathbf{RRfin}_m = \begin{pmatrix} \mathbf{r}_x & \mathbf{r}_y & \mathbf{t} \end{pmatrix} = \begin{pmatrix} r_{x1} & r_{y1} & t_x \\ r_{x2} & r_{y2} & t_y \\ r_{x3} & r_{y3} & t_z \end{pmatrix} \tag{16}$$

On obtient $\mathbf{r}_z$ par le produit vectoriel entre $\mathbf{r}_x$ et $\mathbf{r}_y$:

$$\mathbf{r}_z = \mathbf{r}_x \wedge \mathbf{r}_y \tag{17}$$

Ainsi, on peut écrire :

$$\mathbf{A} = \begin{pmatrix} r_{x1} & r_{x2} & r_{x3} & t_x \\ r_{y1} & r_{y2} & r_{y3} & t_y \\ r_{z1} & r_{z2} & r_{z3} & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{pmatrix} \tag{18}$$

3.1.6 Propriétés de stéréovision pour des capteurs calibrés de manière forte

3.1.6.1 Matrice de passage du repère caméra gauche au repère caméra droite (calibrage stéréoscopique)

Le calibrage stéréoscopique consiste à déterminer la matrice de transformation entre le repère caméra gauche et le repère caméra droite $\mathbf{A_s}$. Cette matrice se calcule à partir des paramètres extrinsèques des deux caméras calibrées en même temps avec la mire aux mêmes positions dans l'espace. Il s'agit en fait de passer du repère caméra gauche au repère mire, puis du repère mire au repère caméra droite (d'où la nécessité de calibrer les deux caméras avec la mire à la même position). À l'aide des paramètres extrinsèques, on construit les matrices $\mathbf{A}$, qui décrit le passage du repère mire au repère caméra gauche, et $\mathbf{A'}$, qui décrit le passage du repère mire au repère caméra droite. On calcule ensuite $\mathbf{A_s}$:

$$\mathbf{A_s} = \mathbf{A'}\mathbf{A}^{-1} \tag{19}$$

Cette matrice s'écrit comme $\mathbf{A}$ et $\mathbf{A'}$:

$$\mathbf{A_s} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & b_x \\ r_{21} & r_{22} & r_{23} & b_y \\ r_{31} & r_{32} & r_{33} & b_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{20}$$

Le vecteur $\mathbf{b}\begin{pmatrix} b_x \\ b_y \\ b_z \end{pmatrix}$ est le vecteur allant du centre de la caméra gauche à la caméra droite. C'est le *baseline* (l'entraxe) entre les deux capteurs configurés pour la stéréovision.

3.1.6.2 Expression de la matrice Essentielle

La matrice essentielle $\mathbf{E}$ décrit la transformation épipolaire gauche/droite dans les repères caméras. Elle permet de calculer l'équation d'une droite épipolaire de paramètres (a', b', c') (d'équation de la forme $a'x' + b'y' + c' = 0$) dans l'image de droite associée à un point de l'image de gauche $\begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$. On écrit :

$$\mathbf{p'}^t\mathbf{E}\mathbf{p} = 0 \tag{21}$$

La transformation épipolaire droite/gauche est donnée par la matrice transposée :

$$\mathbf{p}^t\mathbf{E}^t\mathbf{p'} = 0 \tag{22}$$

La matrice essentielle est le produit de deux matrices, calculées à partir des paramètres $\mathbf{b}\begin{pmatrix} b_x \\ b_y \\ b_z \end{pmatrix}$, $\mathbf{r}_1(r_{11}\ r_{12}\ r_{13})$, $\mathbf{r}_2(r_{21}\ r_{22}\ r_{23})$ et $\mathbf{r}_3(r_{31}\ r_{32}\ r_{33})$:

$$\begin{pmatrix} a' \\ b' \\ c' \end{pmatrix} = \mathbf{E}\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} 0 & -b_z & b_y \\ b_z & 0 & -b_x \\ -b_y & b_x & 0 \end{pmatrix}\begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix}\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \tag{23}$$

3.1.6.3 Expression de la matrice Fondamentale

La matrice fondamentale **F** décrit la transformation épipolaire gauche/droite dans les repères images. La relation entre les coordonnées caméra et les coordonnées image est donnée par (pour la caméra gauche) :

$$\mathbf{m}\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \mathbf{C}\mathbf{p}\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \tag{24}$$

On a la même expression pour la caméra droite :

$$\mathbf{m}'\begin{pmatrix} u' \\ v' \\ 1 \end{pmatrix} = \mathbf{C}'\mathbf{p}'\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} \tag{25}$$

Les éléments des matrices **C** et $\mathbf{C}'$ sont les paramètres intrinsèques des deux caméras. On obtient la matrice fondamentale :

$$\mathbf{F} = (\mathbf{C}'^{-1})^t\mathbf{E}\mathbf{C}^{-1} \tag{26}$$

L'équation :

$$\mathbf{m}'^t\mathbf{F}\mathbf{m} = 0 \tag{27}$$

permet de déterminer l'équation d'une droite épipolaire à partir d'un point dans les repères images.

3.2 Description de la méthode

Nous avons mis en place une procédure de calibrage automatique. De nombreuses méthodes proposent de calibrer de tels systèmes à partir d'une mire présentée manuellement devant les caméras dans plusieurs positions et à différentes distances et en utilisant la toolbox de Scaramuzza. Toutefois ces techniques manuelles posent des problèmes pour notre système. Dans la cas d'Arch 1, l'entraxe entre les caméras est grande ce qui nécessiterait une mire très grande à déplacer dans un espace étendu. De plus, pour Arch 2, un calibrage devant être réalisé pour chaque couple, une procédure manuelle est à proscrire. L'objectif étant de déterminer la géométrie épipolaire du système, le processus proposé cherche à déterminer les valeurs de la matrice fondamentale en s'inspirant de la méthode proposée dans [3] et [1].

Notre méthode propose d'extraire automatiquement un ensemble de points d'intérêt dans un couple d'images (couple stéréo ou couple d'images acquises successivement) puis de les mettre en correspondance. L'estimation des valeurs de la matrice fondamentale est réalisée en minisant l'erreur globale de mise en correspondance par l'algorithme RANSAC. L'erreur globle est calculée à partir de plusieurs points pris aléatoirement dans la liste des points d'intérêt.

3.2.1 Modèle de projection fisheye utilisé

La projection fisheye peut être décrite de manière efficace via le passage par une sphère unitaire.

Dans la figure 5, un point $\mathbf{u}_{2D}$ sur le capteur peut-être représenté sur une rétine en forme de sphère unitaire par le rayon **q**, via le passage à **u** avec $w = 1$. **q** est le rayon normalisé de **p**, qui est obtenu via la fonction de projection dépendant du rayon $g(r)$ ($\mathbf{u}_{2D}(\rightarrow \mathbf{u}) \rightarrow \mathbf{p} \rightarrow \mathbf{q}$). Le w de **p** est donné par $g(r)$ et $r = \sqrt{u^2 + v^2} = \|\mathbf{u}_{2D}\|$. On voit que $w = g(r) = \frac{r}{\tan(\theta)}$ (car $\frac{w}{r} = \tan(\frac{\pi}{2} - \theta) = \frac{1}{\tan(\theta)}$), et donc $\mathbf{p}\left(_{g(r)=\frac{r}{\tan(\theta)}}^{\mathbf{u}_{2D}}\right)$.

Selon le modèle de projection utilisé, les paramètres seront différents. Des modèles plus complexes peuvent avoir plus de paramètres. Jusqu'à présent nous avons essayé deux modèles simples. Ils sont linéaires, à un paramètre et correspondent à deux types de projection fisheye différents.

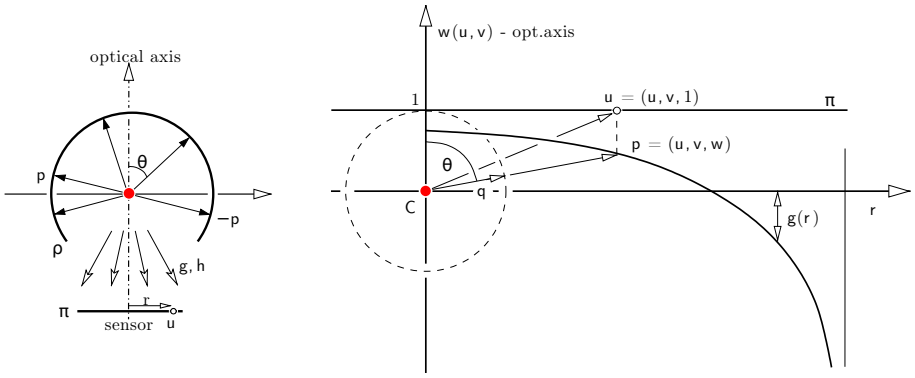


FIGURE 5 – Fonction de projection fisheye avec le modèle sphérique [1].

3.2.1.1 Modèle linéaire à un paramètre de projection équisolide

Un des trois modèles proposés par Mičušìk [1] est le simple modèle linéaire à un paramètre a, pour la projection équidistante. Il correspond bien à une projection équiangulaire (ou équidistante), définie par $r = f.\theta$ ou $\theta = a.r$ avec $a = \frac{1}{f}$. Pour le modèle linéaire à un paramètre équidistant, on obtient donc $g(r, a) = \frac{r}{\tan(a.r)}$. L'objectif des caméras que nous utilisons est le SIGMA 4.5mm F2.8 EX HSM IF. D'après les données du constructeur, il approche la projection à angle équisolide et restitue un angle de vision de 180°. Le modèle proposé par Mičušìk ne semblant pas très adapté à notre matériel, nous avons proposé d'utiliser le modèle linéaire à un paramètre pour la projection à angle équisolide définie par la relation $r = 2f \sin(\frac{\theta}{2})$, que l'on peut encore écrire $\theta = 2 \arcsin(\frac{r}{2f})$

À partir de cette relation, on obtient $g(r, a) = \frac{r}{\tan(2 \arcsin(a.r))}$.

3.2.2 Étape de précalibrage

En pratique, on considère que les images projetées sur les capteurs forment bien des cercles (en réalité ce sont des ellipses très proches d'un cercle) et que les pixels sont bien carrés. L'étape de pré-calibrage consiste à détecter le cercle principal dans les deux images, c'est-à-dire le cercle dans lequel toute l'information projetée est contenue.

L'étape est réasliée de manière manuelle pour le moment. Pour ce faire, on prend trois points du contour, ce qui permet de déterminer l'équation de chaque cercle, et donc leur centre et leur rayon. La position du centre nous donne les translations à appliquer pour passer du repère image au repère centré sur le cercle dans chaque image. En principe, on devrait trouver des cercles de même rayon ou presque. Si ce n'est pas le cas, on considère qu'ils ont bien le même rayon, que l'on fixe à la valeur moyenne des deux rayons estimés.

Avec les essais réalisés pour le moment, en admettant que les trois points utilisés pour estimer les cercles sont bien situés sur les bords (déterminés à la main), on obtient les rayons $r_1 = 1145,4$ et $r_2 = 1149,22$ pixels. Le rayon moyen est $r = 1147,31$ pixels, ce qui fait une erreur maximale de $\frac{|\Delta_r|}{r} = \frac{|r_2 - r_1|}{r} = 0,0033 = 0,33$ % en le choisissant comme rayon de référence.

Soient $\mathbf{t_1}$ et $\mathbf{t_2}$ les vecteurs de translation des repères et (a_1, b_1) et (a_2, b_2) les coordonnées des centres des cercles des deux images, on a :

$$\mathbf{t_1}\begin{pmatrix}-a_1\\-b_1\end{pmatrix} \text{ et } \mathbf{t_2}\begin{pmatrix}-a_2\\-b_2\end{pmatrix}$$

Ainsi dans les deux images, pour chaque point $\mathbf{u'}$ dans le repère image, on peut écrire ses coordonnées dans le repère centré sur le cercle par $\mathbf{u}_{2D} = \mathbf{u'} + \mathbf{t}$.

3.2.3 Extraction des points d'intérêt

Pour pouvoir effectuer un bon calibrage automatique, il est indispensable d'avoir un grand nombre de points d'intérêts et qu'ils soient détectés et appariés avec précision. Pour des images omnidirectionnelles, il est naturel de choisir des détecteurs de points d'intérêt robustes aux changements d'échelles et aux rotations. Certains auteurs utilisent donc des algorithmes robustes, initialement créés pour des images planes, obtenues avec des caméras perspectives classiques. Toutefois, on remarque que ces techniques manquent de précision, car elles ne tiennent pas compte de la géométrie particulière de ces images. La plupart du temps, les auteurs proposent alors de rectifier les images omnidirectionnelles en images planes, afin de les traiter comme des images provenant de caméras perspectives. L'inconvénient de cette pratique est qu'il faut calibrer avec précision le capteur pour pouvoir en rectifier au mieux les images. Plus important, il est nécessaire d'interpoler les valeurs des pixels génèrant ainsi des erreurs dans l'image reconstruite, tout particulièrement dans les régions des pôles. Récemment, certains auteurs ont offert une alternative en développant des algorithmes de détection de points spécifiques adaptés aux images omnidirectionnelles.

Dans la suite de cette section, nous débutons par un état de l'art assez complet des méthodes de détection de points d'intérêt existantes. Nous présentons également une évaluation de ces méthodes sur un ensemble d'images fisheye de synthèse afin de dégager la combinaison détecteur-descripteur la plus adaptée pour assurer un bon appariement.

3.2.3.1 Etat de l'art

Il existe de nombreux algorithmes de détection de points ou d'éléments d'intérêt dans les images. Selon leur nature et le dessein pour lequel ils ont été développés, ils seront plus ou moins efficaces et robustes à certaines conditions. Les types de robustesse les plus courants dans la littérature sont ([4] et sa banque d'images de test) :

- la robustesse au flou (ou plus exactement au manque de netteté qui peut avoir plusieurs causes comme par exemple le flare, une optique de mauvaise qualité, etc),
- la robustesse aux changements de point de vue,
- la robustesse aux changements d'échelle (ou zoom dans le cas de changements de focale),
- la robustesse aux rotations (orientations),
- la robustesse aux changements de luminosité (clair/sombre, variations de la couleur de l'éclairage environnant, ombres, etc),
- ou encore la robustesse à la compression des enregistrements numériques des images avec pertes, et en particulier le format JPEG, ces méthodes d'enregistrement pouvant créer des artefacts.

Pour l'application de stéréovision omnidirectionnelle, les algorithmes employés doivent impérativement être robustes aux :

- changements de rotations/orientations, car les caméras embarquées suivent le mouvement du véhicule et qu'il rencontre forcément des tournants au cours de son parcours ;
- aux changements de point de vue, car le véhicule se déplaçant, la scène sera observée de différents points de vues ;
- et aux changements d'échelle, car un élément n'aura pas la même taille dans l'image en fonction du point de vue, mais aussi car les déformations optiques des objectifs fisheye font que plus les éléments sont situés en périphérie, plus ils sont déformés et grands dans l'image.

Pour toutes ces raisons, le tableau 1 est proposé, récapitulant de nombreux algorithmes de détection de points d'intérêt disponibles, avec l'information sur les notions de robustesse nécessaires aux applications de vision omnidirectionnelle. Un blob désigne une région élémentaire.

Les avantages et inconvénients des algorithmes présentés sont décrits brièvement ci-après :

1. méthode de Harris [5], détecteur de coins très rapide (utiliser un Harris-Laplace/Harris-Affine ou un Hessian-Laplace/Harris-Hessian pour ajouter la robustesse aux changements d'échelle, ces évolutions sont en fait des détecteurs de blobs) ; de nombreux progrès ont été faits depuis dans la détection de points robustes.
2. méthode de Shi-Tomasi [6], (ou Kanade-Tomasi) détecteur de coins très rapide.
3. méthode de Shi-Tomasi par dessus un Harris [6], amélioration du Harris avec la méthode de Shi-Tomasi, plus de stabilité pour le suivi.
4. SIFT [7] (*Scale Invariant Feature Transform*), assez sensible au bruit, mauvais pour de la végétation sinon globalement bon et robuste, utilisation non libre de droits.
5. FAST [8] (*Features from Accelerated Segment Test*), très rapide (utilisation possible en vidéo temps réel), peu robuste au bruit.
6. SURF [9] (*Speeded Up Robust Features*), souvent décrit comme un SIFT en plus rapide (excellent rapport qualité/coût), mauvais pour de la végétation, utilisation non libre de droits.
7. CenSurE [10] (*Center Surround Extremas*), rapide, l'auteur le teste avec une variante du descripteur SURF plus rapide, la robustesse à la rotation est approximative mais normalement meilleure qu'avec SURF, l'auteur montre que le CenSurE est meilleur que SIFT, SURF et Harris pour des applications d'odométrie visuelle grâce à la précision de la localisation des points.
8. MSER [11] (*Maximally Stable Extremal Regions*), robuste aux changements de perspective, le fait de détecter des points au centre des régions peut le rendre légèrement robuste aux changements d'échelle et rotation ; Nistér propose en fait une implémentation de l'algorithme linéaire en temps de calculs.
9. ORB [12] (*Oriented FAST and Rotated BRIEF*), rapide, évolution du FAST en pyramide pour ajouter de la robustesse aux changements d'échelle au détecteur, robuste au bruit.
10. BRISK [13] (*Binary Robust Invariant Scalable Keypoints*), assez rapide, détecteur basé sur AGAST (un FAST déjà amélioré), rendu robuste aux changements d'échelle, plus sensible au bruit que des détecteurs de blobs, bien pour de la végétation.

3.2.4 Mise à l'échelle unitaire

Appliquer une mise à l'échelle sur toutes les coordonnées des points par rapport au repère centré sur le cercle, de sorte à avoir le rayon maximal proche de 1. Travailler avec des grandeurs normalisées permet une meilleure stabilité numérique (voir [1] et [14]). On fait donc pour chaque point : $\mathbf{u}_{2D} = \frac{\mathbf{u}_{2D}}{r}$.

3.2.5 RANSAC (*RANdom SAmple Consensus*)

Appliquer le RANSAC. Pour cela, effectuer les actions suivantes un grand nombre de fois pour augmenter les chances d'obtenir de bons paramètres de calibrage (par exemple 200 itérations donne d'assez bons résultats) :

- Choisir au hasard 9 paires de points correspondants différentes.
- Estimer les paramètres (a) et la matrice fondamentale ($\mathbf{F}$) possibles à partir des points choisis et de paramètres initiaux correspondant à une valeur approximative connue des paramètres à déterminer (voir l'algorithme des 9 points section 3.2.6).
- Évaluer chaque autre paire de points avec ce modèle, garder de côté l'ensemble des paires de points qui s'ajustent bien au modèle (au-dessus d'un seuil d'erreur) en plus des 9 paires choisies au départ.

– Si le nombre de paires de points mises de côté est supérieure à 90 % de l'ensemble des paires de points, mesurer l'erreur moyenne (voir section 3.2.6.2 pour la mesure d'erreur) du modèle sur tous ces points et le conserver si cette erreur est la plus faible trouvée à cette étape.
Par ailleurs, dans le cas où on conserve le modèle obtenu, reprendre aussi les paramètres obtenus à l'estimation comme paramètres initiaux pour l'algorithme des 9 points aux prochaines itérations.

À la fin de ces itérations, le RANSAC fournit le modèle ayant la plus petite erreur moyenne qui a été conservé.

3.2.6 Algorithme des 9 points : Estimation des paramètres et de la matrice fondamentale à partir de 9 correspondances

L'algorithme des 9 points est semblable à l'algorithme des 8 points utilisés pour des caméras pinhole, adapté au modèle sphérique (voir par exemple [14]). Cette estimation se base sur la résolution d'un PEP (*Polynomial Eigenvalue Problem*), problème aux valeurs propres. Pour cela, la création de matrices particulières adaptées au modèle choisi s'impose. Il s'agit de matrices $\mathbf{D_i}$. Ces matrices sont construites à partir d'estimations des projections des points en rayons 3D obtenues grâce à une approximation par développement en série de Taylor de la fonction de projection dont on ne connaît pas encore le paramètre a. Cette approximation se base sur un a_o donné. Pour le modèle de projection équidistante, on peut prendre $a_0 = \frac{1}{f} = \frac{\theta_{max}}{r_{max}} = \theta_{max}$ car les grandeurs ont été normalisées de telle sorte à obtenir $r = 1$ (voir section 3.2.4). θ_{max} est donné par le constructeur (c'est la moitié de l'angle de vision total indiqué).
Observation personnelle : le choix d'une bonne estimation de a_0 est important pour obtenir de bons résultats de calibrage. Un essai avec $a_0 = 1$ a été fait, pour des résultats moins concluants.
À la résolution du PEP à partir de ces matrices, on obtient une matrice $\mathbf{H}$ contenant des colonnes $\mathbf{f}$ avec les valeurs possibles pour $\mathbf{F}$ et un vecteur $\mathbf{a}$ avec les solutions possibles pour a. Un seul couple $(\mathbf{f}, a)$ est correct. On peut premièrement ne garder que les a réalistes, c'est-à-dire qui sont des réels purs (non complexes) positifs et non infini (on peut interpréter cette condition comme inférieur à un seuil). Normalement, il reste entre une et quatre solutions possibles qu'il faut évaluer pour garder la meilleure. Pour chaque a, il y a le $\mathbf{f}$ correspondant, dont les éléments permettent de reconstituer les $\mathbf{F}$ possibles. Il s'agit de mesurer la somme des erreurs angulaires des 9 paires de points utilisés pour l'estimation des paramètres, avec les différents paramètres estimés possibles (voir section 3.2.6.2). Les paramètres qui donnent l'erreur globale la plus faible sont les meilleurs et sont ceux que l'on conserve à l'issue de cette étape.

3.2.6.1 Contribution : modèle linéaire à un paramètre pour la projection à angle équisolide

Il s'agit ici de définir l'écriture des matrices $\mathbf{D_i}$ pour le modèle de projection à angle équisolide. Tout se passe de manière similaire à ce qui est fait dans [1] pour le modèle équidistant. La série de Taylor de degré 1 $\tilde{g}(r, a)$ de $g(r, a)$ pour a proche de a_0 est donnée par :

$$\tilde{g}(r, a) = g(r, a_0) + \frac{\partial g(r, a_0)}{\partial a}(a - a_0)$$

Rappelons que pour le modèle à angle équisolide :

$$g(r, a) = \frac{r}{\tan(2 \arcsin(a.r))}$$

On obtient donc :

$$\frac{\partial g(r, a_0)}{\partial a} = \frac{-2r^2[\tan^2(2 \arcsin(a_0 r)) + 1]}{\tan^2(2 \arcsin(a_0 r))\sqrt{1 - a_0^2 r^2}}$$

Comme l'a décrit Mičušìk [1], le vecteur

$$\tilde{\mathbf{p}} = \begin{pmatrix} \mathbf{u}_{2D} \\ \tilde{g}(r, a_0) \end{pmatrix} = \begin{pmatrix} \mathbf{u}_{2D} \\ g(r, a_0) - a_0 \frac{\partial g(r,a_0)}{\partial a} + a \frac{\partial g(r,a_0)}{\partial a} \end{pmatrix} = \begin{pmatrix} \mathbf{u}_{2D} \\ w \end{pmatrix} + a \begin{pmatrix} \mathbf{0} \\ s \end{pmatrix} = \mathbf{x} + a\mathbf{s}$$

Donc on a :

$$w = g(r, a_0) - a_0 \frac{\partial g(r, a_0)}{\partial a} \text{ et } s = \frac{\partial g(r, a_0)}{\partial a}$$

De la même manière que pour tous les modèles à un paramètre (voir [1]), on utilise l'équation $\tilde{\mathbf{p}}_2^T \mathbf{F} \tilde{\mathbf{p}}_1 = 0$ que l'on arrange pour le PEP :

$$(\mathbf{D_1} + a\mathbf{D_2} + a^2\mathbf{D_3})\mathbf{f} = \mathbf{0},$$

où les $\mathbf{D_i} \in \mathbb{R}^{9\times 9}$ sont composées ligne par ligne à partir des 9 paires de points correspondants utilisées comme suit :

$$\begin{aligned}
&\text{ligne } \mathbf{D_1} = [u_2u_1 \ u_2v_1 \ u_2w_1 \ v_2u_1 \ v_2v_1 \ v_2w_1 \ w_2u_1 \ w_2v_1 \ w_2w_1], \\
&\text{ligne } \mathbf{D_2} = [0 \ 0 \ u_2s_1 \ 0 \ 0 \ v_2s_1 \ s_2u_1 \ s_2v_1 \ w_2s_1 + s_2w_1], \\
&\text{ligne } \mathbf{D_3} = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ s_2s_1].
\end{aligned}$$

Pour ce modèle on prendra $a_0 = \frac{1}{2f} = \frac{\sin(\frac{\theta_{max}}{2})}{r_{max}} = \sin(\frac{\theta_{max}}{2})$ car on a fait en sorte d'avoir $r = 1$ (section 3.2.4).

3.2.6.2 Mesure de l'erreur angulaire pour une paire de points correspondants

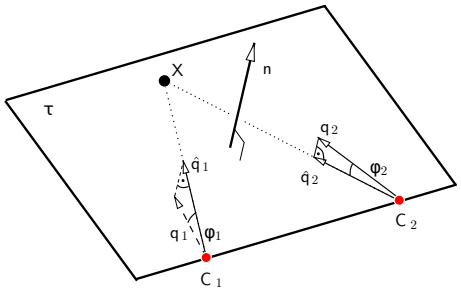


FIGURE 6 – Erreur angulaire dans un plan épipolaire τ. $\mathbf{X}$ est un point 3D de la scène, $\mathbf{n}$ un vecteur normal unitaire à τ, $\mathbf{C}_1$ et $\mathbf{C}_2$ les centres des caméras. $\mathbf{q}_1$ et $\mathbf{q}_2$ sont les rayons unitaires correspondants au point $\mathbf{X}$, et $\hat{\mathbf{q}}_1$ et $\hat{\mathbf{q}}_2$ leurs meilleures estimations à l'intérieur de τ. ϕ_1 et ϕ_2 sont les angles entre les rayons unitaires et leurs meilleures estimations [1].

La première étape est de créer les rayons 3D avec les paramètres estimés pour la projection et la matrice fondamentale, pour les points correspondants dans les deux images. La seconde étape est de calculer l'erreur angulaire selon la formulation donnée par Mičušìk [1], et basée sur les travaux d'Oliensis :

$$\epsilon(\mathbf{q}_1, \mathbf{q}_2, \mathbf{F}) = \frac{\mathbf{A}}{2} - \sqrt{\frac{\mathbf{A}^2}{4} - \mathbf{B}},$$

avec

$$\mathbf{A} = \mathbf{q}_1^T \mathbf{F}^T \mathbf{F} \mathbf{q}_1 + \mathbf{q}_2^T \mathbf{F} \mathbf{F}^T \mathbf{q}_2 \text{ et } \mathbf{B} = (\mathbf{q}_2^T \mathbf{F} \mathbf{q}_1)^2.$$

Généralement le nombre d'itérations du ransac est élevé et offre une première estimation de la matrice qui est ensuite affinée. Pour cela, nous utilisons l'algorithme de Levenberg-Marquardt pour minimiser l'erreur relié à l'appariement des inliers définis à la fin du Ransac en fonction des

paramètres de la matrice fondamentale. La matrice fondamentale permet de déterminer les deux épipoles et de déterminer les courbes épipolaires conjuguées le long desquelles le processus de mise encorrespondance est réalisé. La figure 7 représente la géométrie épipolaire de deux prises de vue et la construction des courbes épipolaires. La figure 8 illustre le résultat obtenu par le calibrage sur un couple stéréoscopique d'images fisheye. La figure 9 illustre le résultat obtenu par le calibrage sur un couple d'images fisheye acquises au cours du déplacement du mobile par une unique caméra. Les courbes épipolaires sont représentées en jaune dans les images. Elles se coupent en deux points qui sont les épipoles. Les points en vert dans les images sont les points d'intérêt ayant servi à l'estimation des paramètres de la matrice fondamentale. Connaissant la matrice fondamentale, il est alors possible de déterminer toutes les courbes épipolaires conjuguées sur chacune d'entre elles et de procéder à la production des nuages de points.

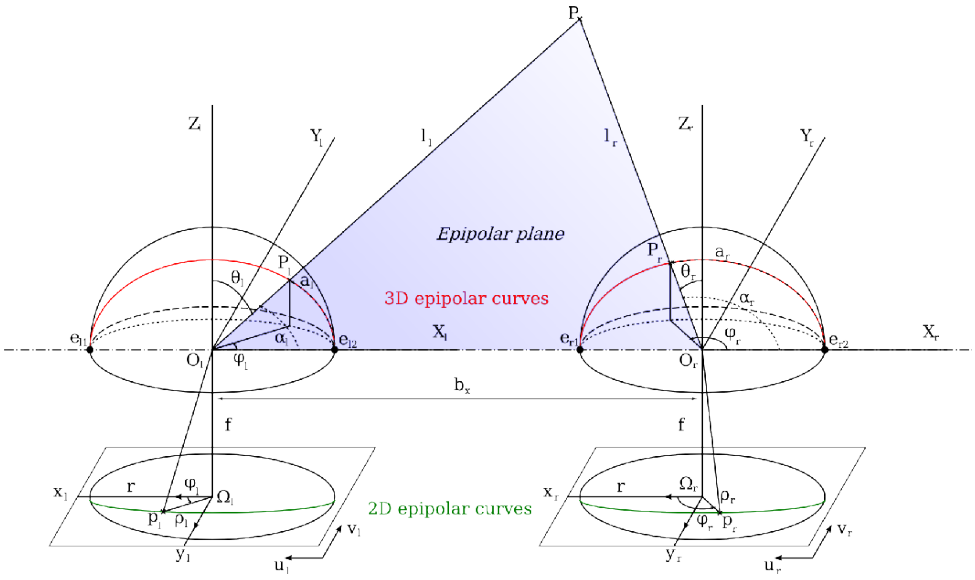


FIGURE 7 – Configuration épipolaire de prises de vue correspondantes à deux images fisheye.

3.3 Evaluation des détecteurs, des descripteurs et des algorithmes d'appariement

Il est important de retrouver les mêmes points d'intéret dans les images d'un couple afin de garantir l'existance de bons appariements. Nous avons fait une étude assez complète sur la robustesse des points d'intérêt à un changement de point de vue. Nous l'avons complété en étudiant avec quels descripteurs de la littérature nous devions les associer pour obtenir les meilleurs résultats. La particularité de notre système étant la projection sphérique des objectifs, nous avons également évalué des détecteurs définis directement dans cet espace.

Il existe de nombreux travaux visant à comparer les différents détecteurs de features et descripteurs : [4, 15, 16, 17] Ces travaux ont été faits pour des images obtenues avec des caméras perspective classiques, on ne peut pas considérer ces résultats valides pour les images omnidirectionnelles. Les bases d'images d'évaluations mise à disposition par la communauté sont des images planes. Nous nous plaçons dans le cadre de la projection sphériques, nous avons donc créé notre propre ensemble d'images. Le temps de calcul étant directement lié à l'implantation plus ou moins optimisée de l'algorithme considéré, nous ne nous attachons pas à son évaluation. Pour créer nos images avec vérité, il n'est pas possible, avec nos moyens actuels, de générer une vérité pour des images fisheye réelle. Nous avons donc pris l'option de synthétiser des images par

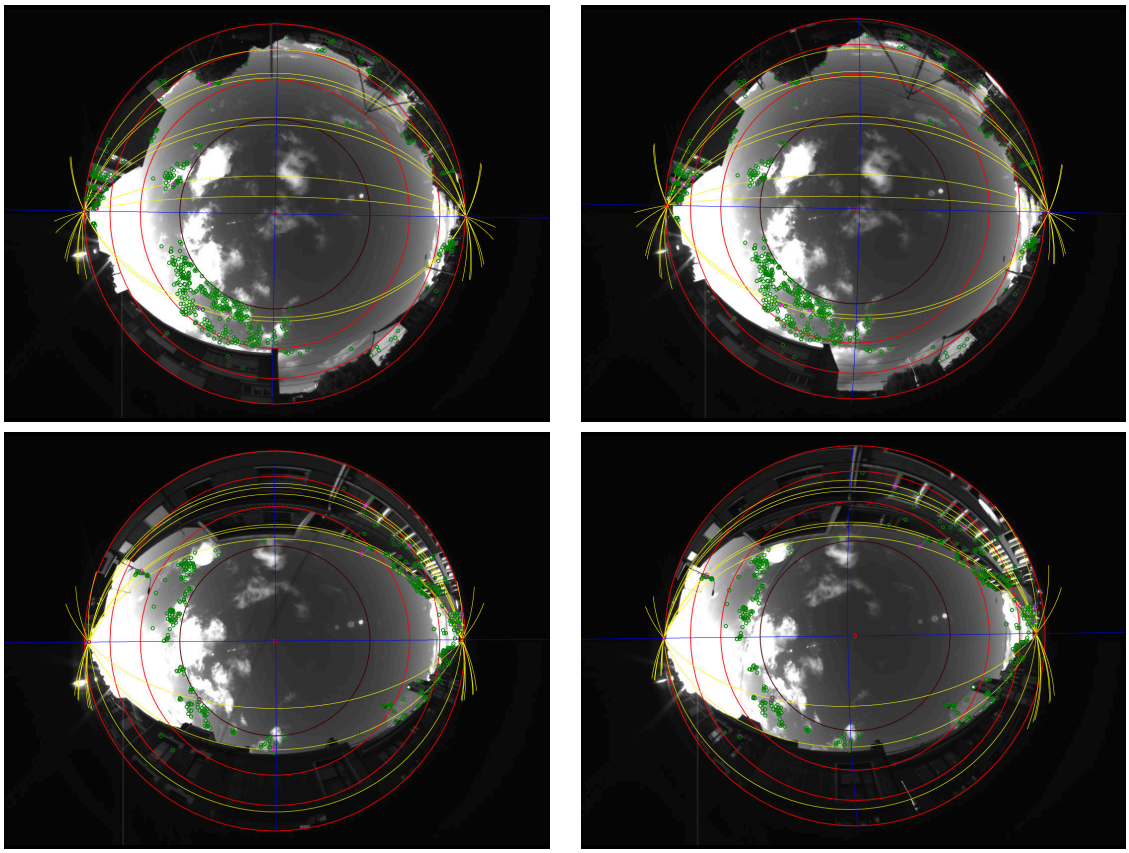


FIGURE 8 – Résultat de calibrage sur un couple d'images stéréoscopiques.

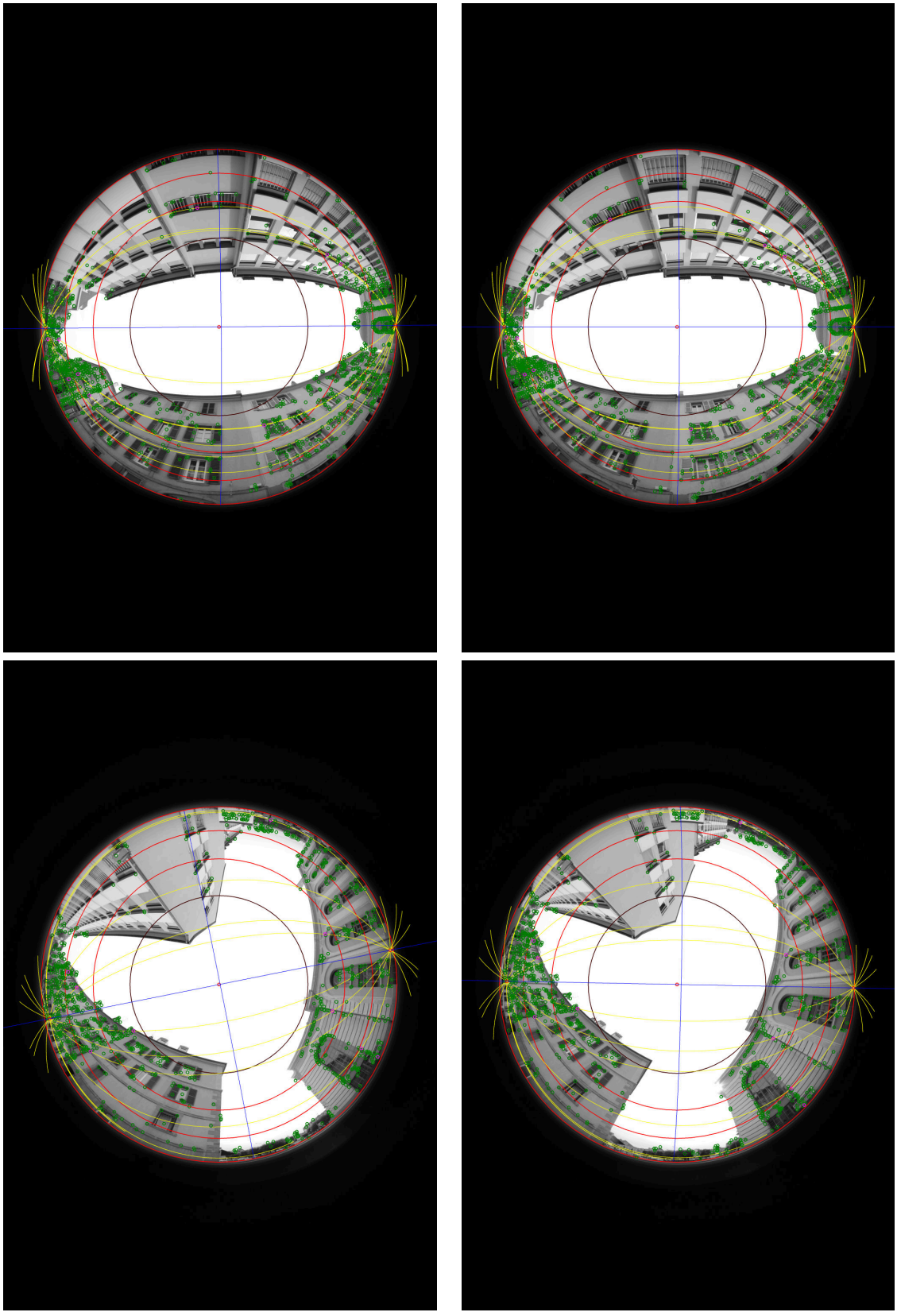


FIGURE 9 – Résultat de calibrage sur un couple d'images mono-caméra.

ordinateur et de simuler au mieux les différents défauts qui existent dans la réalité. Toutefois, dans un deuxième temps, nous avons validé les résultats en évaluant la stabilité des algorithmes en les appliquant sur un ensemble d'images réelles.

La séquence d'images fisheye utilisée a été calculée en images de synthèses. La seule chose que l'on ne peut pas simuler est l'objectif réel, qui ne correspond pas à un modèle central ni exactement à une fonction de projection linéaire à un paramètre. Les images correspondent au modèle de projection à angle équisolide. Nous avons produit 3 types d'images différents (figure 10) :

1. image de translation sans flou de bougé, la première image de démarrage du véhicule (paire n° 0),
2. image de mouvement en translation avec flou de bougé, comme la grande majorité de ces images (paire n° 40),
3. image de rotation et translation, avec flou de bougé, dans un virage, le cas le plus difficile (paire n° 24).



(a) Image 0. (b) Image 40. (c) Image 24.

FIGURE 10 – Images représentatives des différentes conditions rencontrées pour le déplacement du véhicule.

En plus des images, nous avons extrait les distances de chaque point dans les images. Avec la connaissance du modèle de projection précis et des distances de chaque point, il est possible de calculer leur projeté dans le monde 3D. Avec l'aide de la connaissance des positions relatives des caméras virtuelles entre elles, il est possible de repérer les points projetés de l'une par rapport à l'autre. Ainsi, on peut trouver les points qui se superposent, et qui sont en fait les contre-projetés des représentations du même point dans les deux images. En appliquant cette méthode sur les deux ensembles de points d'intérêt détectés, on est capable d'évaluer quels points doivent être appariés. Les évaluations sont faites pour différents rayons de tolérance dans la distance qui sépare les points des deux images en 3D. Il s'agit en fait de considérer les points détectés dans les deux images, de les projeter dans le même repère 3D, et de regarder dans un rayon d'action donné si pour chaque point de l'image de référence on trouve des points de l'image de recherche (on peut considérer cela comme une sphère de recherche autour du point de référence). En cas de choix multiples, les paires de points les plus proches sont conservées, et le choix suivant dans l'ordre de proximité s'impose comme le meilleur candidat pour la correspondance vérité. Ainsi, un rayon de recherche trop élevé peut aboutir à de fausses vérités. Pour cela on travaille avec les rayons de recherche suivants :
5mm, 1cm, 2cm, 3cm, 4cm, 5cm, 6cm, 7cm, 8cm, 9cm, 10cm, et 15cm.

Les grandeurs mesurées durant l'évaluation sont les suivantes :

– le nombre de points détectés ;

– le nombre d'appariement : un appariement est considéré comme valide au regard de la proximité dans la scène 3D des deux points appariés ;
– le nombre de bons appariements.

Ces grandeurs permettent de calculer les taux suivants.

– la répétabilité tirée de [18] ;
– le taux de bons appariements.

La répétabilité est définie pour une tolérance 3D particulière :

$$\text{Répétabilité}_{|tolrance} = \frac{\sharp\text{Points correspondants}_{tolrance}}{\sharp\text{Points détectés}} \tag{28}$$

où ♯Points correspondants est le nombre de paires de points correspondants parmi les points détectés, et ♯Points détectés est le nombre minimal de points détectés entre les deux images.

Le taux de bons appariements (*matching score*) pour une tolérance donnée est défini par :

$$\text{Taux de bons appariements}_{|tolrance} = \frac{\sharp\text{Correspondants corrects}_{tolrance}}{\sharp\text{Points détectés}} \tag{29}$$

où ♯Correspondants corrects est le nombre de paires de points correspondants correctes, et ♯Points détectés est le nombre minimal de points détectés entre les deux images.

Le tableau 1 regroupe, en gras, les détecteurs utilisés dans l'évaluation.

TABLE 1 – Liste des détecteurs étudiés. Les détecteurs marqués en gras sont ceux qui sont donnés assez robustes aux changements d'échelle et d'orientation, et qui ont donc été retenus pour les évaluations sur les images fisheye.

Algorithme	Type d'éléments	Robuste échelles	Robuste orientations
Harris	coins	non	non
Shi-Tomasi	coins	non	non
Shi-Tomasi sur Harris	coins	non	non
SIFT	blobs	oui	oui
FAST	coins	non	non
SURF	blobs	oui	moyen
CenSurE	blobs[1]	oui	moyen+
MSER	régions	faible	faible
ORB	coins	oui	oui
BRISK	coins	oui	oui

[1] : Le CenSurE détecte des blobs, mais dont le but est d'être plus précis que des coins.

Les descripteurs sont un ensemble d'informations, ou de critères, de traits distinctifs, associés aux points détectés. Ils peuvent être calculés par le même algorithme de détection, ou être créés de manière indépendante. Le but est de comparer les informations décrivant les points détectés afin de trouver les points les plus semblables entre eux. On peut effectuer une mesure de distance entre les points, avec une plus grande probabilité pour que les points d'images différentes soient des représentations du même dans la réalité pour une distance faible entre leurs descripteurs. L'appariement des points entre les images sans à-priori ne peut se faire sans les descripteurs.

Les descripteurs de la littérature analysés sont les suivants :

1. SIFT [7] (*Scale Invariant Feature Transform*), robuste aux changements de perspective et de point de vue, nombre de correspondants possibles parfois sur-estimé (trop nombreux), une référence, utilisation non libre de droits.

2. SURF [9] (*Speeded Up Robust Features*), souvent décrit comme un SIFT en plus rapide (excellent rapport qualité/coût), orientations connues approximativement, il existe aussi un *upright* SURF (U-SURF) non robuste aux rotations, utilisation non libre de droits.
3. BRIEF [19] (*Binary Robust Independent Elementary Features*), très rapide, robuste aux changements de perspective, l'auteur l'a testé avec des points SURF et CenSurE, ses résultats avec CenSurE sont bien meilleurs (il dit que BRIEF est plus adapté à des détecteurs rapides comme CenSurE ou FAST).
4. ORB [12] (*Oriented FAST and Rotated BRIEF*), rapide, évolution du BRIEF avec rotation pour le descripteur, robuste au bruit.
5. BRISK [13] (*Binary Robust Invariant Scalable Keypoints*), assez rapide, reprend le principe de base du descripteur DAISY pour l'échantillonnage des infos servant à calculer le descripteur.
6. FREAK [20] (*Fast Retina Keypoint*), descripteur visant à approcher le fonctionnement du système visuel humain, reprend l'idée de la répartition du DAISY adapté au modèle de la rétine, rapide et robuste, l'auteur l'a utilisé avec le détecteur BRISK.

TABLE 2 – Liste des descripteurs étudiés. Les descripteurs marqués en gras sont ceux qui sont donnés assez robustes aux changements d'échelle et d'orientation, et qui ont donc été retenus pour les évaluations sur les images fisheye.

Algorithme	Robuste échelles	Robuste orientations
SIFT	oui	oui
SURF	oui	moyen
BRIEF	non	non
ORB	oui	oui
BRISK	oui	oui
FREAK	oui	oui

De la même manière que pour les détecteurs, la vision omnidirectionnelle nécessite des descripteurs robustes aux changements d'échelles et aux rotations. Le tableau 2 résume les descripteurs disponibles et leurs critères de robustesse utiles pour l'application visée. Ainsi, nous avons choisi d'effectuer des tests pour les algorithmes de description SIFT, ORB, BRISK et FREAK.
De nombreux tableaux et de ombreuses courbes ont été produites identiques à ce que nous présentons aux figures 12.

L'évaluation des détecteurs montre que le SIFT fournit beaucoup plus de points d'intérêts que le CenSurE et que les répétabilités sont comparables. Il apparaît que les performances du CenSurE sont moins stables que celles du SIFT notamment lorsque la complexité des images augmente. Le CenSurE est par exemple inutilisable avec l'image 24, le cas le plus difficile avec rotation et flou de mouvement. Le nombre de points détectés est équivalent que le SIFT soit appliqué sur des images avec ou sans rectification épipolaire. Toutefois, la répétabilité est meilleure si le SIFT est appliqué sur des images sans rectification. Dans le cas difficile de l'image 24, le SIFT avec rectification épipolaire donne la meilleure répétabilité, mais dans ce cas le nombre de points détectés est plus faible.

En conséquence pour l'application décrite et les situtations étudiées, on peut retenir que le détecteur globalement le plus performant pour un bon compromis nombre de points/répétabilité est le détecteur SIFT.

Nous avons évalué la combinaison détecter - descripteur - algorithme de mise en correspondance.

Les 3 méthodes d'appariement comparées sont :

1. RatioTest (méthode d'appariement des points clés proposé par Lowe avec le SIFT dans [7]) basée sur la comparaison des plus proches voisins pour déterminer si le meilleur se distingue vraiment des autres candidats,
2. BruteForce (méthode exhaustive en testant toutes les possibilités de correspondance pour chaque point),
3. et FLANN (Fast Library for Approximate Nearest Neighbors, [21]), méthode rapide mais aboutissant à moins de bons points correspondants.

Pour comparer leur distance, les descripteurs SIFT et SURF emploient la norme euclidienne. Les autres descripteurs présentés utilisent la distance de Hamming. La correspondance FLANN ne fonctionne qu'avec les distances euclidiennes, et donc uniquement avec les descripteurs SIFT et SURF.

De même que pour l'évaluation des détecteurs seuls, seuls les détecteurs CenSurE et SIFT montrent les meilleurs comportements quels que soient leur réglage. La méthode d'apapriement des features la meilleure est le BruteForce. L'utilisation de la rectifiaction polaire ne semble pas augmenter la qualité des résultats. Là encore, le CenSurE aboutit à un petit nombre d'appriements et donne de trés mauvais résultats dans le cas des images complexes(image 24). En comparant les performances du détecteur SIFT pour différents réglages, le meilleurs compromis est d'utiliser le détecteur SIFT sans rectification polaire qui présente trop peu de bons correspondants avec la rectification. L'association SIFT - SIFT et SIFT - FREAK ont des comportements similaires sauf pour l'image 24 où le matching score est supérieur pour un nombre de points quasi identique.

En conclusion, le détecteur SIFT associé au descripteur SIFT ou FREAK apportent les meilleurs résultats et les meilleurs compromis. Les détecteurs sphériques n'apportaient que très peu d'amélioration et augmentaient de temps de traitement de manière très importante.

TABLE 3 – Nombre de correspondants image 0, valeurs numériques.

TolÃ©rance en m	CenSurE nopol 16 10 7	CenSurE nopol 32 20 5	CenSurE nopol 8 10 3	SIFT nopol 0.02 20.0 1.0	SIFT nopol 0.02 20.0 1.6	SIFT nopol 0.04 5.0 1.6	SIFT nopol 0.04 5.0 3.0	SIFT nopol 0.08 5.0 3.0	SIFT pol 0.02 20.0 1.0	SIFT pol 0.02 20.0 1.6	SIFT pol 0.02 5.0 3.0	SIFT pol 0.04 5.0 3.0
0.005	12	37	8	1237	**1239**	329	219	59	847	903	478	210
0.01	47	113	30	**3559**	3255	827	540	156	2752	2729	1324	661
0.02	145	306	110	**6732**	5838	1386	911	274	6090	5552	2173	1132
0.03	221	475	175	**8802**	7384	1672	1081	326	8475	7510	2603	1370
0.04	273	605	216	10205	8419	1841	1174	360	**10238**	8912	2857	1516
0.05	313	693	237	11150	9033	1932	1233	373	**11616**	9955	3014	1611
0.06	329	751	242	11960	9570	1986	1271	385	**12707**	10777	3122	1672
0.07	342	789	256	12559	10003	2024	1297	391	**13596**	11437	3194	1704
0.08	354	824	261	13036	10339	2060	1324	398	**14325**	11949	3249	1729
0.09	363	850	265	13414	10608	2088	1339	400	**14892**	12358	3310	1755
0.1	369	866	267	13762	10862	2109	1352	403	**15422**	12685	3362	1775
0.15	382	913	273	15173	11885	2196	1411	410	**17384**	14007	3591	1840

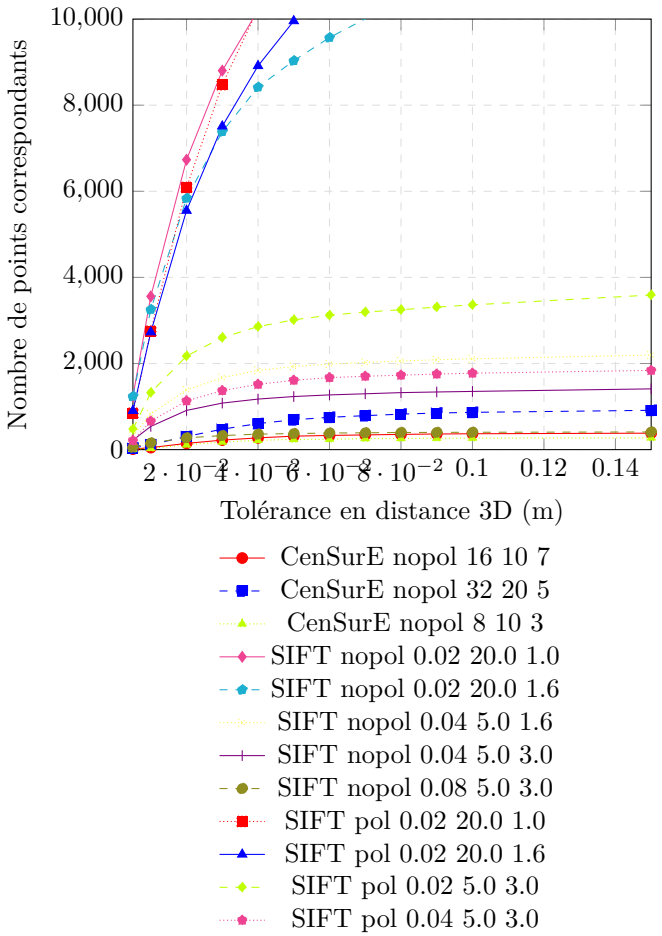


FIGURE 11 – Nombre de correspondants image 0.

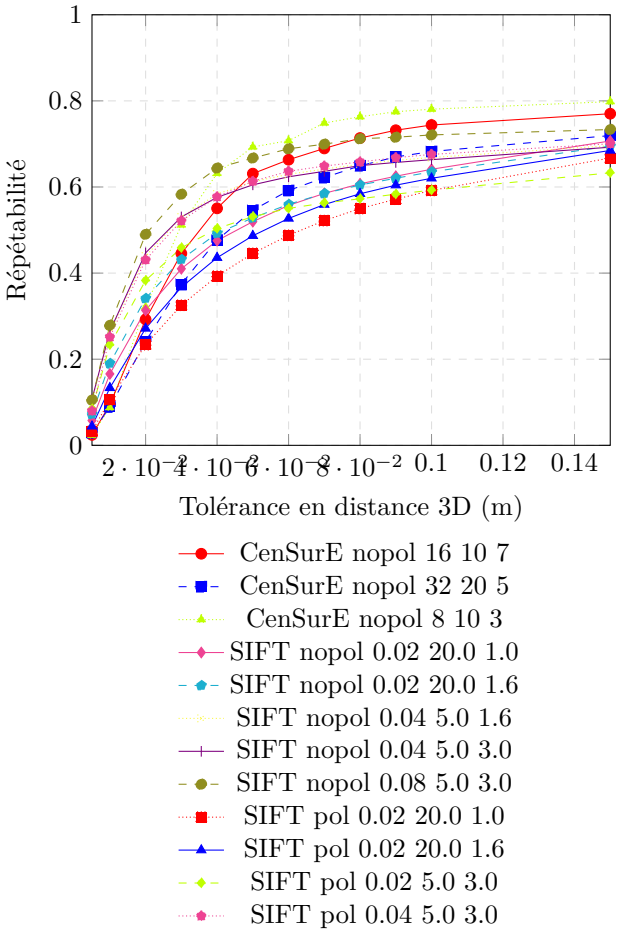


FIGURE 12 – Répétabilité image 0.

TABLE 4 – Répétabilité image 0, valeurs numériques.

TolÃrance en m	CenSurE nopol 16 10 7	CenSurE nopol 32 20 5	CenSurE nopol 8 10 3	SIFT nopol 0.02 20.0 1.0	SIFT nopol 0.02 20.0 1.6	SIFT nopol 0.04 5.0 1.6	SIFT nopol 0.04 5.0 3.0	SIFT nopol 0.08 5.0 3.0	SIFT pol 0.02 20.0 1.0	SIFT pol 0.02 20.0 1.6	SIFT pol 0.02 5.0 3.0	SIFT pol 0.04 5.0 3.0
0.005	0.02	0.03	0.02	0.06	0.07	0.10	**0.11**	0.11	0.03	0.04	0.08	0.08
0.01	0.09	0.09	0.09	0.17	0.19	0.26	0.26	**0.28**	0.11	0.13	0.23	0.25
0.02	0.29	0.24	0.32	0.31	0.34	0.44	0.45	**0.49**	0.23	0.27	0.38	0.43
0.03	0.45	0.37	0.51	0.41	0.43	0.53	0.53	**0.58**	0.33	0.37	0.46	0.52
0.04	0.55	0.48	0.63	0.48	0.49	0.58	0.58	**0.64**	0.39	0.44	0.50	0.58
0.05	0.63	0.55	**0.69**	0.52	0.53	0.61	0.61	0.67	0.45	0.49	0.53	0.61
0.06	0.66	0.59	**0.71**	0.56	0.56	0.63	0.62	0.69	0.49	0.53	0.55	0.64
0.07	0.69	0.62	**0.75**	0.59	0.58	0.64	0.64	0.70	0.52	0.56	0.56	0.65
0.08	0.71	0.65	**0.76**	0.61	0.60	0.65	0.65	0.71	0.55	0.58	0.57	0.66
0.09	0.73	0.67	**0.77**	0.62	0.62	0.66	0.66	0.72	0.57	0.60	0.58	0.67
0.1	0.74	0.68	**0.78**	0.64	0.64	0.67	0.66	0.72	0.59	0.62	0.59	0.68
0.15	0.77	0.72	**0.80**	0.71	0.70	0.70	0.69	0.73	0.67	0.68	0.63	0.70

Pour valider les résultats obtenus avec les images simulées, nous avons étendu l'évaluation à des images réelles en mesurant la stabilité du calibrage au cours d'un déplacement. L'évaluation est réalisée sur grand nombre d'itérations, l'aglorithme utilisé pour le calibrage étant fondé sur un RANSAC, probabiliste et non déterministe. Nous avons exécuté 1000 itérations du calibrage pour une triplet détecteur - descripteur - algorithme d'appariement. Bien entendu, n'avons retenu les résultats que lorsque le nombre de points appariés est élevé i.e. pour les algorithmes ayant eu les meilleurs résultats dans l'évaluation précédente.

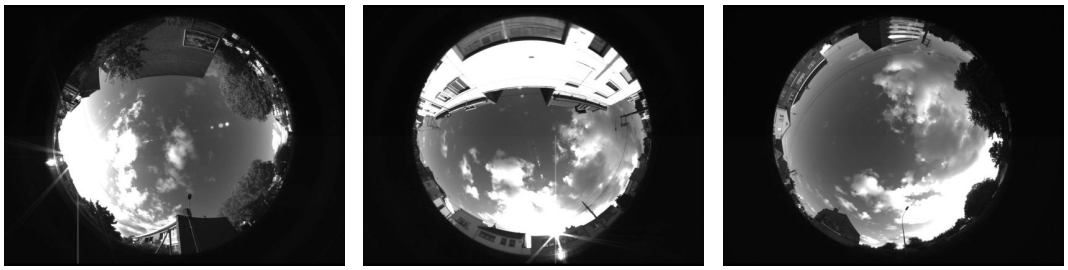
Voici trois images sélectionnées (figure 13) afin d'illustrer les difficultés évaluées lors de l'évaluation :

1. image de translation sans flou de bougé (paire n° 7100),
2. image de mouvement en translation avec flou de bougé (paire n° 10627),
3. image de rotation et translation, avec flou de bougé, dans un virage, le cas le plus difficile (paire n° 11048).

Pour les images réelles, nous proposons également d'employer les critères suivants pour avoir des mesures similaires avec celles utilisées pour les images de simulation :

- Le nombre de points mis en correspondance et le pourcentage par rapport au nombre de points détectés.
- Le nombre moyen d'inliers (car probabiliste) et,
  - le pourcentage par rapport au nombre de points détectés (taux le plus comparable au matching score car la probabilité que les inliers soient justes est la plus forte).
  - le pourcentage par rapport au nombre de points appariés.

Dans le cas réel, il faut bien garder à l'esprit que l'on ne sait pas si les points détectés comme correspondants sont réellement correspondants. Donc nous ne parlons pas de répétabilité, ni de matching score.



(a) Image 7100. (b) Image 10627. (c) Image 11048.

FIGURE 13 – Images représentatives des différentes conditions rencontrées pour le déplacement du véhicule. Images réelles.

Avec le détecteur CenSurE, on remarque que les résultats de stabilité avec le matching FLANN sont parfois mauvais. Autrement, les stabilités avec les différents descripteurs et algorithmes de matching sont globalement homogènes. On voit que généralement, la stabilité avec le détecteur CenSurE est légèrement inférieure à ce que l'on obtient à partir du détecteur SIFT. Pour le détecteur SIFT, l'utilisation du descripteur FREAK aboutit à une mauvaise stabilité de tous les paramètres car l'écart type est très élevé comparativement aux résultats avec le descripteur SIFT. L'algorithme BruteForce mène aux résultats les plus stables.

La conclusion générale nous indique que, le meilleur choix se porte sur le détecteur SIFT avec le descripteur SIFT et la mise en correspondance par BruteForce. Les nouvaux algorithmes de détection adaptés à la projection sphérique ont été également testés. Toutefois, ils apportent beaucoup de complexité compte tenu du faible gain en performance.

4 Production du nuage de points

Dans la section précédente nous avons présenté la méthode de calibrage automatique mise en place afin de déterminer la matrice fondamentale reliant deux images produites de manière stéréoscopique ou en utilisant une seule caméra en mouvement dans une scène. Grâce à cette matrice, il est possible de déterminer les courbes épipolaires dans les deux images et de procéder à la mise en correspondance des pixels de chaque image le long de ces courbes.

Dans la suite de cette section, dans un prmeier temps, nous décrivons la géométrie épipolaire dans le cas d'images fisheye qui nous permet de bien comprendre les ensembles de pixels à mettre en correspondance dans les deux images. Nous poursuivons par une description de l'algorithme de mse en correspondance par programmation dynamique que nous avons mise en oeuvre. Nous terminons par une description des algorithmes de fusion de cartes et de localisation de l'empreinte des bâtiments que nous avons implanté. Nous présentons quelques résultats que nous avons exploités dans le cadre de la tâche 3 sur l'amélioration de la localisation par satellites.

4.1 Géométrie épipolaire fisheye

D'après [22], le principe de la géométrie épipolaire tel qu'il est connu pour le modèle sténopé reste vrai pour le modèle de projection sphérique. En revanche dans le cas sphérique, les lignes épipolaires sont transformées en courbes épipolaires 3D sur la surface de la sphère, c'est-à-dire qu'il s'agit d'arcs de cercle. Comme montré dans les figures 14, la projection sphérique présente deux épipôles e_1 et e_2, au lieu de un seul et unique pour une caméra respectant le modèle sténopé. De la même manière, une étape de calibrage peut permettre de déterminer les courbes épipolaires conjuguées et de rectifier les images.

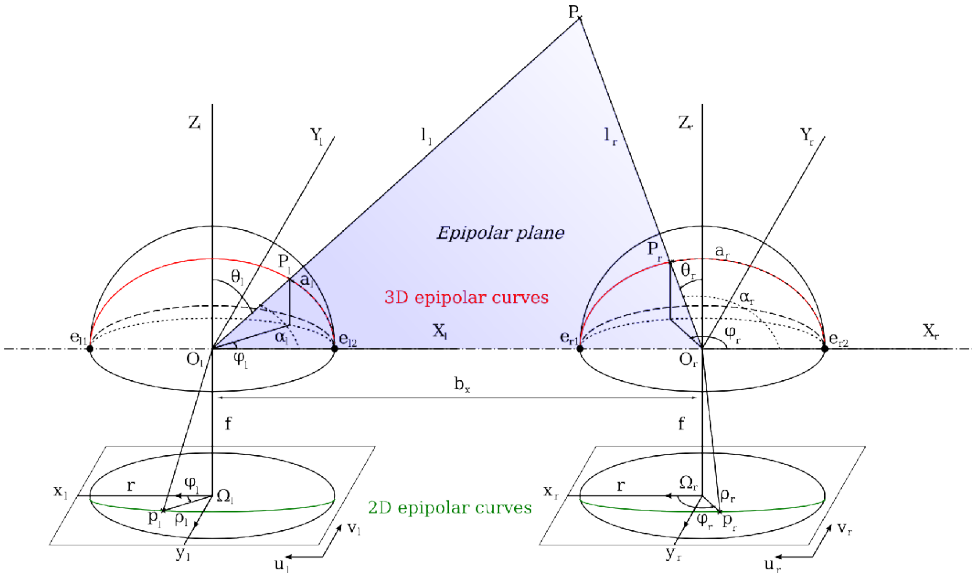


FIGURE 14 – Capteur stéréo fisheye calibré. Le point P de la scène est projeté sur les sphères gauche et droite respectivement en P_l et P_r, et est vu comme p_l et p_r dans les images captées par les deux caméras.

Dans la configuration décrite figure 14 et reprise de la section précédente, les points e_{l1}, O_l, e_{l2}, e_{r1}, O_r et e_{r2} appartiennent à la même ligne formant l'axe épipolaire, et les courbes épipolaires sont les intersections du plan (O_l, P, O_r) avec chaque objectif fisheye (plus exactement des sphères les représentant). Les deux courbes épipolaires sont projetées sur le capteur des caméras suivant l'un des modèles de projection considéré (i.e. modèle de projection equisolide).

Ainsi, il est possible de détecter les courbes épipolaires conjuguées, et de les utiliser pour mettre en correspondance chacun des points leur appartenant. Nous proposons de parcourir chaque paire de courbes épipolaires pour apparier les points entre les deux images fisheye.

4.1.1 Description et obtention des courbes épipolaires fisheye

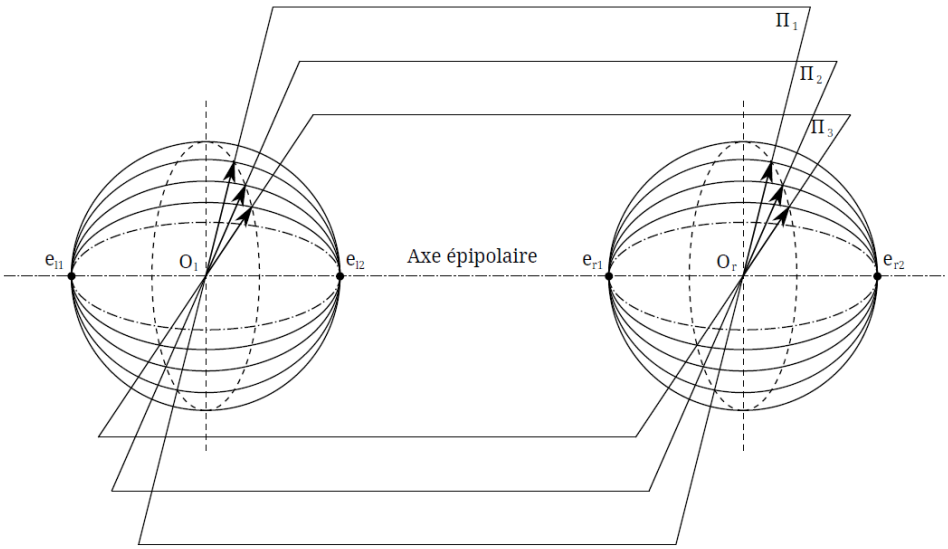


FIGURE 15 – Les courbes épipolaires à l'intérieur du même plan épipolaire (plans Π_i) peuvent être décrites par le même rayon le long de l'arc de scan perpendiculaire à l'axe épipolaire et situé exactement au milieu entre les deux épipôles.

Le but est de connaître les courbes épipolaires afin de pouvoir appliquer des algorithmes de mise en correspondance stéréoscopique basés sur la contrainte épipolaire. Pour des images fish-eye, une courbe épipolaire est un grand cercle de la sphère unitaire, passant par les épipôles et par le point de référence. Pour pouvoir les déterminer, nous avons donc besoin des deux épipôles de chaque image.

Pour obtenir les courbes épipolaires conjuguées, il faut considérer les plans épipolaires tels qu'ils sont décrits dans l'illustration 15. Un plan épipolaire passe par l'axe épipolaire et croise les sphères unitaires en formant de grands cercles : les cercles épipolaires. Le moyen le plus immédiat pour détecter les cercles épipolaires conjugués est de choisir des rayons correspondants issus des deux sphères, c'est-à-dire les rayons qui sont à l'intérieur du même plan épipolaire. Ces rayons peuvent être obtenus en parcourant avec le même angle les deux cercles de scan que l'on peut définir comme l'équateur entre les deux pôles épipolaires. Un arc de scan est projeté en une courbe sur le plan image.

La méthode permettant d'obtenir le cercle de scan nécessite de travailler dans les repères épipolaires, tels que décrits dans les figures 16 et 17. L'important est de bien distinguer le repère épipolaire du repère optique. Le repère optique est le repère 3D directement obtenu à partir de la projection fisheye de l'image enregistrée et il peut être décrit comme le repère de l'objectif. Le repère épipolaire est le repère adapté à la géométrie épipolaire, c'est-à-dire que les repères épipolaires des deux caméras doivent être les mêmes avec uniquement une translation les séparant le long de l'axe X_{epip}, suivant lui-même l'axe épipolaire. Comme on le voit dans la figure 17, déterminer ce repère nécessite la connaissance de l'axe X_{epip}, donné par l'axe épipolaire

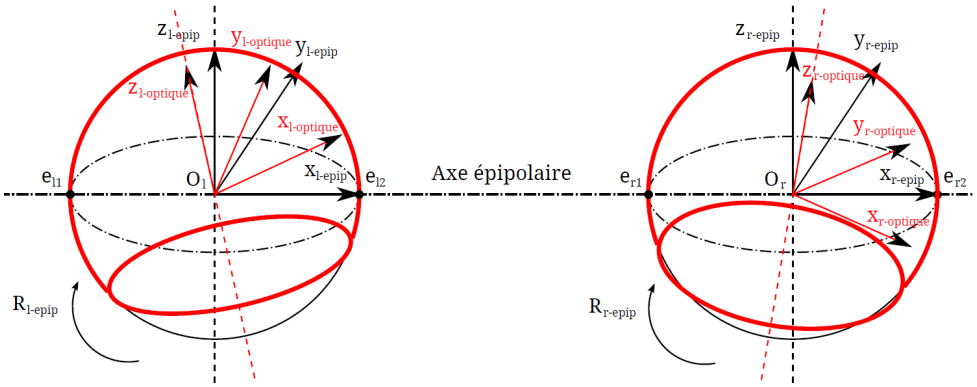


FIGURE 16 – Distinction entre les repères optique et épipolaire. Les repères épipolaire sont obtenus par une rotation des repères optiques, définies par les axes des repères épipolaires.

(simplement grâce à la position d'un épipôle sur la sphère), et par le calcul de l'axe Z_{epip} ou de l'axe Y_{epip}, la connaissance de deux axes étant suffisante pour calculer le troisième dans le cas d'un repère orthonormal. En faisant l'hypothèse que l'axe Z_{epip} se projette lui-même à la position centrale entre les deux épipôles (et non pas au centre du disque image qui correspond à la projection de l'axe Z_{optique}), on obtient une approximation des rotations à appliquer aux repères optiques pour obtenir des repères épipolaires alignés. Avec ces deux axes, on peut calculer Y_{epip} par $Y_{\text{epip}} = Z_{\text{epip}} \wedge X_{\text{epip}}$. Les rotations R_{epip} (figure 16), définies par les axes des repères épipolaires permettent de passer du repère optique au repère épipolaire.

4.1.2 Obtention des cercles épipolaires conjugués

Avec des images calibrées, le cercle de scan est le grand cercle sur la sphère unitaire défini dans le plan $(O, Y_{\text{epip}}, Z_{\text{epip}})$. Le calcul des courbes épipolaires est effectué en utilisant le rayon donnant le point de référence à l'intérieur du cercle de scan, et les positions des deux épipôles. Le point de référence permet de déterminer la rotation d'un grand arc allant d'un épipôle à l'autre. Cet arc est un arc épipolaire, et se projette dans l'image en une courbe épipolaire. En utilisant le même point de référence (c'est-à-dire avec le même angle de rotation) dans les deux images par rapport aux repères épipolaires, on obtient les courbes épipolaires conjuguées (elles sont dans le même plan épipolaire, voir figure 15).

4.1.3 Calcul de la position des points dans l'espace

Cette partie décrit deux manières analytiques de calculer les coordonnées 3D des points à partir des correspondances à l'intérieur des courbes épipolaires conjuguées. Les repères de référence sont les repères épipolaires alignés.

4.1.3.1 Projection des rayons issus des deux capteurs

Dans l'illustration 18, les droites projetées de points correspondants sur les sphères se croisent à la position dans l'espace du point de la scène observé. Les coordonnées de P, dans le repère cartésien de la caméra gauche (O_l, x_l, y_l, z_l), peuvent être exprimées en fonction des coordonnées sphériques de ses projections $P_l(R, \theta_l, \varphi_l)$ et $P_r(R, \theta_r, \varphi_r)$ (voir figure 14).

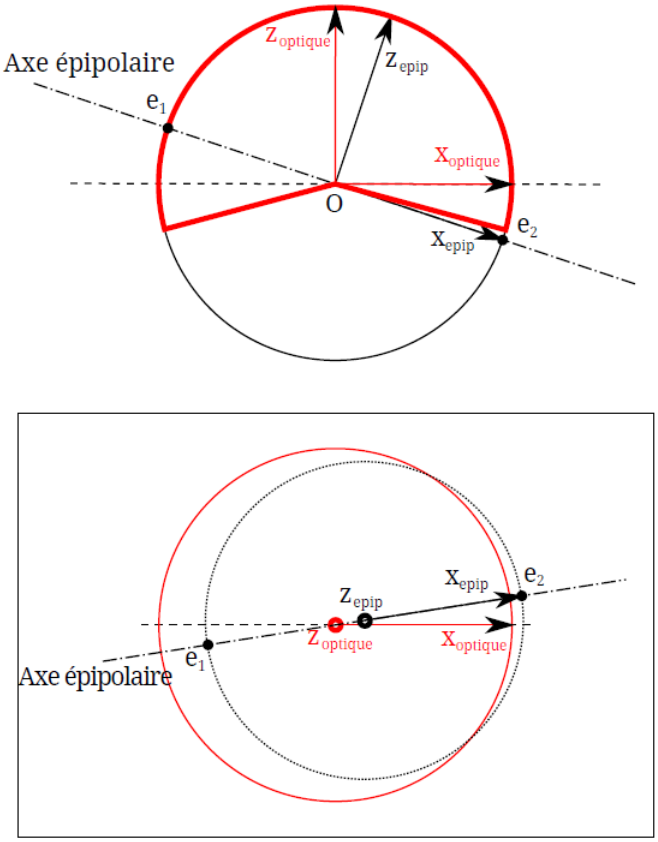


FIGURE 17 – Rotation du repère optique vers le repère épipolaire. En haut : vue de coupe de la sphère le long de l'axe épipolaire. En bas : vue de l'image enregistrée. X_{epip} est le long de l'axe épipolaire, dans une direction proche de X_{optique} (les deux axes forment un angle aigu). Dans le cas d'un objectif de plus de 180° d'angle de vision, le disque image est plus grand que le disuqe défini par les deux épipôles qui lui correspond à 180°. Les deux épipôles ne sont pas nécessairement projetés à l'intérieur du disque image (un épipôles peut très bien être en dehors du champ de vision de la lentille représenté en rouge). Cependant, l'axe épipolaire passe toujours par le centre de la sphère unitaire, et sa droite projetée dans les plans images passe aussi par le centre du disque image.

Dans le repère caméra gauche (O_l, x_l, y_l, z_l), les coordonnées du centre optique sont :

$$O_l \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}_{\text{repère gauche}} \quad \text{et } O_r \begin{pmatrix} b_x \\ 0 \\ 0 \end{pmatrix}_{\text{repère droite}} \tag{30}$$

avec b_x l'entraxe du stéréoscope, c'est-à-dire la distance entre les deux centres optiques.

Les droites O_lP et O_rP sont données par :

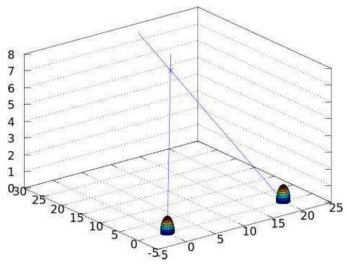


FIGURE 18 – Les sphères représentent les objectifs fisheye. Les deux droites issues des centres optiques et passant par la projection du point de la scène sur les sphères se croisent à la position exacte du point original. Cette illustration a été obtenue en appliquant les principes décrits dans ce document dans un logiciel de mathématiques.

$$O_lP \begin{pmatrix} t\sin(\theta_l)\cos(\varphi_l) \\ t\sin(\theta_l)\sin(\varphi_l) \\ t\cos(\theta_l) \end{pmatrix}_{\text{repère gauche}} \quad \text{avec } t \in [0;+\infty[\tag{31}$$

$$O_rP \begin{pmatrix} b_x + t'\sin(\theta_r)\cos(\varphi_r) \\ t'\sin(\theta_r)\sin(\varphi_r) \\ t'\cos(\theta_r) \end{pmatrix}_{\text{repère gauche}} \tag{32}$$

$$\text{avec } t' \in [0;+\infty[\tag{33}$$

P est à l'intersection de O_lP et O_rP, et ses coordonnées dont obtenues en remplaçant t ou t' respectivement par :

$$t = \frac{b_x}{\sin(\theta_l)\cos(\varphi_l) - \cos(\theta_l)\tan(\theta_r)\cos(\varphi_r)} \tag{34}$$

et :

$$t' = \frac{b_x}{\cos(\theta_r)\tan(\theta_l)\cos(\varphi_l) - \sin(\theta_r)\cos(\varphi_r)} \tag{35}$$

4.1.3.2 À partir des distances du point observé à chaque capteur, à l'intérieur du plan épipolaire

Dans [23], les auteurs proposent de calculer les disparités sphériques (la disparité sphérique normalisée est définie par $\alpha_r - \alpha_l$), et la distance entre le point P et le centre des deux sphères. Soit l_l et l_r ces deux distances, respectivement à partir du centre de l'objectif gauche et de l'objectif droit. Tel que décrit dans la figure 14, en appliquant la loi des sinus, on obtient (voir figure 14) :

$$l_l = \frac{b_x\sin(\alpha_r)}{\sin(\alpha_r - \alpha_l)} \text{ et } l_r = \frac{b_x\sin(\alpha_l)}{\sin(\alpha_r - \alpha_l)} \tag{36}$$

Ainsi on peut exploiter ces résultats pour calculer les coordonnées des points de la scène.
Les valeurs α_i peuvent être obtenues de deux manières :
- par la relation $\alpha_i = \arccos(\sin(\theta_i)\cos(\varphi_i))$,
- ou en reprenant l'angle de parcours des courbes épipolaires conjuguées au moment de l'étape de mise en correspondance, ce qui donne le α du point traité.

Enfin, les coordonnées de P dans le repère de référence gauche sont définies par :

$$P \begin{pmatrix} l_l \cos(\alpha_l) \\ l_l \sin(\alpha_l) \cos(\eta) \\ l_l \sin(\alpha_l) \sin(\eta) \end{pmatrix}_{\text{repère gauche}}, \tag{37}$$

où η est l'angle de rotation du plan épipolaire par rapport à (O, X, Y), ce même angle correspondant à la position dans l'arc de scan pour les points de référence servant à déterminer les courbes épipolaires conjuguées (voir partie 4.1.2).

4.2 Mise en correspondance par programmation dynamique

La méthode d'appariement des points est basée sur l'algorithme de programmation dynamique proposé par [24] qui effectue une mise en correspondance dense des points. La programmation dynamique est une méthode utilisant la notion de graphe de manière simplifiée car exploitant le plus souvent la contrainte épipolaire : un graphe est calculé pour chaque paire de courbes épipolaires. La plupart des travaux existants ([25, 23]) transforment les images sphériques en images rectifiées de telle sorte à ce que les lignes épipolaires soient parfaitement horizontales, dans le but d'appliquer un algorithme de mise en correspondance déjà existant qui s'utilise de la même manière que pour des images rectifiées avec le modèle sténopé. L'approche proposée est différente, car la rectification induit généralement une perte d'informations due aux étapes d'interpolation et de rectification. il s'agit de parcourir les courbes épipolaires en faisant varier l'angle dans les sphères unitaires.

4.2.1 Méthode d'optimisation par programmation dynamique

La programmation dynamique est une méthode d'optimisation des calculs basée sur les graphes. Appliquée à deux courbes épipolaires conjuguées, dans un premier temps, elle crée un graphe en considérant l'une des courbes comme l'axe des ordonnées (correspondant à la courbe épipolaire de l'image de référence) et l'autre courbe comme l'axe des abscisses (correspondant à la courbe épipolaire conjuguée dans l'image de recherche). Puis elle parcourt le graphe afin de définir son plus court chemin.

La programmation dynamique est composée de deux phases :

- La construction du graphe (voir illustration 5a) :
  Les deux axes du graphe correspondent aux lignes épipolaires conjuguées. Il s'agit de comparer les valeurs des pixels composant les deux lignes. L'axe des ordonnées représente la ligne de référence (dans notre exemple l'image de référence est l'image gauche) et l'axe des abscisses représente la ligne de recherche (l'image de recherche est l'image droite). On construit le graphe à partir son origine (à l'origine des deux axes), ligne par ligne, et case par case dans chaque ligne. Chaque case contient la mesure de dissimilarité entre les intensités en niveaux de gris des deux pixels des lignes épipolaires conjuguées correspondants à la position dans le graphe, à laquelle on ajoute la valeur de la case la plus faible parmi les trois précédentes (jusqu'à trois cases sont situées au-dessus et à gauche), pour propager les faibles dissimilarités (c'est-à-dire les plus fortes ressemblances, les meilleures correspondances). Avec les indices i pour les abscisses et j pour les ordonnées, on peut simplement écrire la valeur de chaque case par la formule :
  $A[i,j] = \text{dissimilarité}(\text{référence}[j], \text{recherche}[i]) + \min(A[i-1,j-1], A[i-1,j], A[i,j-1])$.
- La lecture du graphe pour trouver le chemin le plus court (voir illustration 5b) :
  La lecture du graphe, pour la recherche du chemin le moins coûteux, commence à l'extrémité du graphe ; c'est-à-dire à la case correspondant au dernier pixel des deux lignes épipolaires

conjuguées. Le sens de lecture est inverse à celui de la construction. Il s'agit de parcourir le graphe à partir du point extrême pour rejoindre l'origine, et sans jamais revenir en arrière. Pour cela, les seuls mouvements possibles sont ceux pour aller à une case précédente adjacente, ce qui fait trois choix possibles à chaque changement de case (à gauche sur la même ligne, ou la diagonale en haut à gauche ou en haut sur la ligne du dessus). La lecture s'arrête lorsqu'un bord gauche ou haut du graphe est atteint. On connaît de manière très simple les correspondances entre les points des lignes épipolaires conjuguées des deux images, les couples de points servant de coordonnées aux cases traversées par ce chemin constituent l'ensemble des points appariés.

droite ($i \rightarrow$)

($\leftarrow j$) gauche	13	14	15	10	11
10	3	4+3	5+7	0+12	1+12
12	1+3	2+3	3+5	2+8	1+10
14	1+4	0+4	1+4	4+5	3+9
15	2+5	1+4	0+4	5+4	4+9
10	3+7	4+5	5+4	0+4	1+4

(a) Création du graphe.

droite ($i \rightarrow$)

($\leftarrow j$) gauche	13	14	15	10	11
10	3	7	12	12	13
12	**4**	5	8	10	11
14	5	**4**	5	9	12
15	7	5	**4**	9	13
10	10	9	9	**4**	**5**

(b) Lecture du graphe. Les cases noires représentent le chemin le moins coûteux.

TABLE 5 – Exemple simple de graphe de programmation dynamique.

4.2.2 Indexation des courbes épipolaires du modèle sphérique

Avec des caméras fisheye, l'étude de la géométrie épipolaire nous permet de voir que l'on ne dispose pas de lignes épipolaires mais de courbes épipolaires, correspondant aux projections des arcs de cercle épipolaires (ou courbes épipolaires 3D) passant par les deux épipôles, sur le plan image du capteur, en suivant le modèle sphérique approprié. Il s'agit donc de calculer les courbes épipolaires conjuguées et de les traiter de la même manière que les lignes épipolaires dans le cas de caméras classiques représentées avec le modèle sténopé. Ces courbes n'étant pas droites, la création du graphe n'est pas immédiate. La solution consiste à indexer les valeurs des pixels des courbes en lignes, pour pouvoir tracer le graphe de programmation dynamique. Les étapes préliminaires à l'application de la programmation dynamique avec des images fisheye sont décrites dans la figure 19.

La première étape consiste à calculer et mémoriser les angles de longitude et de colatitude pour chacun des pixels des images. Ils ne dépendent pas du contenu de l'image, uniquement des paramètres de calibrage (modèle de projection, centre de projection, longueur focale, rotation, etc). Pour chaque point, on applique l'échantillonnage suivant. L'idée est de retenir les valeurs pour les positions centrales aux pixels, l'échantillonnage consiste donc à prendre la partie entière des coordonnées calculées de manière précise et d'y ajouter 0.5. Nous utilisons les valeurs de ces coordonnées pour le calcul des angles associés aux points (la longitude φ et la colatitude θ).

La seconde étape consiste à calculer un ensemble de courbes épipolaires conjuguées entre les deux images, de façon à couvrir l'ensemble (ou presque) de leurs pixels. Pour obtenir des courbes passant par tous les points des images, on pourrait calculer les courbes conjuguées à chaque pixel. Cette approche est loin d'être optimale car un point peut être traversé par de nombreuses courbes, et on calculerait plusieurs fois les mêmes courbes. Afin d'éviter des calculs inutiles, il

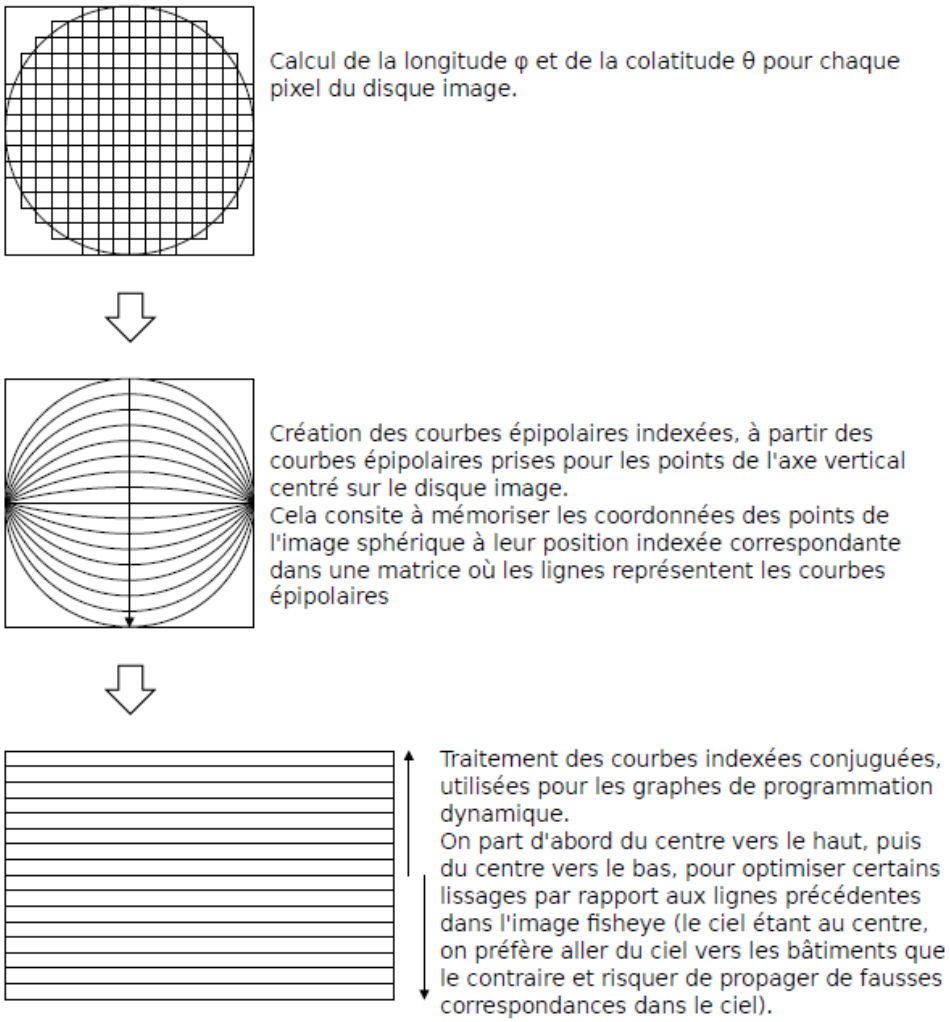


FIGURE 19 – Étapes d'initialisation de la programmation dynamique adaptée aux images sphériques fisheye.

faut calculer les courbes en choisissant des points de référence précis. Une solution satisfaisante consiste à calculer les courbes pour chaque point de l'arc de scan en faisant varier l'angle d'une valeur assez faible pour que les projections dans les images puissent atteindre tous les pixels. De cette manière, on est sûr qu'un maximum de points soient traversés par les courbes épipolaires, et on calcule seulement un nombre de paires de courbes conjuguées inversement proportionnel au pas choisi pour l'angle de parcours de l'arc de scan. Notons tout de même que les points proches des épipôles feront partie de plus de courbes que les autres points. Ensuite, on modélise les courbes calculées de manière discrète en lignes droites indexées, qui seront utilisées pour pouvoir former les axes des graphes de programmation dynamique (on ne peu pas construire un graphe avec des axes incurvés).
Chaque courbe épipolaire ne passe pas par le même nombre de pixels dans les images. Les courbes les plus longues sont celles qui passent près des bords du disque image, et les plus courtes sont celles qui passent au milieu de l'image entre les épipôles.
Pour des raisons pratiques, il faut choisir le même nombre de pixels pour toutes les courbes épipolaires indexées. Ceci signifie que les courbes les plus courtes contiendront des doublons successifs dans leur expression indexée. Dans les faits, on donne une longueur de $\frac{\pi}{2}$ fois le diamètre

en pixels du disque image aux courbes indexées, car cette valeur correspond à la longueur du demi-périmètre du disque (et donc à la longueur de la plus longue courbe).
En connaissant la longueur totale en pixels, on peut faire varier l'angle de parcours γ des courbes épipolaires avec un pas de $\frac{\pi}{\text{longueur des courbes indexées en pixel}}$ pour créer les courbes indexées à partir des courbes épipolaires (π étant la variation totale de γ).

Pour finir, la troisième étape consiste à lire le contenu des couples de courbes épipolaires indexées afin d'appliquer la programmation dynamique. Plutôt que de les analyser du haut en bas ou de la première à la dernière (sens de lecture le plus "naturel"), on choisit de les lire à partir de la position centrale, donc une première partie pour la moitié haute de l'image, puis une deuxième partie pour la moitié basse. Le but est d'éviter des erreurs de propagation du ciel dans les structures dans le cas où des lissages sont utilisés.

4.2.3 Résultats

La figure 20 montre une carte de distances estimée sur un couple d'images réelles stéréoscopiques par programmation dynamique et le nuage de points produit à partir de la carte des distances. La figure 21 montre les mêmes résultats obtenus sur un couple d'images réelles acquises dans le cas mono-caméra.

Une évaluation de l'appariement par programmation dynamique a montré une erreur moyenne en distance inférieure à $2.6m$ sur des objets situés à moins de $50m$. Cette erreur moyenne augmente de manière importante lorsque les objets sont situés à des distances plus importantes dans l'axe du véhicule. L'évaluation a été menée sur 30 couples d'images synthétisées d'une scène urbaine et pour lesquels une vérité terrain a pu être calculée. Chaque couple d'images a été produit à une position différente dans la scène urbaine. La figure 22 présente les résultats obtenus. Sur les images de la dernière colonne les pixels rouge sont les pixels pour lesquels la distance est erronée.

4.3 Post-traitement des nuages de points locaux

4.3.1 Extraction de l'empreinte des bâtiments et de leur hauteur

L'objectif du projet Caploc est d'exploiter la structure du nuages de points que nous obtenons régulièrement le long de la trajectoire du mobile afin de corriger les erreurs de pseudo-distance dues aux réflexions multiples des signaux GNSS sur les objets et les façades autour du mobile. Pour cela, nous avons besoin (1) de connaître la position des surfaces réfléchissantes (principalement celles des bâtiments) et (2) de placer le mobile par rapport à ces surfaces. Nous avons développé une méthode d'extraction de l'empreinte des bâtiments à partir d'un nuage de points. Dans un premier temps, la méthode opère la projection le long de l'axe vertical et sur le plan horizontal de tous les points du nuage. Le résultats obtenu est similaire à un histogramme 2D du nuage. La projection exploite sur une grille régulière définie dans le plan horizontal en comptant, pour chaque éléménet de la grille, les points dont l'altitude est non nulle. Pour un élément de la grille, plus le nombre de points est élevé, plus il est certain de l'existence d'une structure à cet endroit. La figure 23 présente un nuage de points et l'empreinte des bâtiments correspondante. Dans cette figure, l'empreinte est représentée par des segments rouge détectés en appliquant une transformée de Hough probabiliste [26]. Une comparaison avec des mesures prises sur le terrain nous permet de conclure que l'empreinte des bâtiments est localisée avec une erreur maximale de $30cm$ dans des configurations relativement simples. Dans des configrations plus complexe telles que les carrefours, les environnements présentant de la végétation ou les nuages de points présentant des problèmes d'appariement, l'erreur augmente et il est possible d'extraire des empreintes de bâtiments non existants(figure 24). Cette figure, l'image de droite montre le nuage de points

reprojeté dans l'image du couple servant de référence. Il apparait clairement des points dans les nuages qui pourraient provoquer la fausse détection de segments et par conséquent l'extraction d'empreinte de bâtiments qui n'existent pas. Par ailleurs, cette image présente un bâtiment arrondi pour lequel une transformée de Hough donne de mauvais résultats. La connaissance des matrices de projection estimées lors de l'étape de calibrage nous permet alors de localiser le mobile dans ce nuage et de calculer sa position par rapport à chaque surface de réflexion des signaux GNSS. Sur la figure 23 les deux caméras du stéréoscopes sont représentées par deux cercles. L'une des caméras est estimée à une distance de $13.98m$ et $4.96m$ de deux bâtiments situés de part et d'autre de la rue.

La tâche 2 du projet consiste à réaliser un modèle 3D de l'environnement à partir des images et d'en extraire les bâtiments représentrant des surfaces de réflexion des signaux GNSS. Comme nous l'avons présenté dans la section précédente, il est possible de positionner le mobile par rapport aux bâtiments situés autour de lui grâce à l'extraction de leurs empreintes dans le plan horizontal. Nous parvenons ensuite à faire une première estimation de la hauteur des bâtiments dont les empreintes ont été détectées en analysant la distribution des points verticalement au dessus des empreintes détectées. Pour terminer, nous estimons la normale d'un plan que nous cherchons à ajuster sur l'ensemble des points ayant contribué à la détection de chaque segment composant l'empreinte des bâtiments. La figure 25 illustre le résultat obtenu et mmontre qu'il est alors possible de positionner les satellites reçus par rapport aux bâtiments présents dans la scène.

4.3.2 Fusion des nuages de points locaux

Quel que soit l'architecture dans laquelle nous nous plaçons pour produire les nuages de points, il est important de recaler le nuage estimé à partir de deux images avec le nuage de points produit à l'instant précédents. L'algorithe le plus couremment utilisé pour réaliser ce recalage est l'Interative Closest Point (ICP) proposé dans [27] et qui connait désormais beaucoup de variantes. Il opère en minimisant la distance entre deux nuages de points par raffinement des transformations (rotation et translation) liant les deux nuages. Il permet finallement d'obtenir un nuage de points global du trajet emprunté par le mobile.

Nous savons que le repère local de chaque modèle 3D est centré sur la caméra arrière, avec l'axe $\mathbf{X}_{epip}$ selon l'entraxe caméra arrière vers caméra avant. L'axe $\mathbf{Z}_{epip}$ est posé confondu avec l'axe vertical, en direction du ciel (en fait on n'est pas très précis il faudrait un gyroscope, se référer à la partie expliquant la construction du nuage de points). Ainsi on peut se repérer dans un repère orthonormal avec $\mathbf{Y}_{epip} = \mathbf{Z}_{epip} \wedge \mathbf{X}_{epip}$. Appelons O le centre optique de la caméra arrière, on travaille dans le repère "épipolaire" $\mathfrak{R}_{epip} = (O, \mathbf{X}_{epip}, \mathbf{Y}_{epip}, \mathbf{Z}_{epip})$.
Ce repère est à différencier du repère caméra, basé sur le capteur avec les dimensions $\mathbf{X}_{camera}$ et $\mathbf{Y}_{camera}$, et $\mathbf{Z}_{camera}$ normal au plan du capteur. Lors du calibrage stéréo, on définit pour chaque caméra les rotations de leur repère caméra 3D vers leur repère dit épipolaire.
Dans le cas mono, on est capable de générer un nuage de point à chaque nouvelle acquisition. Soit t l'instant de la nouvelle acquisition, cela signifie que le nouveau nuage est basé sur les images acquises aux instants $t-1$ et t. On va appeler $\mathcal{N}_t$ le nuage local de l'instant t. $\mathcal{N}_t$ est centré sur la position caméra de l'instant $t-1$: O_{t-1} ; et sa base est définie par $\mathfrak{B}_{epip\backslash t} = (\mathbf{X}_{epip\backslash t}, \mathbf{Y}_{epip\backslash t}, \mathbf{Z}_{epip\backslash t})$ qui est la base obtenue avec le calibrage effectué à l'acquisition de l'instant t, c'est-à-dire entre les images de $t-1$ et de t. On appelle ainsi $\mathfrak{R}_{epip\backslash t, O(t-1)} = (O_{t-1}, \mathfrak{B}_{epip\backslash t})$ le repère épipolaire dans lequel on définit le nuage local $\mathcal{N}_t$.
De la même manière, le nuage $\mathcal{N}_{t-1}$ est décrit dans le repère $\mathfrak{R}_{epip\backslash t-1, O(t-2)} = (O_{t-2}, \mathfrak{B}_{epip\backslash t-1})$. L'image de l'instant $t-1$ est commune à la création des nuages $\mathcal{N}_{t-1}$ (où elle correspond à l'image avant) et $\mathcal{N}_t$ (où elle correspond à l'image arrière). Pour passer du repère du nuage $\mathcal{N}_t$ au nuage $\mathcal{N}_{t-1}$, il faut se baser sur les trois repères centrés sur la position caméra de l'acquisition $t-1$,

c'est-à-dire O_{t-1} :
- $\mathfrak{R}_{epip\backslash t,O(t-1)}$
- $\mathfrak{R}_{epip\backslash t-1,O(t-1)}$
- $\mathfrak{R}_{camera\backslash t-1}$

À l'aide du calibrage de l'instant t, on connaît la rotation du repère $\mathfrak{R}_{camera\backslash t-1}$ au repère $\mathfrak{R}_{epip\backslash t,O(t-1)}$: $\mathsf{R}(\mathfrak{R}_{camera\backslash t-1} \rightarrow \mathfrak{R}_{epip\backslash t,O(t-1)})$. De même à l'aide du calibrage de l'instant $t-1$, on connaît la rotation du repère $\mathfrak{R}_{camera\backslash t-1}$ au repère $\mathfrak{R}_{epip\backslash t-1,O(t-1)}$: $\mathsf{R}(\mathfrak{R}_{camera\backslash t-1} \rightarrow \mathfrak{R}_{epip\backslash t-1,O(t-1)})$. Ainsi, on est capable de déterminer la rotation du repère épipolaire $\mathfrak{R}_{epip\backslash t,O(t-1)}$ au repère épipolaire $\mathfrak{R}_{epip\backslash t-1,O(t-1)}$:

$$\begin{aligned}\mathsf{R}(\mathfrak{R}_{epip\backslash t,O(t-1)} \rightarrow \mathfrak{R}_{epip\backslash t-1,O(t-1)}) = \mathsf{R}(\mathfrak{R}_{camera\backslash t-1} \rightarrow \mathfrak{R}_{epip\backslash t-1,O(t-1)}) \\ \times \mathsf{R}(\mathfrak{R}_{camera\backslash t-1} \rightarrow \mathfrak{R}_{epip\backslash t,O(t-1)})^T\end{aligned} \tag{38}$$

La figure 26 illustre bien les différents repères. On sait que pour chaque paire d'images, les positions de la caméra sont séparées d'un entraxe donné le long l'axe épipolaire. Cet axe suit $\mathbf{X}_{epip\backslash t}$ pour le calibrage à l'instant t (images de $t-1$ et t), et suit $\mathbf{X}_{epip\backslash t-1}$ pour le calibrage à l'instant $t-1$ (images de $t-2$ et $t-1$). On cherche à placer $\mathcal{N}_t$ par rapport au repère $\mathfrak{R}_{epip\backslash t-1,O(t-2)}$ (le repère local de $\mathcal{N}_{t-1}$, ce qui permet de concaténer les deux nuages locaux successifs dans le même référentiel). On connaît $\mathsf{R}(\mathfrak{R}_{epip\backslash t,O(t-1)} \rightarrow \mathfrak{R}_{epip\backslash t-1,O(t-1)})$, et on sait que $\mathfrak{R}_{epip\backslash t-1,O(t-1)}$ et $\mathfrak{R}_{epip\backslash t-1,O(t-2)}$ ne sont différents que par leur origine. Ainsi le passage de $\mathfrak{R}_{epip\backslash t-1,O(t-1)}$ à $\mathfrak{R}_{epip\backslash t-1,O(t-2)}$ se fait par la translation $\mathbf{t}(O_{t-2} \rightarrow O_{t-1})$:

$$\mathbf{t}(O_{t-2} \rightarrow O_{t-1}) = \begin{pmatrix} entraxe \\ 0 \\ 0 \end{pmatrix} \tag{39}$$

De cette manière, on est capable de décrire la transformation de passage du repère local de $\mathcal{N}_t$ ($\mathfrak{R}_{epip\backslash t,O(t-1)}$) au repère local de $\mathcal{N}_{t-1}$ ($\mathfrak{R}_{epip\backslash t-1,O(t-2)}$) en coordonnées homogènes par :

$$\mathsf{H}(\mathfrak{R}_{epip\backslash t,O(t-1)} \rightarrow \mathfrak{R}_{epip\backslash t-1,O(t-2)}) = \begin{pmatrix} \mathsf{R}(\mathfrak{R}_{epip\backslash t,O(t-1)} \rightarrow \mathfrak{R}_{epip\backslash t-1,O(t-1)}) & \mathbf{t}(O_{t-2} \rightarrow O_{t-1}) \\ \mathbf{0}_{1\times 3} & 1 \end{pmatrix} \tag{40}$$

Il est possible de continuer ainsi de suite jusqu'à se placer dans le premier repère local de la série. Or on choisit comme repère commun (ou global) pour concaténer l'ensemble des nuages ce repère épipolaire local de la première paire d'images traitée. On peut écrire :

$$\begin{aligned}\mathsf{H}(\mathfrak{R}_{epip\backslash t,O(t-1)} \rightarrow \mathfrak{R}_{epip\backslash 1,O(0)}) = \mathsf{H}(\mathfrak{R}_{epip\backslash 2,O(1)} \rightarrow \mathfrak{R}_{epip\backslash 1,O(0)}) \\ \times ... \\ \times \mathsf{H}(\mathfrak{R}_{epip\backslash t-1,O(t-2)} \rightarrow \mathfrak{R}_{epip\backslash t-2,O(t-3)}) \\ \times \mathsf{H}(\mathfrak{R}_{epip\backslash t,O(t-1)} \rightarrow \mathfrak{R}_{epip\backslash t-1,O(t-2)})\end{aligned} \tag{41}$$

Dans le cas mono, on a utilisé un odomètre pour déclencher tous les 2 mètres. Cette méthode est relativement précise. En effet, lors d'une première série d'acquisitions, nous rapportons que sur 2076 images prises, nous mesurons une distance parcourue de 4075m avec le GPS RTK de référence. Ceci revient à une moyenne de 1,963 mètres entre chaque image, soit un écart moyen de 1,85% avec l'entraxe voulu.
Lors d'une seconde série d'acquisition, sur 1255 images acquises, on mesure une distance parcourue de 2487m avec le GPS RTK en mode DGPS. Ceci revient à une moyenne de 1,982 mètres entre chaque image, soit un écart moyen de 0,9% avec l'entraxe voulu.

La figure 27 représente le résultat de la fusion de deux nuages de pints. La figure 28 représente le résultat de la fusion de tous les nuages de points obtenus le long d'une trajectoire.

4.3.3 Rayons GNSS dans le nuage de points 3D

En réglant le problme de syncronisation entre le système d'acquisition vidéo et le récepteur GPS, il est possible de déterminer la position global du récepteur pour chaque acquisition GPS. À l'instant de l'acquisition GPS s, le récepteur est placé à la position $\widetilde{XYZ}_{GPS,s}$ dans le référentiel global. Il est possible de retrouver les positions des satellites avec leur azimut, leur élévation et leur pseudo-distance (non-corrigée, mais l'erreur reste faible relativement à la distance qui est de l'ordre de 20 000km de la surface de la Terre).

Pour représenter de courts rayons correspondants aux rayons GNSS dans le modèle 3D, on a besoin des angles d'azimut et d'élévation de chaque satellite. Par convention, l'azimut est défini dans le sens horaire (ou mathématique indirect) : le nord à 0 rad (0ř), l'est à $\frac{\pi}{2}$ rad (90ř), le sud à π rad (180ř) et l'ouest à $\frac{3\pi}{2}$ rad 270ř. L'élévation est bien définie dans le sens anti-horaire (mathématique direct). Posons $az_{id,s}$ et $el_{id,s}$ respectivement l'azimut et l'élévation du satellite numéro id pour l'acquisition GPS s. Définir un rayon unitaire et le multiplier ensuite par la longueur que l'on veut pour la représentation. Soit $\mathfrak{R}_{epip\backslash 1,s}$ le repère d'orginie la position du récepteur à l'instant de l'acquisition GPS s considérée, et de base la base du repère global. Soit $XYZ^{unit}_{sat,id,s}$ le vecteur unitaire dans la direction du satellite numéro id pour l'acquisition GPS s par rapport au repère $\mathfrak{R}_{epip\backslash 1,s}$ (attention car la référence pour l'azimut est le nord, il faut en tenir compte pour se référer aux axes de la base utilisée ; attention aussi au sens de rotation).

$$XYZ^{unit}_{sat,id,s} = \begin{pmatrix} \cos(\alpha_g.\frac{\pi}{200} - az_{id,s})\cos(el_{id,s}) \\ \sin(\alpha_g.\frac{\pi}{200} - az_{id,s})\cos(el_{id,s}) \\ \sin(el_{id,s}) \end{pmatrix} \tag{42}$$

Ainsi il est possible de représenter les rayons dans le repère global grâce à une translation correspondant à la position du récepteur pour changer de repère (leur base étant la même) par :

$$\widetilde{XYZ}_{GPS,s} + \left(longueur.XYZ^{unit}_{sat,id,s}\right) \tag{43}$$

Pour reprojeter les positions des satellites dans les acquisitions images, il est nécessaire de les considérer au plus près de leur véritable position en 3D. Pour cette raison, on les considère en 3D à une distance donnée du récepteur, qui n'est autre que la pseudo-distance. Si nous posons $pr_{id,s}$ la pseudo-distance du satellite numéro id pour l'acquisition GPS s, alors c'est ce point qui est reprojeté dans l'image choisie.

Dans le cas mono, il faut projeter les satellites dans la dernière image acquise. La position de la caméra est donnée par $XYZ_{APN,n}$. C'est la n-ième image, et elle a été acquise à l'instant t.

Dans le cas stéréo, il faut considéré la dernière paire d'images et connaître la position relative de l'antenne du récepteur aux caméras et notamment de la caméra qui a été définie comme référence.

Il convient de centrer le référentiel sur le repère $\mathfrak{R}_{camera\backslash t}$ Le modèle de projection fisheye et la valeur de ses paramètres régit la manière dont sera reprojeté dans l'image le point représentant le satellite.

Soit $XYZ_{sat,id,s}$ la postion du satellite numéro id, dans le repère $\mathfrak{R}_{epip\backslash 1,s}$.

$$XYZ_{sat,id,s} = pr_{id,s}.XYZ^{unit}_{sat,id,s} \tag{44}$$

Il faut se placer dans le repère optique de la caméra de l'instant t avant d'appliquer la projection inverse tenant compte du modèle de projection fisheye. Pour cela, il faut nécessairement appliquer l'opération suivante (en coordonnées homogènes) :

$$\begin{aligned} &\mathsf{R}(\mathfrak{R}_{camera\backslash t} \to \mathfrak{R}_{epip\backslash t,O(t)})^T \times \mathsf{H}(\mathfrak{R}_{epip\backslash t,O(t)} \to \mathfrak{R}_{epip\backslash 1,O(0)})^T \\ &\times \begin{pmatrix} \mathsf{Id}_3 & \widetilde{XYZ}_{GPS,s} \\ \mathbf{0}_{1\times 3} & 1 \end{pmatrix} \times \begin{pmatrix} XYZ_{sat,id,s} \\ 1 \end{pmatrix} \end{aligned} \tag{45}$$

avec

$$\mathsf{H}(\mathfrak{R}_{epip\backslash t,O(t)} \rightarrow \mathfrak{R}_{epip\backslash 1,O(0)}) = \begin{pmatrix} \mathsf{Id}_3 & \begin{pmatrix} entraxe \\ 0 \\ 0 \end{pmatrix} \\ \mathbf{0}_{1\times 3} & 1 \end{pmatrix} \times \mathsf{H}(\mathfrak{R}_{epip\backslash t,O(t-1)} \rightarrow \mathfrak{R}_{epip\backslash 1,O(0)}) \tag{46}$$

La figure 29 illustre la projection des satellites reçus au moment de l'acquisition d'un couple d'un couple d'images stéréoscopiques dans la ville de Villeneuve d'Ascq. La projection est réalisée dans l'une des images et dans le nuage de points.

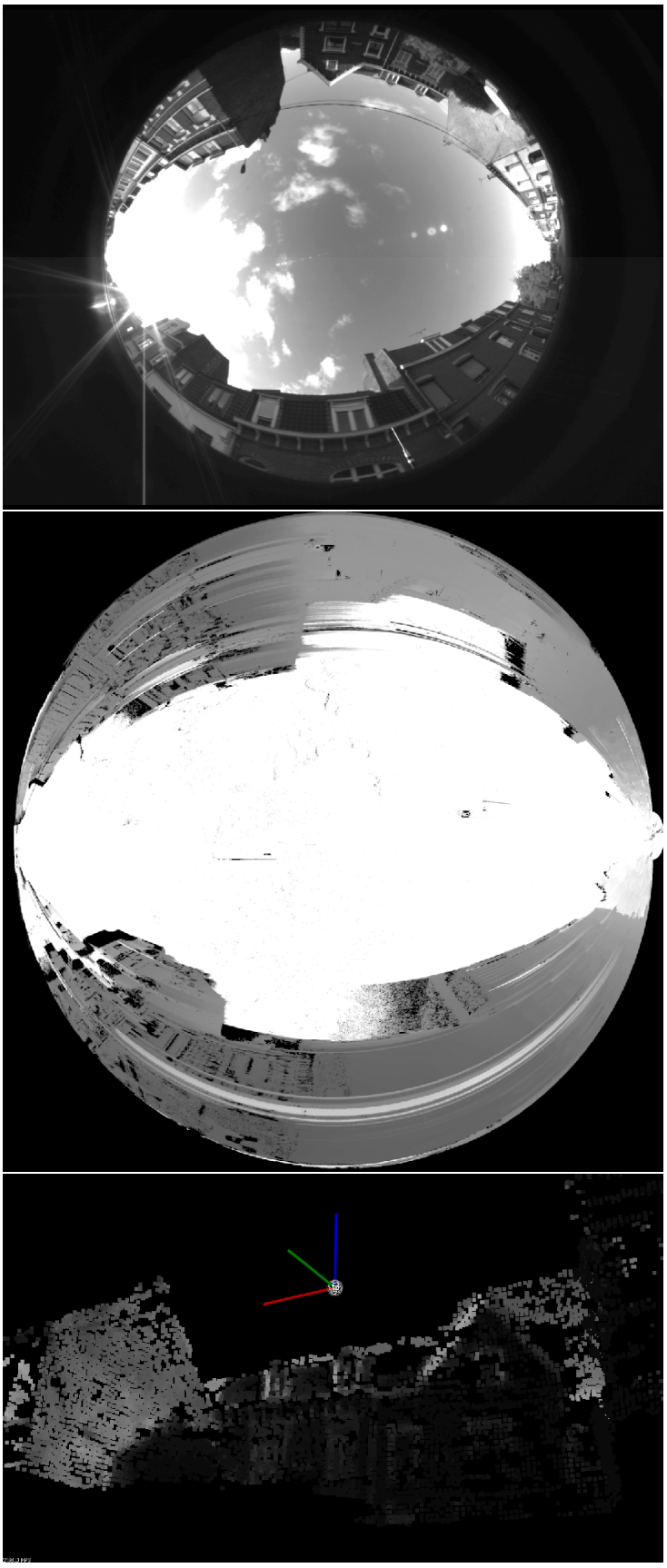


FIGURE 20 – Cas Stéréo. (En haut) une image du couple d'images appariés. (Au milieu) une carte des distances obtenues par programmation dynamique. (En bas) une vue partielle du nuage de points correspondant

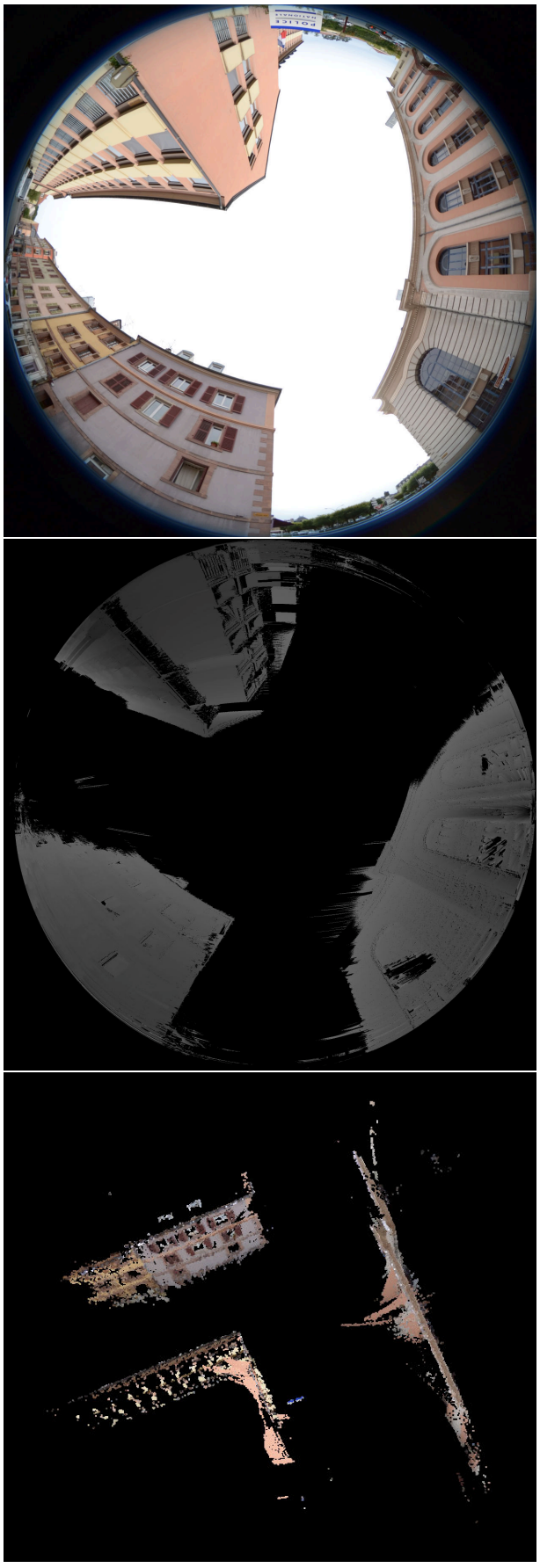


FIGURE 21 – Cas Mono. (En haut) une image du couple d'images appariés. (Au milieu) une carte des distances obtenues par programmation dynamique. (En bas) une vue partielle du nuage de points correspondant

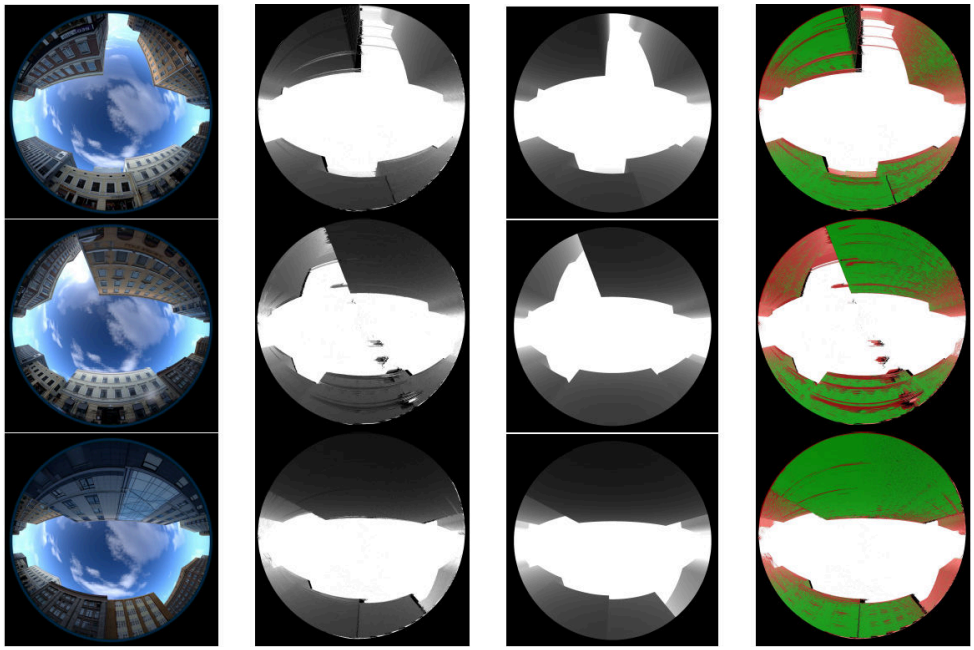


FIGURE 22 – Résultats de l'évaluation menée sur des images de synthèse d'une scène urbaine. De la première à dernière colonne : image de la caméra avant, carte des distances, carte des distances vérité terrain, carte des érreurs.

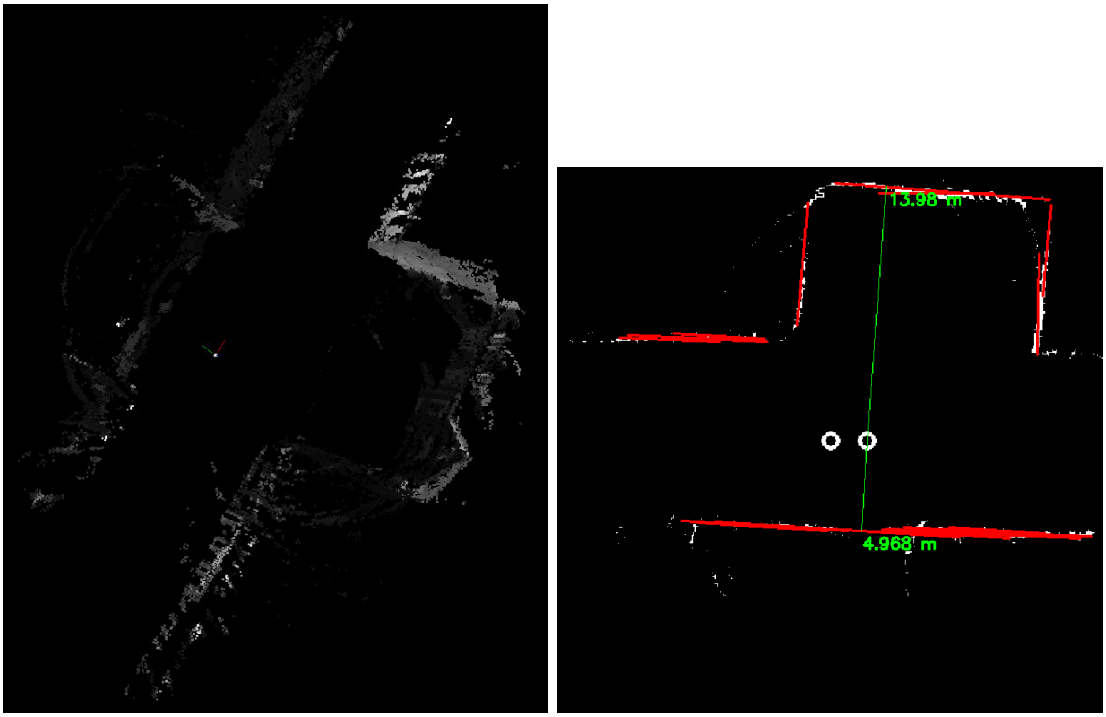


FIGURE 23 – Empreinte de batiments extrait d'un nuage de points. L'enpreinte est présentée après une rotation.

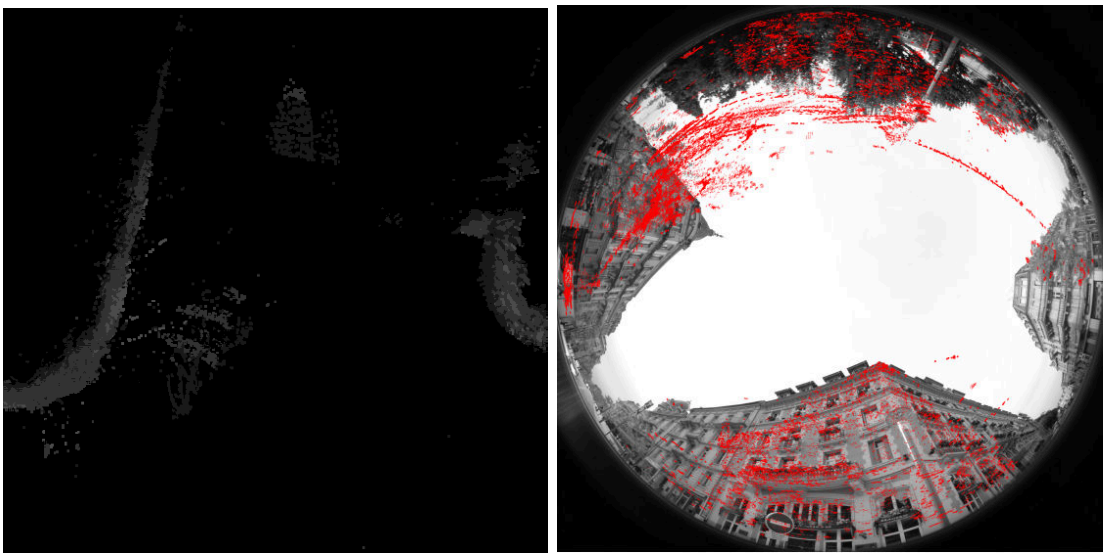


FIGURE 24 – Nuage de points obtenus sur un couple d'images présentant un carrefour avec végétation. Histogramme calculé à partir de ce nuage de points.

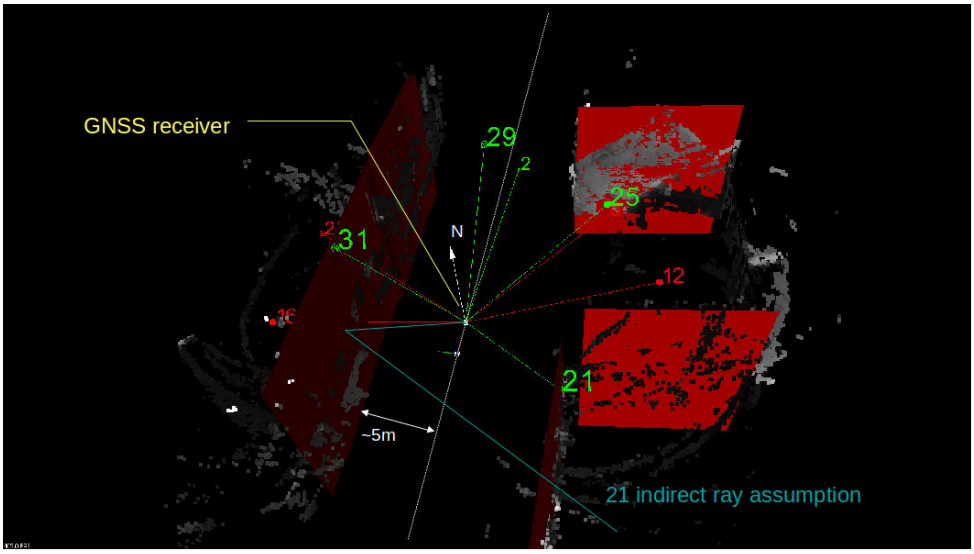


FIGURE 25 – Plane fitting sur le nuages de points à partir de l'empreinte au sol des bâtiments.

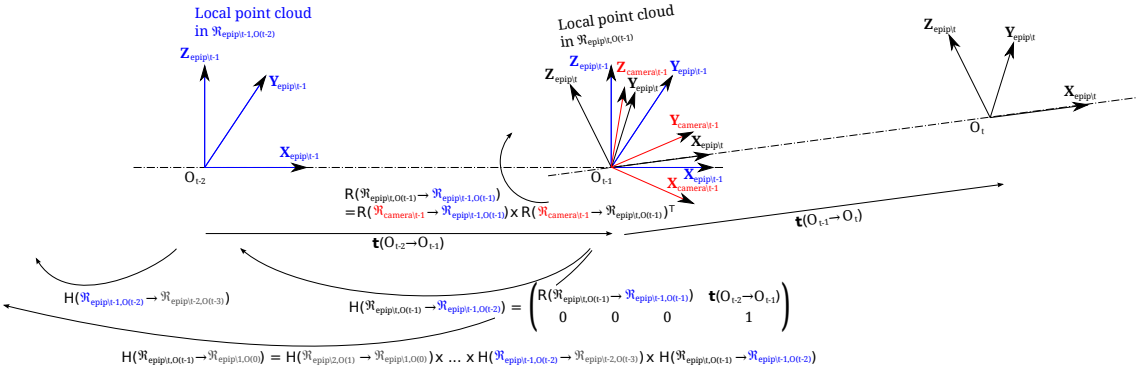


FIGURE 26 – Transformations pour changer de repère et pouvoir concaténer les nuages de points.

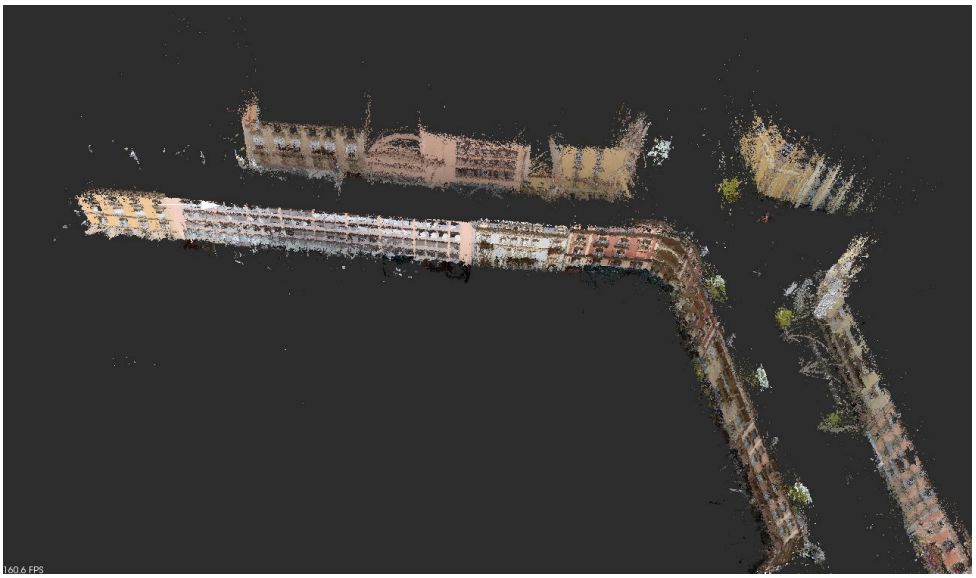


FIGURE 27 – Fusion de plusieurs nuages de points consécutifs.

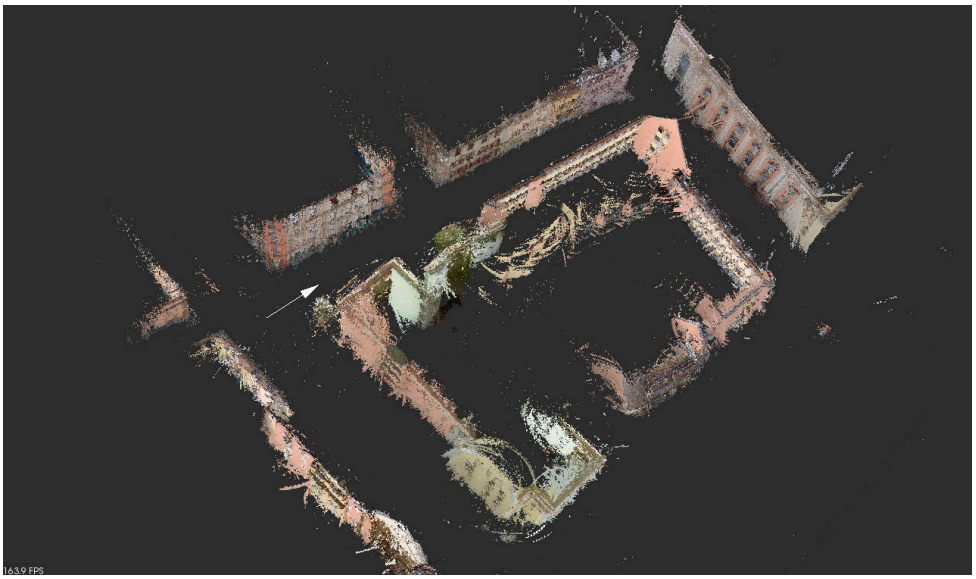


FIGURE 28 – Fusion des nuages de points obtenus le long d'une trajectoire complète.

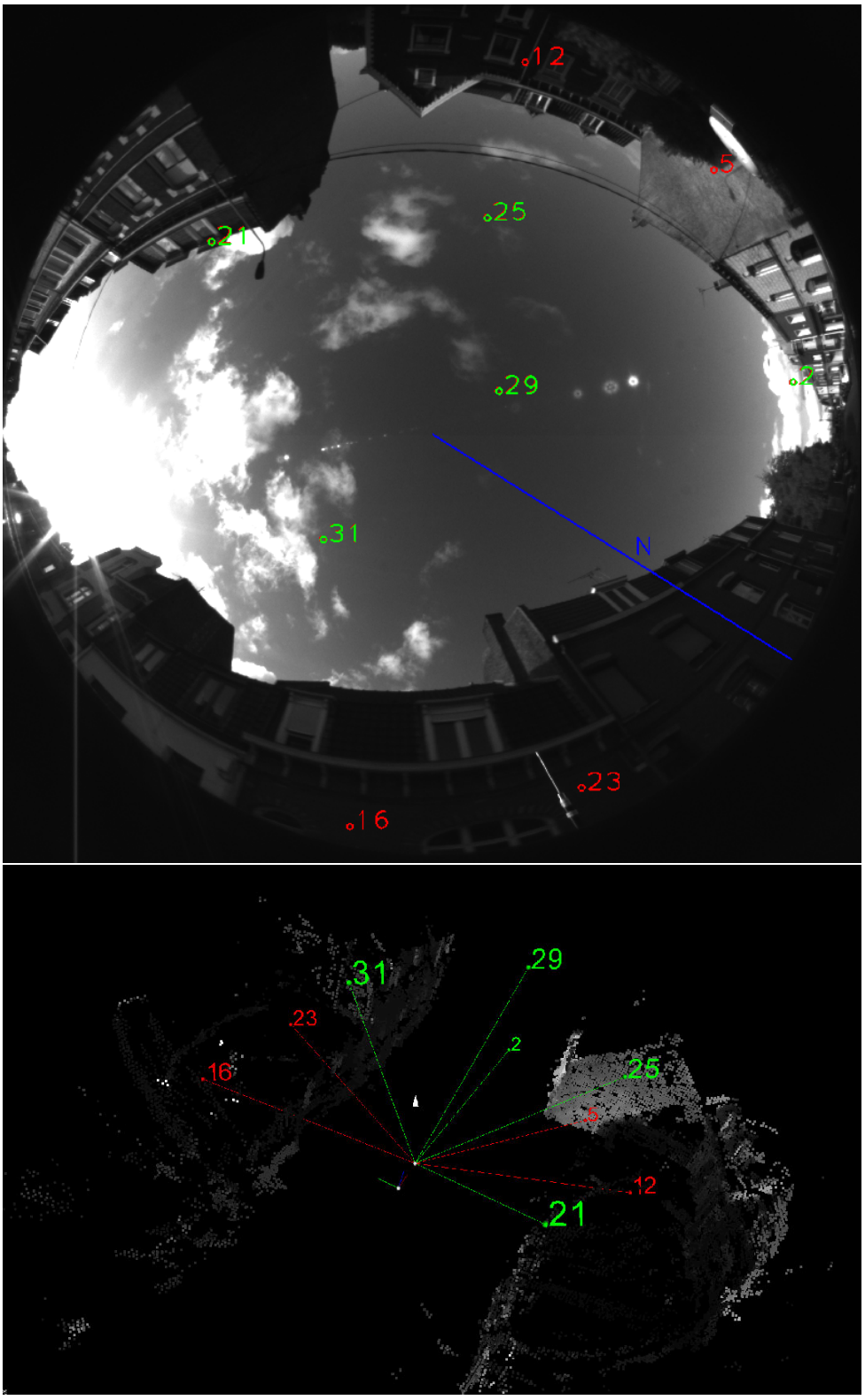


FIGURE 29 – Projection des satellites reçus à l'instant de l'acquisition d'un couple d'images stéréoscopiques.Projection dans l'image (En haut). Projection dans le nuage de points (En bas).

5 Conclusion et perspectives de travail

Ce rapport constitue le livrable 2 de la tâche 2 du projet CAPLOC. Cette tâche est relative à la reconstruction 3D d'un environnement urbain. Sur la période concernée, les équipes se sont focalisées sur la production d'un nuage de points 3D et sur l'extraction d'informations.

Le système d'acquisition mis en place a permis d'étudier deux architectures permettant de produire le nuage de points 3D d'une scène urbaine traversée par un mobile. La première exploite deux caméras vidéo fisheye disposées dans l'axe longitudinal du mobile et produit un nuage à partir de la mise en correspondance du contenu de deux images stéréoscopiques. La seconde architecture exploite une seule caméra et produit un nuage à partir de la mise en correspondance du contenu de deux images acquises à deux instants successifs durant le déplacement une mobile.

Ce livrable a deux sections principales. Dans la première nous exposons la géométrie des deux systèmes étudiés et nous présentons une méthode de calibrage automatique adaptée au modèle de projection sphérique. Elle repose sur l'estimation de la matrice fondamentale en optimisant l'appariement de deux ensembles de points d'intérêt extraits des images à apparier. La méthode proposée a montré de bons résultats sur les images fisheye acquises durant le projet. Toutefois, sachant que la qualité du calibrage dépend fortement de la typologie des points d'intérêt extraits et des descripteurs utilisés, nous avons évalué ce que la littérature nous proposait dans ces deux domaines. Cette évaluation a été réalisée sur des images de synthèse et des images réelles. Nous en avons conlu que les détecteurs définis spécifiquement pour les cas spéhriques ne proposaient pas de résultats meilleurs que ceux obtenus aves le SIFT et son descripteur associé.

Dans la seconde partie, nous décrivons notre algorithme de mise en correspondance par programmation dynamique. Nous présentons les résultats obtenus dans le cas stéréoscopique et dans le cas mmono-caméra. Les résultats sont exploitables en l'état mais nécessiteraient une amélioration afin de réduire les erreurs dans les zones peu texturées (tout particulièrement les zones de ciel). Nous présentons les nuages de points produits à partir des cartes de distances estimées. Nous terminons par trois processus de post-traitement. Le premier processus permet d'extraire l'empreinte des bâtiments et d'interpoler les façades par des plans. Le deuxième propose de fusionner les nuages de points locaux pour produire un nuage de points global d'un trajet suivi par le mobile. Le troisième processus offre la possibilité de replacer les satellites reçus au moment de l'acquisition des images dans le nuage de points 3D. Ensemble ces trois processus nous ont permis d'avoir les premiers résultats sur la correction des pseudo-distances des signaux GNSS réfléchis.

Les perspectives sont nombreuses. Un premier travail consistera à implanter une autre méthode de mise en correspondance respectant mieux des cas de discontinuité dans les zones peu texturées. Nous avons fait des premières expérimentations avec une optimisation par coupure de graphe. Les résultats obtenus méritent que nous nous y attardions dans les mois à venir. Nous travaillerons également à l'implantation des algorithmes développés afin de respecter la contrainte temps-réel sous-jacente à l'application visée. Actuellement la tâche 1, dont l'objectif est de mener à la classification ciel/bâtiment/végétation des éléments présents dans la scène, et la tâche 2 sont très peu mises en relation. Une première action sera de fusionner la segmentation ciel/bâtiment réalisée avec les informations de texture et de couleur et une segmentation identique que nous pouvons réaliser avec la cartes des distances. Une deuxième action pourra être d'intégrer l'information de segmentation de ma tâche 1 dans le processus d'appariement afin d'améliorer la qualité du nuage de points 3D.

Table des figures

1 Dispositif Caploc. . . . 6
2 Couple d'images fisheye fournies par le dispositif. . . . 7
3 Modèle de projection perspective d'une caméra. Issu de [2]. . . . 8
4 Modèle de projection sphérique pour un objectif fisheye (a) et pour un capteur catadioptrique (b). Issu de [1]. . . . 9
5 Fonction de projection fisheye avec le modèle sphérique [1]. . . . 15
6 Erreur angulaire dans un plan épipolaire τ. $\mathbf{X}$ est un point 3D de la scène, $\mathbf{n}$ un vecteur normal unitaire à τ, $\mathbf{C}_1$ et $\mathbf{C}_2$ les centres des caméras. $\mathbf{q}_1$ et $\mathbf{q}_2$ sont les rayons unitaires correspondants au point $\mathbf{X}$, et $\hat{\mathbf{q}}_1$ et $\hat{\mathbf{q}}_2$ leurs meilleures estimations à l'intérieur de τ. ϕ_1 et ϕ_2 sont les angles entre les rayons unitaires et leurs meilleures estimations [1]. . . . 19
7 Configuration épipolaire de prises de vue correspondantes à deux images fisheye. 20
8 Résultat de calibrage sur un couple d'images stéréoscopiques. . . . 21
9 Résultat de calibrage sur un couple d'images mono-caméra. . . . 22
10 Images représentatives des différentes conditions rencontrées pour le déplacement du véhicule. . . . 23
11 Nombre de correspondants image 0. . . . 27
12 Répétabilité image 0. . . . 28
13 Images représentatives des différentes conditions rencontrées pour le déplacement du véhicule. Images réelles. . . . 30
14 Capteur stéréo fisheye calibré. Le point P de la scène est projeté sur les sphères gauche et droite respectivement en P_l et P_r, et est vu comme p_l et p_r dans les images captées par les deux caméras. . . . 31
15 Les courbes épipolaires à l'intérieur du même plan épipolaire (plans Π_i) peuvent être décrites par le même rayon le long de l'arc de scan perpendiculaire à l'axe épipolaire et situé exactement au milieu entre les deux épipôles. . . . 32
16 Distinction entre les repères optique et épipolaire. Les repères épipolaire sont obtenus par une rotation des repères optiques, définies par les axes des repères épipolaires. . . . 33
17 Rotation du repère optique vers le repère épipolaire. En haut : vue de coupe de la sphère le long de l'axe épipolaire. En bas : vue de l'image enregistrée. X_{epip} est le long de l'axe épipolaire, dans une direction proche de X_{optique} (les deux axes forment un angle aigu). Dans le cas d'un objectif de plus de 180° d'angle de vision, le disque image est plus grand que le disuqe défini par les deux épipôles qui lui correspond à 180°. Les deux épipôles ne sont pas nécessairement projetés à l'intérieur du disque image (un épipôles peut très bien être en dehors du champ de vision de la lentille représenté en rouge). Cependant, l'axe épipolaire passe toujours par le centre de la sphère unitaire, et sa droite projetée dans les plans images passe aussi par le centre du disque image. . . . 34
18 Les sphères représentent les objectifs fisheye. Les deux droites issues des centres optiques et passant par la projection du point de la scène sur les sphères se croisent à la position exacte du point original. Cette illustration a été obtenue en appliquant les principes décrits dans ce document dans un logiciel de mathématiques. . . . 35
19 Étapes d'initialisation de la programmation dynamique adaptée aux images sphériques fisheye. . . . 38
20 Cas Stéréo. (En haut) une image du couple d'images appariés. (Au milieu) une carte des distances obtenues par programmation dynamique. (En bas) une vue partielle du nuage de points correspondant . . . 44

21 Cas Mono. (En haut) une image du couple d'images appariés. (Au milieu) une carte des distances obtenues par programmation dynamique. (En bas) une vue partielle du nuage de points correspondant 45
22 Résultats de l'évaluation menée sur des images de synthèse d'une scène urbaine. De la première à dernière colonne : image de la caméra avant, carte des distances, carte des distances vérité terrain, carte des érreurs. 46
23 Empreinte de batiments extrait d'un nuage de points. L'enpreinte est présentée après une rotation. 46
24 Nuage de points obtenus sur un couple d'images présentant un carrefour avec végétation. Histogramme calculé à partir de ce nuage de points. 47
25 Plane fitting sur le nuages de points à partir de l'empreinte au sol des bâtiments. 47
26 Transformations pour changer de repère et pouvoir concaténer les nuages de points. 47
27 Fusion de plusieurs nuages de points consécutifs. 48
28 Fusion des nuages de points obtenus le long d'une trajectoire complète. 48
29 Projection des satellites reçus à l'instant de l'acquisition d'un couple d'images stéréoscopiques.Projection dans l'image (En haut). Projection dans le nuage de points (En bas). 49

Liste des tableaux

1 Liste des détecteurs étudiés. Les détecteurs marqués en gras sont ceux qui sont donnés assez robustes aux changements d'échelle et d'orientation, et qui ont donc été retenus pour les évaluations sur les images fisheye. 24
2 Liste des descripteurs étudiés. Les descripteurs marqués en gras sont ceux qui sont donnés assez robustes aux changements d'échelle et d'orientation, et qui ont donc été retenus pour les évaluations sur les images fisheye. 25
3 Nombre de correspondants image 0, valeurs numériques. 26
4 Répétabilité image 0, valeurs numÃľriques. 29
5 Exemple simple de graphe de programmation dynamique. 37

Références bibliographiques

[1] Branislav Mičušík : *Two-View Geometry of Omnidirectional Cameras*. Thèse de doctorat, Czech Technical University in Prague, 2004. 8, 9, 14, 15, 17, 18, 19, 51

[2] Radu Horaud et Olivier Monga : *Vision par ordinateur : outils fondamentaux*. ÃĽditions HermÃĺs, second édition, 1995. 8, 51

[3] Branislav Mičušík et TomÃąÅą Pajdla : Estimation of omnidirectional camera model from epipolar geometry. *Conference on Computer Vision and Pattern Recognition*, 2003. 14

[4] Krystian Mikolajczyk et Cordelia Schmid : A performance evaluation of local descriptors. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(10):1615–1630, Oct 2005. 16, 20

[5] Chris Harris et Mike Stephens : A combined corner and edge detector. *In Proc. of Fourth Alvey Vision Conference*, pages 147–151, 1988. 17

[6] Jianbo Shi et Carlo Tomasi : Good features to track. *Computer Vision and Pattern Recognition CVPR '94*, 1994. 17

[7] David G. Lowe : Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, novembre 2004. 17, 24, 26

[8] Edward Rosten et Tom Drummond : Machine learning for high-speed corner detection. *Proceedings of the 9th European Conference on Computer Vision - Volume Part I*, ECCV'06, pages 430–443, Berlin, Heidelberg, 2006. Springer-Verlag. 17

[9] Herbert Bay, Tinne Tuytelaars et Luc Van Gool : Surf : Speeded up robust features. *Proceedings of the 9th European Conference on Computer Vision - Volume Part I*, ECCV'06, pages 404–417, Berlin, Heidelberg, 2006. Springer-Verlag. 17, 25

[10] Motilal Agrawal, Kurt Konolige et MortenRufus Blas : Censure : Center surround extremas for realtime feature detection and matching. David Forsyth, Philip Torr et Andrew Zisserman, éditeurs : *Computer Vision âĂŞ ECCV 2008*, volume 5305 de *Lecture Notes in Computer Science*, pages 102–115. Springer Berlin Heidelberg, 2008. 17

[11] David Nistér et Henrik Stewénius : Linear time maximally stable extremal regions. *Proceedings of the 10th European Conference on Computer Vision : Part II*, ECCV '08, pages 183–196, Berlin, Heidelberg, 2008. Springer-Verlag. 17

[12] Ethan Rublee, Vincent Rabaud, Kurt Konolige et Gary Bradski : Orb : An efficient alternative to sift or surf. *Proceedings of the 2011 International Conference on Computer Vision*, ICCV '11, pages 2564–2571, Washington, DC, USA, 2011. IEEE Computer Society. 17, 25

[13] Stefan Leutenegger, Margarita Chli et Roland Y. Siegwart : Brisk : Binary robust invariant scalable keypoints. *Proceedings of the 2011 International Conference on Computer Vision*, ICCV '11, pages 2548–2555, Washington, DC, USA, 2011. IEEE Computer Society. 17, 25

[14] Richard Hartley : In defense of the eight-point algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(6):580 –593, jun 1997. 17, 18

[15] Pierre Moreels et Petro Perona : Evaluation of features detectors and descriptors based on 3d objects. *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 1, pages 800–807 Vol. 1, Oct 2005. 20

[16] Steffen Gauglitz, Tobias HÃűllerer et Matthew Turk : Evaluation of interest point detectors and feature descriptors for visual tracking. *International Journal of Computer Vision*, 94(3):335–360, 2011. 20

[17] You Li : *Stereo vision and LIDAR based Dynamic Occupancy Gird mapping : Application to scenes analysis for Intelligent Vehicles.* Thèse de doctorat, ÃĽcole Doctorale SPIM, 2013. 20

[18] Krystian Mikolajczyk, Tinne Tuytelaars, Cordelia. Schmid, Andrew Zisserman, JiÅŹÃŋ Matas, Frederik Schaffalitzky, Timor Kadir et Luc Van Gool : A comparison of affine region detectors. *International Journal of Computer Vision*, 65(1-2):43–72, 2005. 24

[19] Michael Calonder, Vincent Lepetit, Christoph Strecha et Pascal Fua : Brief : Binary robust independent elementary features. *Proceedings of the 11th European Conference on Computer Vision : Part IV*, ECCV'10, pages 778–792, Berlin, Heidelberg, 2010. Springer-Verlag. 25

[20] Alexandre Alahi, Raphael Ortiz et Pierre Vandergheynst : Freak : Fast retina keypoint. *2012 IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR)*, IEEE Conference on Computer Vision and Pattern Recognition, pages 510–517, 345 E 47TH ST, NEW YORK, NY 10017 USA, 2012. IEEE, IEEE. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Providence, RI, JUN 16-21, 2012. 25

[21] Marius Muja et David G. Lowe : Fast approximate nearest neighbors with automatic algorithm configuration. *International Conference on Computer Vision Theory and Application VISSAPP'09)*, pages 331–340. INSTICC Press, 2009. 26

[22] Zafer Arican et Pascal Frossard : Dense depth estimation from omnidirectional images. *Journal of Computer Vision and Image Understanding*, 2010. 31

[23] Shigang Li : Binocular spherical stereo. *IEEE Transactions On Iintelligent Transportation Systems*, 9:589–600, 2008. 35, 36

[24] Sven Forstmann, Yutaka Kanou, Jun Ohya, Sven Thuering et Alfred Schmitt : Real-time stereo by using dynamic programming. *Computer Vision and Pattern Recognition Workshop*, 2004. 36

[25] Shishir Shah et Jake K. Aggarwal : Mobile robot navigation and scene modeling using stereo fish-eye lens system. *Machine Vision and Applications*, 10:159–173, 1997. 36

[26] Jiri Matas, C. Galambos et J. Kittler : Robust detection of lines using the progressive probabilistic hough transform. *Computer Vision and Image Understanding*, 78(1):119 – 137, 2000. 39

[27] P. N. Belhumeur et D. Mumford : A bayesian treatment of the stereo correspondence problem using half-occluded regions. *IEEE International Conference on Computer Vision and Pattern Recognition*, volume 5, pages 506–512, Champaign, USA, juin 1992. 40

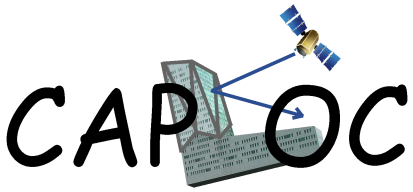
Groupe opérationnel n°2 du Predit 4

« Gestion du trafic et des mobilités »

CAPLOC

Combinaison de l'**A**nalyse d'images
et de la connaissance de la **P**ropagation
des signaux pour la **LOC**alisation

Convention 10-MT-CVS-001



Livrable d'avancement de la tâche 3. Utilisation de l'image pour la caractérisation des signaux reçus et de la disponibilité globale de la constellation.

12 juin 2013

Juliette Marais et al.
Ifsttar

Identification

Acronyme du projet	CAPLOC
Titre du projet	Combinaison de l'Analyse d'images et de la connaissance de la Propagation des signaux pour la LOCalisation
Coordinatrice du projet	Juliette MARAIS (IFSTTAR/LEOST)
Date de début de projet	19 octobre 2010
Date de fin de projet	18 octobre 2014
Partenaires	IFSTTAR/LEOST, UTBM/IRTES-SET
Type de document	Livrable
Date contractuelle de livraison	18 octobre 2013
Accessibilité	La diffusion de ce document est limitée au programme PREDIT

Gestion des versions

Version	**Date**	**Auteurs**	**Modifications**
1.0	12/06/2013	Juliette Marais	Rédaction du corps du livrable
2.0	21/10/2013	Dhouha Attia, Cyril Meurie, Sarab Tay, Sébastien Ambellouis	Modifications
3.0	30/10/2013	Juliette Marais	Approbation

Table des matières

1 Introduction **4**
1.1 Résumé du projet . . . 4
1.2 Objet de ce livrable . . . 4

2 Le concept PREDISSAT **5**
2.1 Contexte . . . 5
2.2 Introduction à la localisation par satellites . . . 5
2.3 Méthodes de détection des multitrajets . . . 5
2.4 Utilisation de l'image : équipement et traitement . . . 6
2.5 Hypothèses liées à la propagation . . . 7
2.6 Résultats issus de PREDISSAT . . . 8
2.7 Conclusions . . . 8

3 Données issues de l'image **10**
3.1 Détection de la ligne d'horizon pour la classification des états de réception des satellites . . . 10
3.1.1 Détection de la ligne d'horizon . . . 10
3.1.2 Calibrage et position des satellites . . . 10

4 Amélioration de la précision et premières pistes pour l'estimation de la confiance dans la position **12**
4.1 Introduction . . . 12
4.2 Données expérimentales utilisées . . . 12
4.3 Exclusion des satellites NLOS dans le calcul de la position . . . 14
4.4 Utilisation combinée du DOP et de l'exclusion . . . 17
4.5 Analyse quantitative. Estimation de la précision d'une position calculée . . . 18
4.6 Perspectives . . . 19
4.6.1 Pourcentage de ciel visible et performances . . . 19
4.6.2 Combinaison état de réception et modèle d'erreur . . . 22
4.6.3 Utilisation du modèle 3D . . . 22

5 Conclusion et perspectives **25**

Bibliographie **27**

1 Introduction

1.1 Résumé du projet

CAPLOC est un projet de recherche amont qui apporte une brique technologique innovante de localisation précise. Il propose d'améliorer la précision d'un système de localisation grâce à la perception vidéo de l'environnement du véhicule et plus particulièrement l'environnement de propagation autour du récepteur GNSS. En effet, les problèmes de disponibilité et d'imprécision sont directement liés à la densité des obstacles notamment en environnement urbain dense. Il s'agit de réduire l'effet des conditions de propagation des signaux GNSS, notamment les multi-trajets, sans dégrader la disponibilité du service, ce qui est souvent le cas lorsque, classiquement, des procédures d'exclusion de défauts sont mises en œuvre.

Deux niveaux d'informations sont étudiés :

1. Le premier vise à indiquer en temps réel, l'état de réception de chacun des satellites, à partir de l'extraction du ciel par un système mono-caméra embarqué ;
2. Le deuxième vise à corriger l'erreur induite par la propagation sur la valeur du retard grâce à la construction d'un modèle 3D de l'environnement par un système multi-caméra.

Le projet CAPLOC s'articule autour de 3 tâches techniques. Les objectifs des deux premières tâches sont respectivement de développer une méthode de segmentation automatique du ciel et des bâtiments obstruant la réception directe des satellites et de construire un modèle 3D de l'environnement situé autour de l'antenne GNSS. La troisième tâche est dédiée à l'exploitation des sorties des deux tâches précédentes afin d'augmenter la précision de la localisation.

1.2 Objet de ce livrable

Ce rapport vise à faire un bilan d'avancement de la tache 3 et en particulier concerne la sous tâche 3.1 : Utilisation de l'image pour la caractérisation des signaux reçus et de la disponibilité globale de la constellation. Il est composé de trois sections :

- la première reprend le concept PREDISSAT qui a servi de base à ces travaux.
- la deuxième illustre les données issues de la tâche 1 (de l'image) qui sont utilisées dans les travaux liés à la localisation.
- le troisième présente l'usage de ces données pour l'amélioration de la précision et de premières pistes pour l'estimation de la confiance dans la position.

2 Le concept PREDISSAT

Cette section reprend le descriptif publié dans [1] et repris en annexe.

2.1 Contexte

L'expérience du LEOST en matière d'utilisation de l'image pour la localisation par satellite remonte à une première étude réalisée dans le cadre du projet DEUFRAKO-P (1999) pour le compte de la SNCF qui souhaitait évaluer la disponibilité de la constallation GPS le long d'une ligne ferroviaire, suivie d'une thèse [2] cofinancée par l'INRETS et la Région Nord-Pas-de-Calais.

2.2 Introduction à la localisation par satellites

Les systèmes de localisation par satellites (GNSS - Global Satellite Navigation System) reposent sur le principe de la mesure du temps de propagation des signaux émis par les satellites de la constellation. Les satellites émettent en continu des signaux que le récepteur utilise pour extraire leur temps de parcours et un certain nombre d'information utiles contenues dans le message de navigation. Ce temps de parcours, multiplié par la vitesse de propagation des ondes, permet d'estimer la distance entre chacun des satellites et le récepteur. On appelle cette mesure la pseudo-distance. La position du récepteur est calculée par le récepteur lui-même à partir de la réception de 4 signaux simultanément, au moins par un principe de trilatération. Mathématiquement, il s'agit de résoudre un système d'équations à 4 inconnues que sont la position du récepteur x,y,z et le décalage d'horloge δt.

En environnement transport, la réception des signaux satellitaires est fréquemment perturbée par les obstacles environnant le véhicule. En effet, la disponibilité des systèmes de localisation des mobiles terrestres utilisant des signaux issus de satellites est considérablement réduite lorsque le canal de propagation est obstrué par la présence de masques, comme cela arrive en milieu urbain dense ou ferroviaire. La précision de la localisation obtenue est également directement liée à la qualité des mesures réalisées. Deux facteurs liés à l'environnement sont à mentionner particulièrement :

- La géométrie des satellites utilisés, caractérisée par le DOP (Dilution of Precision) [3].
- L'imprécision de la mesure de pseudo-distance causée par les multitrajets.

2.3 Méthodes de détection des multitrajets

On trouve dans la littérature plusieurs approches de détection des satellites reçus par multitrajets. Les multitrajets reçus en l'absence de trajet direct sont qualifiés de NLOS (Non Line Of Sight). En environnement contraint, tel qu'en urbain, les principaux perturbateurs sont les obstacles proches de l'antenne de réception (« proche »étant relatif face aux 20000km d'altitude des satellites). L'impact de ces obstacles est directement visible sur l'amplitude des signaux reçus, répercutée sur la valeur du rapport signal sur bruit (CN0). Dans les travaux de [3], une solution basée sur le seuillage de l'amplitude du signal GNSS reçu a été déjà proposée, ainsi que dans [4]. La difficulté de la détection des satellites LOS/NLOS sur ce critère réside dans le choix de la valeur du CN0 seuil. Ce seuil pourra être sensiblement différent d'un équipement à l'autre. De plus, il est fixé à partir de données statistiques et la dispersion des valeurs autour de ce seuil implique de mauvaises détections. C'est pourquoi d'autres techniques sont aujourd'hui étudiées. L'outil PREDISSAT du LEOST a été développé dans cet objectif en 2002 [2] et était fondé sur l'utilisation de caméras classiques et d'un processus de post-traitement. [5] a repris le concept avec une caméra Infra Rouge. Avec le développement des modèles 3D urbains, [6], [7], [8] ou encore [9] ont publié en 2011-2012 des résultats montrant comment ces modèles pouvaient

également permettre de déterminer la présence ou non d'un trajet direct en comparant les positions des obstacles modélisés et les positions des satellites. L'approche fondée sur l'utilisation de l'image est poursuivie dans CAPLOC et décrite ci dessous.

2.4 Utilisation de l'image : équipement et traitement

Le cas le plus favorable au sens de la qualité et du niveau de signal reçu est le cas où le champ électrique est reçu directement, c'est-à-dire lorsque le trajet émetteur-récepteur se fait au-dessus de la ligne de visibilité directe. En effet, lorsque les satellites sont masqués, ils sont soit non reçus, soit reçus par trajets multiples et donc susceptibles d'engendrer une erreur dans le calcul de la position. En nous inspirant des approches appliquées aux satellites de communication [10], [11], nous avons développé un outil fondé sur la connaissance vidéo de l'environnement.

Comme évoqué précédemment, l'originalité de l'outil réside dans l'utilisation de l'image pour analyser l'environnement autour de l'antenne afin de caractériser la qualité de la réception des signaux satellitaires. Afin d'obtenir une vision de l'environnement de l'antenne, le système expérimental testé est composé de deux caméras matricielles analogiques noir et blanc équipées d'un capteur CCD (capteur à transfert de charge). Ces caméras sont placées sur le toit du véhicule pour un trajet seulement, tel qu'illustré par la figure 1. Chaque caméra fournit vingt-cinq images par seconde. Cependant, la synchronisation de l'enregistrement des images avec le capteur odométrique nous permet d'enregistrer les images en fonction de la distance parcourue. Par exemple, nous enregistrerons une image tous les mètres. Cette information spatiale nous permet d'appliquer ensuite un procédé de stéréovision mono-caméra. Ce procédé consiste à considérer deux images successives comme deux images issues de deux caméras parallèles, s'inspirant ainsi des techniques de la stéréovision classique.

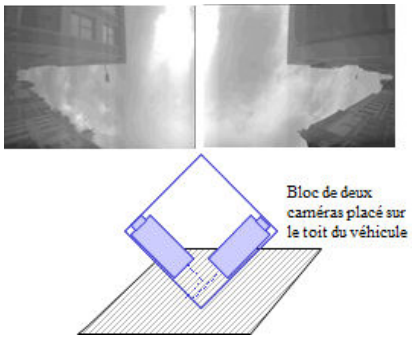


FIGURE 1 – Caméras placées sur le toit du véhicule

Nous pouvons ainsi calculer la distance de chacun des obstacles vus dans l'image, ainsi que leur hauteur. La connaissance de chacune des dimensions des obstacles nous permet alors de reconstituer l'environnement en trois dimensions et de considérer chacun des obstacles masquants autour de l'antenne, ce qui est nécessaire à la détermination des trajets réfléchis. Les zones situées à l'avant et à l'arrière du véhicule sont considérées comme dégagées de masques.

La reconstruction de l'environnement se fonde sur l'hypothèse que les obstacles sont situés le long de la trajectoire, parallèlement à la direction du véhicule. La configuration est donc de type canyon, tel qu'illustré sur la figure 2. En appliquant à ce modèle en trois dimensions les lois de la géométrie optique comme dans le cas de techniques de lancer de rayons, l'outil permet de déterminer non seulement les satellites reçus directement et les satellites bloqués, mais aussi les satellites qui peuvent être reçus après une réflexion [12].

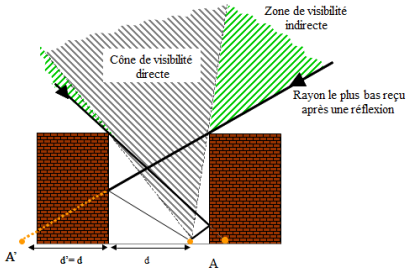


FIGURE 2 – Configuration de type canyon et lois de la géométrie optique

En raison des hypothèses géométriques appliquées, les résultats obtenus avec ce dispositif sont intéressants lorsque les obstacles sont modélisables par des plans verticaux et d'autant plus que leurs arêtes sont marquées. Dans les environnements montagneux par exemple, nos hypothèses sont trop éloignées de la réalité du terrain.

2.5 Hypothèses liées à la propagation

Pour se localiser, un récepteur utilise quatre (ou plus si possible) signaux issus de quatre satellites différents. La position est calculée à partir de la mesure du temps de propagation de chacun de ces signaux par triangulation. Aujourd'hui, la précision de l'information de localisation d'un système classique est de l'ordre de 10 m. Les principales causes des imprécisions relèvent d'une mauvaise estimation de la pseudo-distance : retards liés à la propagation des signaux dans l'atmosphère, erreurs de synchronisation des horloges, mauvaise répartition des satellites disponibles autour de l'antenne... Un certain nombre de modèles mathématiques ou d'équipements permettent de réduire ces erreurs. Cependant, l'environnement autour de l'antenne (immeubles, arbres, ponts...) peut également limiter la précision et la disponibilité. La réception des signaux satellitaires est classée en trois états, tel qu'illustré par la figure 3 :

- l'état V, lorsque le signal est reçu en visibilité directe ;
- l'état R, lorsque le signal est reçu par trajets multiples ; le signal subit réflexion et/ou diffraction sur des arbres et des obstacles environnants ; nous ferons ici l'hypothèse d'un trajet réfléchi au maximum une fois ;
- l'état B, en l'absence de signal lorsque des masques importants obstruent la visibilité.

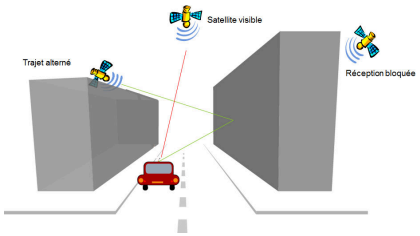


FIGURE 3 – Etats de réception des signaux satellitaires

Les conséquences seront de deux niveaux :

- lorsque moins de quatre satellites sont reçus dans l'état V ou R, le récepteur n'est pas en mesure de calculer sa position ;
- si une majorité de satellites est reçue dans l'état R, la position sera imprécise.

Les élévations et azimuts des satellites sont simulées sur PC afin de comparer élévation et azimut de chaque satellite avec l'élévation et l'azimut du masque issu de l'image et de son traitement.

2.6 Résultats issus de PREDISSAT

La comparaison des positions des satellites avec les positions des masques permet d'obtenir des résultats statistiques le long du trajet complet, ou localisés le long de la ligne. Les résultats statistiques offrent une vision globale de la couverture de la ligne. Les résultats localisés offrent une vision de la disponibilité du service en tout point de la ligne. On montre, pour un trajet donné, le nombre de satellites reçus en chaque point à un instant donné. Ces résultats observés sur une période de vingt-quatre heures offrent une information sur la couverture du système en chaque point de la ligne en termes de :

- gabarit, c'est-à-dire de nombre maximal et de nombre minimal de satellites reçus en chaque point ;
- probabilité qu'un nombre suffisant de satellites soient reçus en chaque point.

A l'issu des travaux de thèse, une utilisation envisagée était la création d'une base de données de prédictions (LUT pour look-up-table) inspirée de [13] en ce qui concerne le canal de propagation satellite et de [14] pour la localisation de mobiles GSM. La chaîne PREDISSAT représentée sur la figure 4 offre les informations relatives à l'état de réception de chacun des satellites en tout point de la trajectoire et à tout instant, ce qui permet ensuite au système de choisir les satellites utilisés en fonction de leur état ou de corriger les retards induits grace à un gabarit des retards, fondé sur la connaissance de la géométrie de l'environnement, en particulier des hauteurs des bâtiments et de la largeur de la rue [12]. Ces informations seront moins précises, mais plus faciles à manipuler.

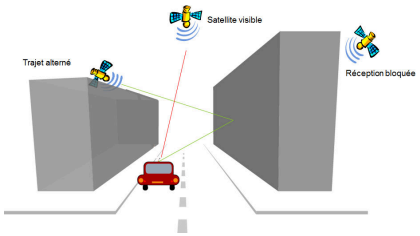


FIGURE 4 – Etats de réception des signaux satellitaires

2.7 Conclusions

PREDISSAT a été éprouvé dans le cadre de deux projets :

- le projet LOCOPROL du $5^{ème}$ PCRD, dont le but était de développer un nouveau système de contrôle-commande ferroviaire, sûr et de faible coût, fondé sur l'utilisation du satellite pour sa fonction de localisation ; est visée ce faisant l'application aux lignes à faible densité de trafic [15].
- le projet TESS (2005) qui se proposait de montrer dans quelle mesure il était possible de développer des services innovants, en mobilité et dans le cadre des transports terrestres, grâce à des moyens de communication et de navigation par satellite.

Le concept a pu être validé. PREDISSAT a montré son intérêt, ses performances et ses limites. La communication réalisée autour de cet outil a aussi révélé un intérêt fort de partenaires tels que CONNEX ou l'UIC pour les études de disponibilité qu'il permettait de réaliser mais aussi dans l'objectif de la validation de Galileo en environnement réel, notamment ferroviaire. Un brevet a été déposé en septembre 2006 pour une version temps réel ainsi que les codes sources existants. L'outil PREDISSAT de 2002 était cependant lourd à mettre en œuvre et surtout son utilisation était jusque là limité à l'analyse statistique de la disponibilité de la constellation le long d'une trajectoire donnée. Son utilisation temps réel doit permettre d'améliorer la précision et la confiance dans l'information de localisation fournie. Et c'est ce que nous avons souhaité développer dans

le projet CAPLOC. L'une des pistes du projet est issue des perspectives présentées (cf. LUT) même si sa forme differera un peu.

3 Données issues de l'image

Cette section reprend des élements des publications [16] et [17].

3.1 Détection de la ligne d'horizon pour la classification des états de réception des satellites

3.1.1 Détection de la ligne d'horizon

Ce travail a fait l'objet du livrable 1.2.

La tâche 1 du projet vise à classer en temps-réel l'état de réception de chacun des satellites reçus (soit directement, soit après réflexion) ; sur la base d'un système mono-caméra, les pixels de l'image sont dans une première phase, étiquetés en deux classes, ciel et non-ciel, permettant ainsi, de déterminer si le satellite, dont la position est reprojetée dans l'image, est localisé dans une région de ciel ou une région de non-ciel. Dans ce dernier cas, les satellites peuvent être considérés comme reçus après réflexion. L'algorithmie associée pourra choisir d'exclure ou de pondérer l'impact de chacun de ces satellites dans le calcul de position. A ce jour, les travaux réalisés permettent d'identifier les satellites localisés dans ces deux régions (ciel et non-ciel), de détecter la ligne d'horizon et d'évaluer le pourcentage de ciel visible dans l'image.

Si l'on considère le premier point, la stratégie permettant d'identifier les satellites localisés dans une zone de ciel et une zone de non-ciel d'une image Fisheye est composée d'une étape d'acquisition de données, de cinq briques étapes de traitement d'images, et d'une brique étape d'évaluation. La première brique étape concerne la phase d'acquisition des données de localisation (à l'aide des récepteurs GNSS embarqués) et des images (à l'aide d'une caméra couleur à objectif fisheye installée sur le toit du véhicule). Pour la partie traitement d'images, nous commençons par une phase de simplification de l'image afin de réduire les effets de luminosité, ombres, etc. Une classification pixellaire est ensuite réalisée afin d'affecter à chaque pixel de l'image, une classe d'appartenance parmi les classes de ciel et de non-ciel. Une étape de post-traitement morphologique est ensuite effectuée pour optimiser la phase de classification ciel/non-ciel. A cette étape de la stratégie, une évaluation des résultats est réalisée par rapport à une vérité terrain (càd par rapport à une classification faite à la main par un expert, visant à identifier les régions de ciel des régions de non-ciel de l'image). Ceci permettra d'évaluer le pourcentage de pixels correctement classés par notre approche par rapport à une classification de référence. Il est donc important de signaler que les algorithmes ont été testés sur la totalité des images de la base et évalués par rapport à une vérité terrain. L'étape suivante consiste à repositionner les satellites dans l'image afin d'identifier ceux situés dans une zone de ciel (satellites Line-Of-Sight (LOS)) ou une zone de non-ciel (satellites Non-Line-Of-Sight (NLOS)). Les détails des traitements ont été publiés dans [17].

3.1.2 Calibrage et position des satellites

Une fois l'étape de classification pixellaire effectuée et la ligne d'horizon détectée, nous cherchons à déterminer le nombre de satellites LOS (resp. NLOS). Pour ce faire, nous projetons les satellites dans l'image classée et identifions les satellites se trouvant dans la région « ciel »(resp. dans la région « non-ciel »). Pour cette raison nous détaillons ci-dessous la méthodologie utilisée pour placer les satellites dans l'image. Celle-ci se décompose en deux sous-étapes. 1) l'acquisition des données ; 2) le calibrage et la mise en forme des données.

1. Acquisition des données. Le dispositif d'acquisition d'images est constitué d'une caméra mono-ccd couleur équipée d'un objectif fisheye fixée sur le toit d'un véhicule et orientée vers le ciel. La fréquence d'acquisition des images est de 27 images par seconde. En parallèle, les trames $NMEA$ issues d'un ensemble récepteur et antenne Septentrio sont enregistrées.

2. Calibrage et mise en forme des données images. Les images présentent un phénomène de distorsion. Cette aberration liée aux propriétés optiques des lentilles de l'objectif fisheye se caractérise par une déformation de l'image. Un calibrage, basé sur une mise en correspondance des points du monde réel avec les pixels de l'image est nécessaire pour corriger cette déformation. Il a été réalisé par David Nowak (stage encadré par Amaury Flancquart) au LEOST et repose sur un dispositif expérimental basé sur un système de mires.
3. Enfin, les trames *GPGGA* et *GPGSV* extraites des données *NMEA* nous informent sur la position du véhicule toutes les secondes, ainsi que sur les positions des satellites reçus en termes d'azimuth et d'élévation. Les informations temporelles associées à chaque système d'acquisition nous permettent de recaler les données de localisation, les images et de placer les satellites dans l'image.

La figure 5 illustre les résultats de ces traitements sur des images acquises à Belfort dans le cadre du projet. Pour chacune des deux images acquises en mobilité et présentées sur cette figure (images de gauche), nous montrons la position des satellites dans l'image (images du milieu) ainsi que la détection de la ligne d'horizon (images de droite). En ce qui concerne, la position des satellites (images du milieu), la zone bleue (respectivement marron) représente une zone de ciel (respectivement de non-ciel). Les satellites numérotés en vert sont reçus en direct (LOS). Ceux numérotés en bleu ou rouge sont soit pas reçus du tout ou reçus après réflexion (NLOS).

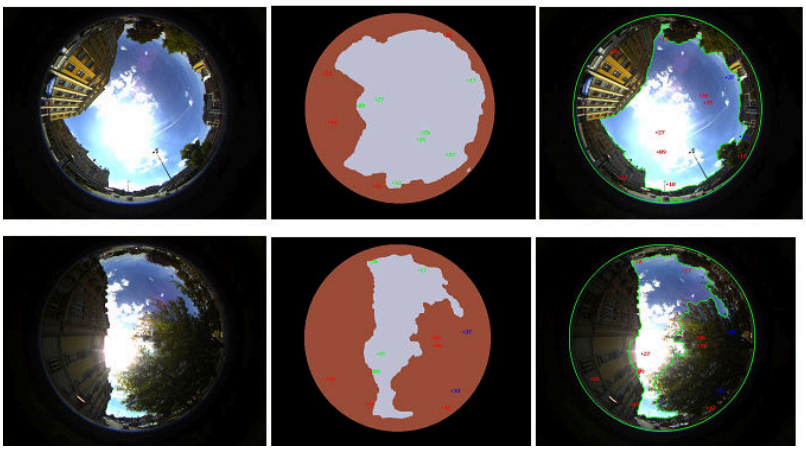


FIGURE 5 – détection du ciel et de la ligne d'horizon (de gauche à droite : l'image acquise en mobilité, la détection de ciel avec le repositionnement des satellites, la détection de la ligne d'horizon)

La sortie de ce travail est ainsi un tableau contenant, pour chaque instant t de mesure, la liste des satellites et leur état de réception.

4 Amélioration de la précision et premières pistes pour l'estimation de la confiance dans la position

La finalité de l'utilisation de l'image dans CAPLOC est l'amélioration de la précision de la localisation dans les environnements fortement perturbés par les multitrajets. Les résultats présentés dans ce chapitre utilisent le résultat du traitement des images fish-eye qui consiste à classer les satellites reçus en LOS (réception directe) et NLOS (réception après réfléxion). Dans ce rapport nous présentons les résultats obtenus à ce jour. Le rapport sera enrichi lors de sa version finale prévue en fin de projet.

4.1 Introduction

Il existe de nombreuses méthodes de mitigation des multitrajets tout au long de la chaîne de réception et de traitement des signaux GNSS [18] : traitement d'antennes, modélisation du canal de propagation, développement de nouveaux corrélateurs, algorithmes de navigation prenant en compte les bruits induits, ajouts de capteurs supplémentaires tels que capteurs inertiels ou même cartographies numériques. Dans le projet CAPLOC, le choix de la méthode s'est porté sur la détection et l'exclusion des satellites potentiellement perturbateurs de la qualité de la mesure.

Dans le projet CAPLOC nous avons évoqué plusieurs pistes de travail :

1. L'exclusion de certains satellites permet d'améliorer la précision et la confiance dans la mesure,
2. l'estimation de la disponibilité permet d'avancer une estimation de la confiance dans la position calculée,
3. la connaissance de l'état de réception des signaux permet d'appliquer des algorithmes de filtrage « intelligents »[19] et améliore la précision grâce à une modélisation fine des bruits de mesure.

A ce stade du projet, la piste la plus aboutie a été réalisée dans le cadre de la thèse de Dhouha Attia et consiste, comme [8], à exclure les satellites classés en zone de non-ciel donc considérés en NLOS. La détection étant effectuée grâce au traitement d'images, nous montrons dans ce chapitre les résultats de l'exclusion.

Cette section est largement rédigée à partir du chapitre 5 de la thèse de D. Attia [20] (soutenue le 3 octobre 2013).

4.2 Données expérimentales utilisées

Le système d'acquisition des données utilisé est composé de différents capteurs vidéo et GNSS comme le montre la figure 6 qui sont :

1. un récepteur GPS classique de marque uBlox, implanté dans un équipement Safedrive,
2. un récepteur GPS/EGNOS de marque Septentrio, plus haut en gamme,
3. un récepteur GNSS-RTK. Le GPS RTK est un système de positionnement précis, que nous utilisons comme source des positions de référence.
4. Une caméra mono CCD couleur équipée d'un objectif à grand angle (fisheye)(figure 6(a)).

Ce système complet a été installé sur un véhicule du laboratoire pour réaliser l'acquisition de ces données en mobilité. La disposition de ces équipements est illustrée par les figures 6(a) et 6(b) décrivant les véhicules de l'UTBM/IRTES-SET et celui du laboratoire Ifsttar-LEOST.



(a) Système d'acquisition du véhicule de l'IRTES-SET



(b) Système d'acquisition du véhicule de l'Ifsttar-LEOST

FIGURE 6 – Plateformes d'acquisition réalisées dans le projet CAPLOC.

L'acquisition réalisée en Juin 2010 a permis d'avoir au total 1885 positions estimées, soit une durée de 31 min 25 sec à raison d'une position par seconde. La zone d'intérêt retenue dans ce rapport est l'intervalle de temps $[13:15:46, 13:16:56]$ soit 71 données GPS. L'enregistrement des images a été effectué à 27 Hz et les trames $NMEA$ sont cadencées à 1 Hz, soit plusieurs images par donnée GPS. Afin de réduire le temps de traitement, un échantillonnage de la séquence d'images a été effectué qui consiste à ne garder qu'une image sur 10 (la première de chaque dizaine). Au final, pour chaque instant, la base de données est constituée de 2 à 3 images par seconde. Le recours à cet échantillonnage est justifié par la faible variation de la répartition des satellites entre les images successives acquises pendant une seconde.

La figure 7 illustre la trajectoire de référence obtenue à l'aide du récepteur GPS-RTK (en rouge). Nous avons exploité une portion de trajectoire de 171 points RTK. La trajectoire en bleue est constituée des positions estimées à partir des signaux reçus depuis tous les satellites disponibles en utilisant une technique basée sur le filtre de KALMAN étendu implémenté par N. VIANDIER [3]. Les pseudo-distances sont fournies par l'équipement Safedrive et utilisées dans le filtre. La trajectoire entière est formée de deux portions quasiment perpendiculaires reliées par un virage. La première ligne droite du trajet est longue d'environ 81 mètres et comprend 32 points de mesure, soit une vitesse moyenne inférieure à 10 km/heure, qui commence par un temps d'arrêt identifiable parce qu'il forme un amas de points. Sur la deuxième ligne droite, on compte 166 mètres et 26 points soit une vitesse moyenne de 23 km/heure.

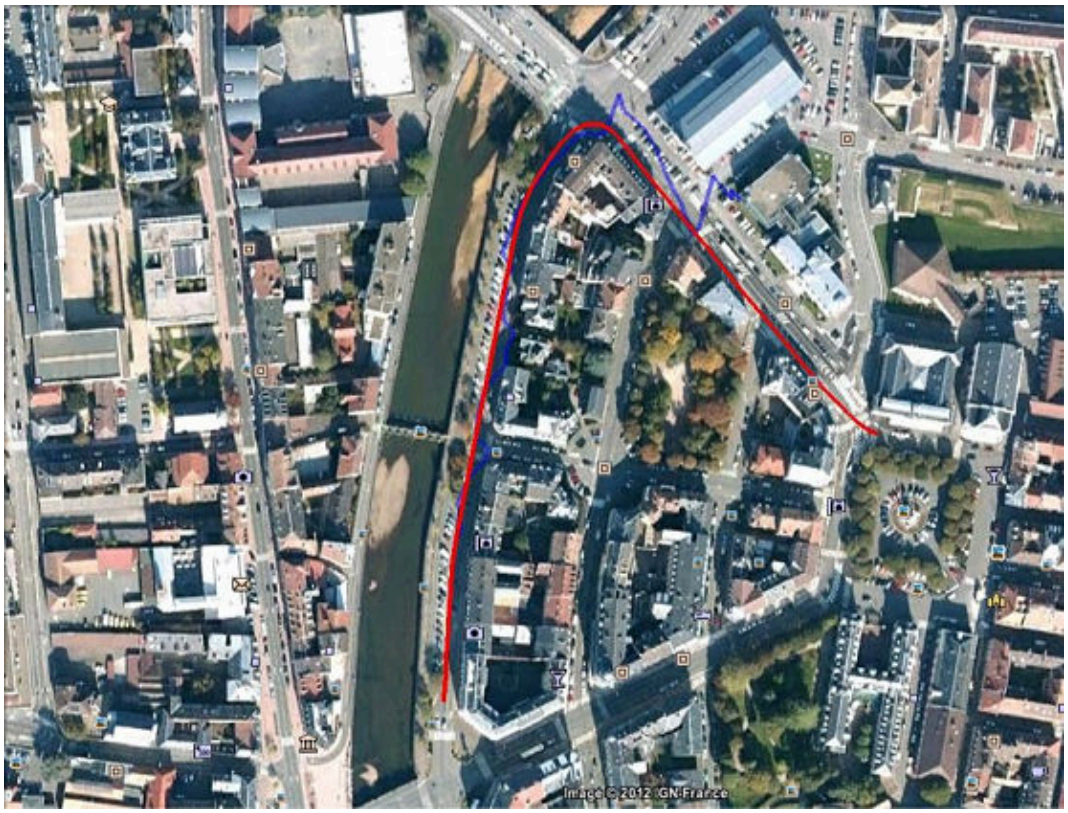


FIGURE 7 – Trajectoire étudiée : positions de référence et positions estimées.

4.3 Exclusion des satellites NLOS dans le calcul de la position

Dans le but d'améliorer la position calculée à partir des signaux disponibles, nous appliquons une politique d'exclusion des satellites dont les mesures peuvent être considérées comme dégradées. Sont exclus les satellites se trouvant dans les régions « non-ciel »afin d'éliminer toutes sources de multitrajets. Dans les figures qui suivent, la trajectoire de référence est toujours tracée en rouge et permet d'évaluer le gain des différentes solutions testées. La courbe bleue nous sert de référence par rapport à une solution classique puisqu'elle représente la solution utilisant tous les satellites disponibles.

Comme décrit plus haut, plusieurs images peuvent être disponibles pour un même point GPS. Nous montrons ici les résultats obtenus lorsque la première image de chaque instant t est utilisée pour la classification des satellites.

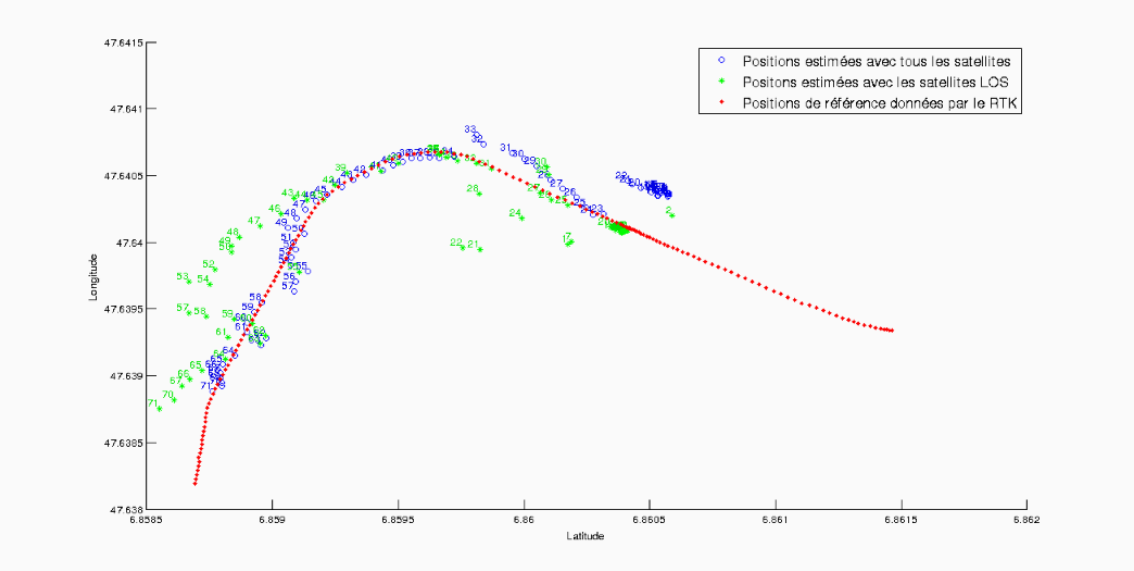


FIGURE 8 – Trajectoires obtenues avec et sans les satellites NLOS pour le calcul des positions.

Figure 8, on observe qu'en éliminant les satellites NLOS, la précision est améliorée sur la première moitiée du trajet ainsi que dans le virage où les positions estimées sont visiblement plus proches de la trajectoire de référence. Son apport reste cependant limité sur la deuxième partie de la trajectoire (la fin de courbe verte). Nous pouvons expliquer ceci par l'impact de la géométrie des satellites utilisés lors du calcul de la nouvelle position. Pour illustrer la relation entre la géométrie des satellites et la précision de la position estimée, nous utilisons le GDOP dont l'évolution est représentée figure 9. Rappellons qu'une valeur de GDOP caractérisant une bonne géométrie est classiquement considérée inférieure à 3. A l'exception de quelques points de GDOP largement supérieur à 2, l'ensemble des points calculés avant $t = 40s$ bénéficient d'une bonne géométrie, même lorsque les satellites NLOS sont exclus (points de couleur jaune et verte). La situation est clairement dégradée sur la période $t = [40, 60]$ et l'impact sur la trajectoire est directement visible. Cette configuration montre les limites de notre stratégie : les satellites disponibles, même s'ils sont directs, ne sont pas disposés de façon optimale et la position est plus imprécise que lorsque tous les satellites sont utilisés.

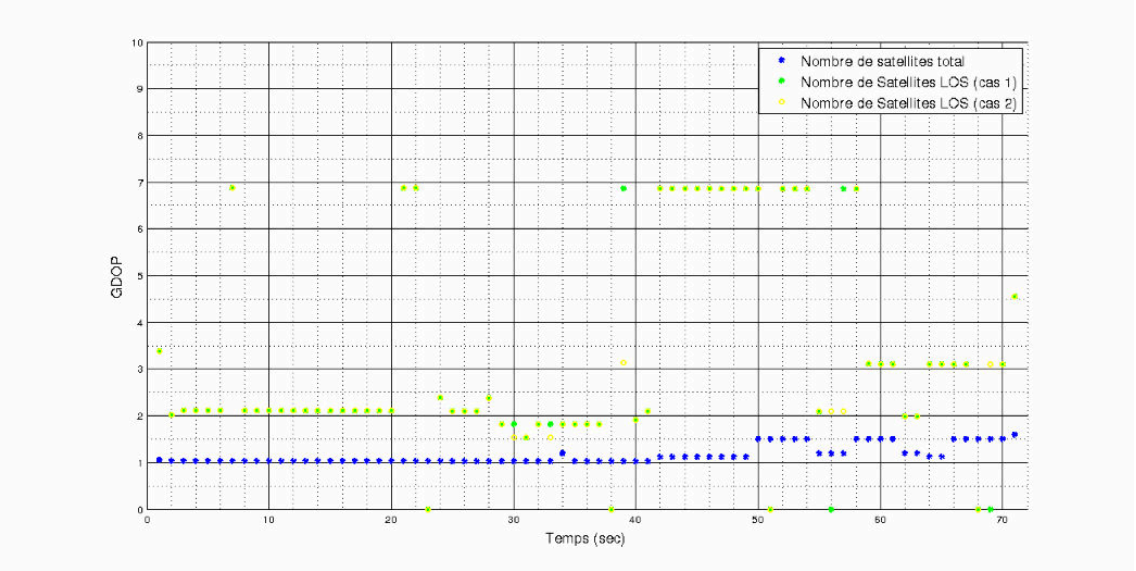
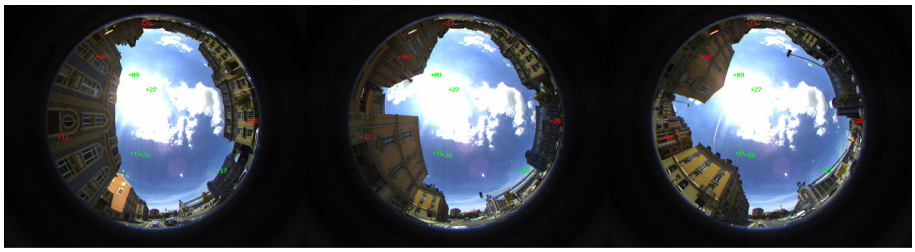


FIGURE 9 – L'évolution dans le temps du DOP des satellites LOS et de tous satellites. Les cas 1 et 2 diffèrent légèrement du fait du choix des images utilisées - les résultats sont décrits dans la thèse de D. Attia

Si l'on observe la configuration des environnements des deux moitié de la trajectoire, on constate que la première moitié de la trajectoire est orientée vers le nord-ouest avec des obstacles symétriques des deux côtés (cf. figures 10(a)). Dans cette première partie, le masque laisse des satellites visibles de part et d'autre de l'antenne de réception. La deuxième partie du trajet, orientée vers le sud-ouest, est bordée à gauche par des immeubles et à droite par un quai longeant la rivière La Savoureuse et parfois masqué par de la végétation (cf. figures 10(b)). La géométrie du masque est ici très déséquilibrée.



(a) Sélection d'images de la première partie du trajet.



(b) Sélection d'images de la deuxième partie du trajet.

FIGURE 10 – Illustration de la répartition géométrique des satellites sur les deux parties du trajet.

4.4 Utilisation combinée du DOP et de l'exclusion

Pour appliquer chaque stratégie de façon optimale, l'approche choisie consiste à alterner entre les deux solutions, c'est-à-dire à choisir la politique d'exclusion des satellites NLOS lorsque le DOP reste suffisamment bon (soit inférieur à un seuil choisi empiriquement - ici 3) et à garder tous les satellites dans le cas contraire.

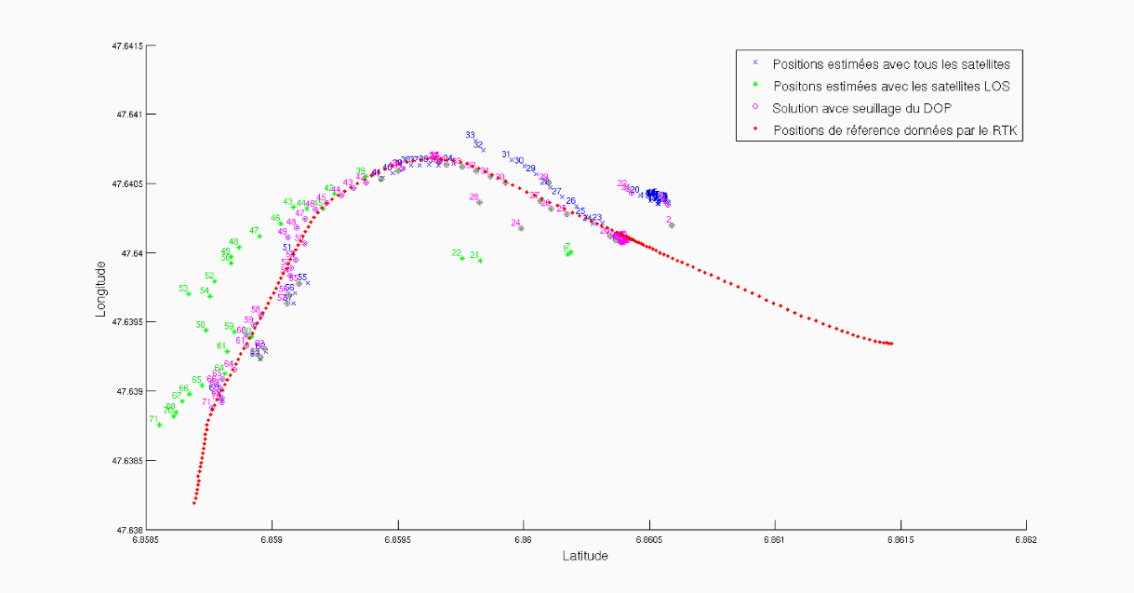


FIGURE 11 – Comparaison entre les trois solutions de calcul de position : la trajectoire rouge est la référence, la trajectoire bleue est la solution qui utilise tous les satellites, la trajectoire verte est la solution qui n'utilise que les satellites LOS, la trajectoire violette est la solution combinée.

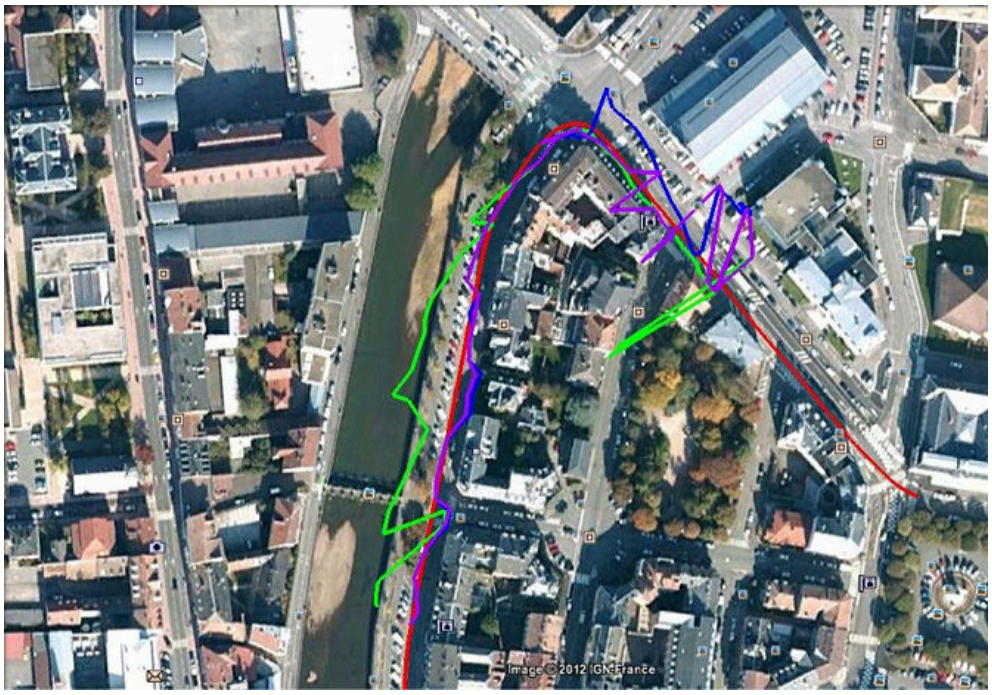


FIGURE 12 – Vue aérienne (Google Earth) des trajectoires obtenues avec les trois solutions (en rouge : la référence, en bleu : tous les satellites utilisés, en vert : les satellites LOS uniquement, en violet : la solution combinée.

La figure 11 montre l'apport de la solution combinée par rapport aux solutions LOS uniquement. Compte tenu de la similarité des approches cas 1 et cas 2, le deuxième cas (la stratégie optimiste) est la solution retenue pour la solution LOS. La figure 12 visualise ces même résultats à l'aide de l'outil Google Earth. Cette approche combinée montre bien l'utilisation de la stratégie la plus avantageuse dans chaque situation.

En perspective : On trouve dans la littérature des articles sur l'impact de la forêt sur la précision des récepteurs GPS [21]. Ils s'adressent à la communauté des utilisateurs forestiers surtout et ne traitent pas directement des retards induits sur les pseudo-distances. L'étude de l'impact des obstacles végétaux sur la réception du signal (ou non) sera cependant abordée dans les mois qui viennent et fera l'objet de résultats dans le livrable final.

4.5 Analyse quantitative. Estimation de la précision d'une position calculée

Afin d'évaluer quantitivement les performances des solutions de calcul de la position proposées, il convient de calculer l'erreur de mesure en chaque point. La précision est calculée par rapport à la trajectoire de référence obtenue avec le récepteur RTK. Dans la campagne d'acquisition, les données acquises n'ont pas été synchrones. Comme la fréquence des points fournis par le RTK est importante, une projection orthogonale de chaque point sur la trajectoire a été appliquée pour estimer cette précision. Celle ci n'est mesurée que lorsque la position peut être calculée (c'est-à-dire s'il y a suffisamment de satellites LOS).

Nous représentons les erreurs sous la forme d'une fonction de répartition sur la figure 13 pour les trois solutions : avec tous les satellites, avec les satellites LOS et la solution combinée. Cette représentation met en évidence l'amélioration apporté epar les différentes méthodes sur la précision d'origine. On constate par exemple que la médiane de l'erreur passe de 5m avec tous les satellites à 3 m avec le choix final soit une amélioration de 40% ou encore que 95% des positions sont inférieures à $24m$ contre 70% seulement des mesures utilisant tous les satellites.

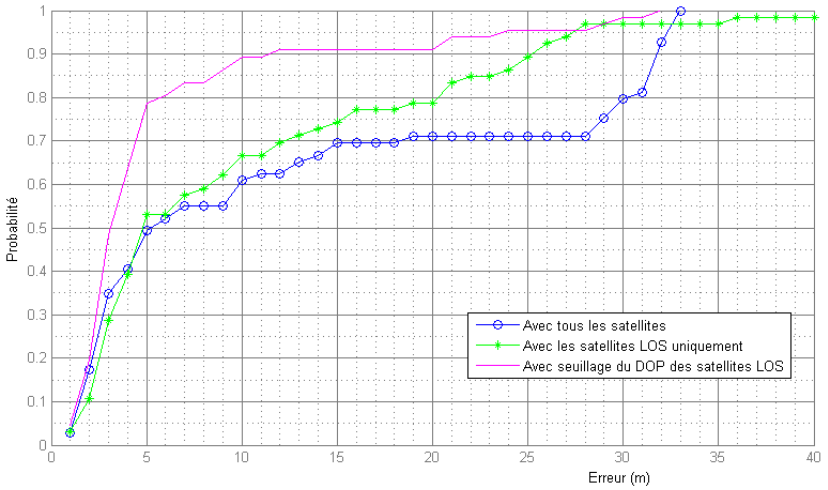


FIGURE 13 – Fonction de répartition de l'erreur de positionnement selon les trois méthodes.

Le tableau 1 reprend trois mesures calculées pour chacune des politiques testées : l'erreur moyenne, l'erreur maximale et l'écart type. Les politiques testée sont : la sélection des satellites LOS uniquement et la solution combinée DOP/tri. Ces résultats montrent que la stratégie basée sur le seuillage du DOP est toujours meilleure comparée aux deux autres stratégies (basée sur tous les satellites et basée sur l'exclusion des satellites NLOS). La précision des positions estimées est améliorée d'un facteur 2 en utilisant la politique du seuillage du DOP. D'autre part, quelle que soit la politique, l'intégration des satellites reçus à travers la végétation montre une augmentation de l'erreur moyenne. Par conséquent, cette mesure améliore la disponibilité mais dégrade légèrement les performances globales. Ainsi nous pouvons conclure que la classification binaire des images est suffisante à ce stade.

	Mesures d'erreur (m)		
Stratégies	Moyenne	Maximale	écart type
Tous les satellites	13.13	33.07	12.75
Tri de satellites LOS	10.5	59.23	12.14
Seuillage du DOP	5.73	32.92	7.66

TABLE 1 – Performance des différentes approches de calcul de position proposées.

4.6 Perspectives

D'ici la fin du projet (octobre 2014) d'autres pistes liées à la localisation seront avancées. Celles-ci sont en cours. Les résultats seront intégrés dans le rapport final de la tâche 3 prévu en fin de projet.

4.6.1 Pourcentage de ciel visible et performances

Le travail en cours consiste à relier le pourcentage de ciel visible et les performances de localisation. La disponibilité satellitaire (le nombre de satellites reçus à chaque instant) affecte les performances de localisation. En effet, moins le nombre de satellites reçus est élevé, plus les chances d'obtenir une bonne distribution géométrique est faible, et plus la précision est dégradée.

Sans compter bien sur les risques d'indisponibilité du service lorsque moins de 4 satellites peuvent être reçus. Par conséquent, il existe un lien entre la quantité de ciel visible et les performances obtenues. La difficulté est de mettre ce lien sous forme mathématique, dans la mesure où plusieurs facteurs interviennent : la quantité de ciel (ici transcrite sous forme de pourcentage de ciel visible dans l'image) mais aussi sa répartition spatiale. On a montré notamment qu'un masquage réparti de façon homogène autour de l'antenne affectait moins les performances qu'un masque très asymétrique (ex : toute une moitié de l'image).

Nous livrons ici les premières pistes de réflexion qui doivent être poursuivies et qui seront notamment éprouvées avec les mesures effectuées lors de la campagne d'acquisition réalisée du 14 au 18 octobre à Lille.

Sur la section expérimentale de Belfort, le parallèle pourcentage de ciel/DOP/imprécision ne permet pas d'établir de règle implicite. La figure 14 représente en parallèle le pourcentage de ciel visible, le nombre de satellites reçus en LOS et le nombre de satellites total. On peut quand même constater que les variations entre le pourcentage de ciel et le nombre de satellites LOS sont cohérentes. La figure 15 montre les variations entre le pourcentage de ciel, l'erreur de mesure et le HDOP lorsque tous les satellites sont utilisés ; La figure 16 montre les mêmes courbes mais lorsque seuls les satellites LOS sont exploités.

Les points où le HDOP est nul dans les courbes utilisant les satellites LOS seulement correspondent aux points ayant une visibilité de ciel inférieure à 30% de l'image. La disponibilité n'est plus suffisante pour calculer la position.

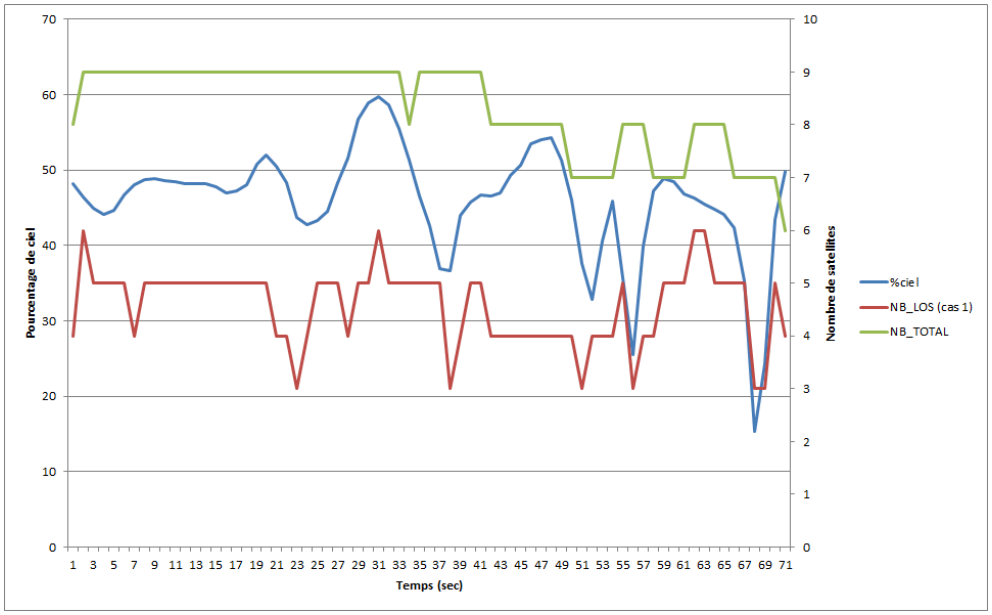


FIGURE 14 – Evolution du pourcentage de ciel visible, du nombre de satellites LOS et du nombre total de satellites reçus en fonction du temps.

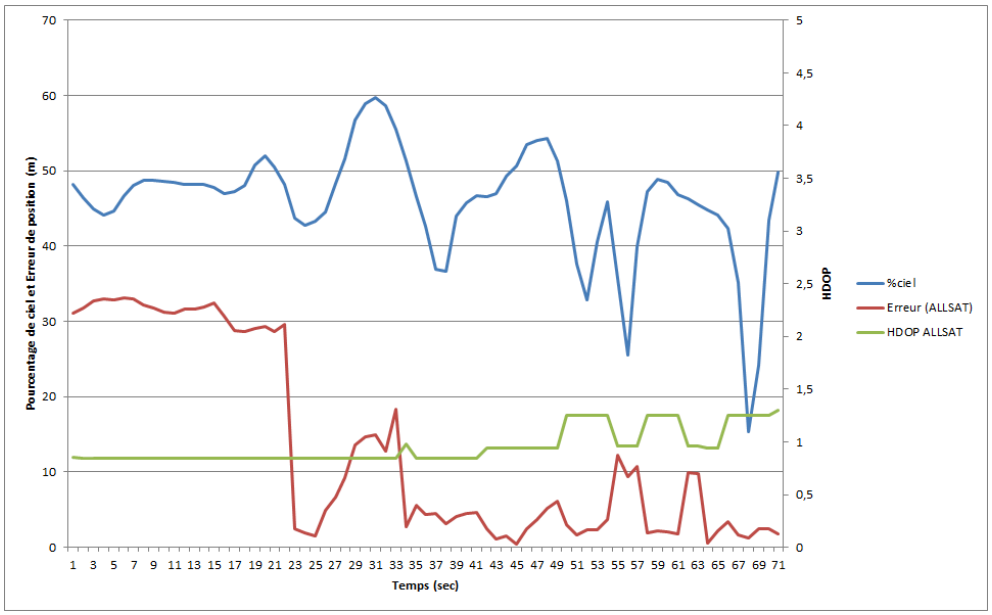


FIGURE 15 – Evolution du pourcentage de ciel visible, de l'erreur de précision et du HDOP obtenu lors de l'utilisation des satellites LOS en fonction du temps.

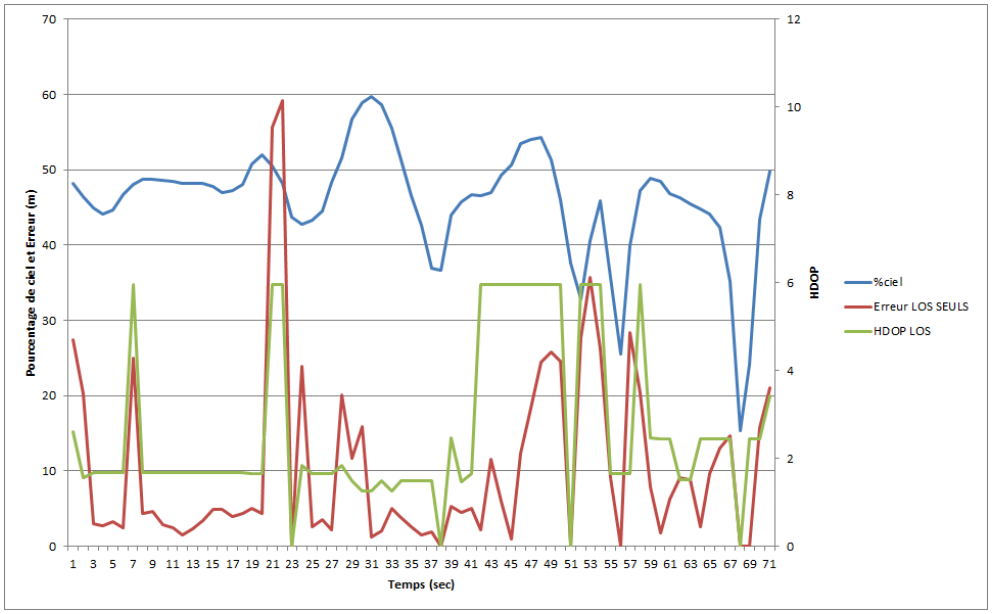


FIGURE 16 – Evolution du pourcentage de ciel visible, de l'erreur de précision et du HDOP obtenu lors de l'utilisation des satellites LOS en fonction du temps.

Sous l'hypothèse que les signaux sont reçus en LOS, l'équation 1 lie GDOP et précision de localisation [22].

$$drms = HDOP \times \sigma_{UERE} \tag{1}$$

où drms est l'erreur horizontale.

Dans [22], la valeur typique de σ_{UERE} est de $7.1m$. Si l'on calcule un sigma moyen ici à partir de $[\sigma = erreurhorizontale(t) \div HDOP(t)]$ $\sigma_{moyen} = 4,05m$ mais avec une variance de $12,5$. On trace sur la figure 17 l'erreur mesurée et la valeur de drms en fonction du temps. Les variations sont comparables mais l'ensemble de ces figures montrent qu'un nombre de critères

plus larges devra vraisemblablement être envisagé pour caractériser de façon correcte ces liens. D'autres essais sur des longueurs de trajet plus importantes seront également nécessaires.

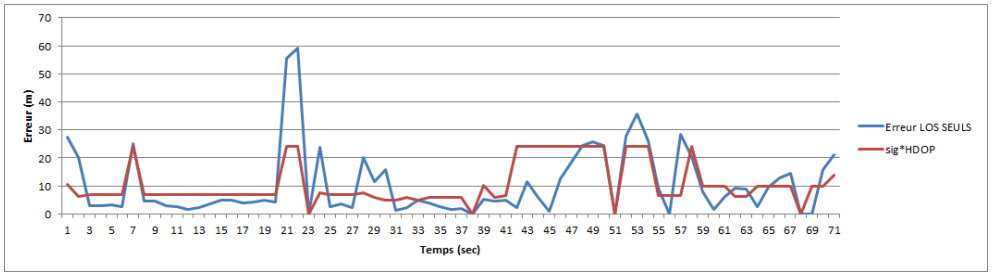


FIGURE 17 – Evolution de l'erreur mesurée en fonction du temps et de la valeur $drms = sigma_{moyen} \times HDOP$.

4.6.2 Combinaison état de réception et modèle d'erreur

Différentes méthodes de filtrage peuvent être appliquées pour estimer la position du récepteur. Dans les applications GNSS, le filtre de Kalman est souvent utilisé mais on trouve également les estimateurs par moindres carrés, et moindres carrés pondérés. L'intérêt des moindres carrés pondérés est qu'il permet d'ajuster le poids appporté par chacun des signaux en fonction de la confiance qui leur est accordée.

L'objectif des moindres carrés est de résoudre une équation du type $A.x = b$ qui n'a pas de solution exacte [23]. b est le vecteur d'observations (pseudo-distances), son nombre doit être supérieur ou égal au nombre d'inconnues (4). Il correspond au nombre de satellites reçus. L'estimation de la position revient à :

$$\Delta\hat{X} = (H^{t}.H)^{-1}.H^{t}.\Delta\rho \tag{2}$$

Pondérer consiste à introduire un poids P, qui est à estimer en fonction des mesures ou en fonction d'un critère de confiance sur la mesure, par exemple l'élévation des satellites ou le rapport signal sur bruit. L'équation 2 devient alors :

$$\Delta\hat{X} = (H^{t}.P.H)^{-1}.H^{t}.P.\Delta\rho \tag{3}$$

[24] utilise le modèle $SIGMA-\epsilon$ qui lie le rapport C/N_0 (valeur normalisée du rapport signal sur bruit SNR) à la variance des observations. Dans ce modèle, P est la matrice de variance covariance (VCM) des observations. L'auteur propose dans cet article une matrice VCM basée sur le rapport C/N_0.
Les résultats présentés dans la section 4.4 ont montré également l'importance du critère géométrique DOP dans le choix des satellites à utiliser. Par conséquent, la proposition d'une solution DOP-pondérée sera également suivie.

Enfin, nous souhaitons utiliser la connaissance nouvelle apportée par l'image, c'est à dire la connaissance déterministe de l'état de réception comme un critère de pondération également, dont la mise en oeuvre mathématique reste à définir.

Les résultats seront présentés dans le rapport final de la tâche localisation.

4.6.3 Utilisation du modèle 3D

La tâche 2 du projet consiste à réaliser un modèle 3D de l'environnement à partir des images. Il s'agit entre autres de l'objectif de la thèse de Julien Moreau réalisée dans le cadre du projet.

Ce modèle donnera les distances qui séparent le véhicule (précisément la position de la caméra) et les obstacles autour de l'antenne. Les premiers résultats concernant la modélisation 3D de l'environnement seront présentés lors de la conférence TRA 2014 à Paris [25].
Comme annoncé dans les objectifs du projet, l'utilisation de ce modèle doit permettre une utilisation constructive de la connaissance des biais induits par la réception en NLOS.

En suivant les pistes développées dans [12], la piste envisagée est de calculer le retard sur le signal propagé causé par la réflexion r (représenté figure 18) afin de corriger l'estimation de cette mesure ρ_{mesure} réalisée par le récepteur. Le calcul de la position pourra ainsi utiliser comme observation la valeur $\rho_{corrige}$:

$$\rho_{corrige} = \rho_{mesure} - r \tag{4}$$

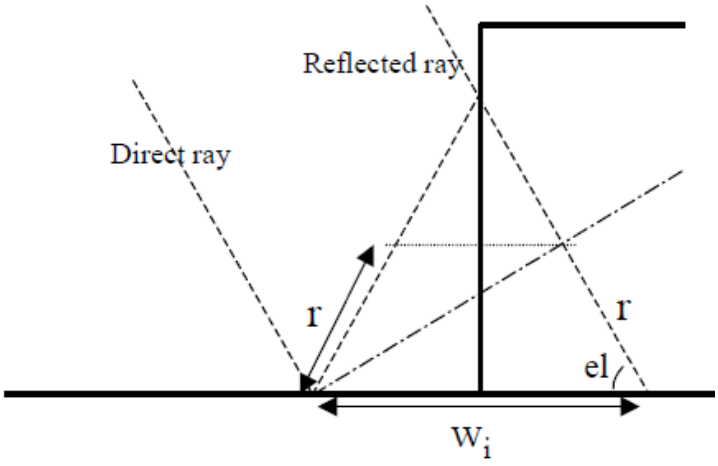


FIGURE 18 – Configuration d'un canyon urbain et retard r causé par la réflexion en 2D

La valeur du retard géométrique dépend à la fois de la configuration du canyon et de la position du satellite (azimuth et élévation). On peut alors écrire r selon l'équation 5 :

$$r = 2 \times d \times cos(el) \tag{5}$$

Où d est la distance le récepteur et la projection du point d'impact sur la surface du sol ; et el est l'élévation du satellite.

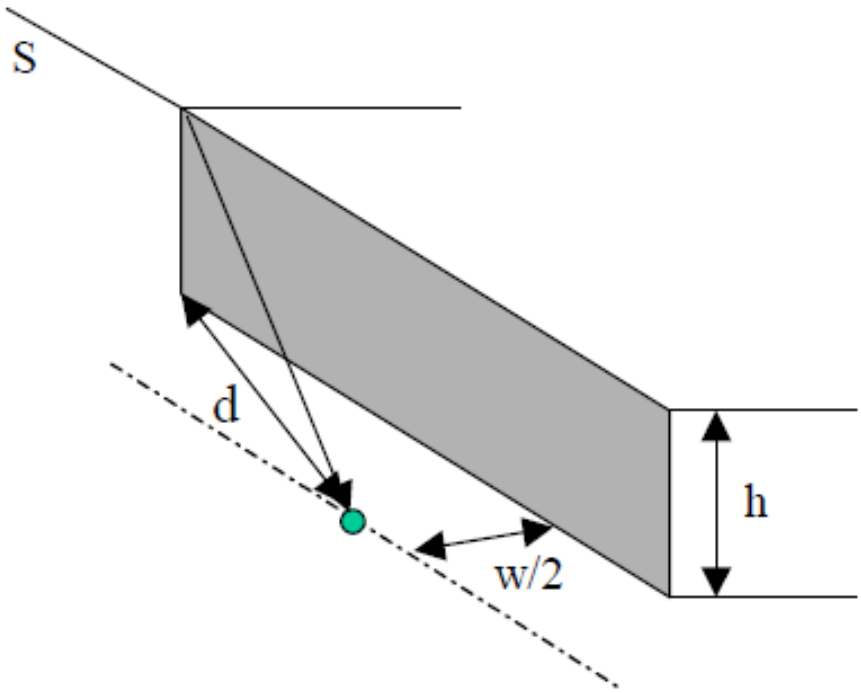


FIGURE 19 – Configuration d'un canyon urbain en 3D

Ces données sont d'autant plus précises que le modèle est précis.

Cet usage des modèles 3D a été récemment présenté sous différentes formes par [7] et [26]. Lors de la rédaction du projet, nous avons évoqué les pistes suivantes, que nous pourrons comparer aux techniques cités ci-dessus :

1. Réutiliser les travaux de Nicolas Viandier et Asma Rabaoui sur l'estimation de la distribution des erreurs de pseudodistances [27] pour les enrichir. L'estimation de l'erreur géométrique doit en effet permettre d'affiner cette estimation.
2. [28] montre qu'une version embarquée d'Ergospace est l'un des éléments qui permet une localisation pertinente dans le centre ville de Toulouse. La connaissance des biais devrait permettre de les compenser au moins en partie. Dans CAPLOC, nous exploiterons conjointement la pseudodistance brute fournie par le récepteur et le biais géométrique obtenu grâce au modèle pour recalculer la position.

Les résultats seront présentés dans les rapports finaux 2.2 et 3.2.

5 Conclusion et perspectives

Ce rapport a pour objectif de présenter les résultats intermédiaires obtenus dans la tâche 3, principalement sur les résultats des sous-tâche T3.1 et T3.2. La première partie présente le concept PREDISSAT et les résultats qui servent de socle au travail dans CAPLOC. Les données issues de l'image sont présentées dans une deuxième section mais ne font pas l'objet d'un large développement puisqu'il s'agit de résultats largement décrit dans le livrable 1.2. Les résultats les plus aboutis à ce stade du projet portent sur l'exclusion des satellites NLOS. Les résultats ont été testés sur un trajet expérimental dans la ville de Belfort. Une campagne de mesure a été réalisée mi-octobre à Lille et permettra de fournir davantage de résultats dans le rapport final.

Plusieurs travaux sont en cours sur la partie localisation du projet dont les principes ont été ébauchés dans ce document :

1. la pondération de l'usage des satellites compte tenu de la connaissance de leur état de réception
2. l'utilisation des données issues du modèle 3D.

Dans les perspectives non détaillées dans ce rapport restent également : le lien avec l'estimation dynamique des erreurs (d'après les travaux de [3]) et le travail sur l'évaluation de la confiance portée à la solution connaissant l'environnement de réception [1].

Depuis le dépôt du projet en 2009, on constate une accélération des travaux liés à l'usage de la connaissance de l'environnement de réception pour l'amélioration de la localisation par satellite. CAPLOC est la seule à utiliser la connaissance vidéo ; les autres embarquent des modèles 3D de l'environnement. On citera notamment [29] et [30] dont les travaux publiés, en parallèle à CAPLOC, présentent des résultats issus de l'exclusion de satellites NLOS. Des contacts ont été ébauchés M. Obst et plus avancés avec les auteurs de [30] pour évaluer les apports et inconvénients de ces méthodes. L'idéal serait de pouvoir comparer les résultats sur une même base de données mais cela fera peut être l'objet d'un travail hors CAPLOC.

1. Ces travaux ont pris un peu de retard en raison du congé maternité de J. Marais (Juin 2013-Décembre 2013) et devront faire l'objet d'un effort particulier en 2014.

Références

[1] J Marais : Predissat, comment évaluer la disponibilité satellitaire en environnement transport ? *Revue Transport et Sécurité, RTS*, 88:175–187, 2005. 5

[2] Juliette Marais : *Localisation de mobiles terrestres par satellites : mise en œuvre d'outils permettant l'analyse de l'influence des conditions de propagation et des effets de masques sur la disponibilité du service offert.* Thèse de doctorat, Université de Lille 1, 2002. 5

[3] Nicolas Viandier : *Modélisation et utilisation des erreurs de pseudodistances GNSS en environnement transport pour l'amélioration des performances de localisation.* Thèse de doctorat, Ecole Centrale de Lille, 2011. 5, 14, 25

[4] Juliette Marais, Sébastien Ambellouis, Amaury Flancquart, Sébastien Lefebvre, Cyril Meurie et Yassine Ruichek : Accurate localisation based on gnss and propagation knowledge for safe applications in guided transport. *Procedia - Social and Behavioral Sciences*, 48(0):796 – 805, 2012. Transport Research Arena 2012. 5

[5] J-I Meguro, Taishi Murata, J-I Takiguchi, Yoshiharu Amano et Takumi Hashizume : Gps accuracy improvement by satellite selection using omnidirectional infrared camera. *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 1804–1810. IEEE, 2008. 5

[6] Paul D Groves : Shadow matching : A new gnss positioning technique for urban canyons. *Journal of Navigation*, 64(03):417–430, 2011. 5

[7] Aude Bourdeau, Mohamed Sahmoudi et Jean-Yves Tourneret : Constructive use of gnss nlos-multipath : Augmenting the navigation kalman filter with a 3d model of the environment. *Information Fusion (FUSION), 2012 15th International Conference on*, pages 2271–2276. IEEE, 2012. 5, 24

[8] Marcus Obst, Sven Bauer et Gerd Wanielik : Urban multipath detection and mitigation with dynamic 3d maps for reliable land vehicle localization. *Position Location and Navigation Symposium (PLANS), 2012 IEEE/ION*, pages 685–691. IEEE, 2012. 5, 12

[9] Francois Peyret, David Bétaille et Florian Mougel : Non-line-of-sight gnss signal detection using an on-board 3d model of buildings. *ITS Telecommunications (ITST), 2011 11th International Conference on*, pages 280–286. IEEE, 2011. 5

[10] PP Robet, BG Evans et A Ekman : Land mobile satellite communication channel model for simultaneous transmission from a land mobile terminal via two separate satellites. *International journal of satellite communications*, 10(3):139–154, 1992. 6

[11] Lars Erling Bråten, César Amaya et David V Rogers : Statistical characterization of land mobile–satellite propagation environments using a photogrammetric technique. *Proc. URSI General Assembly 2002*, 2002. 6

[12] J Marais, M Berbineau et F Nahimana : Effect of alternate path on satellite positioning accuracy in an urban environment. *International Conference on Satellite and Space Communications, 2005. ICSCC 2005*, 2005. 6, 8, 23

[13] FP Fontan, MAV Castro, J Kunisch, J Pamp, E Zollinger, S Buonomo, P Baptista et B Arbesser : A versatile framework for a narrow-and wide-band statistical propagation model for the lms channel. *Broadcasting, IEEE Transactions on*, 43(4):431–458, 1997. 8

[14] D Zimmermann, J Baumann, A Layh, F Landstorfer, R Hoppe et G Wolfle : Database correlation for positioning of mobile terminals in cellular networks using wave propagation models. *Vehicular Technology Conference, 2004. VTC2004-Fall. 2004 IEEE 60th*, volume 7, pages 4682–4686. IEEE, 2004. 8

[15] A Simsky, F Wilms et JP Franckart : Gnss-based failsafe train positioning system for low-density traffic lines based on one dimensional positioning algorithm. *Proc. NAVITEC*, pages 8–10, 2004. 8

[16] Juliette Marais, Cyril Meurie, Sébastien Ambellouis et Yassine Ruichek : Projet caploc. combinaison de l'analyse d'images et la connaissance de la propagation des signaux pour la localisation. Paris, 30-31 janvier 2013. $40^{ème}$ congrès ATEXPO. 10

[17] Dhouha Attia, Cyril Meurie, Yassine Ruichek et Juliette Marais : Counting of satellites with direct gnss signals using fisheye camera : A comparison of clustering algorithms. Washington DC, USA, October 5-7, 2011. 11^{th} Intelligent Transport System Conference. 10

[18] Michael Lentmaier, Bernhard Krach et Patrick Robertson : Bayesian time delay estimation of gnss signals in dynamic multipath environments. *International Journal of Navigation and Observation*, 2008, 2008. 12

[19] N Viandier, DF Nahimana, J Marais et E Duflos : Gnss performance enhancement in urban environment based on pseudo-range error model. *Position, Location and Navigation Symposium, 2008 IEEE/ION*, pages 377–382. IEEE, 2008. 12

[20] Dhouha Attia : *Segmentation dŠimages par combinaison adaptative couleur/texture et classification de pixels. Application à la caractérisation de lŠenvironnement de réception des signaux GNSS.* Thèse de doctorat, Université de Belfort Montbéliard, 2013. 12

[21] Pete Bettinger et Krista L Merry : Influence of the juxtaposition of trees on consumer-grade gps position quality. *Mathematical and Computational Forestry & Natural-Resource Sciences (MCFNS)*, 4(2):Pages–81, 2012. 18

[22] Elliott D Kaplan et Christopher J Hegarty : *Understanding GPS : principles and applications.* Artech House Publishers, 2006. 21

[23] Kai Borre : *A software-defined GPS and Galileo receiver : a single-frequency approach.* Birkhauser, 2007. 22

[24] H. Hartinger et F. K. Brunner : Variances of gps phase observations : The sigma-e model. *GPS solutions*, 2(4):35–43, 1999. 22

[25] Julien Moreau, Sébastien Ambellouis et Yassine Ruichek : 3d modelling of urban environments for enhancing gps localization's accuracy. *Transport Research Arean 2014*, page 9p, 2014. 23

[26] Stéphan Miquel et Leila Fontenay : Amélioration du positionnement par satellites en environnement urbain. Paris, 30-31 janvier 2013. $40^{ème}$ congrès ATEXPO. 24

[27] Asma Rabaoui, Nicolas Viandier, Juliette Marais et Emmanuel Duflos : Using dirichlet process mixtures for the modelling of gnss pseudorange errors in urban canyon. *Proceedings of the 22nd International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS 2009)*, pages 2391–2399, 2009. 24

[28] Baptiste GODEFROY et Philippe SAYERSE : Amélioration du positionnement par satellites, dans un environnement fortement urbanisé, grâce à la prédiction de la propagation des signaux gnss. *Navigation*, 54(213):17–35, 2006. 24

[29] M Obst, C Adam, G Wanielik et R Schubert : Probabilistic multipath mitigation for gnss-based vehicle localization in urban areas. *ION GNSS Conference*, 2012. 25

[30] Sébastien Peyraud, David Bétaille, Stéphane Renault, Miguel Ortiz, Florian Mougel, Dominique Meizel et François Peyret : About non-line-of-sight satellite detection and exclusion in a 3d map-aided localization algorithm. *Sensors*, 13(1):829–847, 2013. 25

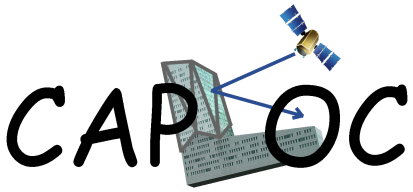
Groupe opérationnel n°2 du Predit 4

« Gestion du trafic et des mobilités »

CAPLOC

Combinaison de l'**A**nalyse d'images
et de la connaissance de la **P**ropagation
des signaux pour la **LOC**alisation

Convention 10-MT-CVS-001



Livrable 3.2 Rapport final de la tâche 3

18 Janvier 2015

Juliette Marais, Sébastien Ambellouis, Sarab Tay
Ifsttar
Julien Moreau
UTBM/Ifsttar

Identification

Acronyme du projet	CAPLOC
Titre du projet	Combinaison de l'Analyse d'images et de la connaissance de la Propagation des signaux pour la LOCalisation
Coordinatrice du projet	Juliette MARAIS (IFSTTAR/LEOST)
Date de début de projet	19 octobre 2010
Date de fin de projet	18 janvier 2015
Partenaires	IFSTTAR/LEOST, UTBM/IRTES-SET
Type de document	Livrable
Date contractuelle de livraison	18 janvier 2015
Accessibilité	La diffusion de ce document est limitée au programme PREDIT

Gestion des versions

Version	**Date**	**Auteurs**	**Modifications**
1.0	05/12/2015	Juliette Marais	Rédaction du corps du livrable et plan
2.0	17/01/2015	Sébastien Ambellouis	Rédaction de la section 4
3.0	18/01/2015	Juliette Marais	Approbation

Table des matières

1 Introduction **4**
1.1 Résumé du projet . . . 4
1.2 Objet de ce livrable . . . 4

2 Synthèse du livrable intermédiaire **5**
2.1 Le concept PREDISSAT . . . 5
2.1.1 Contexte . . . 5
2.1.2 Introduction à la localisation par satellites . . . 5
2.1.3 Méthodes de détection des multitrajets . . . 5
2.1.4 Utilisation de l'image : équipement et traitement . . . 6
2.1.5 Hypothèses liées à la propagation . . . 7
2.1.6 Résultats issus de PREDISSAT . . . 8
2.1.7 Conclusions . . . 8
2.2 Méthode de détection et d'exclusion des satellites NLOS . . . 9
2.3 Conclusions . . . 12

3 Utilisation de l'état de réception dans un calcul pondéré **13**
3.1 Principe des Moindres carrés pondérés . . . 13
3.1.1 Les modèles utilisant l'élévation . . . 13
3.1.2 Les modèles utilisant le rapport $C/N0$ . . . 14
3.1.3 Notre proposition : une combinaison du C/N_0, de l'élévation et de la connaissance de l'état de réception . . . 14
3.2 Application et résultats sur données réelles . . . 15
3.3 Conclusion . . . 18

4 Utilisation du modèle 3D **19**

5 Conclusion **23**
5.1 Perspectives . . . 23

Bibliographie **25**

Article présenté au 6th European Workshop on GNSS Signals and Signal Processing **26**

1 Introduction

1.1 Résumé du projet

CAPLOC est un projet de recherche amont qui apporte une brique technologique innovante de localisation précise. Il propose d'améliorer la précision d'un système de localisation grâce à la perception vidéo de l'environnement du véhicule et plus particulièrement l'environnement de propagation autour du récepteur GNSS. En effet, les problèmes de disponibilité et d'imprécision sont directement liés à la densité des obstacles notamment en environnement urbain dense. Il s'agit de réduire l'effet des conditions de propagation des signaux GNSS, notamment les multi-trajets, sans dégrader la disponibilité du service, ce qui est souvent le cas lorsque, classiquement, des procédures d'exclusion de défauts sont mises en œuvre.

Deux niveaux d'informations sont étudiés :

1. Le premier vise à indiquer en temps réel, l'état de réception de chacun des satellites, à partir de l'extraction du ciel par un système mono-caméra embarqué ;
2. Le deuxième vise à corriger l'erreur induite par la propagation sur la valeur du retard grâce à la construction d'un modèle 3D de l'environnement par un système multi-caméra.

Le projet CAPLOC s'articule autour de 3 tâches techniques. Les objectifs des deux premières tâches sont respectivement de développer une méthode de segmentation automatique du ciel et des bâtiments obstruant la réception directe des satellites et de construire un modèle 3D de l'environnement situé autour de l'antenne GNSS. La troisième tâche est dédiée à l'exploitation des sorties des deux tâches précédentes afin d'augmenter la précision de la localisation.

1.2 Objet de ce livrable

Ce rapport vise à faire un bilan d'avancement de la tache 3 et en particulier concerne la sous tâche 3.3 : Utilisation d'un modèle 3D de l'environnement. Il est composé de trois sections :

- la première reprend le concept PREDISSAT qui a servi de base à ces travaux et résume les résultats présentés dans le livrable intermédiaire 3.1.
- la deuxième présente l'utilisation de la connaissance de l'état de réception dans une solution de moindres carrés pondérés.
- la troisième illustre les résultats de la tâche 3.3 et montre comment les informations issues du modèle 3D pour l'amélioration de la précision.

Enfin la conclusion fera la synthèse et proposera des perspectives liées à la problématique de localisation précise.

2 Synthèse du livrable intermédiaire

2.1 Le concept PREDISSAT

Cette section reprend le descriptif publié dans [1] et repris en annexe.

2.1.1 Contexte

L'expérience du LEOST en matière d'utilisation de l'image pour la localisation par satellite remonte à une première étude réalisée dans le cadre du projet DEUFRAKO-P (1999) pour le compte de la SNCF qui souhaitait évaluer la disponibilité de la constallation GPS le long d'une ligne ferroviaire, suivie d'une thèse [2] cofinancée par l'INRETS et la Région Nord-Pas-de-Calais.

2.1.2 Introduction à la localisation par satellites

Les systèmes de localisation par satellites (GNSS - Global Satellite Navigation System) reposent sur le principe de la mesure du temps de propagation des signaux émis par les satellites de la constellation. Les satellites émettent en continu des signaux que le récepteur utilise pour extraire leur temps de parcours et un certain nombre d'information utiles contenues dans le message de navigation. Ce temps de parcours, multiplié par la vitesse de propagation des ondes, permet d'estimer la distance entre chacun des satellites et le récepteur. On appelle cette mesure la pseudo-distance. La position du récepteur est calculée par le récepteur lui-même à partir de la réception de 4 signaux simultanément, au moins par un principe de trilatération. Mathématiquement, il s'agit de résoudre un système d'équations à 4 inconnues que sont la position du récepteur x,y,z et le décalage d'horloge δt.

En environnement transport, la réception des signaux satellitaires est fréquemment perturbée par les obstacles environnant le véhicule. En effet, la disponibilité des systèmes de localisation des mobiles terrestres utilisant des signaux issus de satellites est considérablement réduite lorsque le canal de propagation est obstrué par la présence de masques, comme cela arrive en milieu urbain dense ou ferroviaire. La précision de la localisation obtenue est également directement liée à la qualité des mesures réalisées. Deux facteurs liés à l'environnement sont à mentionner particulièrement :

- La géométrie des satellites utilisés, caractérisée par le DOP (Dilution of Precision) [3].
- L'imprécision de la mesure de pseudo-distance causée par les multitrajets.

2.1.3 Méthodes de détection des multitrajets

On trouve dans la littérature plusieurs approches de détection des satellites reçus par multitrajets. Les multitrajets reçus en l'absence de trajet direct sont qualifiés de NLOS (Non Line Of Sight). En environnement contraint, tel qu'en urbain, les principaux perturbateurs sont les obstacles proches de l'antenne de réception (« proche »étant relatif face aux 20000km d'altitude des satellites). L'impact de ces obstacles est directement visible sur l'amplitude des signaux reçus, répercutée sur la valeur du rapport signal sur bruit (CN0). Dans les travaux de [3], une solution basée sur le seuillage de l'amplitude du signal GNSS reçu a été déjà proposée, ainsi que dans [4]. La difficulté de la détection des satellites LOS/NLOS sur ce critère réside dans le choix de la valeur du CN0 seuil. Ce seuil pourra être sensiblement différent d'un équipement à l'autre. De plus, il est fixé à partir de données statistiques et la dispersion des valeurs autour de ce seuil implique de mauvaises détections. C'est pourquoi d'autres techniques sont aujourd'hui étudiées. L'outil PREDISSAT du LEOST a été développé dans cet objectif en 2002 [2] et était fondé sur l'utilisation de caméras classiques et d'un processus de post-traitement. [5] a repris le concept avec une caméra Infra Rouge. Avec le développement des modèles 3D urbains, [6], [7], [8] ou encore [9] ont publié en 2011-2012 des résultats montrant comment ces modèles pouvaient

également permettre de déterminer la présence ou non d'un trajet direct en comparant les positions des obstacles modélisés et les positions des satellites. L'approche fondée sur l'utilisation de l'image est poursuivie dans CAPLOC et décrite ci dessous.

2.1.4 Utilisation de l'image : équipement et traitement

Le cas le plus favorable au sens de la qualité et du niveau de signal reçu est le cas où le champ électrique est reçu directement, c'est-à-dire lorsque le trajet émetteur-récepteur se fait au-dessus de la ligne de visibilité directe. En effet, lorsque les satellites sont masqués, ils sont soit non reçus, soit reçus par trajets multiples et donc susceptibles d'engendrer une erreur dans le calcul de la position. En nous inspirant des approches appliquées aux satellites de communication [10], [11], nous avons développé un outil fondé sur la connaissance vidéo de l'environnement.

Comme évoqué précédemment, l'originalité de l'outil réside dans l'utilisation de l'image pour analyser l'environnement autour de l'antenne afin de caractériser la qualité de la réception des signaux satellitaires. Afin d'obtenir une vision de l'environnement de l'antenne, le système expérimental testé est composé de deux caméras matricielles analogiques noir et blanc équipées d'un capteur CCD (capteur à transfert de charge). Ces caméras sont placées sur le toit du véhicule pour un trajet seulement, tel qu'illustré par la figure 1. Chaque caméra fournit vingt-cinq images par seconde. Cependant, la synchronisation de l'enregistrement des images avec le capteur odométrique nous permet d'enregistrer les images en fonction de la distance parcourue. Par exemple, nous enregistrerons une image tous les mètres. Cette information spatiale nous permet d'appliquer ensuite un procédé de stéréovision mono-caméra. Ce procédé consiste à considérer deux images successives comme deux images issues de deux caméras parallèles, s'inspirant ainsi des techniques de la stéréovision classique.

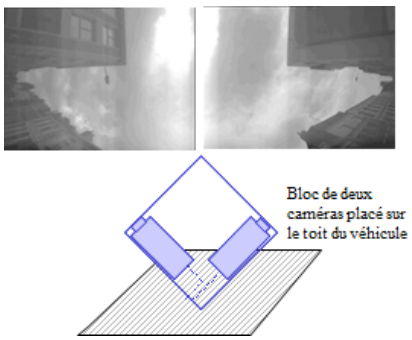


FIGURE 1 – Caméras placées sur le toit du véhicule

Nous pouvons ainsi calculer la distance de chacun des obstacles vus dans l'image, ainsi que leur hauteur. La connaissance de chacune des dimensions des obstacles nous permet alors de reconstituer l'environnement en trois dimensions et de considérer chacun des obstacles masquants autour de l'antenne, ce qui est nécessaire à la détermination des trajets réfléchis. Les zones situées à l'avant et à l'arrière du véhicule sont considérées comme dégagées de masques.

La reconstruction de l'environnement se fonde sur l'hypothèse que les obstacles sont situés le long de la trajectoire, parallèlement à la direction du véhicule. La configuration est donc de type canyon, tel qu'illustré sur la figure 2. En appliquant à ce modèle en trois dimensions les lois de la géométrie optique comme dans le cas de techniques de lancer de rayons, l'outil permet de déterminer non seulement les satellites reçus directement et les satellites bloqués, mais aussi les satellites qui peuvent être reçus après une réflexion [12].

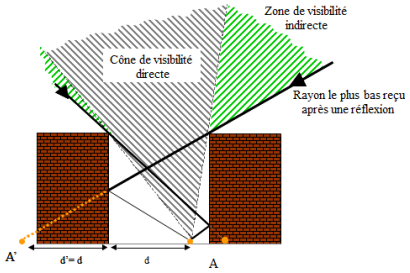


FIGURE 2 – Configuration de type canyon et lois de la géométrie optique

En raison des hypothèses géométriques appliquées, les résultats obtenus avec ce dispositif sont intéressants lorsque les obstacles sont modélisables par des plans verticaux et d'autant plus que leurs arêtes sont marquées. Dans les environnements montagneux par exemple, nos hypothèses sont trop éloignées de la réalité du terrain.

2.1.5 Hypothèses liées à la propagation

Pour se localiser, un récepteur utilise quatre (ou plus si possible) signaux issus de quatre satellites différents. La position est calculée à partir de la mesure du temps de propagation de chacun de ces signaux par triangulation. Aujourd'hui, la précision de l'information de localisation d'un système classique est de l'ordre de 10 m. Les principales causes des imprécisions relèvent d'une mauvaise estimation de la pseudo-distance : retards liés à la propagation des signaux dans l'atmosphère, erreurs de synchronisation des horloges, mauvaise répartition des satellites disponibles autour de l'antenne... Un certain nombre de modèles mathématiques ou d'équipements permettent de réduire ces erreurs. Cependant, l'environnement autour de l'antenne (immeubles, arbres, ponts...) peut également limiter la précision et la disponibilité. La réception des signaux satellitaires est classée en trois états, tel qu'illustré par la figure 3 :

- l'état V, lorsque le signal est reçu en visibilité directe ;
- l'état R, lorsque le signal est reçu par trajets multiples ; le signal subit réflexion et/ou diffraction sur des arbres et des obstacles environnants ; nous ferons ici l'hypothèse d'un trajet réfléchi au maximum une fois ;
- l'état B, en l'absence de signal lorsque des masques importants obstruent la visibilité.

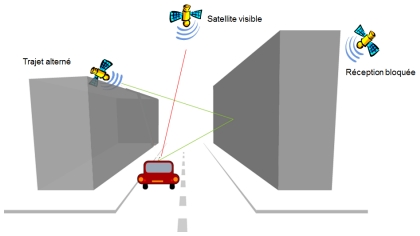


FIGURE 3 – Etats de réception des signaux satellitaires

Les conséquences seront de deux niveaux :

- lorsque moins de quatre satellites sont reçus dans l'état V ou R, le récepteur n'est pas en mesure de calculer sa position ;
- si une majorité de satellites est reçue dans l'état R, la position sera imprécise.

Les élévations et azimuts des satellites sont simulées sur PC afin de comparer élévation et azimut de chaque satellite avec l'élévation et l'azimut du masque issu de l'image et de son traitement.

2.1.6 Résultats issus de PREDISSAT

La comparaison des positions des satellites avec les positions des masques permet d'obtenir des résultats statistiques le long du trajet complet, ou localisés le long de la ligne. Les résultats statistiques offrent une vision globale de la couverture de la ligne. Les résultats localisés offrent une vision de la disponibilité du service en tout point de la ligne. On montre, pour un trajet donné, le nombre de satellites reçus en chaque point à un instant donné. Ces résultats observés sur une période de vingt-quatre heures offrent une information sur la couverture du système en chaque point de la ligne en termes de :

- gabarit, c'est-à-dire de nombre maximal et de nombre minimal de satellites reçus en chaque point ;
- probabilité qu'un nombre suffisant de satellites soient reçus en chaque point.

A l'issu des travaux de thèse, une utilisation envisagée était la création d'une base de données de prédictions (LUT pour look-up-table) inspirée de [13] en ce qui concerne le canal de propagation satellite et de [14] pour la localisation de mobiles GSM. La chaîne PREDISSAT représentée sur la figure 4 offre les informations relatives à l'état de réception de chacun des satellites en tout point de la trajectoire et à tout instant, ce qui permet ensuite au système de choisir les satellites utilisés en fonction de leur état ou de corriger les retards induits grace à un gabarit des retards, fondé sur la connaissance de la géométrie de l'environnement, en particulier des hauteurs des bâtiments et de la largeur de la rue [12]. Ces informations seront moins précises, mais plus faciles à manipuler.

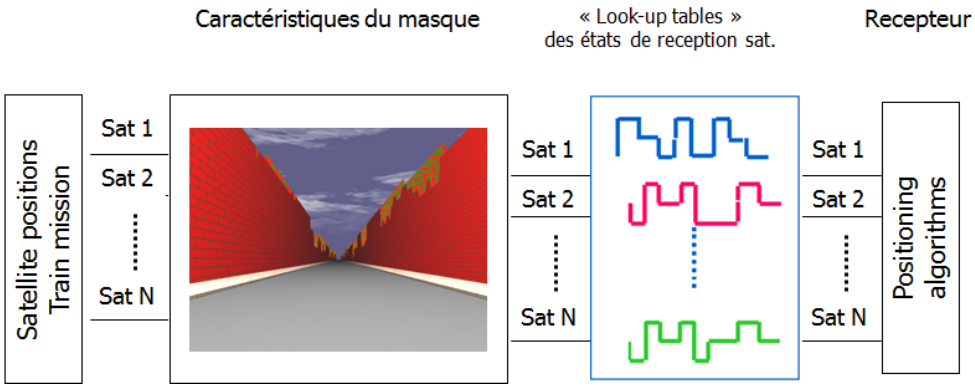


FIGURE 4 – Concept PREDISSAT

2.1.7 Conclusions

PREDISSAT a été éprouvé dans le cadre de deux projets :

- le projet LOCOPROL du $5^{ème}$ PCRD, dont le but était de développer un nouveau système de contrôle-commande ferroviaire, sûr et de faible coût, fondé sur l'utilisation du satellite pour sa fonction de localisation ; est visée ce faisant l'application aux lignes à faible densité de trafic [15].
- le projet TESS (2005) qui se proposait de montrer dans quelle mesure il était possible de développer des services innovants, en mobilité et dans le cadre des transports terrestres, grâce à des moyens de communication et de navigation par satellite.

Le concept a pu être validé. PREDISSAT a montré son intérêt, ses performances et ses limites. La communication réalisée autour de cet outil a aussi révélé un intérêt fort de partenaires tels que CONNEX ou l'UIC pour les études de disponibilité qu'il permettait de réaliser mais aussi dans l'objectif de la validation de Galileo en environnement réel, notamment ferroviaire. Un brevet a été déposé en septembre 2006 pour une version temps réel ainsi que les codes sources existants.

L'outil PREDISSAT de 2002 était cependant lourd à mettre en œuvre et surtout son utilisation était jusque là limité à l'analyse statistique de la disponibilité de la constellation le long d'une trajectoire donnée. Son utilisation temps réel doit permettre d'améliorer la précision et la confiance dans l'information de localisation fournie. Et c'est ce que nous avons souhaité développer dans le projet CAPLOC. L'une des pistes du projet est issue des perspectives présentées (cf. LUT) même si sa forme differera un peu.

2.2 Méthode de détection et d'exclusion des satellites NLOS

Il existe de nombreuses méthodes de mitigation des multitrajets tout au long de la chaîne de réception et de traitement des signaux GNSS [16] : traitement d'antennes, modélisation du canal de propagation, développement de nouveaux corrélateurs, algorithmes de navigation prenant en compte les bruits induits, ajouts de capteurs supplémentaires tels que capteurs inertiels ou même cartographies numériques. Dans le projet CAPLOC, le choix de la méthode s'est porté sur la détection et l'exclusion des satellites potentiellement perturbateurs de la qualité de la mesure.

Cette méthode a été réalisée dans le cadre de la thèse de Dhouha Attia et consiste, comme [8], à exclure les satellites classés en zone de non-ciel donc considérés en NLOS. La détection étant effectuée grâce au traitement d'images, nous montrons dans ce chapitre les résultats de la technique de détection et d'exclusion des satellites NLOS.
Cette section est rédigée à partir du chapitre 5 de la thèse de D. Attia [17] (soutenue le 3 octobre 2013).

Dans le but d'améliorer la position calculée à partir des signaux disponibles, nous appliquons une politique d'exclusion des satellites dont les mesures peuvent être considérées comme dégradées. Sont exclus les satellites se trouvant dans les régions « non-ciel »afin d'éliminer toutes sources de multitrajets. La technique est donc composée d'une première étape de segmentation de l'image en zones ciel et non ciel, suivie d'une étape de classificatioon des signaux reçus selon leur état de réception (dans la zone ciel : LOS, dans la zone non ciel : NLOS). La position est ensuite calculée à partir d'un filtre de Kalman étendu (EKF) classique implémenté par N. Viandier [3] avec les satellites LOS uniquement. La figure 5 illustre cette sélection.

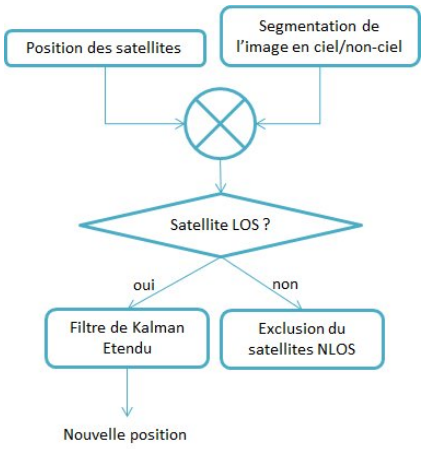


Figure 5 – Processus de sélection des satellites LOS

Dans les figures qui suivent, la trajectoire de référence, obtenue grâce à un récepteur RTK, est toujours tracée en rouge et permet d'évaluer le gain des différentes solutions testées. Nous avons appliqué cette méthode sur une portion de trajectoire de 171 points RTK dans le centre de la ville de Belfort. La trajectoire en bleue est constituée des positions estimées à partir des

signaux reçus depuis tous les satellites disponibles en utilisant une technique basée sur le filtre de KALMAN étendu. Les pseudo-distances sont fournies par l'équipement Safedrive et utilisées dans le filtre. La trajectoire entière est formée de deux portions quasiment perpendiculaires reliées par un virage. La première ligne droite du trajet est longue d'environ 81 mètres et comprend 32 points de mesure, soit une vitesse moyenne inférieure à 10 km/heure, qui commence par un temps d'arrêt identifiable parce qu'il forme un amas de points. Sur la deuxième ligne droite, on compte 166 mètres et 26 points soit une vitesse moyenne de 23 km/heure.

Les résultats de l'exclusion ont montré qu'en éliminant tous les satellites NLOS, la précision était améliorée sur la première moitiée du trajet ainsi que dans le virage où les positions estimées sont visiblement plus proches de la trajectoire de référence mais que la deuxième partie de la trajectoire était dégradée par rapport à la solution utilisant tous les satellites. Nous avons expliqué ceci par l'impact de la géométrie des satellites utilisés lors du calcul de la nouvelle position. Pour illustrer la relation entre la géométrie des satellites et la précision de la position estimée, nous utilisons le GDOP. Rappellons qu'une valeur de GDOP caractérisant une bonne géométrie est classiquement considérée inférieure à 3. A l'exception de quelques points de GDOP largement supérieur à 2, l'ensemble des points calculés avant $t = 40s$ bénéficient d'une bonne géométrie, même lorsque les satellites NLOS sont exclus (points de couleur jaune et verte). La situation est clairement dégradée sur la période $t = [40, 60]$ et l'impact sur la trajectoire est directement visible. Cette configuration montre les limites de notre stratégie : les satellites disponibles, même s'ils sont directs, ne sont pas disposés de façon optimale et la position est plus imprécise que lorsque tous les satellites sont utilisés.

Pour appliquer chaque stratégie de façon optimale, l'approche choisie consiste à alterner entre les deux solutions, c'est-à-dire à choisir la politique d'exclusion des satellites NLOS lorsque le DOP reste suffisamment bon (soit inférieur à un seuil choisi empiriquement - ici 3) et à garder tous les satellites dans le cas contraire. La figure 6 visualise l'ensemble de ces résultats : en rouge la trajectoire de référence ; en bleu la solution utilisant tous les satellites ; en vert les satellites directs uniquement et en violet la solution combinée.

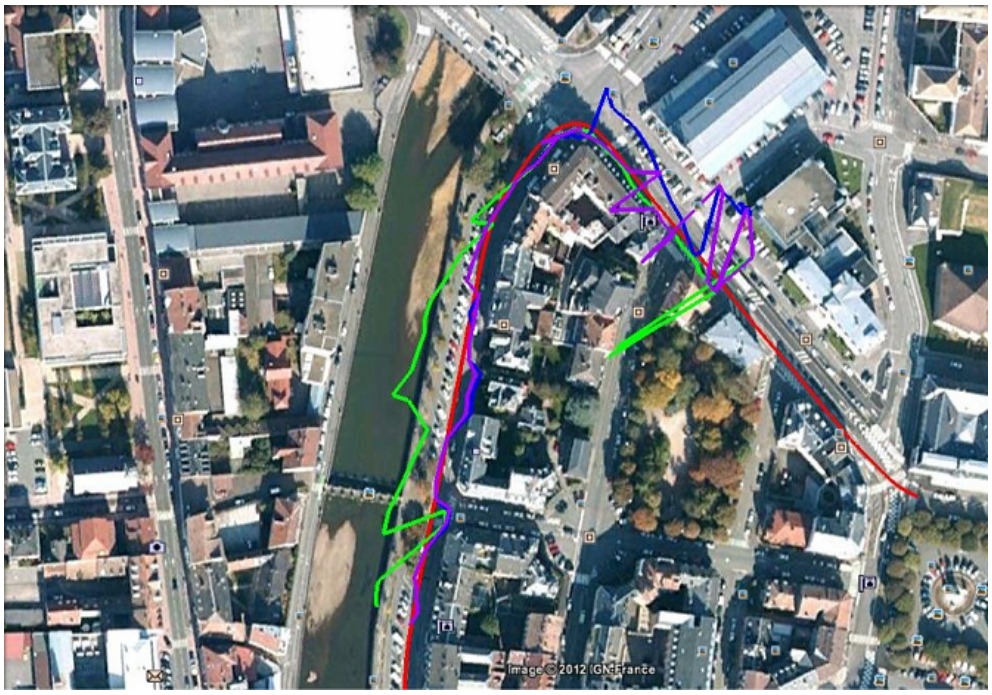


FIGURE 6 – Vue aérienne (Google Earth) des trajectoires obtenues avec les trois solutions (en rouge : la référence, en bleu : tous les satellites utilisés, en vert : les satellites LOS uniquement, en violet : la solution combinée.

L'erreur de localisation est calculée en projetant orthogonalement les points de mesures sur la trajectoire de référence RTK. Nous représentons ces erreurs sous la forme d'une fonction de répartition sur la figure 7 pour les trois solutions : avec tous les satellites, avec les satellites LOS et la solution combinée. Cette représentation met en évidence l'amélioration apporté epar les différentes méthodes sur la précision d'origine. On constate par exemple que la médiane de l'erreur passe de 5m avec tous les satellites à 3 m avec le choix final soit une amélioration de 40% ou encore que 95% des positions sont inférieures à $24m$ contre 70% seulement des mesures utilisant tous les satellites.

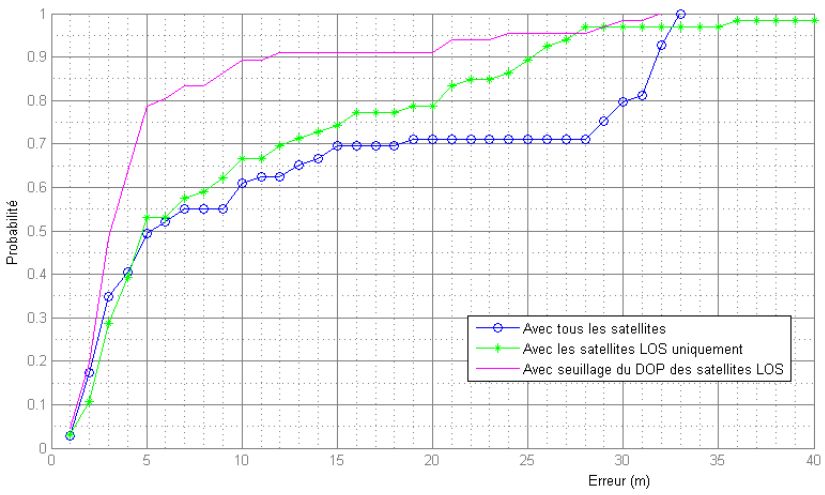


FIGURE 7 – Fonction de répartition de l'erreur de positionnement selon les trois méthodes.

Enfin, le tableau 1 reprend trois mesures calculées pour chacune des politiques testées : l'erreur moyenne, l'erreur maximale et l'écart type. Les politiques testée sont : la sélection des satellites LOS uniquement et la solution combinée DOP/tri. Ces résultats montrent que la stratégie basée sur le seuillage du DOP est toujours meilleure comparée aux deux autres stratégies (basée sur tous les satellites et basée sur l'exclusion des satellites NLOS). La précision des positions estimées est améliorée d'un facteur 2 en utilisant la politique du seuillage du DOP. D'autre part, quelle que soit la politique, l'intégration des satellites reçus à travers la végétation montre une augmentation de l'erreur moyenne. Par conséquent, cette mesure améliore la disponibilité mais dégrade légèrement les performances globales. Ainsi nous pouvons conclure que la classification binaire des images est suffisante à ce stade.

	Mesures d'erreur (m)		
Stratégies	Moyenne	Maximale	écart type
Tous les satellites	13.13	33.07	12.75
Tri de satellites LOS	10.5	59.23	12.14
Seuillage du DOP	5.73	32.92	7.66

TABLE 1 – Performance des différentes approches de calcul de position proposées.

2.3 Conclusions

L'approche présenté dans ce chapitre est celle détaillé dans la thèse de D. Attia [17] et propose d'appliquer un principe connu sous l'acronyme FDE (*Fault Detection and Exclusion*), famille de méthodes couramment appliquées dans l'aéronautique. Les résultats obtenus sur la trajectoire expérimentale de Belfort montre l'intérêt de cette méthode mais aussi ses limites dans les applicatiosn de transport terrestre et en particulier en circulation urbaine. En effet, la géométrie des satellites utilisables souffre de ce principe d'exclusion dans certains environnements et limite la disponibilité de service lorsque le nombre de satellites restant devient inférieur à 4. C'est le cas de 17 points sur les 71 mesures de notre trajectoire de test. Dans le livrable 3.1, une perspective liée à l'utilisation des résolutions de systèmes d'équations par moindres carrés pondérés avait été évoquée. Elle fait l'objet du deuxième chapitre de ce libvrable.

3 Utilisation de l'état de réception dans un calcul pondéré

Cette section est rédigée à partir de l'article [18]

3.1 Principe des Moindres carrés pondérés

Différentes méthodes de filtrage peuvent être appliquées pour estimer la position du récepteur. Dans les applications GNSS, la plupart des solutions utilisent le filtre de Kalman (le plus répandu) ou les algorithmes d'estimation par moindres carrés [19]. Schématiquement, le filtre de Kalman utilise une connaissance a priori et filtre (lisse) les mesures en utilisant la connaissance de l'état précédent pour calculer le nouvel état. L'intérêt des moindres carrés **pondérés** est qu'il permet d'ajuster le poids apporté par chacun des signaux en fonction de la confiance qui leur est accordée. En effet, il est intéressant d'accorder un poids plus fort à une observation précise pour qu'elle contribue davantage à la solution calculée que des mesures imprécises [20].

L'objectif des moindres carrés est de résoudre une équation du type $A.x = b$ qui n'a pas de solution exacte [21]. b est le vecteur d'observations (pseudo-distances), son nombre doit être supérieur ou égal au nombre d'inconnues (4). Il correspond au nombre de satellites reçus. L'estimation de la position revient à :

$$\Delta\hat{X} = (H^{t}.H)^{-1}.H^{t}.\Delta\rho \tag{1}$$

où $\Delta\hat{X}$ est la mise à jour de l'estimation de la position, H est la matrice d'observation et $\Delta\rho$ la mise à jour de la mesure.

Pondérer consiste à introduire un poids P En théorie, P est l'inverse de la matrice de variance-covariance des observations. En pratique, cette matrice est inconnue. Il faut donc soit l'estimer en fonction des mesures ou en fonction d'un critère de confiance sur la mesure, soit la remplacer par un modèle [3]. L'estimation par moindres carrés donnera un résultat précis si une matrice VCM correcte

Lorsque P est introduit dans l'équation 1, celle-ci devient alors :

$$\Delta\hat{X} = (H^{t}.P.H)^{-1}.H^{t}.P.\Delta\rho \tag{2}$$

où $P = \begin{pmatrix} w_1 & \cdots & 0 \\ \vdots & w_i & \vdots \\ 0 & \cdots & w_n \end{pmatrix}$ et, pour chaque satellite i, le poids $w_i = 1/\sigma_i^2$.

On trouve dans la littérature différents modèles de poids dont les plus courant utilisent des paramètres fonction de l'élévation et/ou du rapport $C/N0$.

3.1.1 Les modèles utilisant l'élévation

Le modèle le plus simple utilisant l'élévation exprime la variance comme l'inverse du sinus de l'élévation au carré [20], soit :

$$\sigma_i^2 = \sigma_0/sin(\theta_i)^2 \tag{3}$$

où θ_i représente l'élévation du satellite i et σ_0 est un paramètre du modèle qui dépend du type de récepteur et d'antenne et doit être déterminé à l'avance.

On trouvera également le modèle cité par [22] :

$$\sigma_i^2 = a + b \times \exp -\theta_i/\theta_0 \tag{4}$$

où θ_i représente l'élévation du satellite i et a, b et θ_0 sont des constantes du modèle.

3.1.2 Les modèles utilisant le rapport $C/N0$

D'autres auteurs, comme [23] considèrent qu'un modèle utilisant le rapport $C/N0$ permet de prendre en compte autant l'élévation que les conditions de propagation et proposent les modèles suivants. Le modèle $Sigma - \epsilon$ [20] utilise la valeur du rapport C/N_0 (valeur normalisée du rapport signal sur bruit SNR) des signaux GPS pour estimer les poids à appliquer dans les moindres carrés.

$$\sigma_i^2 = c^2.10^{-CN0_i/10} \tag{5}$$

où c est une constante du modèle.

Wieser [24] a proposé une formule qui exprime la variance de phase σ^2 comme une fonction des valeurs C/N_0 :

$$\sigma_i^2 = v_i + c_i.10^{-0.1\times CN0_i} \tag{6}$$

où v_i (en m^2) et c_i (en m^2Hz) sont les paramètres du modèle, estimés expérimentalement et i le satellite.

Wieser [24] cite également une méthode appelée $Sigma - \Delta$ qui nécessite l'estimation et la modélisation de l'effet de diffraction ou de multitrajets. L'application de cette méthode dans l'article montre qu'il est difficile d'obtenir ces valeurs Δ dans le cas des multitrajets et limite ainsi son intérêt dans notre étude.

3.1.3 Notre proposition : une combinaison du C/N_0, de l'élévation et de la connaissance de l'état de réception

Le modèle $Sigma-\epsilon$ ne tient pas compte de l'élévation des satellites utilisés (équation 5). [24] souligne, et illustre, l'évolution du rapport C/N_0 en fonction de l'élévation du satellite. Il s'agit alors de la mesure du niveau de signal d'un satellite donné, dans le temps, dans un environnement libre de tout obstacle. Dans ces conditions, le rapport C/N_0 d'un satellite augmente avec son élévation. Dans un environnement urbain, en raison des conditions de propagation locales, le lien entre C/N_0 et élévation est moins immédiat, comme l'illustre la figure 8 sur un exemple de mesures. On choisit cependant de conserver cette hypothèse, dans la mesure où un satellite d'élévation haute sera moins sujet aux masquages et multitrajets qu'un satellite d'élévation plus basse.

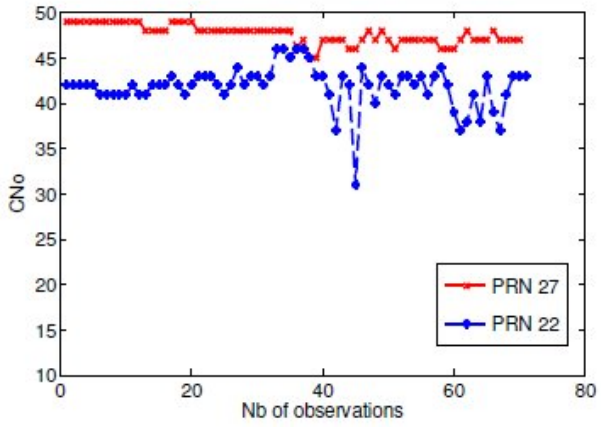


FIGURE 8 – Valeurs de CN0 en fonction du temps pour deux satellites à élévations différentes (PRN 22 à 17 degrés environ et PRN 27 à 70 degrés)

Pour appuyer l'effet de l'élévation sur le poids, nous intégrons dans notre modèle un facteur en $1/(sin\theta_i)^2$. Enfin, pour prendre en considération l'état de réception des satellites utilisés, donnée importante lorsque le véhicule circule en environnement urbain, nous proposons d'utiliser la connaissance déterministe de l'état de réception apportée par l'image comme un critère de pondération également.
Nous proposons d'adapter l'équation 5 en exprimant la variance comme suit :

$$\sigma_i^2 = k \times 10^{-0,1.CN0_i}/sin(\theta_i)^2 \tag{7}$$

avec k variant en fonction de l'état de réception du signal :
- $k = 1$ lorsque le satellite est reçu en trajet direct (LOS),
- $k = 2$ lorsque le satellite est en NLOS.

La valeur de k a été déterminée de façon expérimentale.

3.2 Application et résultats sur données réelles

Nous avons comparé notre modèle aux modèles de la littérature présentés ci-dessus :
- moindres carrés sans pondération,
- modèle d'élévation (éq. 3) avec $\sigma_0 = 1$,
- modèle d'élévation de Groves (éq. 4). Les paramètres appliqués sont les paramètres de [22] qui reprend les éléments de [25] : $a = 0,13m$, $b = 0,56m$, et $\theta_0 = 0,1745rad$,
- modèle $Sigma - \epsilon$ (éq. 5) avec $c = 1,1.10^4 m^2$ selon [26] et [22].

Notre modèle est appelé CN0/Elévation dans les courbes qui suivent. Trois valeurs de k ont été testées et une seule conservée pour la suite des résultats. Les méthodes ont été évaluées sur la base de données de mesures déjà utilisée dans la section 2.2 réalisée à Belfort.

La figure 9 compare les positions obtenues avec une résolution par moindres carrés classique (losanges bleus), et plusieurs solutions par moindres carrés pondérés utilisant le modèle d'élévation de l'équation 3(croix roses), celui de l'équation 4 (ronds bleus ciel), le modèle $Sigma - \epsilon$ (ronds roses), et notre proposition combinant état de réception, $CN0$ et élévation (étoiles rouges). Les performances en termes de précision sont comparées sur la figure 10. L'erreur est estimée par projection des points calculés sur la courbe de référence obtenue grâce aux mesures RTK (courbe verte).

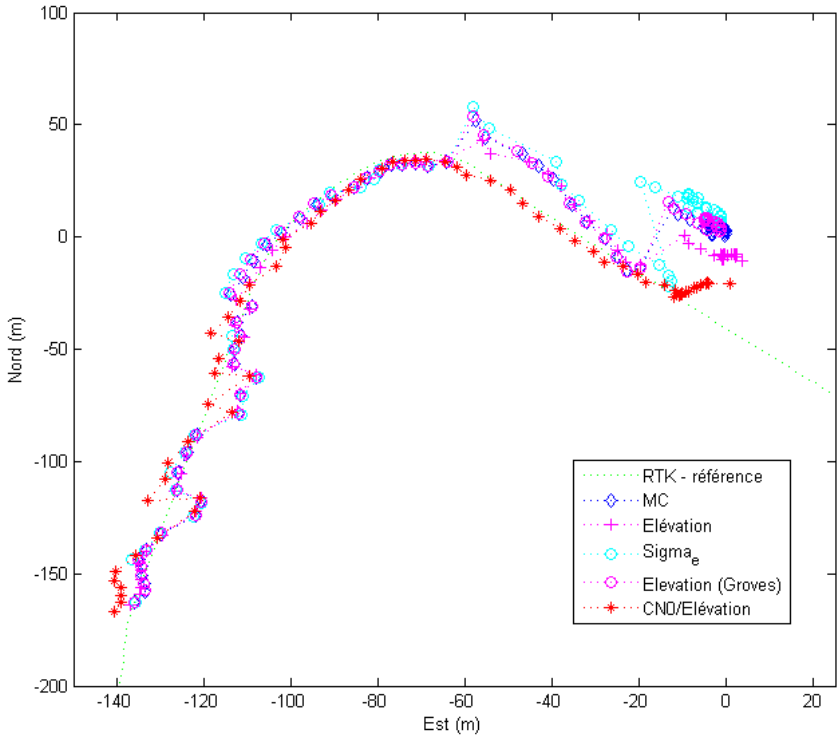


FIGURE 9 – Comparaison des positions obtenues avec les méthodes des moindres carrés ordinaires, les pondérations par des modèles d'élévation, Sigma-ϵ et par notre modèle

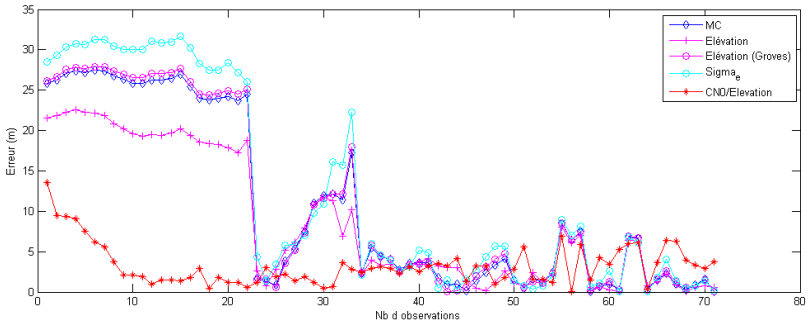


FIGURE 10 – Précisions obtenues avec les méthodes des moindres carrés ordinaires, les pondérations par des modèles d'élévation, Sigma-epsilon et par notre modèle

Les résultats présentent deux phases différentes dans la mesure. La première période de mesure ($t = 0$ à $25s$) est une zone de démarrage, dans une zone fortement bruitée en raison de l'environnement de réception et de l'état stationné du véhicule à $t = 0$. La deuxième partie de la trajectoire montrer des erreurs inférieures à 10m essentiellement, et correspondent typiquement à des erreurs rencontrées par un récepteur GPS en environnement urbain.

Sur la première partie, les résultats diffèrent fortement selon les méthodes : l'erreur obtenue avec le modèle $Sigma - \epsilon$ oscille en effet autour de 30m, contre 25 pour les moindres carrés classiques, 20 pour le modèle d'élévation et décroit très vite de 10 à quelques mètres avec notre méthode. Sur la deuxième partie de trajectoire les différences entre modèles sont moins visibles. Le

tableau 3.2 compare les performances obtenues avec notre modèle et trois valeurs de k différentes : $k = 1$, $k = 2$, $k = 10$. Le tableau 3.2 résume les performances des méthodes de la littérature avec la solution CAPLOC avec $k = 2$, en termes d'erreur moyenne, médiane, d'écart type et de moyenne quadratique (ou RMS - *Root Mean Square*) sur la globalité du trajet.

	$CAPLOC_{k=1}$	$CAPLOC_{k=2}$	$CAPLOC_{k=10}$
Moyenne	3,38	**3,28**	3,97
Médiane	2,87	2,88	3,35
Ecart-type	2,77	2,5	**2,43**
RMS	4,36	**4,11**	4,65

TABLE 2 – Tableau des performances obtenues avec notre proposition et trois valeurs de k

Trois valeurs de k sont présentées ici et montrent que, sur ce jeu de données, $k = 2$ présente des performances intéressantes qu'il conviendra de confirmer sur d'autres jeux de données. Cette solution est celle qui est conservée pour comparaison avec les autres modèles.

	M.C.	$Sigma - \epsilon$	El.	El.(Groves)	$CAPLOC_{k=2}$
Moyenne	10,57	12,22	8,59	10,78	**3,28**
Médiane	4,33	5,82	4,14	4,74	**2,88**
Ecart-type	10,8	12,38	8,24	11,09	2,5
RMS	15,07	17,34	11,86	15,41	**4,11**

TABLE 3 – Tableau des performances obtenues avec les différentes méthodes de pondération

La méthode utilisant l'élévation en $(sin\theta)^2$ est un peu meilleure qu'une solution classique. Celle de [22] ne modifie pratiquement pas les performances. On peut en revanche s'interroger sur nos performances obtenues avec la méthode $Sigma - \epsilon$ qui, telle qu'utilisée ici, dégrade les performances et ce, quelles que soient les valeurs de c testées. Ces résultats montrent clairement l'intérêt de notre approche par rapport à la solution classique des moindres carrés. Trois valeurs de k sont présentées ici et montrent que, sur ce jeu de données, $k = 2$ présente des performances intéressantes qu'il conviendra de confirmer sur d'autres jeux de données.

Enfin, les erreurs sont représentées sous la forme de fonction de répartition selon la méthode utilisée sur la figure 11.

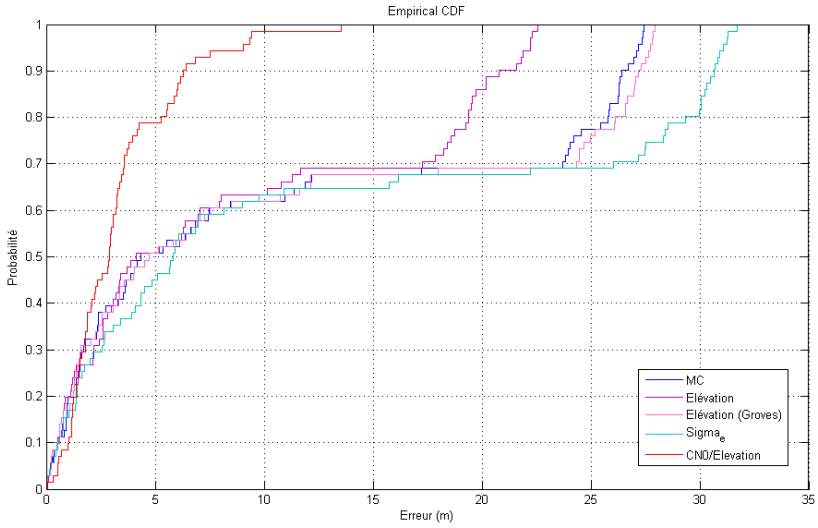


FIGURE 11 – Fonctions de répartition de l'erreur de positionnement selon les différentes méthodes testées

La solution CAPLOC combinant état de réception, C/N_0 et élévation est representée par la courbe rouge. Le tableau 4 présente enfin les probabilité d'erreurs au $90^{ème}$ et $95^{ème}$ percentiles.

	M.C.	$Sigma - \epsilon$	El.	El.(Groves)	$CAPLOC_{k=2}$
90^{me}	26,53	30,68	21,10	27,23	**6,34**
95^{me}	27,19	31,08	22,19	27,7	**8,99**

TABLE 4 – Tableau des performances obtenues aux $90^{ème}$ et $95^{ème}$ percentile

3.3 Conclusion

Dans ce chapitre, nous avons présenté les différents modèles de pondération utilisés dans la littérature, et la solution développée dans le projet. Notre solution repose sur les modèles de la littérature, utilisant à la fois l'élévation du satellite reçu et le rapport signal sur bruit C/N_0 et la connaissance a priori de l'état de réception fournie par les techniques de traitement d'images fish-eyes réalisées dans le projet. Notre modèle a montré des performances plus intéressantes en terme de précision que les solutions classiques, avec notamment une réduction importante de l'erreur au $90^{ème}$ percentile. Si l'on compare ces résultats aux résultats obtenus avec le principe d'exclusion et de seuillage DOP présentés figure 7, on constate que cette nouvelle technique présente des précisions légérement meilleures. la répartition des erreurs (Pour mémoire, l'erreur au $90^{ème}$ percentile était égale à 11m environ.)

4 Utilisation du modèle 3D

La tâche 2 du projet consiste à réaliser un modèle 3D de l'environnement à partir des images et d'en extraire les bâtiments représentant des surfaces de réflexion des signaux GNSS. Comme nous l'avons présenté dans la section précédente, il est possible de positionner le mobile par rapport aux bâtiments situés autour de lui grâce à l'extraction de leurs empreintes dans le plan horizontal. Nous parvenons ensuite à faire une première estimation de la hauteur des bâtiments en analysant la distribution des points verticalement au dessus des empreintes détectées. Pour terminer, nous estimons la normale d'un plan que nous cherchons à ajuster sur l'ensemble des points ayant contribués à la détection de chaque segment composant l'empreinte des bâtiments. La figure 12 illustre le résultat obtenu et montre qu'il est alors possible de positionner les satellites reçus par rapport aux bâtiments présents dans la scène.

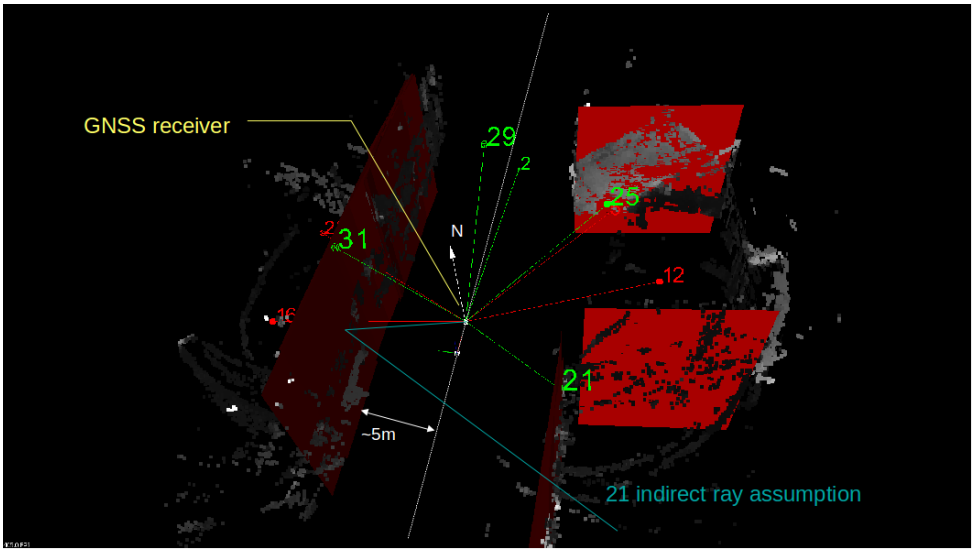


FIGURE 12 – Plane fitting sur le nuages de points à partir de l'empreinte au sol des bâtiments.

Pour cela la connaissance du nord est indispensable afin de correctement orienter le modèle. En l'absence de magnétomètre, nous avons choisi de faire une estimation initiale de manière manuelle en comparant la première image de la séquence avec différents outils de navigation à notre disposition tels que google street. Nous précédons alors au suivi du nord à partir de l'estimation des transformations réalisée au cours de chaque calibrage. Toutefois, des erreurs peuvent apparaitre et se propager d'un couple d'images à l'autre. Une correction manuelle peut réduire ce risque.

Comme annoncé dans les objectifs du projet, l'utilisation de ce modèle doit permettre de corriger l'erreur des pseudo-distances des satellites NLOS reçus après une ou plusieurs réflexions.

En suivant les pistes développées dans [12], la piste envisagée est de calculer le retard r sur le signal propagé par une ou plusieurs réflexion afin de corriger l'estimation de la mesure ρ_{mesure} réalisée par le récepteur. La correction est conforme à l'équation 8.

$$\rho_{corrige} = \rho_{mesure} - r \tag{8}$$

La figure 13 est la représentation d'un retard lorsque le plan azimutal du satellite est orthogonal aux bâtiments. La valeur du retard géométrique dépend à la fois de la configuration du canyon et de la position du satellite (azimuth et élévation). On peut alors écrire r selon l'équation 9 dans laquelle el est l'élévation du satellite.

$$r = w_i \times cos(el) \tag{9}$$

La figure 14 illustre comment déterminer ce retard pour un satellite dont l'angle d'azimut est quelconque. Les retards $r_.$ s'expriment alors de manière générale par les équations 10 conformément aux notations de la figure et avec L représentant la largeur de la rue d'un bâtiment à l'autre.

Il est important de noter qu'il est possible de définir le nombre de réflexions qu'un satellite subira en comparant son élévation à trois seuils donnés par les équations 10. Dans ces equations, H représente la hauteur du bâtiment derrière lequel le satellite est positionné. Soit un satellite d'élévation el. Si $el_{th1} < el < el_{th2}$ alors le satellite subit une seule réflexion. Si $el_{th2} < el < el_{th3}$ alors le satellite subit deux réflexions. Si $el < el_{th3}$ alors le satellite subit trois réflexions.

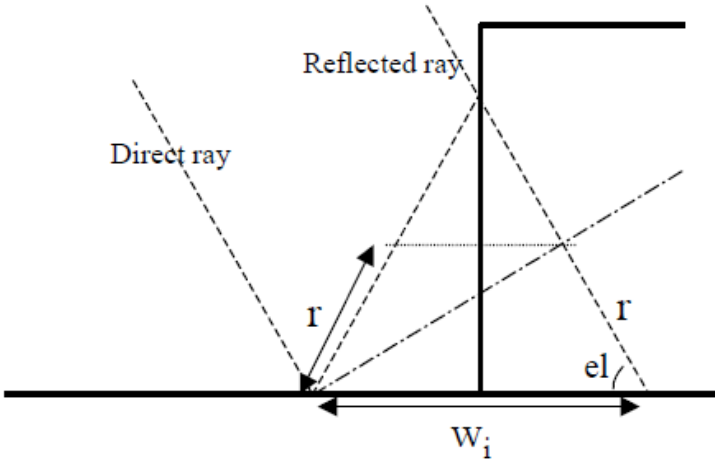


FIGURE 13 – Configuration d'un canyon urbain et retard r causé par la réflexion en 2D.

$$\begin{aligned}
el_{th1} &= \arctan\left|\frac{H \times \cos(\alpha)}{w_i}\right| & r_{1\ ref} &= 2 \times w_i \times \cos(el) \times \cos(\alpha) \\
el_{th2} &= \arctan\left|\frac{H \times \cos(\alpha)}{2 \times L - w_i}\right| & r_{2\ ref} &= 2 \times L \times \cos(el) \times \cos(\alpha) \\
el_{th3} &= \arctan\left|\frac{H \times \cos(\alpha)}{2 \times L + w_i}\right| & r_{3\ ref} &= 2 \times (w_i + L) \times \cos(el) \times \cos(\alpha)
\end{aligned} \tag{10}$$

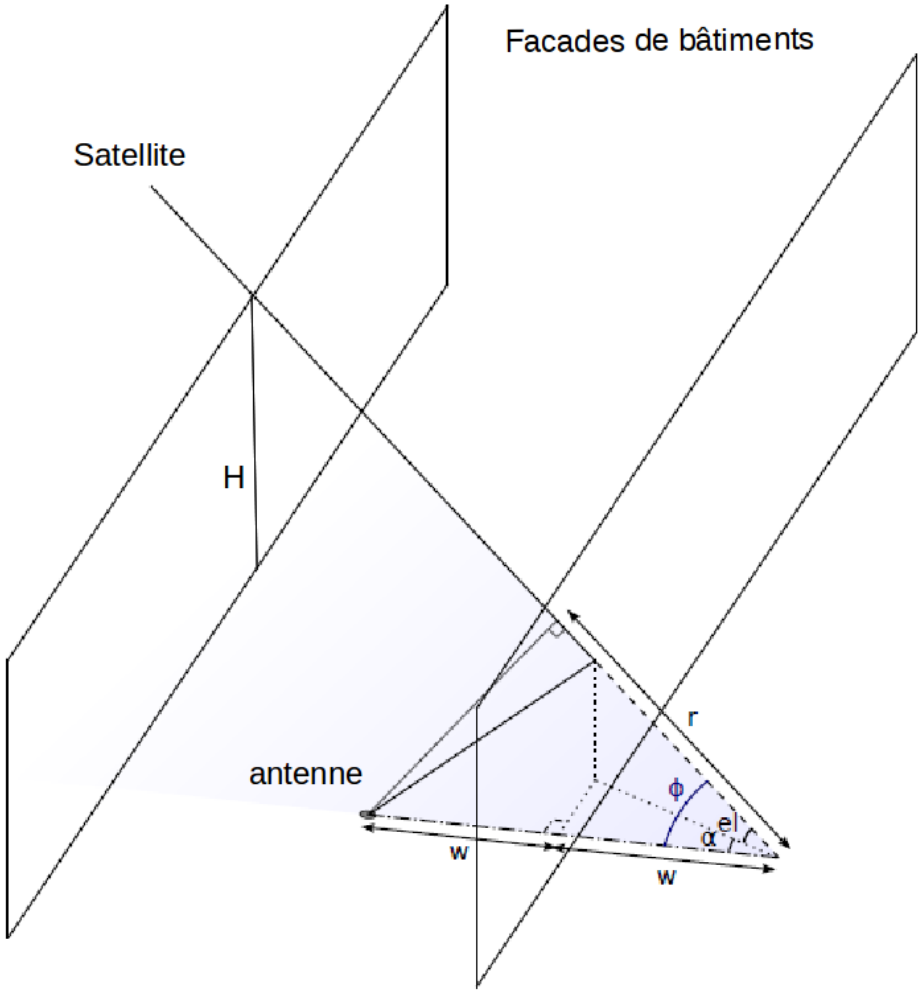


FIGURE 14 – Configuration d'un canyon urbain en 3D et retard d'un satellite NLOS dont l'azimut est quelconque. Dans ce schéma, $w = w_i$

Ainsi, à chaque instant de réception des données GNSS fournies par le récepteur, il est possible de définir le retard associé à chaque satellite après avoir déterminé la nature de sa réflexion. Pour l'Arch 1, la cadence d'acquisition peut atteindre 15 images par seconde. Dans ce cas, le modèle considéré comme valide pour estimer les retards est le dernier modèle calculé. Pour l'Arch 2, un couple d'images est fourni tous les $2m$. Ainsi, lorsque la vitesse du mobile est faible, il est indispensable de prédire la position du mobile dans le dernier modèle valide jusqu'à ce qu'il soit mis à jour. Ainsi pour une vitesse du mobile inférieure à $7.2km/h$ (seuil calculé sur la base d'une acquisition tous les $2m$), le mobile peut recevoir plus d'une donnée GNSS avant de mettre à jour son modèle.

Nous avons validé cette méthode de correction sur des données acquises selon l'Arch 1 sur la ville de Lille et selon l'Arch 2 dans la ville de Belfort. La figure 15 montre le nuage de points en une position sur le trajet lillois du mobile sur lequel nous avons superposé les satellites reçus LOS et NLOS. Sur cette même figure nous représentons (1) la position fournie par le récepteur GNSS en considérant tous les satellites comme LOS (GPS position) et (2) la position après correction des satellites NLOS (PR rectification 2D). La position de référence est donnée par un récepteur RTK. Elle permet de conclure, dans ce cas de figure, que la correction apporte une amélioration.

Sur le trajet effectué à Belfort, nous avons été confronté à la présence importante d'arbres. Lors d'une première phase de correction nous n'avons pas tenu compte de la nature des obstacles

en présence et nous avons appliqué la correction en considérent ces obstacles comme des bâtiments. Nous avons ainsi montré que corriger un satellite reçu directement à travers la végétation induisait une erreur de position plus importante. La méthode de correction que nous proposons considère désormais la nature des obstacles renseignée par les outils développés dans la tâche 1.

Nous avons étudié la distribution des peudo-distances fournies par le récepteur Safedrive. Nous avons constaté que, lorsque le mobile était à l'arrêt, l'écart-type de ces valeurs était bien supérieur à la valeur des corrections que nous pouvions appliquer au signaux GNSS NLOS. Ce phénomène pourrait expliquer la mauvaise qualité des résultats que nous avons obtenus après correction. En mobilité, les résultats obtenus ne sont pas à la hauteur de ce que nous escomptions. La correction que nous appliquons aux pseudo-distances NLOS dégradent souvent la position calculée par le récepteur. La première conclusion serait que le phénomène corrigé par notre méthode n'est pas celui observé par le récepteur. La deuxième conclusion serait de considérer que les pseudo-distances fournies par le Safedrive étant filtrées, nous ne pouvons leur appliquer directement nos corrections.



FIGURE 15 – Amélioration de la localisation après correction du retard sur les pseudo-distances NLOS.

5 Conclusion

Le livrable 3.2 présente les résultats liés à l'amélioration et obtenus depuis le livrable intermédiaire. Il concerne principalement deux méthodes : la pondération des signaux reçus en fonction de leur état de réception et l'utilisation du modèle 3D pour la correction des pseudo-distances des satellites reçus après réfléxion. Le livrable 3.1 avait montré comment le traitement des images fisheye permettait de connaitre l'état de réception (LOS ou NLOS) des satellites disponibles et comment l'exclusion des satellites NLOS permettait alors de réduire l'imprécision de localisation causée par les trajets réfléchis.

Avec les estimations par moindres carrés pondérés, nous avons montré dans ce livrable que la précision pouvait encore être améliorée en utilisant même les satellites NLOS mais avec un poids plus faible que les satellites LOS. Avec la pondération que nous avons proposée, combinant la connaissance de l'état de réception avec le rapport signal sur bruit et l'élévation du satellite, nous avons réduit l'imprecision à 3,28m en moyenne contre 10,57m avec une méthode dite 'classique'.

La méthode proposée pour estimer et corriger la pseudo-distance à partir du modèle 3D réalisé à partir d'images vidéo comme décrit dans le livrable 2.2. Les résultats présentés dans ce rapport montre la validation de l'approche sur des données acquises selon une première technique (Arch 1) sur la ville de Lille et une deuxième (Arch 2) dans la ville de Belfort. Elle permet de conclure, dans le cas lillois, que la correction apporte une amélioration. Le trajet de Belfort, plus complexe en raison de la présence de végétation, n'a pas permis pour le moment de montrer un réel avantage à la correction.

5.1 Perspectives

Les travaux réalisés dans le projet CAPLOC ont montré qu'avec différentes techniques combinant traitement d'images et connaissance de la propagation des signaux GNSS, nous étions capables d'améliorer la précision de localisation GNSS. La faisabilité du concept est donc avérée mais il reste de nombreuses perspectives qu'il sera intéressant de poursuivre pour aller plus loin dans cette direction.

Les moindres carrés pondérés pourront être affinés en continuant à travailler sur le critère de pondération. Nous souhaitons également à court terme utiliser la connaissance a priori de l'état de réception fournie par les traitements CAPLOC à d'autres algorithmes d'estimation dynamique de l'erreur tels que ceux de Viandier [3] développés à l'Ifsttar.

Concernant l'utilisation des modèles 3D, en mobilité, les résultats obtenus ne sont pour l'instant pas à la hauteur de ce que nous escomptions. Différentes solutions identifiées dans la littérature et notamment appliquées par des utilisateurs de modèles 3D pourront cependant être testés pour améliorer les performances actuelles.

Références

[1] J Marais : Predissat, comment évaluer la disponibilité satellitaire en environnement transport ? *Revue Transport et Sécurité, RTS*, 88:175–187, 2005. 5

[2] Juliette Marais : *Localisation de mobiles terrestres par satellites : mise en œuvre d'outils permettant l'analyse de l'influence des conditions de propagation et des effets de masques sur la disponibilité du service offert.* Thèse de doctorat, Université de Lille 1, 2002. 5

[3] Nicolas Viandier : *Modélisation et utilisation des erreurs de pseudodistances GNSS en environnement transport pour l'amélioration des performances de localisation.* Thèse de doctorat, Ecole Centrale de Lille, 2011. 5, 9, 13, 23

[4] Juliette Marais, Sébastien Ambellouis, Amaury Flancquart, Sébastien Lefebvre, Cyril Meurie et Yassine Ruichek : Accurate localisation based on gnss and propagation knowledge for safe applications in guided transport. *Procedia - Social and Behavioral Sciences*, 48(0):796 – 805, 2012. Transport Research Arena 2012. 5

[5] J-I Meguro, Taishi Murata, J-I Takiguchi, Yoshiharu Amano et Takumi Hashizume : Gps accuracy improvement by satellite selection using omnidirectional infrared camera. *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 1804–1810. IEEE, 2008. 5

[6] Paul D Groves : Shadow matching : A new gnss positioning technique for urban canyons. *Journal of Navigation*, 64(03):417–430, 2011. 5

[7] Aude Bourdeau, Mohamed Sahmoudi et Jean-Yves Tourneret : Constructive use of gnss nlos-multipath : Augmenting the navigation kalman filter with a 3d model of the environment. *Information Fusion (FUSION), 2012 15th International Conference on*, pages 2271–2276. IEEE, 2012. 5

[8] Marcus Obst, Sven Bauer et Gerd Wanielik : Urban multipath detection and mitigation with dynamic 3d maps for reliable land vehicle localization. *Position Location and Navigation Symposium (PLANS), 2012 IEEE/ION*, pages 685–691. IEEE, 2012. 5, 9

[9] Francois Peyret, David Bétaille et Florian Mougel : Non-line-of-sight gnss signal detection using an on-board 3d model of buildings. *ITS Telecommunications (ITST), 2011 11th International Conference on*, pages 280–286. IEEE, 2011. 5

[10] PP Robet, BG Evans et A Ekman : Land mobile satellite communication channel model for simultaneous transmission from a land mobile terminal via two separate satellites. *International journal of satellite communications*, 10(3):139–154, 1992. 6

[11] Lars Erling Bråten, César Amaya et David V Rogers : Statistical characterization of land mobile–satellite propagation environments using a photogrammetric technique. *Proc. URSI General Assembly 2002*, 2002. 6

[12] J Marais, M Berbineau et F Nahimana : Effect of alternate path on satellite positioning accuracy in an urban environment. *International Conference on Satellite and Space Communications, 2005. ICSCC 2005*, 2005. 6, 8, 19

[13] FP Fontan, MAV Castro, J Kunisch, J Pamp, E Zollinger, S Buonomo, P Baptista et B Arbesser : A versatile framework for a narrow-and wide-band statistical propagation model for the lms channel. *Broadcasting, IEEE Transactions on*, 43(4):431–458, 1997. 8

[14] D Zimmermann, J Baumann, A Layh, F Landstorfer, R Hoppe et G Wolfle : Database correlation for positioning of mobile terminals in cellular networks using wave propagation models. *Vehicular Technology Conference, 2004. VTC2004-Fall. 2004 IEEE 60th*, volume 7, pages 4682–4686. IEEE, 2004. 8

[15] A Simsky, F Wilms et JP Franckart : Gnss-based failsafe train positioning system for low-density traffic lines based on one dimensional positioning algorithm. *Proc. NAVITEC*, pages 8–10, 2004. 8

[16] Michael Lentmaier, Bernhard Krach et Patrick Robertson : Bayesian time delay estimation of gnss signals in dynamic multipath environments. *International Journal of Navigation and Observation*, 2008, 2008. 9

[17] Dhouha Attia : *Segmentation d'images par combinaison adaptative couleur/texture et classification de pixels. Application à la caractérisation de lŠenvironnement de réception des signaux GNSS.* Thèse de doctorat, Université de Belfort Montbéliard, 2013. 9, 12

[18] Sarab Tay et Juliette Marais : Weighting models for gps pseudorange observations for land transportation in urban canyons. Munich, 5-6 decembre 2013. 6^{th} European Workshop on GNSS Signals and Signal Processing. 13

[19] M Petovello : Gnss filtering options. *Inside GNSS*, 8(2):32–38, 2013. 13

[20] A Wieser : Weighting gnss observations and variations of gnss/ins integration. *Inside GNSS*, 2(1):26–33, 2007. 13, 14

[21] Kai Borre : *A software-defined GPS and Galileo receiver : a single-frequency approach.* Birkhauser, 2007. 13

[22] Paul D Groves et Ziyi Jiang : Height aiding, c/n 0 weighting and consistency checking for gnss nlos and multipath mitigation in urban areas. *Journal of Navigation*, 66(05):653–669, 2013. 13, 15, 17

[23] Jizhong Li et Muqing Wu : The improvement of positioning accuracy with weighted least square based on snr. *Wireless Communications, Networking and Mobile Computing, 2009. WiCom'09. 5th International Conference on*, pages 1–4. IEEE, 2009. 14

[24] Andreas Wieser et Fritz K Brunner : An extended weight model for gps phase observations. *EARTH PLANETS AND SPACE*, 52(10):777–782, 2000. 14

[25] RTCA DO : 229d–minimum operational performance standards for gps waas airborne equipment, 2006. 15

[26] H Hartinger et FK Brunner : Variances of gps phase observations : the sigma- ? model. *GPS solutions*, 2(4):35–43, 1999. 15

Weighting models for GPS Pseudorange observations for land transportation in urban canyons

Sarab Tay
Univ Lille Nord de France, F-59000 Lille,
IFSTTAR, LEOST,
F59650, Villeneuve d'Ascq, France
sarab.tay@ifsttar.fr

Juliette Marais
Univ Lille Nord de France, F-59000 Lille,
IFSTTAR, LEOST,
F59650, Villeneuve d'Ascq, France
juliette.marais@ifsttar.fr

***Abstract*—This paper adresses the problem of position computation in an urban canyon, using signals transmitted by Global Navgation Satellite Systems (GNSS) satellites. Due to the high masking level of buildings in a dense urban environment, the signals arrive at the receiver antenna often by more than one path (multipath).**
Multipath is a source of positioning error, since the receiver solutions for position fail to converge, because of the travel time of the received signal and the moving vehicle.
In this paper, we are presenting a post-processing treatments based on observations variances modelling, in order to solve for vehicle position in an urban canyon.

Keywords*—GNSS signal propagation, urban canyons, multipath, state of reception, weighting models, carrier-to-noise ratio, elevation angle.

I. Introduction

Localization systems are part of the improvement and attractivity efforts for better collective transportation systems. GNSS signals are widely used because this localization solution does not require to enhance the number and thereby the cost of investment and infrastructure. However, localization service quality is strongly degraded in urban environments where the signals are highly affected by diverse types of obstacles located around the receiver antenna. Signals are then subject to multipaths which decrease the accuracy of positioning. Moreover the EGNOS services devoted to increase accuracy by transmitting complementary signals by geostationary satellites suffer here from masking phenomena which make them unavailable. One of the solutions is to reduce the degradations and improve accuracy, it is explored in the COSYS/LEOST team. It consists in estimating a priori the conditions of signal reception.
In this paper, we will present some weighting models based on the GNSS received signals parameters, such as the satellite elevation and the signal strength in terms of the carrier to noise ratio (C/No), along with and the signals' state of reception in an urban canyon.

II. Urban canyon environment

Four satellites at least are necessary to compute a GNSS position. A time of arrival is extracted from each received signal that corresponds to the satellite-to-receiver distance. As summarized in eq. 1, the pseudorange measurement is mainly affected by the propagation through the atmosphere and the environment arround antenna :

$$\rho^s_{t,m} = d^s_t - c(\delta t^s - \delta t^u) + (I^s_t + T^s_t) + \epsilon_t, \quad (1)$$

where :

- d^s_t is the euclidian distance between the satellite s and the user at time t,
- δt^s and δt^u are respectively the offset between the GPS system time and the receiver clock, and the offset between the GPS system time and the receiver clock,
- I^s_t and T^s_t are the ionospheric and the tropospheric errors,
- $\epsilon_t = m^s_t + w_t$ are the observations errors, with m_t is the error due to signal multipath, and w_t is the receiver noise.

The ionospheric and tropospheric errors can be modeled such in [1], whereas the poor satellite geometry and site environment antenna are difficult to treat.
In urban canyons, the presence of buildings of different heights, structures, and materials, dense vegetation, and street elements reduces highly the number of visible satellites and increases the presence of multipath.

Received signals may be classified into three types:

- Line-of-sight (LOS) signals (direct),
- Non Line-of-sight (NLOS) signals (reflected),
- Blocked signals.

Several techniques have been developped for multipath mitigation, for instance the Narrow Spacing Correlator [2], Multipath estimation Delay-Locked-Loop algorithm (MEDLL) [3], some Bayesian approaches [4] and array antenna algorithms [5]. Other solutions such as in [6] and [7] are made to exlcude all the NLOS received signals. However, in [6] it was proved that a simple exclusion decreases the accuracy of user position in an urban area.

III. Observation Weighting Models

In order to improve user position computation accuracy, we are using all received signals, jointly with the information on the state of reception.
Concretly, we propose a weighting model depending on the

quality of the signals and the geometrical form of the surrounding antenna environment. The strength quality of the signal is defined by the Carrier-to-Noise Ratio C/No, which is the ratio if the power level of the signal in 1 Hz bandwidth. The geometrical form of the city canyon defines the masking level of surrounding buildings. This affects directly signals transmitted from a low elevation angle.

Thus, C/No and elevation angles values, as recorded by the receiver, are used to model the pseudorange observations noise variance.

Further information about the geometrical form of the canyon for a defined vehicle path allow us to determine, epoch by epoch, if the signals are received within multipath or not. This information provided from the Caploc Project [8] will be used to introduce the our weighting model.
Before presenting these models, let us remember some GNSS signal basis about user position computation.

The pseudorange observations equation can be written :

$$\Delta\rho = H\Delta X + \epsilon \tag{2}$$

H is the design matrix of the unit-vectors pointing from the linearization point to the location of the i-th satellite, and ΔX is the vector offset of the user's true position and time bias from the values at linearization point [9].
These observations are uncorrelated, and the errors are assumed to be normally distributed according to $\epsilon \sim \mathcal{N}(0, \Sigma)$. The inverse of the covariance matrix Σ will be used as the weighting matrix W, with the assumption that $H^T\Sigma^{-1}H$ is non-singular. The weighted least squares solution is :

$$\Delta\hat{X} = (H^T\Sigma^{-1}H)^{-1} \times H^T\Sigma^{-1}\Delta\rho, \tag{3}$$

The matrix Σ is a diagonal matrix, where the diagonal elements are constituted of the rms of the σ^2, for a specified model. Whatever the used model, the geometric and position DOP can be calculated using the expression $(H^T\Sigma^{-1}H)^{-1}$.

We can find in literature some weighting models like the sigma-ϵ [10], sigma-Δ [11] and exponential model [12]. These models link the C/No to the variance of the pseudorange measurements errors by means of some specified parameters.

A. Sigma-ϵ and elevation model

Concretly, the sigma-ϵ model uses the CNo values of the GPS signals to estimate weights for least square adjustment [10]. Langley [13] has derived a formula that expresses the phase variance σ^2 as a function of the recorded C/No values. This formula was used for geodetic receiver.
Wieser [14] has proposed a completed formula that goes with all receiver types :

$$\sigma^2_{se} = v_i + c_i \cdot 10^{-0.1\times CNo_{measured}} \tag{4}$$

where v_i (m^2) and c_i (m^2Hz) are model parameters and i is the frequency index (L_1 or L_2).

As in [10], one should notice that the function $1/\sin^2(\theta_i)$ (θ_i is the elevation angle of the satellite) can be a good estimate for the error variance and weighting function. The weights increase with the elevation angle.

These two models present some lacks in a strong multipath envionment. Actually, in the sigma-ϵ model, the variance σ^2_{se} does not change with the signal being LOS or NLOS, it changes only with C/No changing.
On the other hand, the elevation model is not always beneficial for low elevation angles (multipath). For instance we can find a very low satellite with a relatively high C/No (fig. 1).

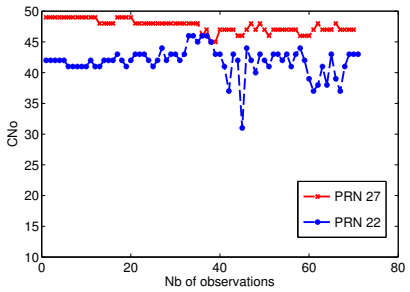


Figure 1. C/No values for PRNs 22 (elev. angle $\approx$ 17 deg.) and 27 (elev. angle $\approx$ 70 deg.)

That is why we have developed a complete model that consider both C/No and the elevation angle of a received signal at the same time. In this model, the variance σ^2_{sr} is changing with the elevation angle of the received signal and its carrier-to-noise density power. Since variance increases with multipath, this ratio is then multiplied by a coefficient k, k is determined whether the signal is LOS or NLOS.
The state of reception of the received signal can be determined as in the Caploc Project, presented in the next paragraph.

B. C/No - Elevation model : Image analysis contribution

The Caploc project proposes a solution based on combining the image analysis and knowledge about the signal propagation for localization applications. The aim of this project is to classify, in a deterministic way, the received signals depending on the state of reception.
The system developped in this project uses a camera with a fisheye lens. During the acquisition campaign, the camera records images of the surrouding environment. The pixels of the recorded images are classified into two zones : sky and non-sky (fig. 2) and (fig. 3).
This classification permits to detect the mask caused by the buildings and other elements (fig. 2), and therefore to determine the position of the satellites in this sky map (fig. 3).The sky zone denotes the LOS received signals, while in the non-sky zone, signals are supposed to be received after one or many reflections.

The variance of the C/No - Elevation model is :

$$\sigma^2_{sr} = k \times \frac{10^{(-0.1\times CNo_{measured})}}{\sin^2(\theta_i)} \tag{5}$$

When the received signal is LOS, k is supposed equal to 1. Whereas, when it is NLOS, k changes. In this paper we have computed the weights $w_{sr} = 1/\sigma^2_{sr}$ for $k = 1, 2$ and $k = +\infty$.

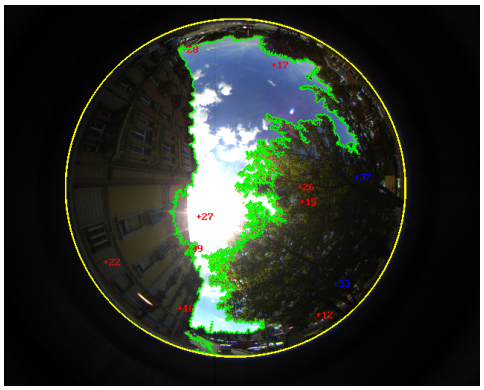


Figure 2. Fish-eye camera image : detection of objects and vegetation edges [15]

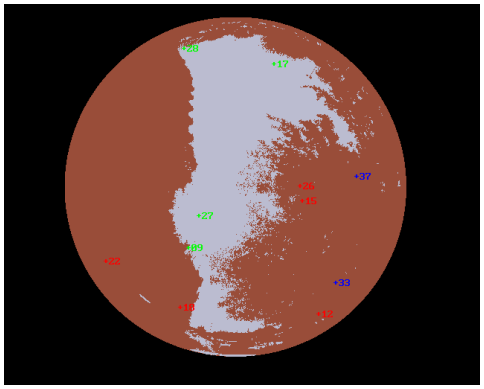


Figure 3. Fish-eye camera image : classification of the image pixels into Sky/Non-Sky zones. Different state of reception satellites are also presented (Green = LOS), (Red = NLOS) and (Blue = Blocked)

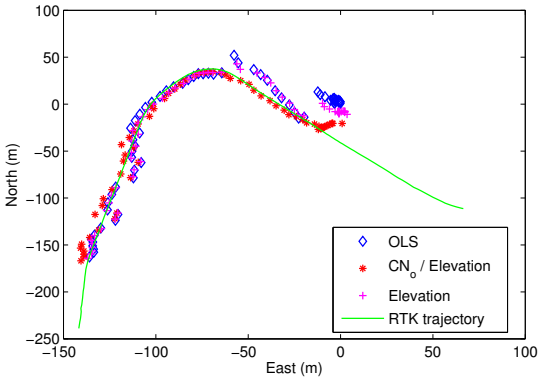


Figure 4. Reference trajectory RTK, CE model, elevation model and OLS vehicle position.

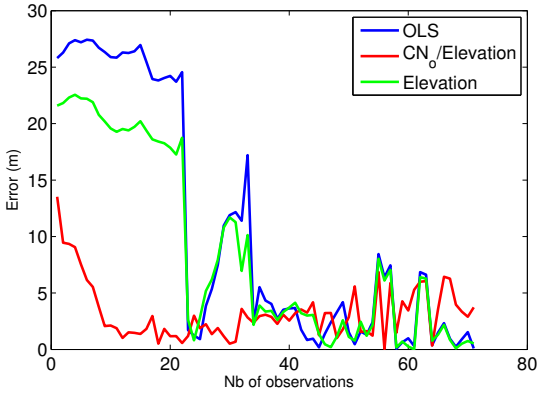


Figure 5. Position errors (all models) with respect to the RTK trajectory, $k = 2$

IV. Measurements campaign and first results

To validate the proposed algorithms, we have used measurements realized on June 2010 in the city of Belfort, France. Data acquisition was performed with a 1 Hz frequency Septentrio receiver for the NMEA data and 27 Hz for the camera. The path we are using here has a duration of about 71 GPS positions for Septentrio and 171 points for the RTK system.

For illustration, fig. 4 shows the RTK trajectory, along with GPS positions computed with our C/No - elevation model (CE model), elevation model and ordinary least squares method. This figure shows a good improvement of the vehicle position computation when the CE model is used.

The figure 5 shows the vehicle position errors with respect to the RTK trajectory. Positions computed with the CE model ($k = 2$) present a very low error level. In table I, we have drawn the standard deviation (σ) and the root mean square errors values for all methods. We can see that there is a deviation improvement of 76.89% and RMS improvement of 72%.

Table I. Standard deviation and RMS error values for the three models, with $k = 2$

	OLS	CNo / Elevation	Elevation only
Std deviation (m)	10.82	2.5	8.24
RMS(m)	15.074	4.11	11.86

When $k = 1$, a good position is obtained also (table II), but it still less important that $k = 2$.

We have also tried the solution about excluding the NLOS signals by using the CE method with $k = +\infty$. This means that the variance σ^2_{sr} is infinite and the weights w_{sr} are equal to zero. Thereby, NLOS signals are excluded from position computation.

When only methods excluding GPS, such as OLS or CE model ($k = +\infty$) are used, the position accuracy is highly degraded. Actually, as we can see in fig. 7, the acquisition zone constitutes a typical urban canyon with high and large buildings, dense vegetation, cross streets, ...
The yellow spot in this same figure indicates the map zone where NLOS are highly present. These multipath result in the outliers observed in fig. 6 (observations $[0 : 23]$) and fig. 4 (East $[-20m : 20m]$).

Our CE model including NLOS signals presents a very good solution for multipath problem in an urban canyon. On the same figures above, position accuracy is clearly improved when the CE model is used, even in the outliers zone.

V. Conclusion

The aim of our work is to develop a weighting model able to solve for multipath problem in an urban canyon

Table II. STANDARD DEVIATION AND RMS ERROR VALUES FOR THE THREE MODELS, WITH $k = 1$

	OLS	CNo / Elevation	Elevation only
Std deviation (m)	10.82	2.76	8.24
RMS (m)	15.074	4.4	11.86

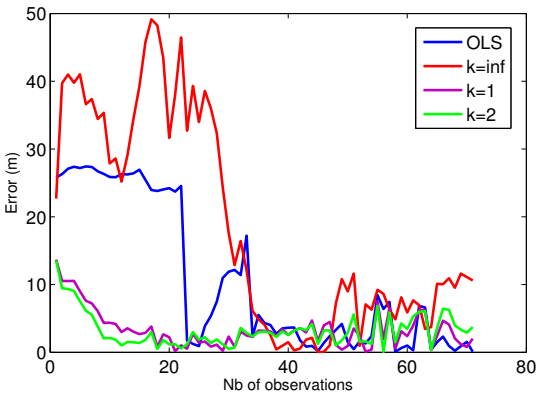


Figure 6. Comparison of position errors for some given values of k.

Table III. STANDARD DEVIATION AND RMS ERROR VALUES FOR THE THREE MODELS, WITH $k = +\infty$

	OLS(m)	CNo / Elevation	Elevation only
Std deviation (m)	10.82	15.74	8.24
RMS	15.074	24.4	11.86

environment.
In this paper, we have presented the CE weighting model which uses data about both carrier-to-noise density power of the received signal and its elevation angle.
An image classification algorithm permitted us to decide if the received signal is LOS or NLOS. The CE model is then applied with a scalar (k) that changes according to the signal type.
The first results are very promoting, since we have noticed an improvement of 76.89% on the standard deviation value between the CE weighting model and OLS method.

In order to validate this model, we believe that it should be applied in other urban environments with strong multipath profiles. We have already performed another measurement campaign in a very dense district of Lille. On the other hand, it would be interesting to compare the performances of kalman filter and CE model in the same environment conditions, and to to implement real-time processors to compute vehicle position.

REFERENCES

[1] J. Klobuchar, "Ionospheric effects on gps," *Global Positioning System : Theory and Applications*, vol. 1, 1996.

[2] A. V. Dierendonck, P. Fenton, and T. Ford, "Theory and performance of narrow correlator spacing in a gps receiver," *Journal of the institute of Navigation*, vol. 39, pp. 265–283, 1992.

[3] R. V. Nee, J. Siereveld, P. Fenton, and B. Townsend, "Multipath estimating delay lock loop : approaching theoretical accuracy limits." the 1994 IEEE Position Location and Navigation Symposium, April 1994, pp. 246–251.

[4] M. Lentmaier, B. Krach, and P. Robertson, "Bayesian time delay estimation of gnss signals in dynamic multipath environments," *International Journal of Navigation and Observation*, vol. 2008, 2008.

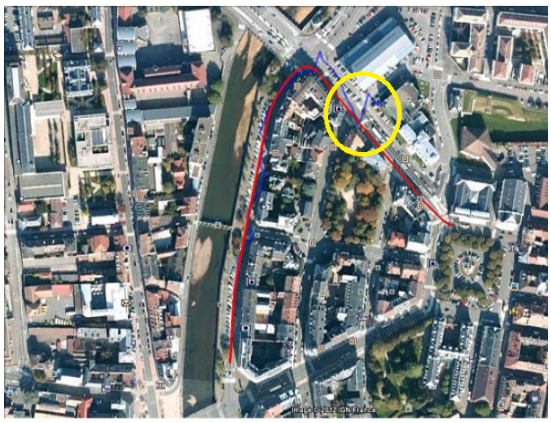


Figure 7. Part of Belfort city where GPS signal acquisitions were performed.

[5] A. Konovaltsev, F. Antreich, and A. Hornbostel, "Performance assessment of antenna array algorithms for multipath and interference mitigation." in Proceedings of the 2-nd workshop on GNSS Signals and Signal Processing, 2007.

[6] D. Attia, "Segmentation d'images par combinaison adaptative couleur/texture et classification de pixels. application à la caractérisation de l'environnement de réception des signaux gnss." Ph.D. dissertation, Université de Belfort Montbéliard, 2013.

[7] S. Peyraud, D. Betaille, S. Renault, M. Ortiz, F. Mougel, and F. Peyret, "About non-line-of-sight satellite detection and exclusion in a 3-d map aided localization algorithm," *Sensors*, vol. 13(1), pp. 829–847, 2013.

[8] J. Marais, S. Ambellouis, A. Flancquart, S. Lefebvre, C. Meurie, and Y. Ruichek, "Accurate localization based on gnss and propagation knwoledge for safe applications in guided transport," *Procedia - Social and Bahavorial Sciences*, vol. 48, pp. 796–805, 2012.

[9] E. Kaplan and C. Hegarty, *Undersatnding GPS : Principles and Applications*. Artech House Publishers, 2006.

[10] H. Hartinger and F. Brunner, "Variances of gps phase observations : The sigma-e model," *GPS Solutions*, vol. 2, pp. 35–43, 1999.

[11] F. Brunner, H. Hartinger, and L. Troyer, "Gps signal diffraction mode-model : the stochastic sigma-δ model," *Journal of Geodesy*, vol. 73, pp. 259–267, 1999.

[12] J. Li and M. Wu, "The improvement of positioning accuracy with weighted least square based on snr," in *Wireless Communications, Networking and Mobile Computing, 2009. WiCom'09. 5th International Conference on*. IEEE, 2009, pp. 1–4.

[13] R. Langley, "Gps receiver system noise," *GPS World*, vol. 8, pp. 40–45, 1997.

[14] A. Wieser and F. Brunner, "An extended weight model for gps phase observations," *Earth Planet Space*, vol. 52, pp. 777–782, 2000.

[15] D. Attia, C. Meurie, Y. Ruichek, and J. Marais, "Counting of satellites with direct gnss signals using fisheye camera : A comparison of clustering algorithms." Washington DC, USA: 11^{th} Intelligent Transport System Conference, October 5-7, 2011.

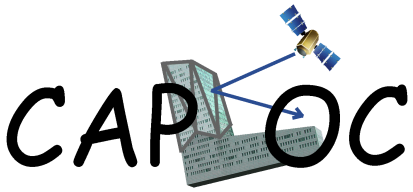
Groupe opérationnel n°2 du Predit 4

« Gestion du trafic et des mobilités »

CAPLOC

Combinaison de l'**A**nalyse d'images et de la connaissance de la **P**ropagation des signaux pour la **LOC**alisation

Convention 10-MT-CVS-001



Livrable L4.2 Bilan des publications

18 janvier 2015

Juliette Marais, Cyril Meurie, Sébastien Ambellouis, Amaury Flancquart
IFSTTAR

Julien Moreau, Dhouha Attia, Yassine Ruichek
UTBM

Identification

Acronyme du projet	CAPLOC
Titre du projet	Combinaison de l'Analyse d'images et de la connaissance de la Propagation des signaux pour la LOCalisation
Coordinatrice du projet	Juliette MARAIS (IFSTTAR/LEOST)
Date de début de projet	19 octobre 2010
Date de fin de projet	18 octobre 2014
Partenaires	IFSTTAR/LEOST, UTBM/IRTES-SET
Type de document	Livrable
Date contractuelle de livraison	18 janvier 2015
Accessibilité	La diffusion de ce document est public

Gestion des versions

Version	**Date**	**Auteurs**	**Modifications**
1.0	02/10/2014	Yassine Ruichek	Rédaction du corps du livrable
1.1	14/01/2015	Juliette Marais, Cyril Meurie, Julien Moreau et Sébastien Ambellouis	Ajouts et corrections
2.0	16/01/2015	Yassine Ruichek	Finalisation

Table des matières

1 Introduction **4**

2 Bilan quantitatif **5**

3 Publications CAPLOC **6**
3.1 Revues à comité de lecture internationale 6
3.2 Revues à comité de lecture nationale 6
3.3 Conférences internationales avec actes 7
3.4 Conférences nationales avec actes 9
3.5 Mémoires de thèse 11
3.6 Communications orales nationales et internationales 12

4 Conclusion **14**

1 Introduction

Ce document dresse le bilan de l'activité de publications des travaux réalisés dans le cadre du projet CAPLOC. Nous commençons par décliner un bilan quantitatif par type de publications. Ensuite, nous donnons, pour chaque article, sa référence accompagnée d'un résumé en anglais ou en français.

2 Bilan quantitatif

Revues à comité de lecture internationale	1
Revues à comité de lecture nationale	1
Conférences internationales avec actes	7
Conférences nationales avec actes	3
Mémoires de thèse	1 (Soutenance prévue en juin 2015)
Communications orales nationales et internationales	6

3 Publications CAPLOC

3.1 Revues à comité de lecture internationale

Toward accurate localization in guided transport : Combining GNSS data and imaging information
J. Marais, C. Meurie, D. Attia, Y. Ruichek, A. Flancquart,
Transportation Research Part C : Emerging Technologies, December 2013

Résumé : Global Navigation Satellite Systems (GNSS) are widely spread (with Global Positioning System - GPS) in intelligent transport systems and offer a low cost, continuous and global solution for positioning. Unfortunately, urban users are often the most demanding of accurate localization but receive a degraded service because of signal propagation conditions. Several mitigation solutions can be developed. We propose, within CAPLOC project (2010-2013) to deal with inaccuracy by associating image processing techniques and signal propagation knowledge. In this paper, we focus on the contribution of image processing in more accurate position estimation. Thus, we use a laboratory vehicle, which is equipped with a fisheye camera and two GNSS receivers. The camera is located on the roof and oriented upwards to capture images of the sky. The GNSS receivers are used to obtain raw data, the position of the vehicle and the reference trajectory. The proposed approach consists in determining where satellites are located in the fisheye image, and then excluding those located in non-sky regions when calculating the position. For that, the strategy is based on an image simplification step coupled with a pixels classification. The image-based exclusion procedure is compared with the classical one based on the application of a threshold on carrier-to-noise (CN0) ratio to separate LOS and NLOS signals. Accuracy improvement is satisfying with the CN0-based method and show an improvement from 13 m to 4.5 m. Image-based detection shows mixed improvements but promising : good in a static area and too harsh in another configuration of the scenario.

3.2 Revues à comité de lecture nationale

Projet CAPLOC : Combinaison de l'analyse d'images et la connaissance de la Propagation des signaux pour la Localisation
J. Marais, C. Meurie, S. Ambellouis, Y. Ruichek,
TEC Transport Environnement Circulation, 218, 4p, 2013

Résumé : Améliorer la capacité et l'attractivité des transports collectifs est une solution pour réduire l'empreinte écologique de nos déplacements. Cela passe par le développement de systèmes de transport intelligents au coeur desquels l'information de localisation tient une place très importante.
Dans le domaine du transport de voyageurs et de fret, la volonté des opérateurs est de réduire les équipements attachés à l'infrastructure notamment en raison de l'augmentation des actes de vandalisme qu'ils subissent et de la hausse des coûts de leur maintenance. Aujourd'hui, les systèmes GNSS (Global Navigation Satellite System) constituent une réponse très adaptée à cette problématique. Parmi eux et en attente de la solution européenne Galileo, le système de radionavigation par satellites Américain GPS offre le meilleur compromis coût/simplicité/ performance. Par ailleurs, les performances d'un récepteur GPS sont améliorées lorsque les services EGNOS (European Geostationary Navigation Overlay Service) sont utilisés.
De nombreuses études ont cependant montré que le service de localisation est sensiblement dégradé en environnement urbain. Dans ces milieux, les conditions de propagation des signaux sont fortement altérées par la présence et la densité des obstacles situés autour de l'antenne de réception. Les signaux subissent des réflexions multiples qui affectent sensiblement les résultats fournis

par l'algorithme d'estimation de position. Les services EGNOS, portés par une constellation de satellites géostationnaires dont la réception est très difficile dans les canyons urbains, s'avèrent également souvent inexploitables pour améliorer la précision de l'information de localisation.
La plupart des solutions visant à rendre les solutions satellitaires de localisation plus précises et plus robustes reposent sur la multiplication des capteurs embarqués. Les solutions multicapteurs peuvent cependant s'avérer plus coûteuses si on multiplie le nombre de capteurs. L'objectif du projet CAPLOC est, avant de combler les lacunes du GNSS avec d'autres capteurs, d'explorer les performances du GNSS et de les enrichir à l'aide de la connaissance de l'environnement de propagation. L'ambition n'est pas de développer un nouveau capteur mais d'étudier comment il est possible d'améliorer les performances des récepteurs grand publics actuels à partir de l'analyse des images fournies par deux caméras grand angle installées sur le toit du mobile.
Dans CAPLOC, seul le système GPS est utilisé pour évaluer les développements. Toutefois, le futur système GALILEO souffrira des mêmes phénomènes locaux de propagation et les solutions CAPLOC pourront donc être appliquées rapidement et sans modification profonde.

3.3 Conférences internationales avec actes

3D modelling of urban environment for enhancing GPS localization's accuracy
J. Moreau, S. Ambellouis, Y. Ruichek,
TRA (Transport Research Arena) 2014, Paris, avril 2014

Résumé : Described project aims at accessing accurate localization. It consists in improving accuracy of localization systems with video perception of the environment around equipped vehicles. The environment corresponds to the propagation area of GNSS receiver installed on top of the vehicle. Accuracy problems are linked to obstacles density, especially in urban environments. The objective is to reduce multipath effect on the measurements, without lowering the availability of the service, which is typically the case for errors detection procedures. The presented work aims at using two fisheye cameras in a stereoscopic configuration in order to compute 3D coordinates of the scene's points. A reconstructed 3D model allows then to correct the propagation error thanks to the knowledge of the satellites' reception status.

Equisolid fisheye stereovision calibration and point cloud computation
J. Moreau, S. Ambellouis, Y. Ruichek,
ISPRS-SSG 2013, conference on Serving Society with Geoinformatics, Antalya, Turquie, 6p, novembre 2013

Résumé : This paper deals with dense 3D point cloud computation of urban environments around a vehicle. The idea is to use two fisheye views to get 3D coordinates of the surrounding scene's points. The first contribution of this paper is the adaptation of an omnidirectional stereovision self-calibration algorithm to an equisolid fisheye projection model. The second contribution is the description of a new epipolar matching based on a scan-circle principle and a dynamic programming technique adapted for fisheye images. The method is validated using both synthetic images for which ground truth is available and real images of an urban scene.

Quantify and improve GNSS quality of service in land transportation by using image processing
J. Marais, C. Meurie,
First CNES-ONERA Workshop on Earth-Space Propagation, Toulouse, France, 5p, janvier 2013

Résumé : Most of the Intelligent Transport Systems rely on GNSS positioning information.

Unfortunately, most of their uses occur in dense urban areas where signal propagation is degraded by surrounding obstacles because of the reception of NLOS (Non Line-Of-Sight) signals. Since the end of the 90's, as a research institute concerned by transport applications, the LEOST team of IFSTTAR pursues research on the impact of the surroundings of a GNSS antenna on the receiver's performances and in particular on NLOS cases when the direct path is not present or it is undetectable due to its low power compared to the rest of path. NLOS detection and mitigation remains an open research problem that can be studied from different angles : at signal processing level, at observable level and at PVT level.
First works of LEOST have dealt to the development of a satellite availability prediction tool based on image processing already (PREDISSAT) for guided transport modes. Then, in collaboration with the Ecole Centrale of Lille, signal processing techniques have been developed to model the evolution of the satellite states of reception versus time and propose pseudo-range error models linked to the different states and in particular the NLOS case where reflected signal can be received without a direct ray. Recent activities are performed in the context of the CAPLOC research project. The main objective of CAPLOC, in collaboration with the University of Technology of Belfort Montbéliard is to provide an innovative tool for the positioning function, relying on satellite-based technologies, GPS and EGNOS, and mitigating the difficulties linked to the constricted environment of reception. In the same way like users of 3D models, our goal is to compare satellite positions to obstacles in order to detect satellite states of reception. Our approach relies on video records of the environments. In this paper, we will present how the images sequences provided by several cameras installed on the roof of the vehicle allow characterizing GNSS performance and we will give a quick overview of how they are used for accuracy enhancement in the CAPLOC project.

Weighting models for GPS Pseudorange observations for land transportation in urban canyons
S. Tay, J. Marais,
6th European Workshop on GNSS Signals and Signal Processing, Munich, December 5-6, 2013

Résumé : This paper adresses the problem of position computation in an urban canyon, using signals transmitted by Global Navgation Satellite Systems (GNSS) satellites. Due to the high masking level of buildings in a dense urban environment, the signals arrive at the receiver antenna often by more than one path (multipath).
Multipath is a source of positioning error, since the receiver solutions for position fail to converge, because of the travel time of the received signal and the moving vehicle. In this paper, we are presenting a post-processing treatments based on observations variances modelling, in order to solve for vehicle position in an urban canyon.

3D reconstruction of urban environments based on fisheye stereovision
J. Moreau, S. Ambellouis, Y. Ruichek,
8th International Conference on Signal Image Technology and Internet Based Systems, Naples, Italy, 2012

Résumé : In this paper, we show some results about 3D urban scenes reconstruction using a fisheye stereovision setup. We propose an analytical analysis of epipolar geometry of the system and an analytical description of tools to compute a 3D point cloud from matched pixels. The novelty is that we do not rectify the images and that we match points along 3D or 2D epipolar curves. The matching process is based on a global dynamic programming algorithm that we adapt to take into account continuous epipolar curve equation. We show 3D point cloud in the case of synthetic images.

Accurate Localisation Based on GNSS and Propagation Knowledge for Safe Applications in Guided Transport
J. Marais, S. Ambellouis, A. Flancquart, S. Lefebvre, C. Meurie, Y. Ruichek,
Transport Research Arena 2012, Athens, Greece, April 23-26, 2012

Résumé : Global Navigation Satellite Systems (GNSS) are widely spread (with Global Positioning System - GPS) in intelligent transport systems and offer a low cost continuous and global solution for positioning. Unfortunately, urban users are often the most demanding of accurate localization but receive a degraded service because of signal propagation conditions. Several mitigation solutions can be developed. We propose, within the CAPLOC project (2010-2013) to deal with inaccuracy by associating image processing techniques for a 3D reconstruction of the surrounding environment and the signal propagation knowledge.

Counting of satellites with direct GNSS signals using Fisheye camera : A comparison of clustering algorithms
D. Attia, C Meurie, Y Ruichek, J. Marais,
ITSC 2011, Washington DC, USA, October 5-7, 2011

Résumé : This paper investigates the problem of accuracy of localization with GNSS in constraint environments. The ultimate goal is to provide a first confidence index on the accuracy of the position given by the GNSS. In this paper, we propose to use the complementarity between the GNSS signals and the development in image processing to count satellites with direct reception state. It consists to use a vehicle equipped with a GPS-RTK and a camera oriented upwards to capture images and count after repositioning, the satellites with direct signals (resp. with blocked/reflected signals) i.e. located in the sky region of the image (resp. located in the not-sky region). The proposed approach is based on an optimal clustering applied on simplified images. More preciously, the acquired image is simplified using a geodesic reconstruction with an optimal contrast parameter. Then, a clustering step is made in order to classify the regions into two classes (sky and not-sky). For that, a set of unsupervised (KMlocal, Fuzzy C-means, Fisher and Statistical region Merging) and supervised (Bayes, K-Nearest Neighbor and Support Vector Machine) clustering algorithms are compared in order to define the best classifier in terms of good classification rate and processing time. Experimental results are shown for hundred images taken in different conditions of acquisition (illumination changes, clouds, sun, tunnels, etc).

3.4 Conférences nationales avec actes

Caractérisation et Structure 3D de l'environnement urbain pour une position GNSS plus précise
J. Marais, S. Ambellouis, C. Meurie, Y. Ruichek,
Congrès ATEXPO, Paris, 30 janvier 2014

Résumé : L'information de localisation est une donnée primordiale pour augmenter la capacité des systèmes de transport, améliorer leur qualité de service et proposer de nouveaux services propices à l'amélioration de l'attractivité des transports collectifs. Face à des actes de vandalisme récurrents et toujours plus nombreux, le coût lié à la maintenance des systèmes augmente et les opérateurs de transport souhaitent réduire les équipements attachés à l'infrastructure. Les systèmes GNSS (Global Navigation Satellite System) peuvent constituer une réponse adaptée à cette problématique. Cependant, de nombreuses études ont montré que le service de localisation est sensiblement dégradé en environnement urbain où les conditions de propagation des signaux

sont fortement altérées par la présence dense des obstacles situés entre le satellite et l'antenne de réception. La réception des signaux s'effectue parfois après plusieurs réflexions et se manifeste par un allongement du temps de trajet du signal entre le satellite et l'antenne. Le récepteur utilise alors les signaux reçus en leur attribuant des "pseudodistances" erronées affectant sensiblement l'estimation de la position. Dans ces canyons urbains, EGNOS (European Geostationary Navigation Overlay Service), normalement utilisé pour améliorer la précision de localisation, présente une disponibilité de service trop faible et s'avère souvent inexploitable. Cette communication est le second volet de la présentation des résultats obtenus dans le cadre du projet CAPLOC dont les objectifs sont : 1) développer un système capable d'extraire et de modéliser des connaissances sur l'environnement traversé par le mobile à partir d'une perception vidéo afin d'inférer les caractéristiques du canal de propagation entre les satellites et l'antenne du récepteur ; 2) proposer des techniques d'estimation plus précise de la position du mobile à partir de ces nouvelles connaissances.

Estimation des états de réception des satellites GNSS pour une position plus précise
J. Marais, S. Tay,
Journées Nationales des Communications dans les Transports, Nevers, France, 2p, mai 2013

Résumé : L'amélioration de la capacité et de l'attractivité des transports collectifs passe par le développement de systèmes de transport intelligents au coeur desquels l'information de localisation occupe une place de plus en plus importante. Les systèmes GNSS (Global Navigation Satellite System) constituent une réponse très adaptée parce qu'ils limitent l'équipement sur l'infrastructure. Parmi eux et en attente de Galileo, le système Américain GPS offre le meilleur compromis coût/simplicité/performance, encore amélioré par les services EGNOS (European Geostationary Navigation Overlay Service).
De nombreuses études ont cependant montré que le service de localisation est sensiblement dégradé en environnement urbain où les conditions de propagation des signaux sont fortement altérées par la présence et la densité des obstacles situés autour de l'antenne de réception. Les signaux subissent des réflexions multiples qui affectent sensiblement les résultats fournis par l'algorithme d'estimation de position. En outre, les services EGNOS, transmis par satellite géostationnaire, souffrent de difficultés liées au masquage dans les canyons urbains et rendent souvent inexploitables leurs informations destinées à améliorer la précision de la localisation.
Si la plupart des solutions visant à rendre les solutions GNSS plus précises et plus robustes reposent sur la multiplication des capteurs embarqués, elles s'avèrent souvent plus coûteuses puisqu'elles font le choix de multiplier le nombre de capteurs. Les travaux menés à l'Ifsttar/Leost visent à explorer les performances potentielles du GNSS sans prendre en compte d'autres capteurs. Pour ce faire, nous exploitons l'information relative aux conditions de réception des signaux satellites disponibles. Dans ces journées, nous présenterons les techniques étudiées pour estimer les états de réception des satellites et les erreurs associées, ainsi que les méthodes d'amélioration de la précision qui en découlent.

Projet CAPLOC - Combinaison de l'Analyse d'images et la connaissance de la Propagation des signaux pour la LOCalisation
J. Marais, C. Meurie, S. Ambellouis, Y. Ruichek,
40ème congrès ATEXPO, Paris, 30-31 janvier 2013
Lauréat du prix de la meilleure intervention

Résumé : Améliorer la capacité et l'attractivité des transports collectifs est une solution pour réduire l'empreinte écologique de nos déplacements. Cela passe par le développement de systèmes de transport intelligents au coeur desquels l'information de localisation tient une place très im-

portante.
Dans le domaine du transport de voyageurs et de fret, la volonté des opérateurs est de réduire les équipements attachés à l'infrastructure notamment en raison de l'augmentation des actes de vandalisme qu'ils subissent et de la hausse des coûts de leur maintenance. Aujourd'hui, les systèmes GNSS (Global Navigation Satellite System) constituent une réponse très adaptée à cette problématique. Parmi eux et en attente de la solution européenne Galileo, le système de radionavigation par satellites Américain GPS offre le meilleur compromis coût/simplicité/performance ; par ailleurs, les performances d'un récepteur GPS sont améliorées lorsque les services EGNOS (European Geostationary Navigation Overlay Service) sont utilisés.
De nombreuses études ont cependant montré que le service de localisation est sensiblement dégradé en environnement urbain. Dans ces milieux, les conditions de propagation des signaux sont fortement altérées par la présence et la densité des obstacles situés autour de l'antenne de réception. Les signaux subissent des réflexions multiples qui affectent sensiblement les résultats fournis par l'algorithme d'estimation de position. Les services EGNOS, portés par une constellation de satellites géostationnaires dont la réception est très difficile dans les canyons urbains, s'avèrent également souvent inexploitables pour améliorer la précision de l'information de localisation.
La plupart des solutions visant à rendre les solutions satellitaires de localisation plus précises et plus robustes reposent sur la multiplication des capteurs embarqués. Les solutions multicapteurs peuvent cependant s'avérer plus coûteuses si on multiplie le nombre de capteurs. L'objectif du projet CAPLOC est, avant de combler les lacunes du GNSS avec d'autres capteurs, d'explorer les performances du GNSS et de les enrichir à l'aide de la connaissance de l'environnement de propagation. L'ambition n'est pas de développer un nouveau capteur mais d'étudier comment il est possible d'améliorer les performances des récepteurs grand publics actuels à partir de l'analyse des images fournies par deux caméras grand angle installées sur le toit du mobile.
Dans CAPLOC, seul le système GPS est utilisé pour évaluer les développements. Toutefois, le futur système GALILEO souffrira des mêmes phénomènes locaux de propagation et les solutions CAPLOC pourront donc être appliquées rapidement et sans modification profonde.

3.5 Mémoires de thèse

Construction de modèles 3D à partir de données vidéo - Application à la localisation en milieu urbain
Julien Moreau,
Encadrement : Sébastien Ambellouis (IFSTTAR), Yassine Ruichek (UTBM)
Date de soutenance : juin 2015

Résumé : Cette thèse s'intéresse à la reconstruction de modèles 3D à partir d'acquisitions vidéo d'images fisheye pour une application à la localisation en milieu urbain. L'objectif est d'améliorer la précision de la localisation GNSS de véhicules en environnement urbain, où l'effet multi-trajets est très prononcé (les structures urbaines forment un canyon urbain et autant d'obstacles pour les rayons GNSS, causant occultations et réflexions). L'intérêt de la démarche est que la génération du modèle 3D est dynamique et ne dépend pas de données préenregistrées. Un modèle 3D peut être créé avec des données télémétriques ou vidéo avec le principe de la stéréovision. L'intérêt de la vidéo est qu'elle peut retranscrire les couleurs, et qu'il est possible d'y adjoindre différents types d'objectifs pour étendre les possibilités.
On propose une méthode de stéréovision basée sur des images fisheye, les objectifs fisheye étant la manière la plus simple d'exploiter la vision omnidirectionnelle. Pour l'application présentée, les capteurs sont orientés vers le ciel afin d'observer tous les bâtiments autour de leur position. Dans ce contexte, on choisit l'option du calibrage automatique par soucis de simplicité d'utilisation et pour rester généraliste. Le calibrage automatique nécessite la détection, la description et la mise

en correspondance de points d'intérêts avec une grande fiabilité.
À partir de ces correspondances, on est capable de sélectionner certains points pour appliquer l'algorithme des 9 points dans une boucle RANSAC jusqu'à obtenir une bonne estimation des paramètres. On présente deux systèmes d'acquisition : un système stéréo avec entraxe connu, et un système mono caméra avec déclenchement basé sur un odomètre, et donc aussi avec entraxe connu.
Le calibrage stéréo permet d'exploiter la contrainte épipolaire pour faciliter la mise en correspondance des points. Grâce à cela, on peut utiliser toute méthode de correspondance stéréo existante, locale ou globale, éparse ou dense. Les nuages de points 3D obtenus sont ensuite assemblés par la méthode de registration ICP (Iterative Closest Point). Le nuage de points 3D étant lourd à traiter, il est important de détecter et de tracer les murs par des surfaces (plan fitting). Enfin, le modèle obtenu est exploité dans l'application de localisation CAPLOC. La position GNSS est calculée à partir des pseudo-distances des satellites reçus (distance du satellite au récepteur). Or, cette information est calculée en fonction de l'heure d'émission par le satellite et l'heure de réception, considérant une visibilité directe. En milieu urbain, de nombreux obstacles vont cacher les satellites pour lesquels l'information sera souvent non plus reçue de manière directe mais de manière réfléchie. Ainsi, la distance parcourue par le rayon sera plus grande, et donc le satellite sera considéré plus loin qu'il ne l'est en réalité. La localisation s'en retrouve faussée. Il s'agit par conséquent de détecter les rayons GNSS NLOS (reçus de manière indirecte), et de mesurer et corriger leur retard grâce au modèle généré. Il peut y avoir une ou plusieurs réflexions, en fonction de l'élévation du satellite, de la hauteur des obstacles et de leur distance au récepteur.
Pour valider les résultats de localisation, on utilise un GPS bas coût comme base (pour corriger les positions) et un GPS RTK pour la vérité terrain. Des expérimentations, validations, évaluations et mesures de précision sont proposées pour toutes les étapes, avec différentes méthodes et des scènes simulées et réelles.

3.6 Communications orales nationales et internationales

CAPLOC : Combinaison de l'Analyse d'images et de la connaissance de la Propagation des signaux pour la LOCalisation
J. Marais, C. Meurie,
Séminaire PREDIT, Connaissance de la mobilité, Paris, 13/11/2014

Résumé : Présentation générale du projet et des résulats intermédiaires.

CAPLOC : Combinaison de l'Analyse d'images et de la connaissance de la Propagation des signaux pour la LOCalisation
C. Meurie,
Présenté à la commission Géo-Pos de la CNIG (Conseil National de l'Information Géographique), 16/10/2014

Résumé : Présentation générale du projet et des résulats intermédiaires.

CAPLOC : Combinaison de l'Analyse d'images et de la connaissance de la Propagation des signaux pour la LOCalisation
J. Marais,
Workshop du COST SaPPART, Londres, septembre 2014

Résumé : Présentation générale du projet et des résulats intermédiaires.

CAPLOC : Combinaison de l'Analyse d'images et de la connaissance de la Propagation des signaux pour la LOCalisation
J. Marais,
Séminaire interne du GIS ITS, Nantes, 4 avril 2012

Résumé : Présentation générale du projet et des résulats intermédiaires.

CAPLOC : Combinaison de l'analyse d'images et la connaissance de la propagation des signaux pour la localisation
C. Meurie, J. Marais,
Séminaire de valorisation GO 2 - Gestion de trafic, Paris, 28 mars 2012

Résumé : Présentation générale du projet et des résulats intermédiaires.

CAPLOC : Combinaison de l'analyse d'images et la connaissance de la propagation des signaux pour la localisation
S. Ambellouis,
Mi-parcours du Predit 4, Bordeaux, 10-12 mai 2011

Résumé : Présentation générale du projet et des résulats intermédiaires.

4 Conclusion

Les résultats des travaux menés dans le cadre du projet CAPLOC ont été présentés et publiés dans de nombreux revues, conférences nationales et internationales, et différents séminaires. En lien étroit avec les travaux de CAPLOC, d'autres résultats (3 conférences internationales avec actes, 3 conférences nationales avec actes et une thèse de doctorat) obtenus par le même consortium dans le cadre du projet collaboratif ViLoc, soutenu par la Région Franche Comté. D'autres résultats, en cours d'évaluation et de finalisation, donneraient certainement lieu à d'autres publications (revues, conférences nationales et/ou internationales).

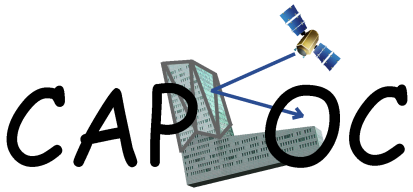
Groupe opérationnel n°2 du Predit 4

« Gestion du trafic et des mobilités »

CAPLOC

Combinaison de l'**A**nalyse d'images
et de la connaissance de la **P**ropagation
des signaux pour la **LOC**alisation

Convention 10-MT-CVS-001



Livrable L4.3 Bilan des résultats et recommandations

18 janvier 2015

Juliette Marais, Cyril Meurie, Sébastien Ambellouis, Amaury Flancquart
IFSTTAR

Julien Moreau, Dhouha Attia, Yassine Ruichek
UTBM

Identification

Acronyme du projet	CAPLOC
Titre du projet	Combinaison de l'Analyse d'images et de la connaissance de la Propagation des signaux pour la LOCalisation
Coordinatrice du projet	Juliette MARAIS (IFSTTAR/LEOST)
Date de début de projet	19 octobre 2010
Date de fin de projet	18 octobre 2014
Partenaires	IFSTTAR/LEOST, UTBM/IRTES-SET
Type de document	Livrable
Date contractuelle de livraison	18 janvier 2015
Accessibilité	La diffusion de ce document est limitée au programme PREDIT

Gestion des versions

Version	**Date**	**Auteurs**	**Modifications**
1.0	06/01/2015	Yassine Ruichek	Rédaction du corps du livrable
1.1	16/01/2015	Juliette Marais, Cyril Meurie et Sébastien Ambellouis	Ajouts et corrections
2.0	17/01/2015	Yassine Ruichek	Finalisation

Table des matières

1 Introduction **4**

2 Bilan et recommandations sur la détection d'obstacles dans l'environnement de réception des signaux GNSS **5**
2.1 Bilan 5
2.2 Recommandations et perspectives 5

3 Bilan et recommandations sur la construction d'un modèle 3D de l'environnement de réception des signaux GNSS **6**
3.1 Bilan 6
3.2 Recommandations et perspectives 6

4 Bilan et recommandations sur l'exploitation des informations de perception dans le processus de localisation par GPS **8**
4.1 Bilan 8
4.2 Recommandations et perspectives 8

5 Conclusion **9**

1 Introduction

Ce document dresse le bilan des résultats du projet CAPLOC. Ce bilan peut être décomposé selon trois sous-programmes : 1) Détection d'obstacles dans l'environnement de réception des signaux GNSS ; 2) Construction d'un modèle 3D de l'environnement de réception des signaux GNNS ; 3) Exploitation des informations de perception (détection d'obstacles et modèle 3D) dans le processus de localisation par GPS. En sortie de chaque sous-programme, des recommandations et des perspectives sont proposées en s'appuyant sur l'analyse des résultats obtenus.

2 Bilan et recommandations sur la détection d'obstacles dans l'environnement de réception des signaux GNSS

L'objectif du sous-programme "détection d'obstacles dans l'environnement de réception des signaux GNSS" est d'estimer dans un premier temps la ligne d'horizon par l'identification des zones ciel et non ciel à partir d'images fisheye fournies par une caméra grand angle installée sur le toît d'un véhicule, en circulation dans un environnement urbain. Cette information de ligne d'horizon permet ensuite de classer les satellites en 2 classes : LOS (Line-Of-Sight) qui représente les satellites dont le signal est reçu via un trajet direct et NLOS (Non-Line-Of-Sight) qui représente les satellites dont le signal est bloqué ou reçu suite à une ou plusieurs réflexions par des obstacles (phénomène de multi-trajet). Dans un deuxième temps, l'objectif est d'identifier dans la zone non ciel deux classes d'objets : la classe bâtiment et la classe végétation. Ceci permettra d'affiner la prise en compte des satellites NLOS dans le calcul de la position du véhicule instrumenté.

2.1 Bilan

Pour la détection de la ligne d'horizon, nous avons développé deux méthodes permettant d'identifier les zones ciel et non ciel à partir d'images fisheye. La première méthode consiste à segmenter l'image par combinaison des informations de couleur et de texture, puis appliquer une classification binaire permettant d'obtenir les deux classes ciel et non ciel. Cette méthode fournit de bons résultats, mais son inconvénient majeur est qu'elle nécessite un coût calculatoire la rendant incompatible avec la contrainte temps-réel. la deuxième méthode est basée sur une simplification de l'image, suivie d'une classification pixelaire adaptée. En plus de ses résutats très satisfaisants, cette méthode est très rapide, en comparaison avec la première, et permet un traitement en temps réel.

Suite à cette première classification, nous avons caractérisé plus finement l'environnement en développant une procédure permettant d'identifier deux classes d'objets (dans la zone non ciel) : bâtiment et végétation. La détection de la classe bâtiment est basée sur l'extraction et l'analyse de segments dans l'image fisheye. La classe végétation est obtenue par une simple soustraction.

2.2 Recommandations et perspectives

Les résultats obtenus sur la question de la caractérisation de l'environnement de réception des signaux GNSS (séparation du contenu de l'image fisheye en classes d'objets : ciel, bâtiment, végétation) suscitent des réflexions et ouvrent la voie à plusieurs perspectives qui peuvent faire l'objet de travaux futurs, afin d'améliorer les performances des différentes phases de traitement.

Les travaux réalisés sont effectués en considérant l'espace de représentation colorimétrique RVB. Il serait très intéressant de mener un travail de test et d'évaluation sur les espaces et les invariants colorimétriques afin d'identifier ceux qui permettent une meilleure identification des différentes classes d'objets de l'image, notamment en situations dégradées (faible ou fort éclairage).

En terme de description de la texture, l'utilisation d'autres descripteurs tels LBP (Local Binary Pattern), les matrices de covariance et leur combinaison pourrait améliorer les performances des procédures de segmentation et de classification.

Concernant la classification, il serait très intéressant d'exploiter les techniques à base d'apprentissage semi-supervisé, notamment pour l'identification des trois classes principales (ciel, bâtiment et végétation), voire même d'autres classes d'objets souvent présents dans les images tels que la route et les objets mobiles (véhicules, piétons, etc.).

Il serait également intéressant d'étendre l'étude en vision nocturne par l'utilisation par exemple de caméra infrarouge ou de caméra multi-spectrale.

3 Bilan et recommandations sur la construction d'un modèle 3D de l'environnement de réception des signaux GNSS

Le sous-programme "construction d'un modèle 3D de l'environnement de réception des signaux GNSS" a pour objectif de construire un modèle 3D des obstacles (en général des bâtiments) autour de l'antenne de réception du GPS, à partir d'images fisheye. Grâce aux distances séparant le véhicule instrumenté (plus précisément la position de la caméra) et les obstacles, nous pouvons déterminer les biais de pseudo-distances induits par les signaux GNSS NLOS reçu de manière indirect (object du troisième sous-programme).

3.1 Bilan

Nous avons développé deux systèmes de vision pour la reconstruction d'un modèle 3D de l'environnement de réception des signaux GNSS. Le premier est un système stéréoscopique et est composé de deux caméras munies d'un objectif fisheye dont les deux axes optiques sont parallèles et orientés verticalement vers le ciel. Une procédure de calibrage automatique a été mise en oeuvre pour estimer les paramètres intrinsèques et extrinsèques. Suite au calibrage du système stéréoscopique, les traitements consistent à mettre en correspondance le contenu des couples d'images acquises simultanément. La procédure de calibrage est appliquée à l'installation du système sur le véhicule, puis ré-appliquée ponctuellement au cours du déplacement du mobile afin de compenser l'effet des vibrations des parties mécaniques du système.

Le deuxième système exploite une seule caméra fisheye dont l'axe optique est vertical et orienté vers le ciel. Ce système est moins encombrant que le premier et permet (1) d'estimer la structure 3D de l'environnement et (2) de renseigner directement (sans nécessité de calibrage) la nature des éléments de la scène urbaine à partir des résultats de segmentation et de classification ciel/bâtiment/végétation (outil développé dans le premier sous-programme). Les propriétés 3D des surfaces observées sont estimées à partir de la mise en correspondance du contenu d'images acquises à des instants successifs au cours du déplacement du mobile (depth from motion). Cette architecture nécessite un calibrage tout au long du trajet puisque la transformation (paramètres extrinsèques) entre les repères caméra à des positions successives n'est pas fixe. Seuls les paramètres intrinsèques de la caméra restent constants.

La procédure de mise en correspondance des images fisheye (stéréo ou temporel) fournit un nuage de points 3D à partir duquel nous construisons l'empreinte des bâtiments, c'est-à-dire les surfaces 3D autour du mobile. Ceci permet ensuite de placer le mobile par rapport à ces surfaces, ce qui est nécessaire pour déterminer les biais des pseudo-distances induits par les satellites NLOS reçu de manière indirecte (objet du troisième sous-programme).

3.2 Recommandations et perspectives

Le premier sous-programme représente le contenu d'une image fisheye par trois classes d'objets : ciel, bâtiment et végétation. Il serait intéressant d'exploiter cette information dans le processus de construction du modèle 3D de l'environnement de réception des signaux GNSS. En effet, l'identification de ces classes pourraît être utilisée dans le processus de mise en correspondance afin d'améliorer les résultats en termes de qualité et de coût calculatoire. En identifiant par exemple les bâtiments dans les images à apparier, on peut réduire les ambiguïtés de mise en correspondance en excluant lors de l'appariement les autres parties dans les deux images. Ceci conduirait aussi à réduire la complexité de mise en correspondance, et donc à réduire le temps de traitement.

Cette première voie d'améliotration pourrait se faire dans le cadre d'un système de vision unique à une seule caméra fisheye. Ce dernier traiterait à la fois la classification des objets et la reconstruction du modèle 3D avec une approche structure from motion. Ceci pemettrait d'éviter

un calibrage global nécessaire entre un système mono dédié à la classification des objets et un système stéréo dédié à la construction du modèle 3D.

L'utilisation d'un système de vision mono pour la reconstruction du modèle 3D permet également de s'affranchir des effets dus aux vibrations des parties mécaniques d'un système de vision stéréo. Cependant, ceci nécessite l'estimation précise du déplacement de la caméra entre deux positions successives. Ce point devrait être traiter avec beaucoup d'attention, notamment dans des situations de changement de direction (véhicule qui tourne à gauche/droite). Différentes approches peuvent être utilisées pour estimer ce déplacement, par exemple l'utilisation de l'odométrie visuelle à partir du même système de perception ou l'utilisation d'autres capteurs dédiés.

Un autre point à améliorer dans notre démarche de reconstruction du modèle 3D est lié au temps de calcul. En effet, notre procédure n'opère pas en temps-réel, comme exigé par l'application CAPLOC. Aux recommandations annoncées plus haut (adoption d'un système de vision mono unique et utilisation des résultats du premier sous-programme) s'ajoutent deux autres possibilités pour parvenir à la réduction du temps de calcul. la première consiste à optimiser les algorithmes de mise en correspondance (par exemple avec une approche eparse ou semi-dense). La deuxième possibilité pourraît être l'implémentation des algorithmes sur des architectures spécialisées (GPU par exemple).

L'estimation des surfaces 3D à partir des nuages de points obtenus par mise en correspondance serait également un point à approfondir, notamment en utilisant des méthodes plus avancées telles que les techniques par construction de plans (techniques par optimisation permettant de regrouper les points 3D en plans).

4 Bilan et recommandations sur l'exploitation des informations de perception dans le processus de localisation par GPS

En se basant sur les résultats des deux premiers sous-programmes "détection d'obstacles dans l'environnement de réception des signaux GNSS" et "construction d'un modèle 3D de l'environnement de réception des signaux GNSS", le sous-programme "exploitation des informations de perception dans le processus de localisation par GPS" s'intéresse d'une part à l'utilisation de la classification des signaux GNSS LOS (Line-Of-Sight, c'est à dire reçus directement) et NLOS (Non Line-Of-Sight, c'est à dire bloqués ou reçus indirectement après réflexion), résultat issu du premier sous-programme, dans le calcul de la position du véhicule. D'autre part, l'objectif est d'utiliser le modèle 3D, résultat du second sous-programme, pour estimer les erreurs de pseudo-distances causées par le phénomène de multi-trajet (signaux GNSS reçus après réflexion).

4.1 Bilan

Dans l'objectif d'améliorer la position du mobile, nous avons développé une procédure basée sur le maintien ou l'exclusion des satellites NLOS en fonction de l'information GDOP relative aux satellites LOS. L'information GDOP est délivrée par le GPS et donne un indication sur la répartition géométrique des satellites. Lorsque le GDOP des satellites LOS est jugé suffisant, l'exclusion des satellites NLOS est appliquée, puis un filtre de Kalman étendu est implémenté pour le calcul de la position. A l'inverse, si le GDOP des satellites LOS est insuffisant, tous les satellites (LOS et NLOS) sont considérés.

Lors de l'exclusion des satellites NLOS, le système peut se retrouver dans la situation où des points d'indisponibilité sont rencontrés. Pour éviter cette situation, nous avons développé une méthode de calcul de position qui pondère la contribution de chaque satellite en fonction de critères tels que le rapport signal sur bruit, l'élévation des satellites et leur état de réception (LOS ou NLOS).

Le troisième résultat de ce sous-programme est l'estimation des erreurs de pseudo-distances causées par le phénomène de multi-trajet (signaux NLOS reçus après réflexion) à partir du modèle 3D développé dans le deuxième sous-programme. Ce modèle 3D fournit les distances qui séparent le véhicule (précisément la position de la caméra) et les obstacles (en général des bâtiments) autour de l'antenne de réception du GPS. Ces distances sont utilisées pour pouvoir déterminer les biais induits par les signaux GNSS NLOS reçus de manière indirecte.

4.2 Recommandations et perspectives

Comme nous l'avons précisé plus haut, de nombreuses informations provenant de la perception de l'environnement de réception des signaux GNSS ont été exploitées pour améliorer la position du véhicule. Cependant, il existe d'autres voies à explorer. On pourrait par exemple intégrer l'information relative à la classe végétation et de développer un modèle de pondération tenant compte aussi des signaux GNSS issus des satellites NLOS se trouvant cachés par la végétation.

La réflexion des signaux GNSS dépend de la nature des surfaces reconstruites. Il serait ainsi intéressant d'intégrer cet aspect dans le calcul de la position du mobile en définissant la nature des surfaces à travers leur texture via des descripteurs de texture, qui peuvent être extraits à partir de l'analyse des images.

Il serait également intéressant d'associer à la position estimée une incertitude, en modélisant les incertitudes induites par les différentes étapes du processus de calcul.

5 Conclusion

Nous avons dressé dans ce livrable un bilan synthétique des résultats produits par le projet CAPLOC. Les principaux résultats sont : 1) la caractérisation de l'environnement de réception des signaux GNSS par analyse d'images fisheye. Cette caractérisation consiste à extraire différentes classes d'objets : ciel, bâtiment et végétation. Ceci permet de détecter la ligne d'horizon et d'identifier alors les satellites LOS et NLOS ; 2) la construction d'un modèle 3D de l'environnement de réception des signaux GNSS. Cette tâche permet d'extraire les surfaces 3D autour de l'antenne de réception GPS. Les distances séparant le mobile instrumenté et ces surfaces permettent d'estimer le biais induits par les signaux des satellites NLOS reçus suite à une réflexion ; 3) l'exploitation des résultats de perception pour l'amélioration du calcul de la position du mobile. Dans ce cadre, plusieurs stratégies ont été proposées pour accroitre la précision de la position.

Les différents résultats obtenus ont été testés et évalués sur des données de synthèses et réelles. Leur analyse a donné lieu à des recommandations et des perspectives quant aux solutions proposées par le projet CAPLOC.

Nous rappelons qu'une présentation détaillées des solutions et des résultats CAPLOC est donnée dans les livrables fournis par les différentes tâches.