Localisation de mobiles par construction de modèles en 3D en utilisant la stéréovision

Sergio Nogueira

UNIVERSITE DE TECHNOLOGIE DE BELFORT – MONTBELIARD

Ecole doctorale SPIM

Thèse

pour obtenir le titre de

Docteur en Sciences

de l'université de Belfort - Montbéliard

Spécialité : INFORMATIQUE

présentée et soutenue par

Sergio NOGUEIRA

LOCALISATION DE MOBILES PAR CONSTRUCTION DE MODELES 3D EN UTILISANT LA STEREOVISION

Thèse dirigée par Yassine RUICHEK et François CHARPILLET

Equipe d'accueil : SeT (UTBM) / MAIA (INRIA Lorraine)

Thèse soutenue publiquement le 9 décembre 2009

devant le jury composé de :

M.	Jacques	JACOT	Professeur, EPFL, Lausanne	Président
M.	Louahdi	KHOUDOUR	Chargé de Recherche, INRETS-LEOST, Villeneuve d'Ascq	Rapporteur
M.	Abdelaziz	BENSRHAIR	Professeur des Universités, INSA, Rouen	Rapporteur
M.	Maan	EL BADAOUI	Maître de Conférences, LAGIS, USTL, Lille	Examinateur
M.	François	CHARPILLET	Directeur de Recherche, LORIA-INRIA, Nancy	Co-directeur
M.	Yassine	RUICHEK	Professeur des Universités, UTBM, Belfort	Directeur

REMERCIEMENTS

En premier lieu, je tiens à remercier M. Ruichek, mon directeur de thèse, qui a dirigé mes travaux dans la continuité de mon stage de fin d'études. Tout au long de ces quatre années, il a su orienter mes recherches tout en me laissant une grande liberté. Je remercie également M. Charpillet, mon co-directeur de thèse, pour l'encadrement, le soutien et l'intérêt qu'il porte à mes recherches.

Merci également à M. Jacot d'avoir accepté de présider le jury de cette thèse et les rapporteurs de cette thèse, M. Khoudour et M. Benshair pour l'intérêt qu'ils ont porté à mon travail. Merci également M. El Badaoui, examinateur qui a accepté de juger cette thèse.

Je remercie également l'équipe du SET de Belfort et du LORIA de Nancy avec lesquelles j'ai eu le plaisir de travailler. Un merci particulier à Arianne qui m'a soulagé de beaucoup de tâches administratives et à Frédéric pour tout le travail qu'il a accompli autour du véhicule.

Je profite de cette occasion pour saluer mes collègues de l'entreprise Mikron, qui m'ont soutenu.

J'ai aussi une pensée pour mes amis, thésards ou non qui m'ont aidé au cours de ces quatre dernières années. Un grand merci au couple d'amis qui m'a soutenu depuis le début et qui a su être de bons conseils le moment venu.

Je souhaite une bonne continuation à Pawel et à Jean-Michel avec qui j'ai partagé de bons moments durant ces dernières années. Je n'oublie pas ma famille, particulièrement mes frères et mon épouse, qui ont su m'encourager, me soutenir et me motiver lorsqu'il le fallait.

RESUME

Les travaux présentés dans cette thèse contribuent aux systèmes de localisation pour un robot mobile en utilisant la stéréovision. Ces travaux s'inscrivent dans le cadre d'une collaboration entre le LORIA-INRIA de Nancy et le laboratoire SeT de l'UTBM. L'approche proposée est décomposée en deux étapes. La première étape constitue une phase d'apprentissage qui permet de construire un modèle 3D de l'environnement de navigation. La deuxième étape est consacrée à la localisation du véhicule par rapport au modèle 3D.

La phase d'apprentissage a pour objectif de construire un modèle tridimensionnel, à partir de points d'intérêt pouvant être appariés sous différentes contraintes géométriques (translation, rotation, changement d'échelle) et/ou contraintes de changements d'illumination. Dans l'objectif de répondre à toutes ces contraintes, nous utilisons la méthode SIFT (Scale Invariant Feature Transform) permettant des mises en correspondance de vues éloignées. Ces points d'intérêt sont décrits par de nombreux attributs qui font d'eux des caractéristiques très intéressantes pour une localisation robuste. Suite à la mise en correspondance de ces points, un modèle tridimensionnel est construit, en utilisant une méthode incrémentale. Un ajustement des positions est effectué afin d'écarter les éventuelles déviations.

La phase de localisation consiste à déterminer la position du mobile par rapport au modèle 3D représentant l'environnement de navigation. Elle consiste à apparier les points 3D reconstruits à partir d'une pose du capteur stéréoscopie et les points 3D du modèle. Cet appariement est effectué par l'intermédiaire des points d'intérêt, issus de la méthode d'extraction SIFT.

L'approche proposée a été évaluée en utilisant une plate-forme de simulation permettant de simuler un capteur stéréoscopique, installé sur véhicule naviguant dans un environnement 3D virtuel. Par ailleurs, le système de localisation développé a été testé en utilisant le véhicule instrumenté du laboratoire SeT afin d'évaluer ses performances en conditions réelles d'utilisation.

Mots-clés : Localisation, reconstruction, stéréovision, analyse d'images

ABSTRACT

The work presented in this thesis contribute to global localization systems for mobile robots using stereovision. This works are in line with the collaboration between LORIA-NANCY from Nancy and the SeT laboratory from UTBM. The proposed localization method is composed of two steps. The first one is a learning environment phase that allows a three-dimensional model generation of the navigation environment. The second step is devoted to the vehicle localization by the three-dimensional model.

During the learning phase a three-dimensional model is build on features. These features can be match into geometric constraints (translation, rotation, scale) and/or illumination changes. Constraints are respected using an invariant approach to these variations that allow distant point of views matching. Features are described using several attributes which make them useful for robust localization. From extracted points, the three-dimensional model is build using an incremental method with position adjustment to remove deviations.

The localization process computes an accurate vehicle position by matching the three-dimensional features extracted using stereovision and the three-dimensional model data. This matching is done through interest points obtained by a scale invariant feature transform (SIFT) method.

The proposed localization method has been evaluated using a simulation platform that allows stereoscopic sensor simulation onto a vehicle navigating into virtual environment. Moreover, some experiments into real conditions have been made to evaluate the method performance.

Keywords: Localization, reconstruction, stereovision, image processing, matching process.

NOTATIONS ET ACRONYMES

$\mathbb{R}^n$	:	Espace projectif de dimension n
$\mathcal{P}$	:	Espace projectif de dimension n
M	:	Point de l'espace euclidien
L	:	Droite de l'espace euclidien
m	:	Point du plan projectif
l	:	Ligne du plan projectif
$\tilde{m}$	:	Coordonnées non-homogènes du point du plan projectif
Π	:	Plan de la scène
C	:	Coordonnées du repère caméra dans le repère global
K	:	Matrice des paramètres intrinsèques
E	:	Matrices essentielles
F	:	Matrices fondamentales
H	:	Matrice d'homographie
I_n	:	Matrice identité de dimension $n \times n$
E	:	Matrice essentielle
T	:	Matrice de translation
I	:	Image d'une caméra
u	:	Abscisse de l'image
v	:	Ordonnée de l'image
f	:	Focale de la caméra
u_0	:	Abscisse du point central
v_0	:	Ordonnée du point central
$p_1 \wedge p_2$	:	Produit vectoriel entre les vecteurs p_1 et p_2

$p_1 \simeq p_2$: Egalité projective entre les vecteurs p_1 et p_2

$[T]_\times$: Matrice antisymétrique construite à partir de T

TABLE DES MATIERES

Remerciements ..... **i**

Résumé ..... **iii**

Abstract ..... **v**

Notations et acronymes ..... **vii**

Table des matières ..... **ix**

Table des figures ..... **xv**

Liste des tableaux ..... **xxi**

Introduction générale **1**

Problématique et contexte **2**

Contributions **5**

Organisation du manuscrit **6**

1. **Etat de l'art de la localisation de robots mobiles** **9**

1.1. Contexte 10

1.2. La localisation relative 10

1.2.1. L'odométrie 11

1.2.2. La centrale inertielle 12

1.3. La localisation absolue 15

1.3.1. Triangulation 15

1.3.2. Trilatération 16

1.3.3. Les amers 17

1.3.3.1. Les amers naturels 17

1.3.3.2. Les amers artificiels 18

1.3.4. Localisation sur carte 19

1.3.4.1. Présentation 19

1.3.4.2. Marquage au sol 20
1.3.4.3. Localisation à partir d'amers sur une carte 20
1.3.5. Les systèmes de géo-référencement satellitaire 23
1.3.5.1. Introduction 23
1.3.5.2. Composantes du GPS 23
1.3.5.2.1. Segment spatial 24
1.3.5.2.2. Segment de contrôle 25
1.3.5.2.3. Segment utilisateur 26
1.3.5.3. Fonctionnement du GPS 26
1.3.5.3.1. Signal émis 26
1.3.5.3.2. Principe de localisation 27
1.3.5.4. Les autres systèmes de géo-référencement satellitaire 28
1.3.5.4.1. Galileo 28
1.3.5.4.2. GLONASS 28
1.3.5.4.3. Compass 29
1.3.5.4.4. IRNSS 29
1.3.5.4.5. QZSS 29
1.4. La localisation hybride 29
1.5. Conclusion 31

2. La localisation par vision artificielle 33

2.1. Introduction 34
2.2. Les extracteurs de points 35
2.3. Approches par invariants locaux 38
2.3.1. Le descripteur SIFT 39
2.3.1.1. Détection des extrema dans l'espace échelle 40
2.3.1.2. Localisation des points clés 42
2.3.1.3. Calcul de l'orientation 43
2.3.1.4. Calcul du descripteur 44
2.4. Mise en correspondance des points 46
2.4.1. Méthode par corrélation 46
2.4.2. Méthode SIFT 47
2.5. Géométrie de la vision 48
2.5.1. Les espaces projectifs 48

2.5.2. Homographies 49
2.5.3. Modèle géométrique des caméras 49
2.5.4. Transformations interne et externe 50
2.5.4.1. Changement de repère 51
2.5.4.2. Projection centrale 52
2.5.4.3. Mise à l'échelle 53
2.5.4.4. Transformation complète 54
2.6. Modèle géométrique des stéréoscopes 54
2.6.1. Centres de projection 55
2.6.2. Géométrie épipolaire 55
2.7. Reconstruction 3D 57
2.7.1. Matrice fondamentale 57
2.7.2. Matrice essentielle 58
2.7.3. Estimation de la matrice fondamentale 59
2.7.3.1. Méthodes linéaires 59
2.7.3.2. Méthodes itératives 61
2.7.3.3. Méthodes robustes 62
2.7.3.3.1. M-Estimateur 62
2.7.3.3.2. RANSAC 64
2.7.3.3.3. LMedS : Least Median of Squares 66
2.7.4. Estimation de la matrice essentielle 67
2.7.4.1. Matrice essentielle et fondamentale 67
2.7.4.2. Calcul de la matrice de rotation et de translation relatives à partir de la matrice essentielle 68
2.8. Calibration d'un système de stéréovision 69
2.9. Conclusion 71

3. Simulateur 73

3.1. Introduction 74
3.2. Approches de modélisation urbaine 3D 75
3.2.1. Déterminer la forme des bâtiments 76
3.2.2. Formes complexes d'un groupe de bâtiments 77
3.2.3. Méthode d'unification par séquences d'images 77
3.3. Le modèle virtuel de la place Stanislas 77

3.4. Simulateur 83
3.4.1. Fonctionnement des modèles virtuels 83
3.4.1.1. Vertex shaders 84
3.4.1.1.1. Changements d'illumination 86
3.4.1.1.2. Bruits d'images 87
3.4.1.2. Backface culling 87
3.4.1.3. Clipping 88
3.4.1.4. Rastérisation 89
3.5. Conclusion 90

4. Localisation par stéréovision 91

4.1. Introduction 92
4.2. Reconstruction 92
4.2.1. Mise en correspondance stéréoscopique 93
4.2.1.1. La contrainte épipolaire 93
4.2.1.2. La contrainte de position 94
4.2.1.3. La contrainte d'orientation 96
4.2.1.4. La contrainte d'échelle 96
4.2.1.5. La contrainte d'unicité 96
4.2.2. Calcul du modèle 96
4.2.3. Construction 3D 97
4.2.3.1. Estimation de la pose du stéréoscope 99
4.2.3.2. Sélection des points SIFT appariables 99
4.2.3.3. Mise en correspondance robuste 101
4.2.4. Ajustement de faisceaux local 101
4.2.5. Modèle reconstruit 103
4.3. Localisation 105
4.3.1. Calcul robuste de la pose de la caméra 107
4.3.2. Estimation de la position de la caméra à partir de trois points 3D 108
4.4. Conclusion 111

5. Résultats expérimentaux 113

5.1. Plate-forme expérimentale 114
5.1.1. Caractéristiques techniques du Gem Car 115

5.2. Les extracteurs de points 116
5.2.1. Critère de stabilité géométrique 117
5.2.2. Comparaison des extracteurs dans le modèle virtuel 117
5.2.2.1. Evalution aux changements de points de vue 117
5.2.2.2. Changements d'illumination avec images virtuelles 122
5.2.2.3. Occultation de l'image 123
5.2.3. Comparaison des extracteurs dans le modèle réel 124
5.2.3.1. Images réelles floues 126
5.2.3.2. Changements d'illumination avec images réelles 127
5.2.3.3. Changements de points de vue sur des images réelles 128
5.2.4. Conclusion des extracteurs 130
5.3. Evaluation de la reconstruction 130
5.4. Localisation 132
5.4.1. Localisation dans le modèle virtuel 132
5.4.2. Localisation dans le modèle virtuel 134
5.4.3. Localisation dans le modèle réel 136
5.4.3.1. Transformation WGS84 vers Lambert II étendue 136
5.4.3.2. Localisation sur des données réelles 138
5.5. Conclusion 138

Conclusion et perspectives 141

Conclusion 142

Perspectives 142

Annexes 145

A.1. Changements de point de vue avec des images virtuelles 146

A.2. Changements d'illumination avec des images virtuelles 147

A.3. Images avec différents niveaux de flous 148

A.4. Changements d'illumination avec des images réelles 149

A.5. Changements de points de vue avec des images réelles 150

A.6. Capteurs et ordinateur embarqués 151

Bibliographie 155

TABLE DES FIGURES

Figure 1.1 : Schéma d'une unité de mesure inertielle avec une structure à cadrant qui permet de déterminer une vitesse et une attitude. Les éléments X_a, Y_a, Z_a représentent respectivement les accéléromètres sur les axes X,Y,Z. De même, les éléments X_g, Y_g, Z_g représentent respectivement le roulis, le tangage et le lacet. (Image de Courtesy NASA) 13

Figure 1.2 : Schéma de l'algorithme de calcul de la position et de la vitesse dans le repère de navigation de l'unité de mesure à structure arrimée. 14

Figure 1.3 : Principe de la localisation par triangulation 16

Figure 1.4 : Principe de la localisation par trilatération 17

Figure 1.5 : Juxtaposition d'une carte de navigation et d'une carte d'amers 21

Figure 1.6 : Utilisation d'une carte d'amers pour identifier la position d'un mobile.... 22

Figure 1.7 : Constellation du segment spatial, aujourd'hui composé de 30 satellites (image extrait de Courtesy Hans Toft) 24

Figure 1.8 : Combinaison de trois satellites pour le géopositionnement (image de la planète extrait de Google Earth) 25

Figure 1.9 : Stations de contrôle au sol des satellites du GPS 26

Figure 1.10 : Types de signaux émis par le système GPS (source : Wikipédia) 27

Figure 2.1 : Exemple des niveaux d'échelle pour une octave. 41

Figure 2.2: Diagramme des images lissées à différents espaces d'échelle et calcul des images issues de l'opérateur DoG. Image extraite de l'artice de Lowe [38]. 42

Figure 2.3 : Détection des extrema locaux; le point en vert est comparé avec les 26 voisins marqués par un X : ses 8 points connexes + les 9 points des 2 niveaux d'échelles adjacents. 43

Figure 2.4 : Image du descripteur du SIFT. La partie gauche de la figure présente l'extraction des gradients autour de la position des points clés de l'image. Chaque gradient est pondéré par un noyau gaussien de forme circulaire, présenté en jaune, puis accumulé dans l'histogramme des orientations, décomposé en 4x4 sous-régions, (voir sur la partie de droite de la figure). 45

Figure 2.5 : Extraction des points de SIFT sur une image de résolution 760x740. 1741 points clés (représentés par cercles rouges) ont été extraits. Les cercles sont proportionnels à leurs échelles respectives, avec en vert l'orientation des points SIFT. 46

Figure 2.6 : Mise en correspondance par corrélation. Un point détecté dans l'image de gauche est cherché dans une région autour des coordonnées du point dans l'image de droite. 47

Figure 2.7: Modèle sténopé..... 50

Figure 2.8: Repère image par rapport au repère du plan image..... 53

Figure 2.9: Géométrie épipolaire. 57

Figure 2.10 : Exemples de fonctions de poids. 63

Figure 2.11 : Combinaisons possibles pour la position des caméras..... 69

Figure 2.12 : Images de mires utilisées pour réaliser une calibration d'une caméra. 70

Le modèle 3D de la place Stanislas (Nancy) a été mis à disposition pour la présentation de la rénovation de la place lors des journées d'expérimentation du projet PREDIT MobiVIP. Ce modèle a été conçu avec une précision centimétrique pour pouvoir mettre en correspondance des images réelles et virtuelles [7]. La figure Figure 3.1 illustre un ensemble de vue 3D de cet environnement. 78

Figure 3.2 : Vues du modèle 3D de la place Stanislas modélisé par la société Tecnomade. Images réalisées par le moteur graphique propriétaire de 3D temps réel de Tecnomade appelé PortEye. 79

Figure 3.3 : Echantillon de textures du modèle 3D de la place Stanislas..... 80

Figure 3.4 : Extrait de la définition des maillages constituant le modèle 3D. Les données sont stockées dans le format XML. 81

Figure 3.5 : Extrait de la configuration des objets du modèle virtuel. Les données sont stockées dans le format XML..... 82

Figure 3.6 : Pipeline graphique du rendu d'une image virtuelle, permettant de transformer des données brutes en une image. 84

Figure 3.7 : Saturation de la lumière sur une partie de l'objet pour la simulation d'effets de réflexion spéculaire. 86

Figure 3.8 : Changement de l'éclairage par pixel shader. 87

Figure 3.9 : Zone de clipping..... 88

Figure 3.10 : Projection d'une scène 3D sur un plan image 89

Figure 4.1 : Extraction des points SIFT d'un stéréoscope virtuel..... 93

Figure 4.2 : Zones de recherche du point Pi dans l'image 2 ; cas où la matrice fondamentale est connue, on obtient une bande de recherche autour de la droite épipolaire. 94

Figure 4.3 : Configuration alignée d'un stéréoscope. 94

Figure 4.4 : Zones de recherche du point Pi dans l'image 2 avec un stéréoscope aligné ; la zone de recherche est une bande horizontale limitée par la contrainte de position. 95

Figure 4.5 : Configuration stéréoscopique particulière. Les caméras sont alignées optiquement avec une distance inter-caméra E 95

Figure 4.6 : Schéma de la reconstruction d'une paire stéréoscopique ; Les images a) et b) sont respectivement l'image gauche et droite du stéréoscope. Les images c) et d) représentent l'extraction des points SIFT respectivement de a) et b). L'image e) est la reconstruction stéréoscopique des points SIFT par rapport au stéréoscope. L'image f) montre les points SIFT placés sur le repère global. 98

Figure 4.7 : Champ de vue (Field Of View) de la caméra défini par la focale et la dimension du capteur. 100

Figure 4.8 : Zones de recherche du point Pi dans l'image 2 ; cas où le point 3D reconstruit de Pi est projeté dans l'image 2 avec une incertitude connue de la pose : la zone de recherche est limitée à un rectangle dans l'image 2. 101

Figure 4.9 : Ajustement de faisceaux local. 102

Figure 4.10 : Reconstruction des points SIFT 104

Figure 4.11 : Méthode de localisation robuste 106

Figure 5.1 : Plate-forme expérimentale électrique automatisée : « Gem Car » équipée par le laboratoire SeT de l'UTBM. 115

Figure 5.2 : Dimensions du Gem Car 115

Figure 5.3: Base d'images pour la comparaison de l'extraction de primitives sur différents points de vue. 119

Figure 5.4 : Critère de stabilité pour les images 1 à 5 des opérateurs SIFT et Harris 121

Figure 5.5 : Critère de stabilité pour les images 6 à 10 des opérateurs SIFT et Harris121

Figure 5.6 : Critère de stabilité pour les images 11 à 15 des opérateurs SIFT et Harris 121

Figure 5.8 : Critère de stabilité pour les images 1 à 6 des opérateurs SIFT et Harris 122

Figure 5.7 : Images virtuelles avec différents niveaux d'éclairage 122

Figure 5.9 : Occultation de l'image. 123

Figure 5.10 : Critère de stabilité pour les images 1 à 4 des opérateurs SIFT et Harris 124

Figure 5.11 : Extrait d'images de la base de données de tests 125

Figure 5.12 : Image d'une même scène réelle : à gauche, l'image est nette ; à droite, l'image est floutée par un changement de focus de l'appareil de prise de vue. 126

Figure 5.13 : Critère de stabilité pour les images 1 à 5 des opérateurs SIFT et Harris sur les différents niveaux de floue. 127

Figure 5.14 : Images réelles de la même scène avec un changement d'illumination .. 127

Figure 5.15 : Critère de stabilité pour les images 1 à 5 des opérateurs SIFT et Harris aux changements d'illumination sur des images réelles. 128

Figure 5.17 : Critère de stabilité pour les images 1 à 5 des opérateurs SIFT et Harris aux changements de points de vue sur des images réelles. 129

Figure 5.16 : Images réelles avec changements de point de vue d'une même scène.. 129

Figure 5.18 : Construction du modèle de l'environnement réalisé à partir de 80 acquisitions stéréoscopiques. Le modèle est constitué de 10488 points sur une distance d'environ 142m. 131

Figure 5.19 : Calcul de la position des points de la caméra reconstruit en rouge. En jaune la trajectoire effectuée. 131

Figure 5.20 : Conditions de tests pour la localisation en virtuelle. a) modèle virtuel normale ; b) sous exposée ; c) sur exposée ; d) occlusion du point de vue 132

Figure 5.21 : Erreur (en mètres) dans la condition d'éclairage normale..... 133

Figure 5.22 : Erreur (en mètres) dans la condition d'éclairage sous-exposée..... 133

Figure 5.23 : Erreur (en mètres) dans la condition d'éclairage sur-exposée 133

Figure 5.24 : Erreur (en mètres) dans le cas de la présence d'objets proche des objets de la trajectoire. 134

Figure 5.26 : Erreur (en mètres) des trajectoires parallèles. En rouge, déviation de la trajectoire intérieure (en magenta sur la figure des trajectoires). En vert, déviation de la trajectoire apprise (en jaune sur la figure des trajectoires) ; En bleu, la déviation de la trajectoire extérieure (en cyan sur la figure des trajectoires). 135

Figure 5.25 : Trajectoires parallèles autour de la trajectoire principale. 135

Figure 5.27 : Trajectoire réelle effectué par le GemCar. En rouge les données GPS RTK. Les points bleus correspondent à l'évaluation de la localisation par stéréovision. 138

LISTE DES TABLEAUX

Table 2.1 : Algorithme d'extraction des points de Harris 38

Table 2.2 : Algorithme normalisé des 8 points 61

Table 2.3 : Algorithme M-Estimateur 64

Table 2.4 : Algorithme de RANSAC 65

Table 2.5 : Algorithme LMedS (Least Median of Squares) 67

Table 4.1 : Méthode de la localisation robuste 110

Table 5.1 : Performances du Gem Car 114

Table 5.2 : Caractéristiques propres du Gem Car 116

Table 5.3 : Résultats de l'extraction et la mise en correspondance des points de Harris et de SIFT sur les différents points de vue. 120

Table 5.4 : Résultats de l'extraction et la mise en correspondance des points de Harris et de SIFT sur différentes illumination. 123

Table 5.5 : Résultats de l'extraction et la mise en correspondance des points de Harris et de SIFT sur des images avec une partie de l'image occultée 124

Table 5.6 : Résultats de l'extraction et la mise en correspondance des points de Harris et de SIFT sur les différents niveaux de floue. 127

Table 5.7 : Résultats de l'extraction et la mise en correspondance des points de Harris et de SIFT avec différents éclairages. 128

Table 5.8 : Résultats de l'extraction et la mise en correspondance des points de Harris et de SIFT d'images réelles avec différents points de vue. Le nombre maximal de points de Harris extraits est limité à 4096 points. 129

Table 5.9 : Conversion de données WGS84 en coordonnées projectives Lambert II étendue 138

INTRODUCTION GENERALE

PROBLEMATIQUE ET CONTEXTE

Depuis une vingtaine d'années, de nombreux programmes de recherche se sont penchés sur le concept de « voiture intelligente ». L'étude de ce concept est, de nos jours, bien avancée grâce au développement des technologies de l'information et de la communication (TIC). Dotée de capteurs qui lui permettent de percevoir son environnement, de tester son propre état et celui de son conducteur, une telle voiture doit être capable de prévenir, d'assister et/ou de remplacer l'usager dans des situations dangereuses de conduite. Plus sophistiqués encore, des systèmes de sécurité interactifs permettront de connecter l'automobiliste à l'infrastructure routière ou à un autre conducteur pour un échange d'informations portant sur l'état de la route, le trafic et les conditions de circulation.

Le véhicule intelligent et son intégration dans la ville de demain sont des sujets qui préoccupent de nombreux acteurs sociaux. Récemment, la Commission Européenne a adopté en décembre 2008 un plan de déploiement pour les systèmes de transports intelligents (STI) afin d'améliorer la sécurité routière, réduire la congestion du trafic et les émissions de CO_2. L'utilisation des technologies de l'information et de la communication telles que les satellites, les ordinateurs, les étiquettes RFID[1] et autres appareils de géolocalisation, comme un téléphone, joue donc un rôle majeur pour proposer de nouveaux outils de planification de voyages multimodaux.

Le but est, d'une part, de créer des véhicules plus sûrs, munis d'assistances automatisées (contrôle des vitesses et des distances, contrôle latéral, détection d'obstacles, parking automatique, conduite automatique sur certains trajets, etc.) et, d'autre part, de limiter le recours aux véhicules privés en facilitant l'utilisation de systèmes modernes de transports collectifs, partagés ou mis à la libre disposition du public. Ce système de transport public est fondé sur une flotte de petits véhicules électriques spécifiquement conçus pour les zones où la circulation automobile doit être fortement restreinte. De façon à pouvoir utiliser ce type de transport, il faut (1) concevoir des fonctions d'aide à la navigation en fusionnant les informations délivrées par des capteurs proprioceptifs et/ou extéroceptifs disposés à bord du véhicule (2)

[1] RFID : Radio Frequency Identification est une méthode pour mémoriser et récupérer des données à distances en utilisant des marqueurs appelés « radio-étiquettes ».

concevoir un système permettant à un passager d'accéder à une palette de services embarqués ou distants.

Le travail de cette thèse s'inscrit dans le cadre du concept du véhicule intelligent et son intégration dans la ville du futur. L'étude de ce concept se fait à travers plusieurs collaborations entre différents laboratoires autour du Cycab, le projet MobiVIP[2] du PREDIT, le pôle de compétitivité « Véhicule du futur »[3] et le projet CRISTAL[4]. Il s'agit de déployer dans une ville une flotte de véhicules autonomes pouvant servir de véhicules individuels aux usagers. C'est un projet à long terme qui nécessite de résoudre de nombreuses problématiques comme la localisation, la détection d'obstacles et la planification de trajectoires.

L'objectif économique de l'étude de ce concept est de proposer des systèmes de transports alternatifs qui doivent « s'accommoder » des réseaux urbains. Ils doivent également cohabiter parfaitement avec les autres types de véhicules conventionnels, qui ne peuvent disparaître du jour au lendemain. L'avenir immédiat des cybercabs réside dans les véhicules publics « bimodes », c'est-à-dire pouvant être indifféremment conduits dans le « réseau banalisé » par une personne, remorqués automatiquement par un véhicule tracteur, rapatriés/redistribués à distance par un téléopérateur (à terme, automatiquement par intelligence artificielle). L'opérateur public dispose ainsi d'un certain nombre de « briques de mobilité » complémentaires à son offre de transport, qu'il pourra composer instantanément selon l'évolution de la demande, dans

---

[2] http://www-sop.inria.fr/mobivip/ : Projet de recherche dans le cadre du Programme de Recherche Et D'expérimentation et d'Innovation dans les Transports terrestres (PREDIT) initié par l'ADEME , l'ANVAR et le ministère chargé de la recherche, des transports, de l'environnement et de l'industrie.

[3] http://www.vehiculedufutur.com/ : Le Pôle de compétitivité Véhicule du Futur a pour vocation à faire travailler en synergie les entreprises, établissements d'enseignement supérieur et organismes de recherche publics ou privés, pour mettre en œuvre des projets de R&D par l'innovation.

[4] http://projet-cristal.net/company.html : Cellule de Recherche Industrielle en Systèmes de Transports Automatisés Légers est un système de transport public « bi-mode », individuel (mode libre-service) ou semi-collectif (mode navette), adaptable à l'évolution de la demande de mobilité dans le temps et dans l'espace.

le temps ou dans l'espace. C'est le concept intégré à la mobilité intégrale. Toutes les villes sont susceptibles de recourir à ce dispositif.

Les besoins croissants de fiabilité et de sécurité des systèmes de transport ont incité le Conseil des Transports de l'Union Européenne à lancer en 2000 son propre système de géo-référencement satellitaire appelée Galileo[5]. Ce système se compose de 30 satellites qui seront mis sur une orbite moyenne d'environ 20000Km avant la fin 2013. Ce nouveau système sera compatible avec le système américain de géo-référencement appelé GPS (Global Positioning System) pour améliorer la précision et la sécurité des signaux, en cas de défaillances. Par rapport aux autres systèmes, Galileo offre une couverture mondiale en zones urbaines et des messages sur les erreurs de fonctionnement ainsi que la fiabilité du signal. Galileo garantit également un fonctionnement permanent. Il ne peut pas être interrompu pour des besoins géopolitiques d'un pays.

Conçu pour les besoins civils, Galileo permettra aux utilisateurs du monde entier de se localiser avec une précision métrique par un service ouvert. Une précision inférieure au mètre pourra être fournie dans le cas d'une utilisation du service commercial de l'opérateur Galileo. Grâce à ce système, la couverture de localisation en milieu urbain passera de 50% à 95%, offrant de bonnes perspectives pour la navigation dans les centres urbains. Avec les technologies actuelles telles que le GPS RTK (Real Time Kinematic), la précision obtenue approche le centimètre, ce qui permettra un contrôle précis d'un véhicule automatisé.

Cependant, les possibilités offertes par les systèmes de géo-référencement satellitaire ne sont pas valides pour tous les milieux urbains. En effet, les infrastructures telles que les ponts, les grands immeubles et les tunnels empêchent leur bon fonctionnement. En ne détectant pas les satellites, à cause de leur non-visibilité ou des reflets des signaux sur les bâtiments, un système de géo-référencement satellitaire perd de sa précision, et peut même fournir des positions erronées. Ces problèmes sont bien connus et portent le nom de « canyons urbains ». On ne peut donc pas envisager la navigation d'un véhicule automatisé en se basant uniquement sur un tel système. Ceci implique alors la nécessite de chercher des informations supplémentaires à l'aide d'autres capteurs de localisation relative tels que l'odomètre

[5] http://ec.europa.eu/transport/galileo/index_en.htm

ou la centrale inertielle. Ces types de capteurs permettent d'obtenir une estimation précise des déplacements, mais uniquement sur de faibles distances à cause des dérives au cours du temps. Il est donc nécessaire de pouvoir ajuster les dérives par un système de la localisation globale tel que le système de géo-référencement satellitaire.

La complémentarité des capteurs locaux/globaux ne peut fonctionner que si le système de localisation globale est disponible fréquemment. Or, en milieu urbain dense, on ne peut pas garantir la disponibilité régulière d'un tel système. Afin de palier à cette limitation, notre idée est de proposer un système embarqué de localisation globale, basé sur la vision artificielle.

CONTRIBUTIONS

Les travaux présentés dans cette thèse contribuent aux systèmes de localisation pour un robot mobile en utilisant la stéréovision. L'approche proposée est décomposée en deux étapes. La première étape constitue une phase d'apprentissage et permet de construire un modèle 3D de l'environnement de navigation. La deuxième étape est consacrée à la localisation du véhicule par rapport au modèle 3D.

La phase d'apprentissage a pour objectif de construire un modèle tridimensionnel, à partir de points d'intérêt pouvant être appariées sous différentes contraintes géométriques (translation, rotation, changement d'échelle) et/ou contraintes de changements d'illumination. Dans l'objectif de répondre à toutes ces contraintes, nous utilisons la méthode SIFT (Scale Invariant Feature Transform) permettant des mises en correspondance de vues éloignées. Ces points d'intérêt sont décrits par de nombreux attributs qui font d'eux des caractéristiques très intéressantes pour une localisation robuste. Suite à la mise en correspondance de ces points, un modèle tridimensionnel est construit, en utilisant une méthode incrémentale. Un ajustement des positions est effectué afin d'écarter les éventuelles déviations.

La phase de localisation consiste à déterminer la position du mobile par rapport au modèle 3D représentant l'environnement de navigation. Elle consiste à apparier les points 3D reconstruits à partir d'une pose du capteur stéréoscopique et les points 3D du modèle. Cet appariement est effectué par l'intermédiaire des points d'intérêt, issus de la méthode d'extraction SIFT.

L'approche proposée a été évaluée en utilisant une plate-forme de simulation permettant de simuler un capteur stéréoscopique, installé sur véhicule naviguant dans un environnement 3D virtuel. Afin d'évaluer ses performances en conditions réelles d'utilisation, le système de localisation développé a été également testé en utilisant le véhicule instrumenté du laboratoire SeT.

ORGANISATION DU MANUSCRIT

Le chapitre 1 propose un état de l'art sur la localisation de robots mobiles en utilisant différents capteurs. Les méthodes proposées peuvent être distinguées en différentes catégories :

- La localisation relative : ensemble des méthodes permettant de déterminer un déplacement relatif d'un véhicule par des capteurs dits proprioceptifs (odomètres, capteurs inertiels,…) ;
- La localisation globale : ensemble des méthodes permettant de déterminer la position d'un véhicule dans un repère global (lié à l'environnement). Ainsi, ces méthodes utilisent des capteurs, dits extéroceptifs (caméra, GPS, …), permettant d'obtenir des informations clés sur l'environnement ;
- La localisation hybride : ensemble de techniques basées sur la fusion des méthodes locales et globales dont l'objectif de tirer profit des avantages de chacune d'elles ;

Le chapitre 2 décrit les concepts mathématiques et algorithmiques utilisés dans les processus de reconstruction 3D à partir de plusieurs images d'une même scène prises sous différents points de vue. Ce chapitre détaille les méthodes d'extraction des primitives et de leur mise en correspondance. Il présente ensuite les points essentiels de la reconstruction tridimensionnelle.

Le chapitre 3 présente l'utilisation de la modélisation 3D dans le cadre de la simulation 3D. Nous décrirons l'utilité et l'importance de cette simulation dans le développement et l'évaluation des projets dans différents domaines. Ce travail sur la modélisation nous a permis de développer une plateforme de simulation destinée à l'évaluation et aux tests de nos algorithmes.

Le chapitre 4 décrit l'approche mis en œuvre pour localiser un mobile à partir d'une reconstruction de primitives stéréoscopiques. L'approche proposée est décomposée en deux étapes. La première étape constitue une phase d'apprentissage et permet de construire un modèle 3D de l'environnement de navigation. La deuxième étape est consacrée à la localisation du véhicule par rapport au modèle 3D.

Le chapitre 5 expose les résultats obtenus lors des expérimentations. Dans un premier temps, une étude est menée sur les différentes méthodes d'extraction de points clés. Des résultats de comparaison à partir d'images virtuelles et réelles sont présentés. Dans un deuxième temps, les résultats expérimentaux de la reconstruction et de la localisation sont présentés et analysés sous différentes contraintes dans des environnements virtuels, obtenus grâce à la plateforme de simulation mise en œuvre dans le chapitre 3. De plus, un essai sur les données réelles est présenté réalisé avec le véhicule instrumenté.

Ce mémoire se termine par une conclusion qui soulignent les contributions principales des travaux réalisés et proposent quelques perspectives de recherche.

CHAPITRE 1

ETAT DE L'ART DE LA LOCALISATION DE ROBOTS MOBILES

1.1. Contexte

Un système de navigation autonome permet à un agent mobile de se déplacer dans un environnement. Pour pouvoir accomplir la fonction de navigation, un agent mobile doit répondre aux questions suivantes :

- Quelle est ma position ?
- Quelle est ma destination ?
- Quel chemin choisir ?

La position et la destination se réfèrent à la même notion de localisation. Il est nécessaire pour se localiser de réaliser des observations d'un environnement afin de déterminer une pose de l'agent dans un référentiel connu. La localisation est un élément clé [76,3] qui implique donc que l'agent ait des capteurs afin de récupérer des informations liées à l'environnement et de les traiter pour en déduire sa position par rapport aux éléments structurant la scène ou par rapport à un référentiel global.

Le choix d'un chemin pour un agent se réfère à la notion de navigation. Pour se déplacer, un agent mobile doit pouvoir établir un lien entre sa position et sa destination, impliquant une représentation en mémoire de l'environnement.

1.2. La localisation relative

La localisation relative, également appelée localisation à l'estime, permet à un agent mobile de calculer sa position à un instant t, par intégration des données capteurs qui lui sont propres et de la position à l'instant $t-1$. Les capteurs utilisés pour cette localisation, dits proprioceptifs, permettent de déterminer la position, la vitesse, l'accélération et l'orientation d'un mobile. L'intégration d'erreurs, à cause de l'imprécision du calcul ou encore de glissement du véhicule, fait dériver le système de localisation. Une étape de recalage est alors nécessaire en utilisant un système de localisation absolue. Le principal avantage de cette méthode est que les capteurs proprioceptifs fournissent des données à des fréquences élevées et que l'incertitude sur les erreurs de mesure puisse être bornée.

Dans le cadre de la robotique mobile, les capteurs employés pour la localisation relative sont l'odométrie, les mesures inertielles, le recalage successif d'images par caméra ou les profils télémétriques. Les paragraphes suivants détaillent le

fonctionnement, les avantages et les inconvénients des différents capteurs de localisation relative.

1.2.1. L'odométrie

L'odométrie permet de déterminer le nombre de tours d'une roue, en utilisant un codeur optique (type de codeur le plus répandu en robotique mobile) au niveau de celle-ci, sur un certain laps de temps. Connaissant la vitesse de rotation de la roue, on peut déduire la distance parcourue par le véhicule par intégration du temps. L'odométrie est un capteur très utilisé en robotique [4] de par sa simplicité d'intégration et de son faible coût. Les principaux avantages de l'odomètre sont :

- La simplicité d'intégration à un robot ou un véhicule ;
- La fréquence élevée d'acquisition des données, ce qui permet une précision élevée à court terme ;
- Le faible coût d'achat ;
- Une utilisation possible dans toutes les conditions climatiques ;
- Un modèle de données simple, ce qui permet de réaliser les calculs en temps réel ;
- La fiabilité et l'autonomie du capteur.

L'odométrie présente les inconvénients suivants :

- L'odométrie permet une localisation relative. En conséquence, il est nécessaire d'initialiser à nouveau le système relatif par un système absolu en utilisant par exemple un GPS (Global Position System). Ce capteur n'est donc pas utilisable seul sur de longs trajets et nécessite des informations provenant d'autres objets ;
- L'odométrie est sensible aux conditions d'utilisation. Dans le cadre d'un véhicule, l'estimation de la distance parcourue n'est pas précise dans le cas de glissement sur des surfaces non lisses puisque l'odomètre ne peut pas prendre en compte les irrégularités du terrain. De plus, une faible erreur sur l'orientation du véhicule engendre une erreur importante sur la position estimée ;

Les sources d'erreurs peuvent être de deux types en référence à l'article de Borenstein [4] : les erreurs systématiques et les erreurs non-systématiques.

Les erreurs systématiques définissent des erreurs liées au fonctionnement du véhicule et de l'odomètre. Elles induisent une dérive et peuvent être dues à :

- La différence de diamètre entre les roues ;
- La différence de diamètre d'une roue et de sa valeur nominale ;
- La différence de la distance inter-essieux et sa valeur nominale ;
- La différence d'alignement des roues ;
- La résolution finie des codeurs.

Les erreurs non-systématiques induisent une erreur immédiate importante. Elles sont liées au véhicule et à son environnement comme par exemple :

- Le patinage des roues ;
- Le glissement sur une surface qui n'est pas lisse.

1.2.2. La centrale inertielle

Le déplacement d'un agent mobile peut être mesuré indirectement par l'intégration de la vitesse ou de l'accélération. Ces données peuvent être mesurées à partir d'une centrale inertielle appelée également centrale à inertie (*inertial unit* en anglais). Le capteur inertiel est un dispositif muni de trois gyroscopes, de trois accéléromètres et d'un calculateur en temps réel. Les accéléromètres mesurent l'accélération sur chacun des axes et les gyroscopes mesurent l'attitude (roulis, tangage, cap). Les capteurs inertiels retournant directement ces mesures sont connus sous le nom d'*unité de mesure inertielle* (*inertial Measurement Unit* en anglais). Certains systèmes permettent de déterminer directement une position, une orientation à partir de l'accélération et de l'attitude. De tels systèmes sont appelés *systèmes de navigation inertielle* (*Inertial Navigation System* en anglais).

Les structures inertielles existantes peuvent avoir l'une des deux structures suivantes : la *structure à cadran* (*Gimbaled*) ou la *structure arrimée* (*strapdown*). La première consiste à maintenir les accéléromètres alignés dans l'axe de la navigation et permet ainsi de calculer directement la vitesse et la position (cf. Figure 1.1).

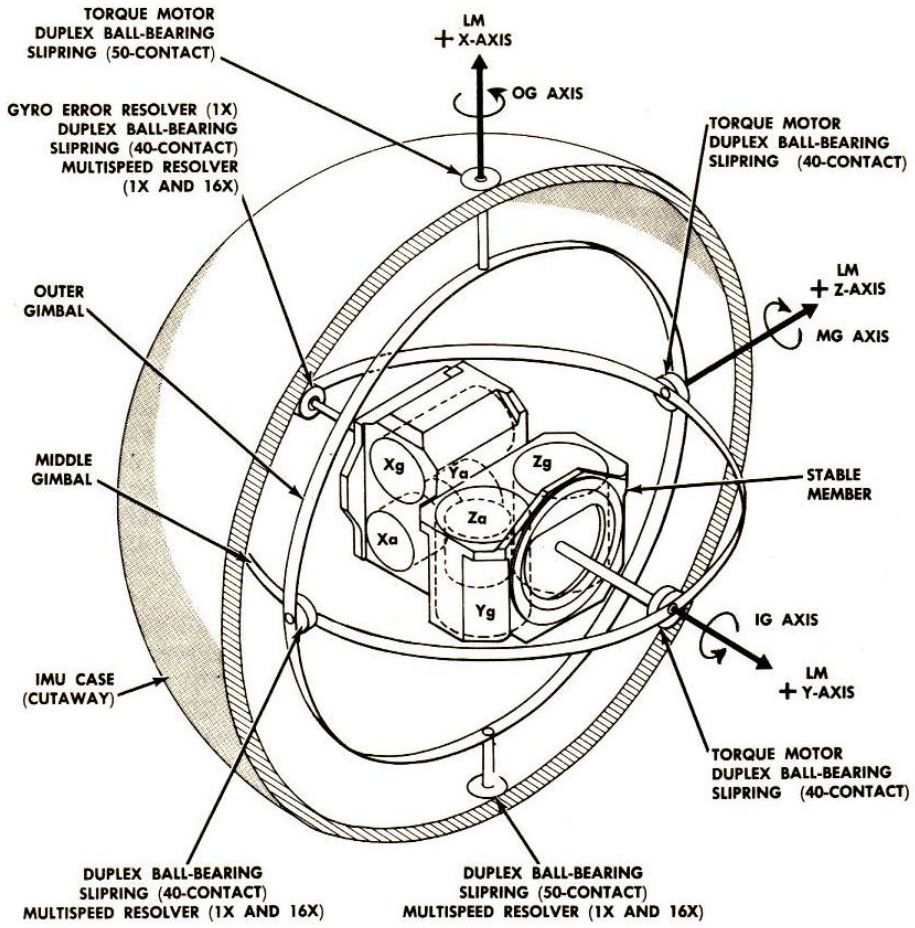


Figure 1.1 : Schéma d'une unité de mesure inertielle avec une structure à cadrant qui permet de déterminer une vitesse et une attitude. Les éléments X_a, Y_a, Z_a représentent respectivement les accéléromètres sur les axes X,Y,Z. De même, les éléments X_g, Y_g, Z_g représentent respectivement le roulis, le tangage et le lacet. (Image de Courtesy NASA[6])

La deuxième structure est telle que les accéléromètres et les gyroscopes sont fixés, en donnant des mesures dans le repère du véhicule. Un algorithme utilisant les vitesses de rotation (cf. Figure 1.2) permet de déterminer la vitesse, la position et l'attitude du véhicule. Ce système est moins précis que la structure à cadran, mais présente l'intérêt d'être moins coûteux et plus résistant aux chocs.

[6] http://www.nasa.gov/

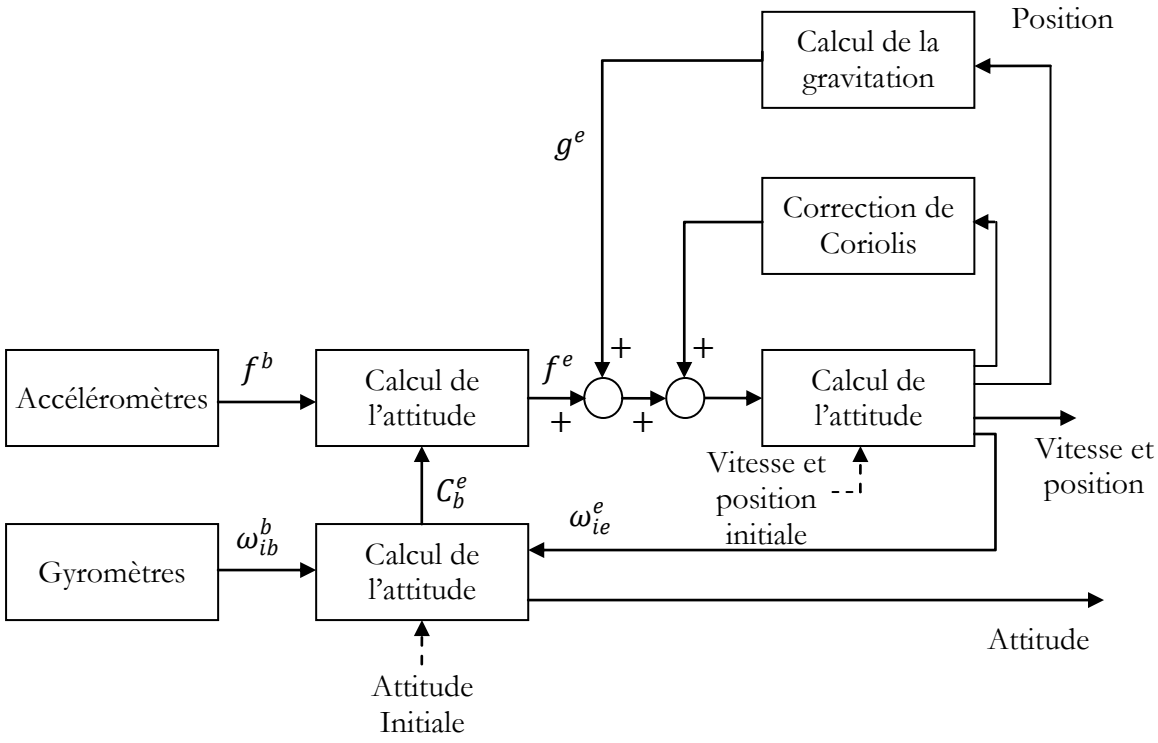


Figure 1.2 : Schéma de l'algorithme de calcul de la position et de la vitesse dans le repère de navigation de l'unité de mesure à structure arrimée.

Les centrales inertielles utilisées et détaillées dans différentes approches [2,29,85] ont principalement les avantages suivants :

- La simplicité d'intégration du capteur dans un robot ou un véhicule ;
- La fréquence d'acquisition (de 100 à 150Hz) des données est élevée, ce qui permet une précision élevée à court terme ;
- L'utilisation possible dans différents environnements et conditions, notamment en aéronautique ;
- De multiples fonctions : déterminer la position, la vitesse, l'accélération et l'orientation.

Elles présentent cependant les inconvénients suivants :

- Comme l'odométrie, une centrale inertielle est un système de localisation relative. Par conséquent, il est nécessaire d'initialiser à nouveau le système relatif par un système absolu. Elle n'est donc pas utilisable seul sur de longs trajets et nécessite des informations provenant d'autres capteurs ;

- Une centrale inertielle doit être initialisée par l'utilisation d'un capteur externe pour déterminer la position et la vitesse initiales;
- Une centrale inertielle est sensible à la gravité.

1.3. La localisation absolue

La localisation absolue permet de déterminer la position d'un agent mobile dans le repère global (lié à l'environnement). La réalisation de ce type de localisation nécessite l'utilisation de capteurs extéroceptifs. Ces capteurs peuvent être différenciés en deux catégories. D'une part, il existe des capteurs dédiés à la localisation permettant de détecter des balises artificielles actives ou passives qui sont connues dans l'environnement. Ils permettent de mesurer une position ou une distance. L'exemple type de cette catégorie de capteurs est le système GPS. D'autre part, il existe des capteurs dédiés à la perception qui réalisent des mesures sur l'environnement proche d'un agent mobile. Ces capteurs traitent différentes sources d'informations (vidéo, données télémétriques, etc.) pour détecter des amers naturels ou artificiels.

1.3.1. Triangulation

La position et le cap d'un véhicule peuvent être déterminés à partir de trois balises. La Figure 1.3 présente le principe de la localisation par triangulation. Soient B_1, B_2 et B_3 trois balises définissant les angles de gisement relatifs α_{21} et α_{32}. Les trois balises ont pour coordonnées $(x_i \;\; y_i)^t$ (avec i variant de 1 à 3) dans le repère (B_1, I, J). La position du véhicule (x, y) est définie par :

$$\begin{aligned} x &= W_y + x_2 \\ y &= -x_2 . \frac{W + cot\, \alpha_{21}}{W^2 + 1} \end{aligned} \tag{1.1}$$

avec :

$$W = \frac{(x_3 - x_2)\, cot\, \alpha_{32} - x_2\, cot\, \alpha_{21}}{x_3} \tag{1.2}$$

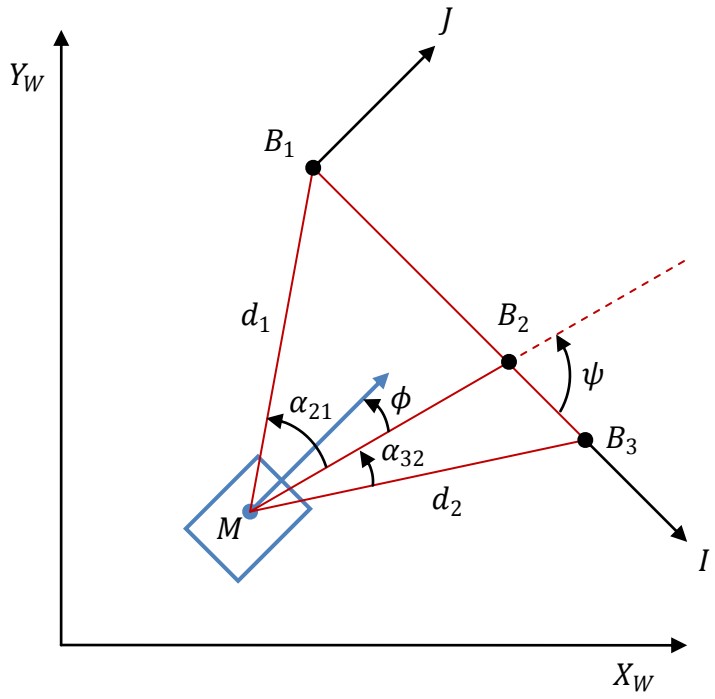


Figure 1.3 : Principe de la localisation par triangulation

Le cap θ est défini par :

$$\theta = \psi + \phi \tag{1.3}$$

avec :

$$\phi = arctan\frac{1}{W} \tag{1.4}$$

1.3.2. Trilatération

La trilatération est une technique permettant de déterminer une position à partir de trois distances. La Figure 1.4 schématise le fonctionnement de cette technique de localisation. L'intersection des cercles (B_1, d_1) et (B_2, d_2) met en évidence deux solutions possibles : M et M' . En ajoutant l'intersection avec un troisième cercle (B_3, d_3) nous obtenons une solution unique pour la position du mobile M.

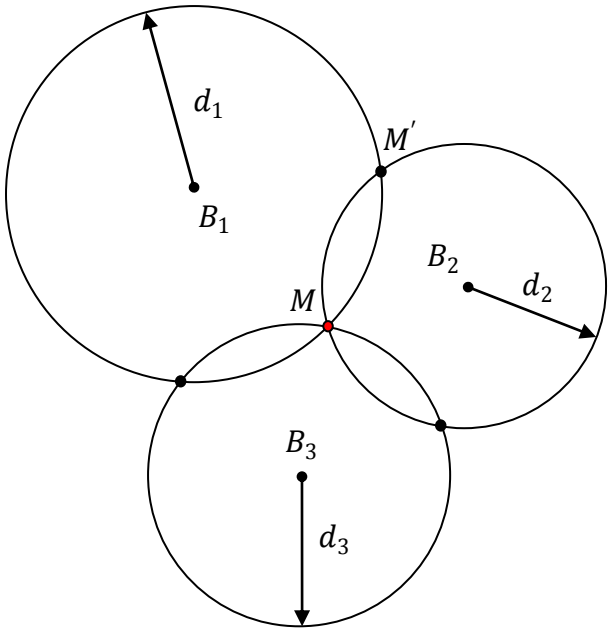


Figure 1.4 : Principe de la localisation par trilatération

La trilatération est une technique très connue. Elle est utilisée dans le système de positionnement par satellite GPS.

1.3.3. Les amers

Les amers sont des repères qui vont permettre au véhicule de se localiser. Cette localisation ne pourra se faire qu'à partir de l'utilisation d'une carte donnant tous les renseignements des amers, que ce soit sa position ou bien ce qui le caractérise. Si nous devons aménager l'environnement, car nous ne trouvons pas d'objet permettant de réaliser le rôle d'amer naturel, nous utiliserons des amers artificiels. Voyons plus en détail ces deux types d'amers.

1.3.3.1. Les amers naturels

Les amers naturels sont des repères fixes terrestres facilement identifiables qui vont permettre au véhicule de se repérer dans son environnement et ainsi de calculer

sa position. On peut identifier différents objets qui serviront d'amer naturel, des exemples de ceux-ci sont :

- Amers visuels (exemple les points de Harris) ;
- Amers télémétriques (exemple les trottoirs).

Ces amers naturels sont utilisés car l'environnement est déjà mis en place pour effectuer le calcul de localisation, nul besoin d'autres éléments. Mais pour utiliser cette méthode pour la localisation, il nous faut pallier un problème qui subsiste. En effet, si nous devons utiliser les amers naturels pour permettre la localisation, encore faut-il connaître la position de ceux-ci. Nous disposons d'une base de données des amers dans le véhicule et si nous n'arrivons pas à discerner l'amer détecté, il faut résoudre ce problème pour que le véhicule soit capable de se localiser et réussisse à associer l'amer capté avec la base de données du véhicule.

1.3.3.2. Les amers artificiels

Les amers artificiels sont aussi des repères facilement identifiables qui vont permettre au véhicule de se repérer dans son environnement, mais à la différence de ceux qui sont naturels, ils ne sont pas terrestres, mais construits ou inventés par l'homme. Nous distinguons dans les amers artificiels 2 types séparés :

- **Les balises actives** : ces balises contiennent de l'électronique qui permet de transmettre un signal aux récepteurs du véhicule. Le véhicule va donc devoir être équipé du nécessaire pour pouvoir recevoir les différents signaux émis des balises. Le signal des balises peut être par exemple sonore (balise acoustique), électromagnétique (radar) ou bien lumineux. La balise active la plus connue de toutes est le satellite GPS qui émet en continu et qui permet de se localiser.

- **les balises passives** : ces balises par rapport à celles qui sont actives ne contiennent pas d'électronique et n'émettent donc pas de signal. Cela est gênant pour le véhicule qui doit alors dans un premier temps détecter la balise avant d'entreprendre tout calcul de position. Le véhicule va donc devoir être équipé du nécessaire pour détecter les balises et ainsi se localiser par la suite. On équipera le véhicule de capteurs extéroceptifs pour la perception des

amers artificiels (télémètre laser, caméras ordinaires, stéréovision, capteurs ultrason,…).

L'utilisation de ces amers artificiels donne une localisation plus fine et robuste que celle avec l'utilisation d'amers naturels. Nous obtenons des résultats d'une bonne précision par l'utilisation des amers artificiels mais ceux-ci présentent des inconvénients indubitables. En effet, l'utilisation de ces capteurs amène à une quantité énorme de données, de plus il faut aménager l'environnement pour les utiliser, ce qui n'était pas le cas des amers naturels, et c'est ce qui paraît inconcevable à grande échelle au vu du coût d'un télémètre par exemple.

1.3.4. Localisation sur carte

1.3.4.1. Présentation

La localisation sur carte est une technique qui permet d'associer une estimation de la position d'un véhicule et de sa trajectoire sur une carte routière numérique. Elle permet à l'automobiliste de savoir sur quelle voie il circule et d'utiliser les informations de guidage du navigateur.

Comme la structure du réseau est complexe, la localisation, qui estime une position ou une trajectoire, est insuffisante. Il faut donc y ajouter des éléments géographiques représentant les routes sur la carte numérique pour compenser les imperfections et les erreurs issues de différentes sources d'information.

La carte routière numérique sera composée d'une base de données géographiques vectorielles 2D incluse dans un SIG (Système d' Informations Géographiques) où les routes sont représentées par des points décrivant l'axe central de la route (connaissance de leur position et orientation associée), les points d'intersection sont modélisés par des nœuds et les points de forme pour décrire la géométrie. On est loin de la réalité et on en obtient ici une vue déformée en raison du déplacement du véhicule sur une surface 3D et non 2D (ce qui est représenté sur la carte). Le problème de cette localisation consiste à savoir où est le véhicule de la manière la plus précise sur cette carte routière numérique. Ainsi, pour se localiser sur une carte numérique, il faut :

- Sélectionner le segment représentant la route sur laquelle le véhicule se situe ;

- Estimer l'abscisse curviligne depuis une des deux extrémités.

Il s'agit d'une tâche complexe. Pour venir à bout de ce problème de localisation, on peut utiliser des algorithmes ou divers moyens. Toutes ces méthodes ont des degrés de technicité différents et permettent d'obtenir des résultats variables. Selon les cas, une méthode conviendra mieux qu'une autre.

Parmi les méthodes les plus connues, on peut citer la méthode floue, le filtrage de Kalman, les méthodes de résolution par les mathématiques (géométrie, probabilité, statistiques) entre autres.

1.3.4.2. Marquage au sol

Le marquage au sol va servir à délimiter et donc à détecter les routes. On peut recenser différentes méthodes pour reconnaître le marquage au sol. En effet, on peut utiliser la méthode de détermination de contours pour détecter les bords de route avec les ombres portées ou bien d'autres méthodes de recherche de segments forts sur l'image montrant la route à emprunter. Le marquage au sol pour la localisation est une méthode souvent utilisée. Il est très facile de détecter les bords de route à travers l'image de par le contraste de ceux-ci par une segmentation de contours.

Des méthodes de détection de contours ont été réalisées à partir du changement de l'intensité lumineuse. Un exemple de méthode très utilisée est la transformée de Hough [13] qui va permettre de reconnaître dans l'image les lignes qui constituent la bordure de la route.

La difficulté à identifier les contours grâce au marquage au sol dépend des modèles utilisés qui peuvent être plus ou moins compliqués selon la profondeur de champ. D'autres méthodes recherchent quels éléments varient selon les images obtenues et ainsi identifient des points ou droites en comparant deux à deux les images grâce à un algorithme.

1.3.4.3. Localisation à partir d'amers sur une carte

Pour pouvoir utiliser ces amers, nous sommes obligés d'utiliser des cartes. Celles-ci sont de deux types :

- Carte locale : elle contient des amers situés localement dans une zone géographique proche ;
- Carte globale : elle contient des amers qui pourront être utilisés sur toute la Terre.

Ces cartes peuvent nous renseigner, d'une part, sur la position exacte des amers par leurs coordonnées et leur représentation par des objets ponctuels. D'autre part, elles nous livrent des informations sur ces derniers comme leur hauteur par rapport au sol.

Dans notre cas, nous n'utiliserons que des cartes locales. Il faut tout d'abord faire des acquisitions d'images sur un parcours pour en faire un traitement afin de définir les amers. Il est évident que le calcul de position et de trajectoire par le biais d'une carte globale amènerait une quantité énorme d'informations à traiter. Or, dans notre cas, le véhicule est équipé d'une mémoire limitée et n'a pas la capacité à gérer autant de données. Il est donc judicieux de placer les amers sur une carte locale qui va permettre d'obtenir le minimum d'informations nécessaires à traiter.

Afin de pouvoir utiliser cette carte d'amers, il va être nécessaire de juxtaposer celle-ci avec une carte de navigation pour pouvoir se diriger (cf. Figure 1.5 et Figure 1.6).

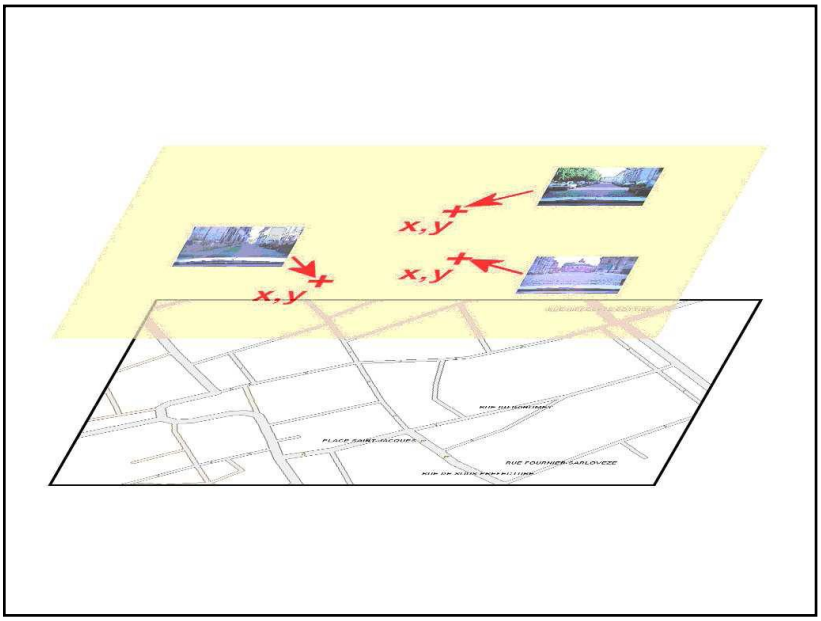


Figure 1.5 : Juxtaposition d'une carte de navigation et d'une carte d'amers

Plusieurs approches peuvent être utilisées selon le type d'amers :

- L'utilisation d'amers en tant qu'images clés : ici, nous allons à partir de plusieurs images retrouver la position 3D du véhicule grâce à la détection de points d'intérêt sur ces images. A chaque image correspond une position estimée du véhicule et différentes informations comme les paramètres de la pose de la caméra, les matrices de projection associées, les coordonnées homogènes pour chaque point dans un référentiel commun.
- L'utilisation d'amers en tant qu'images clés comme précédemment mais ici nous ne prenons que l'information 2D et non plus 3D. Les coordonnées dans l'image suffisent pour la position. On extrait les points Harris (cf §2.2) communs entre les images, ce qui permet au véhicule d'identifier le déplacement à réaliser.
- L'utilisation d'amers plans : cette fois-ci, on détecte également les points d'intérêt 3D, mais contenus dans un même plan d'équation connue. A ces différents points, on va associer une image caractéristique.

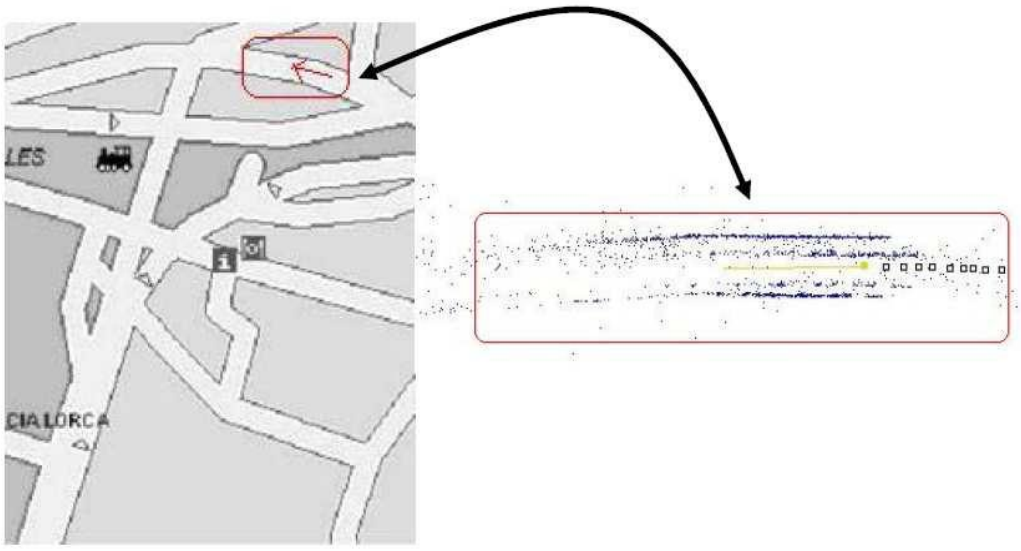


Figure 1.6 : Utilisation d'une carte d'amers pour identifier la position d'un mobile.

1.3.5. Les systèmes de géo-référencement satellitaire

GLONASS, Compass, IRNSS, QZSS sont des systèmes bien moins connus en Europe que le leader des systèmes de géo-référencement satellitaire GPS et le futur projet système européen Galileo. Pourtant, ces systèmes ont la même fonction : localiser.

1.3.5.1. Introduction

Le système américain de géo-référencement appelé GPS signifiant « Global Positioning System » traduit en français par « Système de positionnement mondial » est un système de positionnement par satellites. Ce système est capable de donner n' importe où sur le globe la position et la vitesse d'un mobile avec une précision de quelques dizaines de mètres et ceci à tout moment avec une précision temporelle de l'ordre de la microseconde. C'est le principal système totalement opérationnel qui est actuellement utilisé par tous. Il permet de déterminer les coordonnées géographiques d'un point situé n'importe où dans le monde 24 h sur 24 h. Grâce à la mesure de distances depuis les satellites, qui émettent en permanence des informations codées, il est possible d'identifier ces positions géographiques.

Ce système a été théorisé par le physicien D. Fanelli et mis en place par le département de la Défense des Etats-Unis d'Amérique dans un cadre strictement militaire. Ce projet de recherches a été conçu uniquement pour les besoins propres des Etats-Unis. Mais le système a été rendu public à partir de 1985.

C'est ainsi que le système GPS a connu un énorme succès dans le domaine civil et engendré un énorme développement commercial dans de nombreux domaines comme la localisation, la navigation routière ou maritime, les déplacements pédestres, etc. L'utilisation du système GPS est omniprésente dans notre vie, ce qui en fait un des produits les plus populaires au monde. Nous pouvons voir que même dans le milieu scientifique, le GPS a été propice au développement et à l'exploitation des propriétés des signaux transmis pour de nombreuses applications telles que la géodésie, l'étude de l'atmosphère, ...

1.3.5.2. Composantes du GPS

Le GPS est constitué de trois parties distinctes appelées aussi segments :

- Segment spatial ;
- Segment de contrôle ;
- Segment utilisateur ;

1.3.5.2.1. Segment spatial

C'est une partie constituée actuellement d'une constellation de 31 satellites (cf. Figure 1.7). Ces satellites évoluent sur 6 plans orbitaux ayant une inclinaison d'environ 55° sur l'équateur. Ces satellites effectuent le tour de la Terre à une altitude assez basse aux alentours de 20 000 km qu'ils parcourent en 12h, soit une demi-journée.

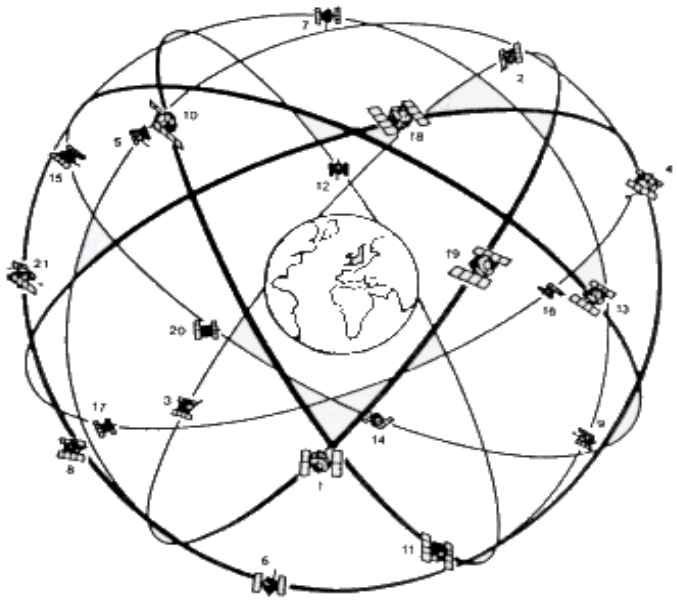


Figure 1.7 : Constellation du segment spatial, aujourd'hui composé de 30 satellites (image extrait de Courtesy Hans Toft)

Les satellites sont des émetteurs qui envoient en permanence des informations sous forme d'ondes électromagnétiques L1 et L2 de fréquences 1,6 et 1,2 GHz. Celles-ci sont constituées de plusieurs informations dont :

- Leur position orbitale ;
- Les heures exactes d'émission des messages ;
- La position des autres satellites.

Pour qu'un récepteur GPS sur terre puisse calculer sa position, il doit recevoir des signaux d'au moins trois satellites à un instant donné (cf. Figure Figure 1.8).

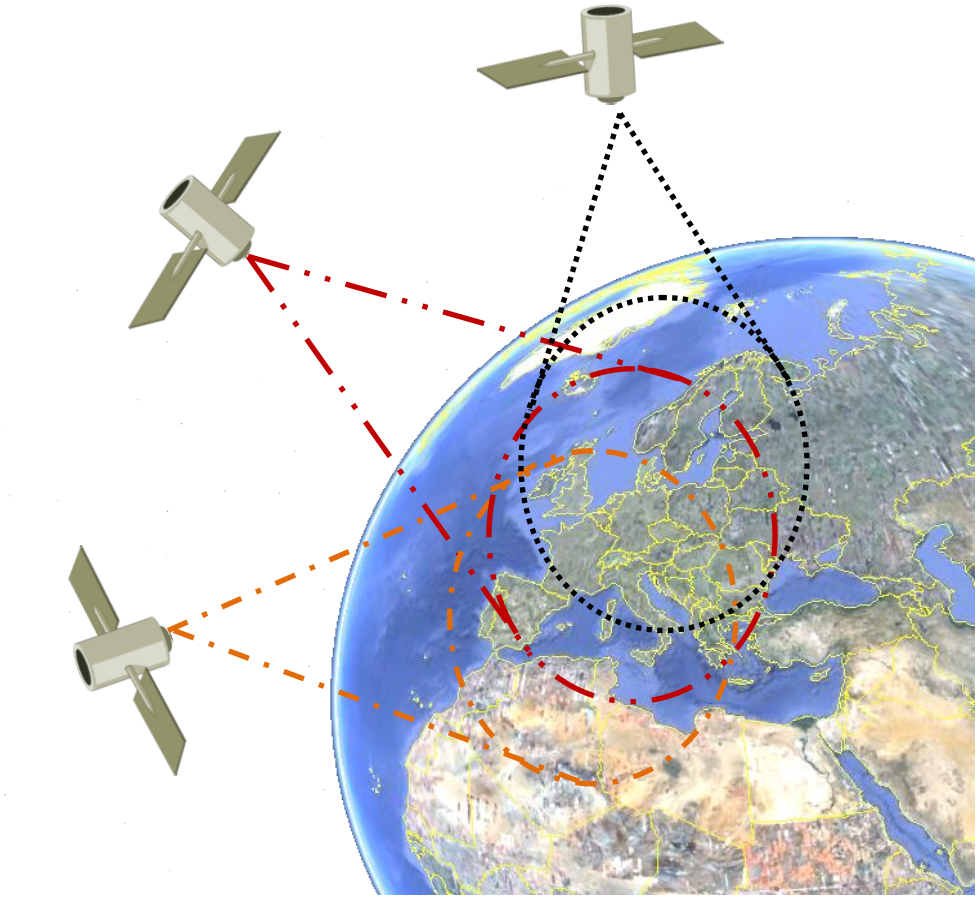


Figure 1.8 : Combinaison de trois satellites pour le géopositionnement (image de la planète extrait de Google Earth)

1.3.5.2.2. Segment de contrôle

Le segment de contrôle permet de piloter et de surveiller le système. Il est composé de 5 stations de surveillance au sol (voir Figure 1.9) réparties tout au long de l'équateur. Leur fonction est de contrôler les satellites GPS. Le segment de contrôle reçoit les signaux des satellites, les analyse et met à jour les informations transmises par ceux-ci (éphémérides, paramètres d'horloge) tout en contrôlant leur bon fonctionnement. La station principale de contrôle est située au Colorado.

Figure 1.9 : Stations de contrôle au sol des satellites du GPS

1.3.5.2.3. Segment utilisateur

Ce segment regroupe l'ensemble des utilisateurs civils et militaires (marine, armée,…) qui ne font que recevoir et exploiter les informations des satellites en utilisant leur récepteur GPS.

1.3.5.3. Fonctionnement du GPS

Les positions géographiques sont déterminées à partir des distances qui séparent un récepteur GPS et les différents satellites. Ceux-ci émettent en permanence des informations codées.

1.3.5.3.1. Signal émis

Chaque satellite possède une horloge atomique et émet à deux fréquences élevées en simultané en bande L de L1=1575.42 MHz et L2=1227.6 MHz. Malheureusement, pour pouvoir utiliser ces fréquences, il faut que le récepteur soit dans une zone dégagée.

Trois types de signaux sont émis (cf. Figure 1.10) :

- Un message de navigation avec l'almanach du système (état, identification, positions, temps), sur L1 à un faible débit de 50Hz. Le message complet faisant 1500 bits sera donc transmis en 30 secondes.

- Un code C/A. (Coarse/Acquisition) qui module L1, répété toutes les millisecondes, permettant la mesure de la distance. Ce code a une longueur de 1023 bits, il est émis à 1.023 Mbits/s et dure donc 1 seconde.
- Un code P (Precision) qui module L1 et L2, à intervalles longs et réservé uniquement aux utilisateurs privilégiés du GPS. Ce code a une longueur bien supérieure et il est émis à une fréquence 10 fois plus grande. Sa durée est de 7 jours. Les clients utilisent des clés de décryptage.

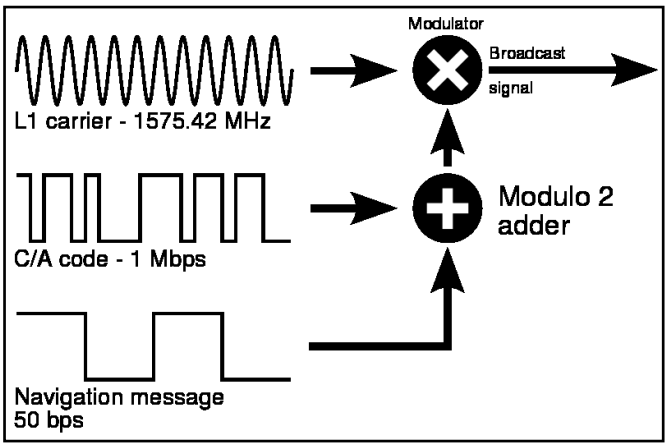


Figure 1.10 : Types de signaux émis par le système GPS (source : Wikipédia)

1.3.5.3.2. Principe de localisation

Le principe du positionnement GPS est très proche du principe de triangulation. La constellation a été conçue de telle manière que partout sur Terre, on puisse voir au moins 3 satellites à tout moment. La vitesse de transmission des signaux émis par les satellites est égale à celle de la lumière. Chaque satellite envoie deux éléments :

- Son numéro d'identification ;
- L'éphéméride avec l'heure exacte d'émission du signal ;

A partir de l'horloge du récepteur supposée synchronisée sur celle des satellites et la vitesse connue des signaux, on détermine le temps de propagation du signal, ce qui

permet de déduire la distance au satellite. De là, on définit ainsi des sphères centrées sur des satellites et dont l'intersection donne la position.

Le récepteur GPS identifie chaque satellite qu'il utilise grâce au signal pseudo aléatoire émis par chaque satellite. Il charge, à l'aide de ce signal, les informations sur l'orbite et la position du satellite.

Lorsque l'on veut déterminer l'altitude en plus de la latitude et de la longitude, on utilise un quatrième satellite. Plus celui-ci est proche de la verticale du point où se situe le récepteur, plus l'altitude sera une mesure fiable. Plus le nombre de satellites utilisés pour la localisation est grand, plus la précision de la localisation est bonne. Le positionnement de la constellation des satellites permet d'utiliser jusqu'à 12 satellites au même moment.

1.3.5.4. Les autres systèmes de géo-référencement satellitaire

Parmi les systèmes de géo-référencement satellitaire nous pouvons citer cinq autres systèmes : Galileo, GLONASS, Compass, IRNSS et QZSS. Voyons brièvement leurs spécificités.

1.3.5.4.1. Galileo

Lancé par l'Union Européenne, ce système se compose de 30 satellites qui seront mis en orbite à 23616Km avant la fin 2013. Ce nouveau système sera compatible avec le GPS pour améliorer la précision et la sécurité du système en cas de défaillances de l'un des deux. Par rapport aux autres systèmes, Galileo offre une couverture mondiale en zones urbaines et des messages sur les erreurs de fonctionnement ainsi que la fiabilité du signal. Galileo garantit également un fonctionnement permanent, il ne peut pas être interrompu pour des besoins géopolitiques d'un pays.

1.3.5.4.2. GLONASS

Cet acronyme russe signifiant Système Global de Navigation par Satellite est le nom du système de positionnement utilisé actuellement par la Russie. Né dans les années 1980, ce système est opérationnel depuis 1996 avec 24 satellites mis en orbite à 19130 km. Depuis 2008, seuls 17 satellites sont fonctionnels, mais le programme de

l'agence spatiale fédérale russe prévoit le déploiement à terme de 30 satellites pour couvrir tout le globe en 2011.

1.3.5.4.3. Compass

Ce système de positionnement chinois aussi appelé Beidou doit, dans un premier temps, fournir des services régionaux pour 2011 avec une constellation de 12 satellites. A terme, la Chine veut se munir d'un système de localisation globale composé de 35 satellites pour 2020. Aujourd'hui, seuls 2 satellites sont en orbite. Ce système permettra d'améliorer fortement la précision du positionnement du système de navigation globale. Par ailleurs, des négociations sont en cours afin d'apporter une compatibilité avec les autres systèmes pour proposer un service de localisation civil, fiable et précis de 120 satellites.

1.3.5.4.4. IRNSS

Construit et contrôlé par le gouvernement indien, IRNSS (Indian Regional Navigational Satellite System) est un système de positionnement régional. Composé de 7 satellites, il permet de déterminer une position absolue avec une précision de 20 mètres en l'Inde. Ce système est strictement réservé à l'Inde et développé dans ce pays.

1.3.5.4.5. QZSS

L'acronyme QZSS (Quasi-Zenith Satellite System) désigne le système de positionnement régional japonais. Basé sur trois satellites géostationnaires, ce système complémentaire au GPS permet d'améliorer sa précision sur le Japon et sa région. Ce système sera opérationnel pour 2013 avec un premier lancement de satellite prévu pour 2010.

1.4. La localisation hybride

La localisation hybride va consister à estimer la position du véhicule et sa trajectoire à partir de plusieurs capteurs extéroceptifs (GPS, caméra,…) mais aussi proprioceptifs (capteur d'angle au volant, odomètre,…). Cette technique va permettre de pallier les inconvénients de la localisation absolue et relative, et d'envisager de

prendre les avantages de chacun des deux types de localisation pour trouver une meilleure solution, que ce soit en terme de précision et de fiabilité pour la localisation du véhicule.

En effet, que ce soit la localisation absolue ou bien la localisation relative, on obtient des mesures qui ne sont pas parfaites. Elles peuvent effectivement être erronées ou incomplètes. On peut voir, à travers les points suivants montrant les avantages et les inconvénients de ces deux techniques, qu'il est possible de les combiner afin de produire une solution de localisation bien meilleure.

	Inconvénients	**Avantages**
Localisation relative	▪ Accumulation d'erreurs avec la distance ▪ Utilisation d'un repère attaché à la position initiale du véhicule	▪ Autonomie vis-à-vis de l'environnement ▪ Fonctionnement à fréquence élevée
Localisation absolue	▪ Aménagement parfois nécessaire de l'environnement ▪ Fonctionnement à très faible fréquence	▪ Utilisation d'un repère attaché à l'environnement ▪ Erreurs indépendantes de la distance

On peut voir à travers ce comparatif que les deux techniques de localisation se complètent parfaitement. Par exemple, la distance est une cause d'erreurs pour la localisation relative, mais pas pour la localisation absolue.

L'utilisation d'une localisation hybride va donc permettre de cumuler les différents avantages des techniques de localisation et de pallier les différents défauts qu'elles présentent chacune. La fusion des données des différents capteurs sera réalisée par le biais de l'utilisation du filtre de Kalman [28] ou bien d'un filtre particulaire.

1.5. Conclusion

A travers ce chapitre, nous avons pu dresser un état de l'art de la localisation de robots mobiles à partir de différents capteurs. La localisation de robots mobiles reste de nos jours un problème complexe qui tend à être résolu par les méthodes hybrides qui pallient les défauts de la localisation relative (erreurs dues à la distance, repère attaché à la position initiale du robot mobile) ainsi que ceux de la localisation absolue (faible fréquence, aménagement de l'environnement).

La précision nécessaire à la localisation de robots mobiles doit être de l'ordre du décimètre, voire du centimètre. Les techniques les plus récentes de systèmes de géo-référencement satellitaire permettent d'atteindre une telle précision, mais un problème subsiste puisque ces systèmes ne sont pas accessibles de manière continue en milieu urbain.

En effet, grâce aux progrès au niveau des technologies de la réalité virtuelle, il est possible d'utiliser une nouvelle source d'informations, à savoir un modèle virtuel 3D de l'environnement qui va être reconstruit par l'acquisition d'images par stéréovision. Notre étude se place donc dans un contexte de localisation globale d'un véhicule en milieu urbain. L'objectif est de localiser un véhicule à partir des seules informations extraites d'un capteur stéréoscopique.

CHAPITRE 2

LA LOCALISATION PAR VISION ARTIFICIELLE

2.1. Introduction

La vision par ordinateur est un domaine de recherches très actif qui n'a pas cessé d'évoluer depuis les années 40. On retrouve de nos jours des systèmes d'imagerie très avancés qui contribuent à différents domaines d'activités tels que l'industrie (contrôles, mesures de qualité), la production (ligne de fabrication automatisée), la communication (télévision numérique, cinéma 3D, réalité virtuelle) ou encore la médecine (scanners, échographie, IRM).

Au fur et à mesure des grandes avancées réalisées par la vision artificielle, de nombreuses applications ont vu le jour, mais l'évolution de nos moyens de communication et de technologie pose de nouveaux problèmes et contraintes. D'un point de vue global, la vision par ordinateur peut être considérée comme un processus de traitement de l'information [42]. Il convient alors de déterminer la forme de ces informations et leur représentation pour faciliter l'extraction et l'interprétation de ces données.

Toute application définit sa propre manière de représenter les informations, mais le processus de traitement et d'analyse est identique et peut être décomposé selon les étapes suivantes [64] :

- Extraction des informations pertinentes de l'image ;
- Mesures des propriétés de la représentation des informations ;
- Corrélation entre les mesures extraites ;
- Interprétation du contenu de l'image ;

Ces étapes sont nécessaires à toute application afin d'extraire les informations représentant des caractéristiques précises. Celles-ci sont généralement employées afin de résoudre un problème fondamental, qui est l'appariement ou la mise en correspondance d'éléments extraits d'images. L'association de plusieurs sources d'informations (images, primitives géométriques, régions, …) permet de réaliser des tâches complexes telles que la reconstruction de structure 3D, la reconnaissance d'image, le suivi de mouvements, l'estimation de la pose d'une vue ou bien de distances, etc.

La mise en correspondance d'images permet de réaliser ce que le système visuel humain effectue instinctivement. Il s'agit d'établir une association entre les objets extraits des images et les objets en mémoire. Agissant comme le système binoculaire

humain, un système stéréoscopique permet de déterminer la structure 3D ou le positionnement spatial de l'objet d'une scène.

Cependant, pour réaliser des tâches complexes telles que la localisation 3D, la mise en correspondance doit résoudre le problème de la perte de propriétés (angle, distance, …) liée à la projection d'informations tridimensionnelles (univers 3D) dans un plan bidimensionnel (image 2D). De plus, un objet 3D de l'univers peut avoir une même projection 2D et inversement un objet 3D peut avoir plusieurs projections 2D, en fonction du point de vue.

Ces contraintes nécessitent l'extraction de primitives caractéristiques invariantes aux transformations entre chaque image. De plus, il faut pouvoir distinguer chaque primitive en associant un descripteur robuste permettant d'identifier un correspondant.

Dans le cas de la stéréovision, le problème de la mise en correspondance est généralement simplifié. En effet, les variations entre les deux images stéréoscopiques sont généralement faibles. Ainsi, les variations autour du voisinage de chaque primitive caractéristique sont faibles entre les images, ce qui permet l'utilisation de méthodes de corrélation. Dans le cas d'un système stéréoscopique calibré, nous pouvons utiliser la géométrie épipolaire pour réduire l'espace de recherche des correspondants entre les deux images. Cependant, un changement de point de vue rend difficile la mise en correspondance et nécessite des méthodes plus robustes. Celles-ci sont basées sur l'utilisation d'invariants locaux.

2.2. Les extracteurs de points

Les premiers travaux sur les points d'intérêt sont issus de Moravec [49]. Le détecteur de Moravec est un détecteur de « coin » qui met en évidence les fortes variations bidirectionnelles dans le voisinage d'un point. Ces fortes variations constituent les caractéristiques principales de ce point d'intérêt que l'on retrouve dans les intersections de contours dans l'image.

Harris et Stephens [24] améliorent le détecteur de Moravec en calculant les courbures principales de la fonction d'autocorrélation du signal (cf. équation (2.1)). La mesure des variations locales de l'image I au point $\mathrm{x} = (x, y)^t$, associé à un déplacement $\Delta\mathrm{x} = (\Delta x, \Delta y)$, est donc fournie par :

$$\chi(x) = \sum_{x \in W} \left(I(x) - I(x + \Delta x)\right)^2 \tag{2.1}$$

où W est une fenêtre centrée sur le point x ;

En utilisant une approximation du premier ordre :

$$I(x + \Delta x) \simeq I(x) + \left(\frac{\delta I}{\delta x}(x) \quad \frac{\delta I}{\delta y}(x)\right).\Delta x$$

On obtient :

$$\begin{aligned}\chi(x) &= \sum_{x \in W} \left[\left(\frac{\delta I}{\delta x}(x) \quad \frac{\delta I}{\delta y}(x)\right).\Delta x\right]^2 \\ &= \Delta x^t M(x) \Delta x\end{aligned}$$

où $M(\mathrm{x})$ représente la variation locale de l'image de I en x :

$$M(x) = \begin{pmatrix} \sum_{(x_k, y_k) \in W} \left(\frac{\delta I}{\delta x}(x_k, y_k)\right)^2 & \sum_{(x_k, y_k) \in W} \frac{\delta I}{\delta x}(x_k, y_k).\frac{\delta I}{\delta y}(x_k, y_k) \\ \sum_{(x_k, y_k) \in W} \frac{\delta I}{\delta x}(x_k, y_k).\frac{\delta I}{\delta y}(x_k, y_k) & \sum_{(x_k, y_k) \in W} \left(\frac{\delta I}{\delta y}(x_k, y_k)\right)^2 \end{pmatrix}$$

Si, pour tout déplacement $\Delta \mathrm{x}$, la quantité $\chi(\mathrm{x})$ est grande, alors le point $\mathrm{x} = (x, y)$ est considéré comme un point d'intérêt. En d'autres termes, les points d'intérêts sont les points pour lesquels la matrice d'autocorrélation présente deux valeurs propres élevées.

La mesure d'autocorrélation est généralement estimée par les dérivées premières calculées sur un support gaussien $G(\sigma_I)$ de dimension σ_D :

$$M(x, \sigma_I, \sigma_D) = G(\sigma_I) \otimes \begin{bmatrix} I_x^2(x, \sigma_D) & I_x(x, \sigma_D).I_y(x, \sigma_D) \\ I_x(x, \sigma_D).I_y(x, \sigma_D) & I_y^2(x, \sigma_D) \end{bmatrix} \tag{2.2}$$

avec : σ_D est l'écart type de la gaussienne utilisée pour calculer les dérivées de l'image, il définit la taille de la fenêtre de dérivation ; σ_I est la taille de la fenêtre d'intégration.

Pour éviter le calcul des valeurs propres de la matrice M, Harris et Stephens proposent [24] de calculer la réponse k_H du détecteur :

$$k_H = Det(M) - \alpha.Trace^2(M) \quad (2.3)$$

α est un coefficient déterminé de manière empirique. Dans [24], α prend la valeur 0.04.

Avec cette approche, les points d'intérêt sont les points correspondant aux maxima locaux de l'opérateur k_H. L'utilisation pratique du détecteur de Harris est détaillée dans la Table 2.1.

De nombreuses applications utilisent le détecteur de Harris qui s'est développé également sur les images couleurs par Montesinos et Gouet [48,21]. Malgré son adaptation dans de nombreuses applications de reconstruction et de localisation [77,87], le détecteur de Harris montre ses limites lorsque le point de vue entre deux images est différent. En effet, le détecteur de Harris est très sensible aux changements d'échelle. Ainsi, lors de la mise en correspondance des images, les fenêtres de corrélation ne couvrent pas la même surface, ce qui ne permet pas d'obtenir des résultats satisfaisants. Etudié dans [12], la perte de performance du détecteur de Harris entre deux points vue éloignés est induite par la projection d'une scène tridimensionnelle qui ne conserve ni les angles ni les distances ni les formes. Les zones autour des points homologues ne sont donc pas de même forme, ce qui diminue la qualité des appariements en utilisant les algorithmes par corrélation.

Pour s'affranchir de ces difficultés, il convient d'utiliser un détecteur invariant aux changements d'échelle et aux transformations géométriques du point de vue (rotations, translations).

Algorithme Harris

Objectif : Extraire les « coins » d'une image I.

Algorithme :

1. Calcul des gradients de l'image I en x et y notés respectivement I_{dx} et I_{dy}.
2. Calcul des images I_{dx}^2, I_{dy}^2 et le produit $I_{dx}I_{dy}$.

3. Lissage gaussien des images I_x^2, I_y^2 et $I_x . I_y$.
4. Pour chaque pixel :
   4.1. Calculer la matrice des variations locales M (cf. équation (2.2)) :

$$M = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}$$

   4.2. Calculer la réponse du détecteur (cf. équation (2.3)) avec :

$$k_H = M_{11}M_{22} - M_{12}M_{21} - 0.04 * (M_{11} + M_{22})^2$$

5. Extraire les maxima locaux de la fonction k_H.
6. Extraire les N meilleurs points de Harris. Généralement une condition de proximité entre les points est ajoutée pour éviter d'avoir un ensemble de points trop concentrés sur la même partie de l'image.

Table 2.1 : Algorithme d'extraction des points de Harris

2.3. Approches par invariants locaux

Les premiers travaux sur les approches par invariants locaux portent sur les techniques multi-échelles pour estimer l'échelle entre deux images. Hanson et Morse [22] proposent de calculer la trace d'échelle comme étant l'ensemble des valeurs calculées sur différents niveaux de résolution consécutifs en utilisant des filtres gaussiens. De même, Dufournaud *et al.* proposent [14] une méthode robuste d'extraction de points et de descripteurs à plusieurs niveaux d'échelle permettant de sélectionner le facteur d'échelle entre deux images. Ces deux approches nécessitent d'apparier sur plusieurs échelles de chaque image les points détectés. Le processus de mise en correspondance devient alors complexe, en particulier en terme de coût calculatoire.

L'approche de Lindeberg [36] appelée « automatic scale detection » permet de sélectionner l'échelle de chaque point d'intérêt automatiquement. Son principe est de calculer sur plusieurs niveaux d'échelle les points d'intérêt et de sélectionner ceux ayant une mesure locale maximale, comme le Laplacien dans une dimension d'échelle

définie. Lindeberg [35] a, par ailleurs, montré que des structures caractéristiques sont liées aux extrema locaux des dérivées normalisées dans l'espace échelle.

Cette méthode permettant de détecter l'échelle caractéristique de chaque point est la base de différents détecteurs [1,38,43,44,70,81]. Les derniers travaux sur les invariants locaux, notamment ceux de Schmid et Mohr [72], démontrent une grande efficacité et robustesse pour l'appariement de primitives extraites de vues différentes d'une même scène. Les descripteurs issus de ces travaux assurent une robustesse face aux changements de fond, aux occultations de point de vue et d'échelle. Parmi ces descripteurs, nous pouvons citer le descripteur SIFT (Scale Invariant Feature Transform) introduit par Lowe [39]. Il est considéré actuellement comme étant le descripteur le plus performant [47].

2.3.1. Le descripteur SIFT

Le Scale Invariant Feature Transorm (SIFT) est une méthode proposée par Lowe [40]. Les primitives extraites de l'image par cette méthode ont pour particularité d'être invariantes aux changements d'échelle et aux rotations. Elles sont partiellement invariantes aux changements d'illumination et aux changements 3D du point de vue de la caméra. De type point, ces primitives sont localisées à la fois dans les domaines spatial et fréquentiel, rendant aussi la mise en correspondance moins sensible aux occlusions et aux bruits dans l'image. De plus, ces primitives permettent d'effectuer une recherche aisée des correspondances d'une primitive dans une base d'images représentant un même environnement, avec une probabilité élevée de trouver les bons appariements.

La génération de l'ensemble des primitives d'une image pour la méthode SIFT se décompose en 4 parties :

1. **Détection des extrema dans l'espace échelles** : Cette première étape est effectuée sur l'ensemble de l'image pour détecter les points qui sont stables dans les espaces échelles pour obtenir une invariance aux changements d'échelle de l'image.

2. **Localisation des points clés** : Pour chaque extrema détecté, des conditions supplémentaires sont appliquées afin de retenir les points clés et d'améliorer leur répétitivité aux changements de point de vue.

3. **Calcul de l'orientation** : Afin de rendre les points clés invariants aux rotations, une ou plusieurs orientations sont attribuées à chaque point clé en calculant la direction locale des gradients autour de celui-ci.

4. **Calcul des descripteurs** : Les gradients locaux autour de chaque point clé sont calculés pour le niveau d'échelle sélectionné, puis transformés dans un histogramme d'orientation pour obtenir un vecteur caractéristique de dimension 128 invariant aux changements d'illumination.

Cette approche génère un grand nombre de points qui couvrent toute l'image sur l'ensemble des niveaux. Ceci permet d'identifier de petits objets dans une image composée de plusieurs éléments.

2.3.1.1. Détection des extrema dans l'espace échelle

La première étape pour l'extraction des points clés est d'identifier les positions et les niveaux d'échelle qui sont associés de manière répétitive sur un même objet dans différentes vues. Cette étape consiste à détecter les points qui sont stables dans l'espace échelle. Pour calculer l'espace d'échelle, Lowe [38] définit la fonction $L(x, y, \sigma)$ (cf. Figure 2.1) comme étant le produit de la convolution d'une Gaussienne $G(x, y, \sigma)$ et d'une image $I(x, y)$:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \tag{2.4}$$

où $*$ est l'opérateur de convolution en (x, y) ; et :

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \tag{2.5}$$

Pour réaliser la détection des points stables, Lowe [39] propose d'utiliser les extrema locaux de l'opérateur DoG (Difference of Gaussian) dans l'espace d'échelle. La DoG est déterminée par la différence de deux espaces proches séparés par une constante multiplicative k :

$$DoG(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \tag{2.6}$$

donc :

$$DoG(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma) \quad (2.7)$$

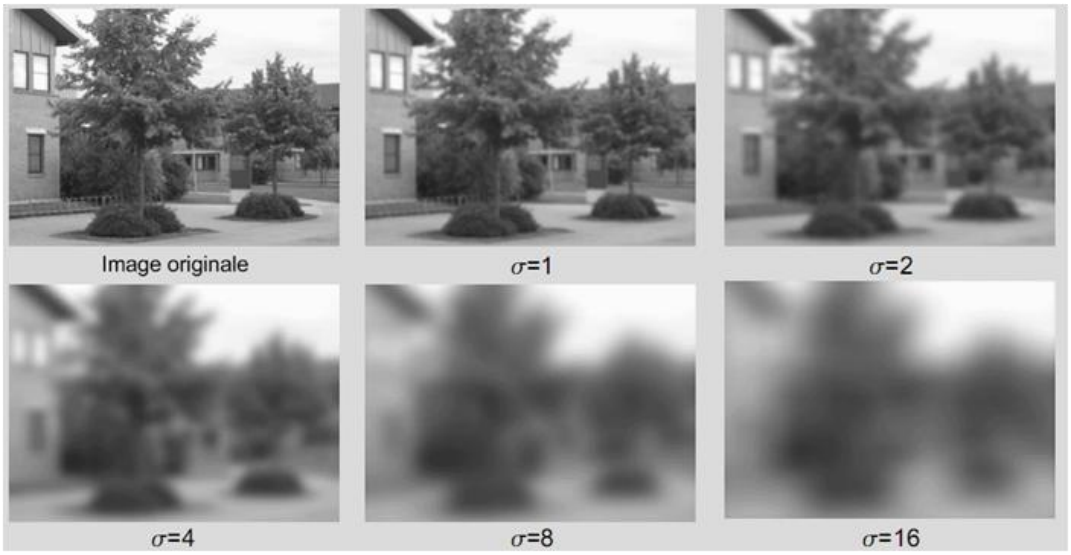


Figure 2.1 : Exemple des niveaux d'échelle pour une octave.

Le choix de l'opérateur DoG est judicieux puisqu'il fournit une bonne approximation de l'opérateur normalisé en échelle du Laplacien de Gaussienne, $\sigma^2\nabla^2\mathrm{G}$, décrit par Lindeberg [34]. De plus, s'agissant d'une soustraction d'images, la DoG est rapide à calculer.

La Figure 2.2 présente le schéma de construction des espaces d'échelle et le calcul de la DoG. On répète à l'image initiale la convolution avec une gaussienne pour réaliser des images séparées par un facteur d'échelle k vu dans la pile à gauche de la Figure 2.2.

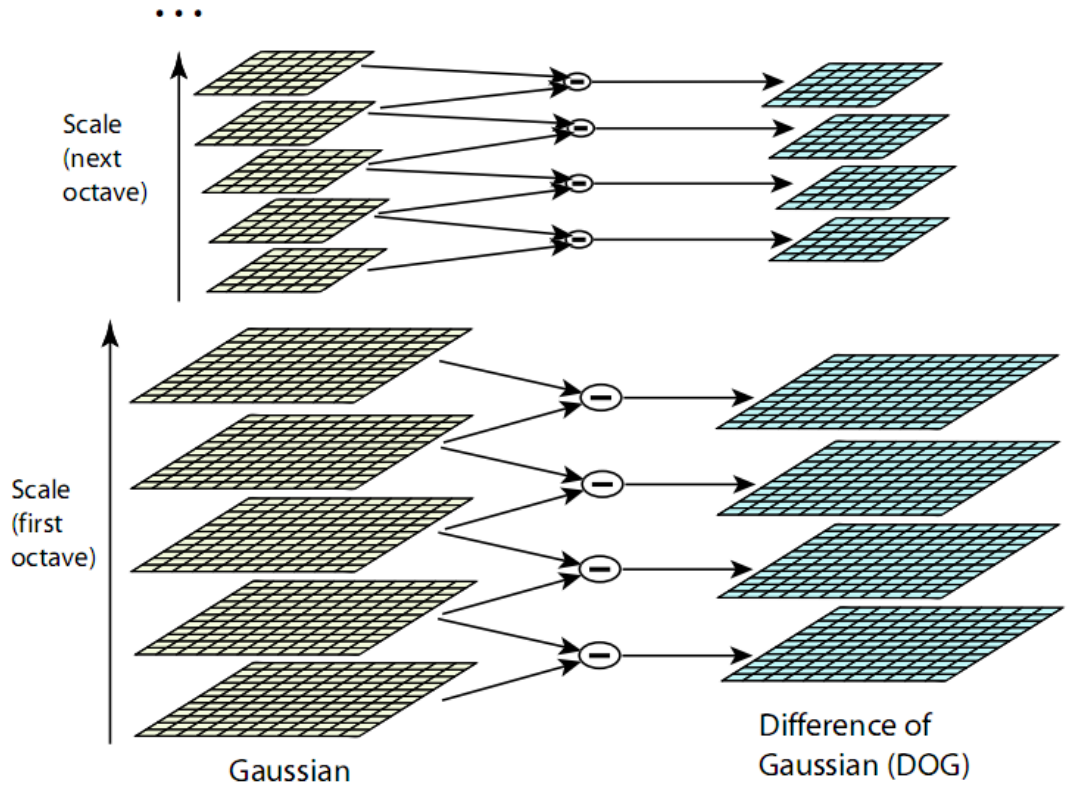


Figure 2.2: Diagramme des images lissées à différents espaces d'échelle et calcul des images issues de l'opérateur DoG. Image extraite de l'artice de Lowe [38].

2.3.1.2. Localisation des points clés

Les points d'intérêt, appelés points clés, sont identifiés comme étant les maxima locaux de la DoG entre les échelles (cf. Figure 2.3).

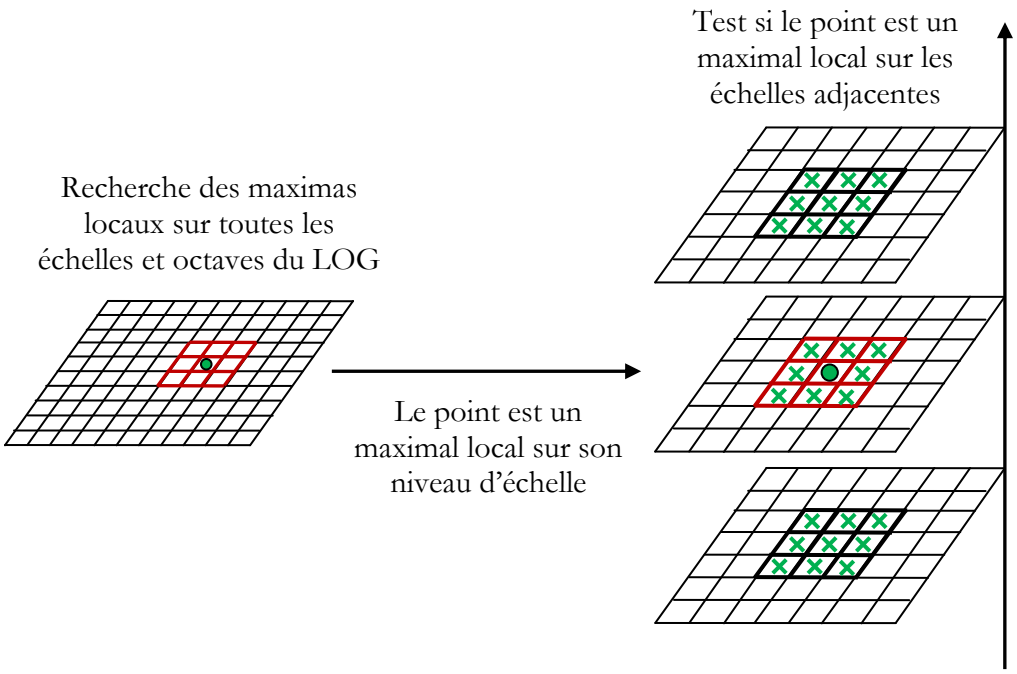


Figure 2.3 : Détection des extrema locaux; le point en vert est comparé avec les 26 voisins marqués par un X : ses 8 points connexes + les 9 points des 2 niveaux d'échelles adjacents.

Afin d'améliorer la robustesse de la détection des points clés, Lowe propose d'ajouter différentes contraintes :

- Déterminer de manière plus précise la position en utilisant l'interpolation des données voisines ;
- Ne pas conserver les points clés ayant un contraste faible ;
- Rejeter tous les points clés le long des contours ;

A chaque point clé retenu, une orientation et un descripteur sont associés.

2.3.1.3. Calcul de l'orientation

Le calcul de l'orientation d'un point clé a pour but de déterminer son descripteur, afin de fournir une invariance des descripteurs aux rotations de l'image. Le calcul de l'orientation locale d'un point clé se compose des étapes suivantes :

- On utilise l'échelle du point clé pour choisir l'image lissée, L, la plus proche de l'échelle du point clé pour conserver l'invariance aux changements d'échelle des calculs réalisés ;
- Pour chaque image $L(x, y)$, dans l'échelle sélectionnée, on détermine la magnitude $m(x, y)$:

$$m(x,y) = \sqrt{\big(L(x,y) - L(x-1,y)\big)^2 + \big(L(x,y+1) - L(x,y-1)\big)^2} \quad (2.8)$$

et l'orientation $\theta(x, y)$:

$$\theta(x,y) = tan^{-1}\left(\frac{L(x,y+1) - L(x,y-1)}{L(x+1,y) - L(x-1,y)}\right) \quad (2.9)$$

- L'histogramme de l'orientation est formé en utilisant l'orientation des gradients dans la région autour du point clé, correspondant aux flèches du schéma de droite sur la Figure 2.4. Chaque valeur ajoutée dans l'histogramme est pondérée par la magnitude du gradient et lissée en utilisant un noyau gaussien de forme circulaire (voir le cercle en jaune de la Figure 2.4) avec σ égale à 1.5 fois l'échelle du point clé ;
- Les pics de l'histogramme créé correspondent aux directions majeures des gradients locaux. Un point clé est créé dans la direction maximale, mais aussi pour tous les autres pics supérieurs à 80% du pic maximal ;
- Une parabole est interpolée entre les 3 valeurs les plus proches d'un pic pour déterminer avec plus de précision le pic maximal.

2.3.1.4. Calcul du descripteur

Le descripteur est composé d'un vecteur contenant les valeurs de toutes les orientations de l'histogramme précédemment décrit (cf. §2.3.1.3). La Figure 2.4 montre sur la droite un tableau de dimension 4x4 de l'histogramme des orientations (composées de 8 orientations pour chaque case). Chaque point clé se compose donc de 4*4*8 = 128 caractéristiques composant le vecteur caractéristique du descripteur.

Pour rendre le descripteur invariant aux changements d'illumination, le vecteur caractéristique est normalisé pour avoir une taille unitaire. En effet, un changement de

contraste entraîne un changement des gradients qui sont tous multipliés par la même constante de variation. La normalisation annule ainsi cet effet. Un changement d'éclairage entraîne également un changement de valeur des pixels de l'image. La valeur de chaque pixel est augmentée par la même constante de variation. Cependant, les gradients de l'image restent inchangés. Les points clés sont donc invariants aux changements d'illuminations homogènes. Dans le cas non affine, comme dans le cas de la saturation d'une caméra ou de reflets lumineux sur certaines surfaces d'objets de la scène, la magnitude d'une partie des gradients peut varier fortement. Cependant, l'orientation des gradients subit de faibles variations. Afin de réduire l'effet d'amplification de la magnitude, les gradients normalisés ayant une forte magnitude sont limités à 0.2 (valeur donnée empiriquement par Lowe [38]). Une nouvelle normalisation est effectuée sur les gradients.

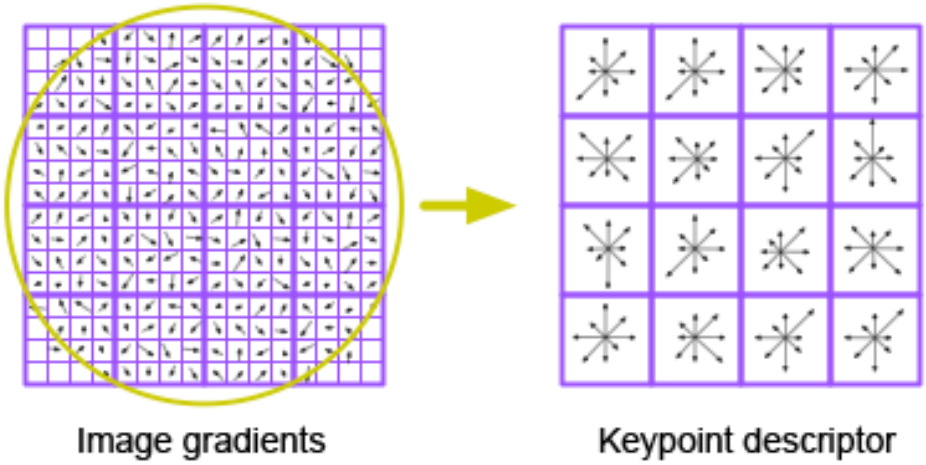


Figure 2.4 : Image[7] du descripteur du SIFT. La partie gauche de la figure présente l'extraction des gradients autour de la position des points clés de l'image. Chaque gradient est pondéré par un noyau gaussien de forme circulaire, présenté en jaune, puis accumulé dans l'histogramme des orientations, décomposé en 4x4 sous-régions, (voir sur la partie de droite de la figure).

[7]http://www.cg.tu-berlin.de/fileadmin/fg144/Courses/06WS/scanning/Jonas/html/index.html

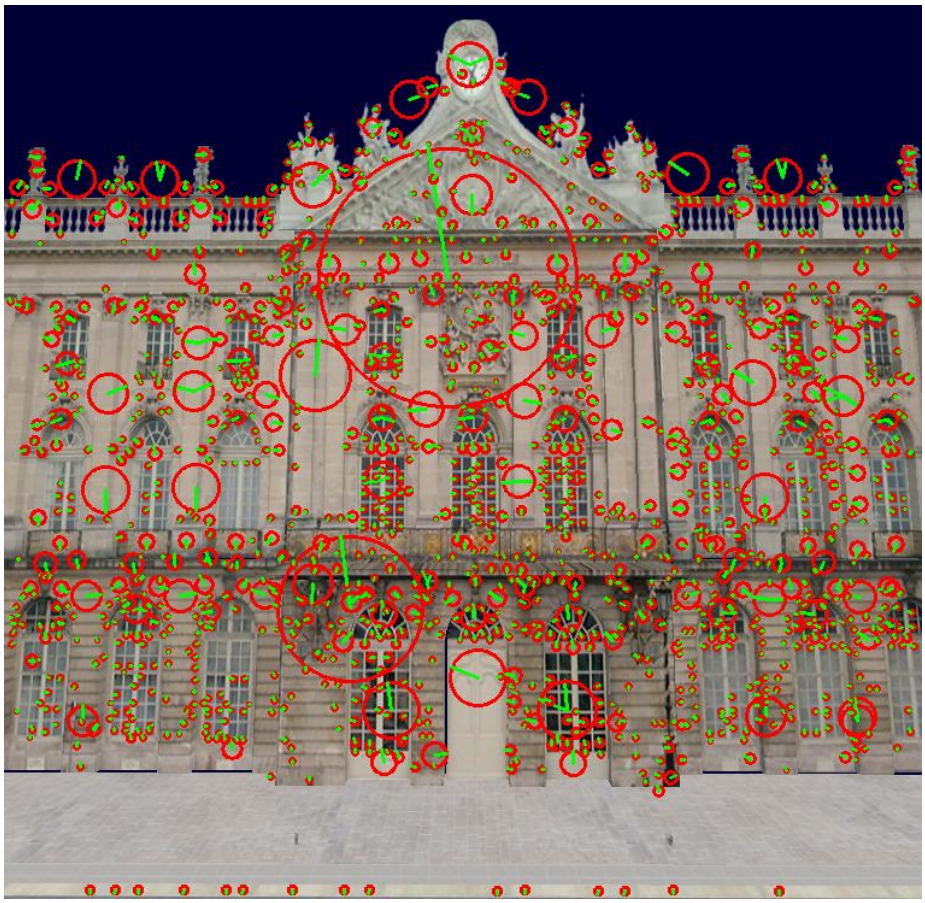


Figure 2.5 : Extraction des points de SIFT sur une image de résolution 760x740. 1741 points clés (représentés par cercles rouges) ont été extraits. Les cercles sont proportionnels à leurs échelles respectives, avec en vert l'orientation des points SIFT.

2.4. Mise en correspondance des points

La mise en correspondance des primitives de l'image consiste à déterminer la liste des couples de primitives similaires entre deux images. Pour réaliser la mise en correspondance, nous pouvons utiliser différentes méthodes. Celles-ci varient en fonction de la nature de la primitive et du descripteur associé.

2.4.1. Méthode par corrélation

A partir d'informations sur l'intensité des images, il est possible d'utiliser la méthode par corrélation qui va mettre en correspondance les pixels des deux images pour pouvoir calculer la similarité de deux images par rapport à leurs pixels. L'utilisation de la primitive du pixel sur l'image est ce qu'il y a de plus adapté pour la mise en correspondance.

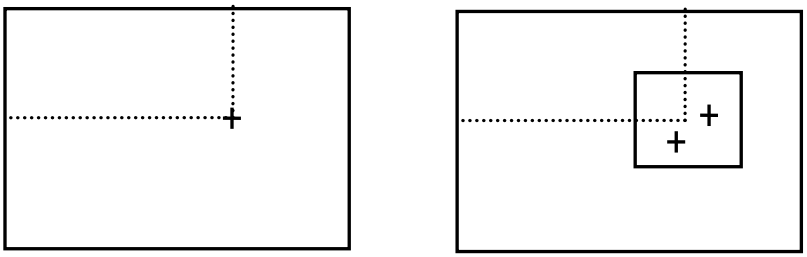


Figure 2.6 : Mise en correspondance par corrélation. Un point détecté dans l'image de gauche est cherché dans une région autour des coordonnées du point dans l'image de droite.

Pour calculer le score associé à cette méthode de mise en correspondance, on utilise le plus couramment le ZNCC (corrélation croisée, centrée et normée). Ce score de corrélation permet d'éviter les pertes d'appariements dues aux changements d'éclairage, elle est définie par :

$$zncc(P_1, P_2) = \frac{\sum_{d \in W}\left(I_1(P_1 + d) - \overline{I_1}(P_1)\right)\left(I_2(P_2 + d) - \overline{I_2}(P_2)\right)}{\sqrt{\sum_{d \in W}\left(I_1(P_1 + d) - \overline{I_1}(P_1)\right)^2}\sqrt{\sum_{d \in W}\left(I_2(P_2 + d) - \overline{I_2}(P_2)\right)^2}} \quad (2.10)$$

avec W la fenêtre de dimension $N \times N$.

$$\overline{I_1}(P_1) = \frac{1}{N^2}\sum_{d \in W} I_1(P_i + d) \quad (2.11)$$

2.4.2. Méthode SIFT

La mise en correspondance des points SIFT est rapide et simple grâce aux vecteurs caractéristiques. En effet, deux points homologues ont un vecteur caractéristique très proche. Pour rechercher le correspondant d'un point clé dans une autre image, il suffit de rechercher le point ayant la distance euclidienne minimum entre leurs vecteurs caractéristiques.

Même si cette méthode donne dans la plupart des cas de bons résultats, elle possède quelques inconvénients :

- Pour caractériser le contenu de la base de données, seuls les deux plus proches voisins sont retenus ;
- Si une structure est présente plus d'une fois dans la base de données, on ne peut la détecter. Si on prend une énorme base de données, cela peut engendrer des problèmes quant à l'utilisation de cette méthode. Ceci est vraiment gênant lors de la détection d'objets multiples ou de tout objet possédant des structures qui sont répétées.

Ces inconvénients ne gênent pas son utilisation dans notre cas.

2.5. Géométrie de la vision

2.5.1. Les espaces projectifs

Soit $\mathcal{P}^n$ l'espace projectif de dimension n composé de l'espace $\mathbb{R}^{n+1}$ sans le vecteur $[0, \dots, 0]^t$. Deux points x et x' sont liés par une relation d'équivalence si et seulement s'il existe $\lambda \neq 0$ tel que :

$$(x_1, \dots, x_{n+1})^t = \lambda . (x_1', \dots, x_{n+1}')^t$$

Alors x et x' représentent le même point de l'espace $\mathcal{P}^n$.

Les coordonnées homogènes (projectives) d'un point x de l'espace projectif $\mathcal{P}^n$ s'écrit comme une combinaison linéaire de $n+1$ points parmi les $n+2$ points e_i de la base canonique $\mathcal{B}_0$:

$$\boldsymbol{x} = \sum_{i=1}^{n+1} x_i \boldsymbol{e_i} \tag{2.12}$$

Les points e_i sont définis tels qu'aucun sous-ensemble de $n+1$ points de ces points n'appartienne à un hyperplan de $\mathcal{P}^n$ et comme étant l'ensemble des points suivants :

$$e_i = (1, \dots, 0)$$
$$e_i = (0, \dots, 1, \dots, 0)$$
$$e_{n+1} = (0, \dots, 1)$$
$$e_{n+2} = (1, \dots, 1)$$

2.5.2. Homographies

Une homographie est une transformation linéaire $\mathcal{P}^n$ dans $\mathcal{P}^n$. On représente généralement par le symbole H, la matrice associée à une homographie. Les matrices homographiques sont définies à un coefficient près, d'où :

$$y = \lambda H x \tag{2.13}$$

On parle également d'égalité projective :

$$y = H x \tag{2.14}$$

Une homographie possède $n \times (n+2)$ dégrés de liberté. L'égalité projective définit donc n rapports et il faudra $n+2$ égalités projectives pour définir tous les dégrés de liberté de la matrice H. Pour déterminer la matrice H, il faut donc $n+2$ correspondances $(y \leftrightarrow x)$.

Soient $x_1, x_2, \dots, x_{n+2}$ des points de l'espace projectif formant une base projective. Il existe alors un ensemble de matrices H régulières :

$$\lambda_i x_i = H e_i \tag{2.15}$$

tel que $i = 1$ à $n+2$ et $\lambda_i \neq 0$.

Toute les matrices de l'ensemble de H se distinguent uniquement par un facteur d'échelle λ différent. Dans la base canonique, chaque base projective peut être transformé par une homographie.

2.5.3. Modèle géométrique des caméras

Une caméra est un capteur qui fournit une information limitée de l'environnement observé. Cette information correspond à la projection de points tridimensionnels sur le capteur. Le modèle mathématique le plus utilisé pour représenter le processus de projection est le modèle sténopé, appelé aussi le modèle de la projection centrale (cf. Figure 2.7).

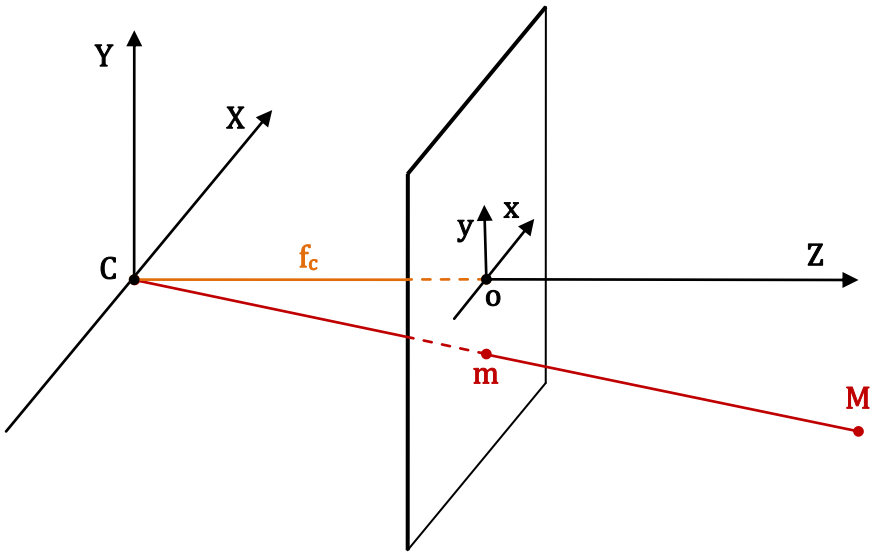


Figure 2.7: Modèle sténopé

La Figure 2.7 représente la projection centrale d'un point M de l'espace sur le plan image, appelé également rétine ou plan focal, dont le centre de projection est le point C. Nous définissons C comme étant l'origine du repère de la caméra, noté $\mathcal{R}_c$. Ce repère se compose d'un axe principal (ou axe optique) Z, d'un axe horizontal X et d'un axe vertical Y. Le plan image se situe à $Z = f_c$, où f_c est la distance focale de l'objectif de la caméra. L'intersection de l'axe principal avec le plan image définit le point o, appelé point central, de coordonnées $o = (0,0,f_c)$.

Tout point M de l'espace admet un point projeté m sur le plan image représentant l'intersection de la droite (ou rayon optique) (CM) avec le plan image. Le centre C de la caméra (ou centre optique) est le seul point à ne pas admettre de point projeté sur le plan image.

2.5.4. Transformations interne et externe

La transformation d'un point M de l'espace projectif, exprimé en coordonnées cartésiennes, en un point de l'image acquise, exprimé en pixel, se décompose en trois sous-transformations :

- Un changement de repère permettant de passer du repère de référence (ou repère absolu) au repère de la caméra ;

- Une projection en perspective pour passer de l'espace euclidien $\mathbb{R}^3$ à l'espace euclidien $\mathbb{R}^2$. Cette transformation vise à passer du point M de l'espace en un point m du plan image (cf. Figure 2.7) ;
- Une mise à l'échelle pour transformer les coordonnées métriques, liées au repère de la caméra, en coordonnées pixelliques dans le repère image.

La transformation permettant d'effectuer le changement de repère pour passer au repère lié à la caméra est appelée transformation externe. Elle est définie à partir des paramètres extrinsèques de la caméra. La projection ainsi que la mise à l'échelle sont appelées transformations internes. Elles sont définies à partir des paramètres intrinsèques de la caméra.

Considérons les notations suivantes :

- $\mathcal{R}_0$: le repère de référence (ou repère absolu) de l'espace projectif.
- $\mathcal{R}_c$: le repère lié à la caméra. Il est défini par $\mathcal{R}_c = (C, X, Y, Z)$ où C est le centre optique de la caméra (cf. Figure 2.7).

Dans la suite, tous les points sont exprimés dans le système de coordonnées homogènes.

2.5.4.1. Changement de repère

Le changement de repère permet d'exprimer un point M dans le repère $\mathcal{R}_c$, à partir de ses coordonnées dans le repère $\mathcal{R}_0$ de l'espace projectif. Caractérisée par les paramètres extrinsèques de la caméra, cette transformation est composée de :

- Une translation $\overline{T}$ correspondant à la position de la caméra, c'est-à-dire la position du centre C de la caméra, dans le repère $\mathcal{R}_0$, où $\overline{T}$ est une matrice de dimension 3 x 1, exprimée dans le système de coordonnées classiques ;
- Une orientation de la caméra qui correspond à la rotation du repère $\mathcal{R}_0$ vers le repère $\mathcal{R}_c$. Cette rotation est caractérisée par une matrice $\bar{R}$ de dimension 3×3, exprimée dans le système de coordonnées classiques ;

Etant donné un point M_0, exprimé dans le repère $\mathcal{R}_0$ de l'espace projectif. Les coordonnées de ce point dans le repère $\mathcal{R}_c$ lié à la caméra sont données par la transformation euclidienne ou rigide :

$$\overline{M_c} = \bar{R}.\overline{M_0} + \overline{T} \tag{2.16}$$

où $\overline{M_0}$ et $\overline{M_c}$ sont respectivement les coordonnées des points M_0 (dans $\mathcal{R}_0$) et M_c (dans $\mathcal{R}_c$) dans le système de coordonnées classiques. En utilisant le système de coordonnées homogènes, la transformation de changement de repère peut s'écrire de la manière suivante : $M_c = S.M_0$, où S est une matrice de dimension 4 x 4, définie par:

$$S = \begin{pmatrix} \bar{R} & \bar{T} \\ 0 & 1 \end{pmatrix} \tag{2.17}$$

M_c et M_0 représentent les coordonnées dans le système de coordonnées homogènes :

$$M_0 = \left(X_{M_0}, Y_{M_0}, Z_{M_0}, T_{M_0}\right)^t = \begin{pmatrix} \overline{M_0} \\ T_{M_0} \end{pmatrix} \tag{2.18}$$

$$M_c = \left(X_{M_c}, Y_{M_c}, Z_{M_c}, T_{M_c}\right)^t = \begin{pmatrix} \overline{M_c} \\ T_{M_c} \end{pmatrix} \tag{2.19}$$

où T_{M_0} (respectivement T_{M_c}) représente la coordonnée homogène associée au point M_0 (respectivement M_c). Elle est généralement égale à 1.

2.5.4.2. Projection centrale

La projection centrale d'un point M est définie par l'intersection de la droite (CM) et le plan image (cf. Figure 2.7). Elle est caractérisée par une matrice P permettant d'associer à un point M_c (exprimé dans le repère $\mathcal{R}_c$) de l'espace euclidien $\mathbb{R}^3$, un point $\widetilde{m}$ de l'espace euclidien $\mathbb{R}^2$. Exprimé dans le repère (o, x, y) lié au plan image, le point $\widetilde{m}$ a pour coordonnées dans le système de coordonnées homogènes :

$$\widetilde{m} = \begin{pmatrix} f_c X_{M_c}/Z_{M_c} \\ f_c Y_{M_c}/Z_{M_c} \\ 1 \end{pmatrix} = \frac{1}{Z_{M_c}} . P . \begin{pmatrix} X_{M_c} \\ Y_{M_c} \\ Z_{M_c} \\ 1 \end{pmatrix} \quad (2.20)$$

avec :

$$P = \begin{pmatrix} f_c & 0 & 0 & 0 \\ 0 & f_c & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad (2.21)$$

2.5.4.3. Mise à l'échelle

La mise à l'échelle consiste à transformer le point $\widetilde{m}$ projeté sur le plan image en un point image m, exprimé en pixels dans le repère image $\mathcal{R}_i$ = (Oi, u, v) (cf. Figure 2.8).

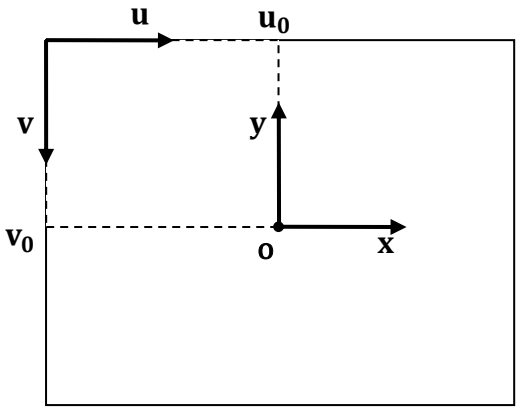


Figure 2.8: Repère image par rapport au repère du plan image

Les coordonnées du point image m dans le repère image $\mathcal{R}_i$, exprimées avec le système de coordonnées homogènes, sont obtenues par l'intermédiaire de la matrice K de dimension 3×3, appelée matrice de calibrage :

$$m = \begin{pmatrix} u_m \\ v_m \\ 1 \end{pmatrix} = K . \widetilde{m} = \frac{1}{Z_{M_c}} . K . P . \begin{pmatrix} X_{M_c} \\ Y_{M_c} \\ Z_{M_c} \\ 1 \end{pmatrix} \quad (2.22)$$

avec :

$$K = \begin{pmatrix} k_u & -k_u.cot(\tau) & u_0 \\ 0 & k_v.sin(\tau) & v_0 \\ 0 & 0 & 1 \end{pmatrix}.P_f = \begin{pmatrix} f_c.k_u & -f_c.k_u.cot(\tau) & u_0 \\ 0 & f_c.k_v.sin(\tau) & v_0 \\ 0 & 0 & 1 \end{pmatrix} \quad (2.23)$$

La matrice K utilise les paramètres intrinsèques de la caméra :

- τ représente l'angle entre les vecteurs u et v du repère image (cf. Figure 2.8) ;
- k_u et k_v sont respectivement les facteurs d'échelles horizontale et verticale, exprimés en pixel/mm ;
- $(u_0, v_0)^t$ sont les coordonnées du point principal p, exprimées en pixel dans le repère image ;
- f_c est la distance focale.

2.5.4.4. Transformation complète

Etant donné un point M_0, exprimé dans le repère absolu $\mathcal{R}_0$ de l'espace projectif, le point image m, exprimé en pixels dans le repère image $\mathcal{R}_i$, est obtenu en appliquant :

- Le changement de repère permettant de passer du repère $\mathcal{R}_0$ au repère $\mathcal{R}_c$, lié à la caméra. On obtient alors $M_c = S.M_0$;
- Puis, la projection centrale permettant d'obtenir le point projeté $\widetilde{m}$ du plan image : $\widetilde{m} = \frac{1}{Z_{M_c}}.P.M_c$;
- Enfin, la mise à l'échelle qui fournit le point image m : $m = K.\widetilde{m}$.

La transformation complète s'écrit alors :

$$\begin{pmatrix} u_m \\ v_m \\ 1 \end{pmatrix} = \frac{1}{Z_{M_c}}.K.P.S.\begin{pmatrix} X_{M_0} \\ Y_{M_0} \\ Z_{M_0} \\ 1 \end{pmatrix} = \frac{1}{Z_{M_c}} P_{proj} \begin{pmatrix} X_{M_0} \\ Y_{M_0} \\ Z_{M_0} \\ 1 \end{pmatrix} \quad (2.24)$$

2.6. Modèle géométrique des stéréoscopes

Un stéréoscope est un dispositif de prise de vue composé de plusieurs caméras. Nous traitons le cas le plus simple où le système se compose uniquement de deux caméras. Ce système permet d'obtenir 2 images différentes $\mathcal{I}_1$ et $\mathcal{I}_2$ de la même scène.

On peut aussi considérer un dispositif prenant 2 images à des instants différents à condition que l'environnement observé soit statique.

Etant données deux caméras de centres optiques C_1 et C_2. Soit $\mathcal{R}_{c1}(C_1, X_1, Y_1, Z_1)$, respectivement $\mathcal{R}_{c2}(C_2, X_2, Y_2, Z_2)$, le repère lié à la caméra 1 (respectivement caméra 2). Soit P_1, respectivement P_1, la matrice de projection associée à la caméra 1 (respectivement caméra 2). Les bases associées aux repères $\mathcal{R}_{c1}$ et $\mathcal{R}_{c2}$ sont considérées comme étant des bases orthonormées.

2.6.1. Centres de projection

Comme nous l'avons vu (cf. §2.5.3), la projection centrale projette tout point de l'espace projectif sur le plan image de la caméra, sauf pour le centre optique (ou centre de projection) C_1 de la caméra. Cette projection indéfinie est caractérisée par la relation $P.C = 0$. Le centre de projection s'obtient simplement à partir de l'équation suivante, en utilisant le système de coordonnées homogènes :

$$(\bar{P} \quad p)\begin{pmatrix}\bar{C}\\1\end{pmatrix} = 0 \tag{2.25}$$

où $\bar{P}$ et $\bar{C}$ sont respectivement la matrice de projection et le centre de projection, exprimés dans le système de coordonnées classiques.

On obtient alors le centre de projection $\overline{C_1}$ de la caméra 1:

$$\overline{C_1} = -\overline{P_1}^{-1}.p_1 \tag{2.26}$$

et le centre de projection C_2 de la caméra 2 :

$$\overline{C_2} = -\overline{P_2}^{-1}.p_2 \tag{2.27}$$

2.6.2. Géométrie épipolaire

On appelle épipôle, la projection du centre optique de la première caméra dans l'image de la deuxième caméra. L'épipôle e_1 (resp e_2) correspond donc à la projection du centre de projection C_2 (resp. C_1) dans l'image $\mathcal{I}_1$ (resp. $\mathcal{I}_2$) :

$$e_1 = P_1.C_2 = P_1 \begin{pmatrix} -\overline{P_2}^{-1}.p_2 \\ 1 \end{pmatrix} = -\overline{P_1}\,.\overline{P_2}^{-1}\,.p_2 + p_1 \tag{2.28}$$

$$e_2 = -\overline{P_2}\,.\overline{P_1}^{-1}\,.p_1 + p_2 \tag{2.29}$$

La géométrie épipolaire représente la relation entre deux caméras. Cette relation est indépendante de la structure de la scène, elle dépend de la position relative entre les deux caméras et des paramètres intrinsèques. Elle modélise la relation entre les points image m_1 et m_2, issus respectivement de la projection d'un point M sur les plans image des deux caméras (cf. Figure 2.9). Les points image, les centres optiques des deux caméras ainsi que le point M définissent un plan, appelé plan épipolaire et noté Π_M. L'intersection du plan Π_M avec les deux plans image forme deux ligne l_1 et l_2, appelées lignes épipolaires. La ligne épipolaire gauche (respectivement droite) passe par l'épipôle gauche (respectivement droit) et le point projeté sur le plan image gauche (repectivement droit). Par ailleurs, toutes les lignes épipolaires gauches (respectivement droites) ainsi que la droite (C_1C_2), appelée ligne de base, se coupent en un seul point, qui n'est autre que l'épipôle gauche (respectivement droit) (cf. Figure 2.9).

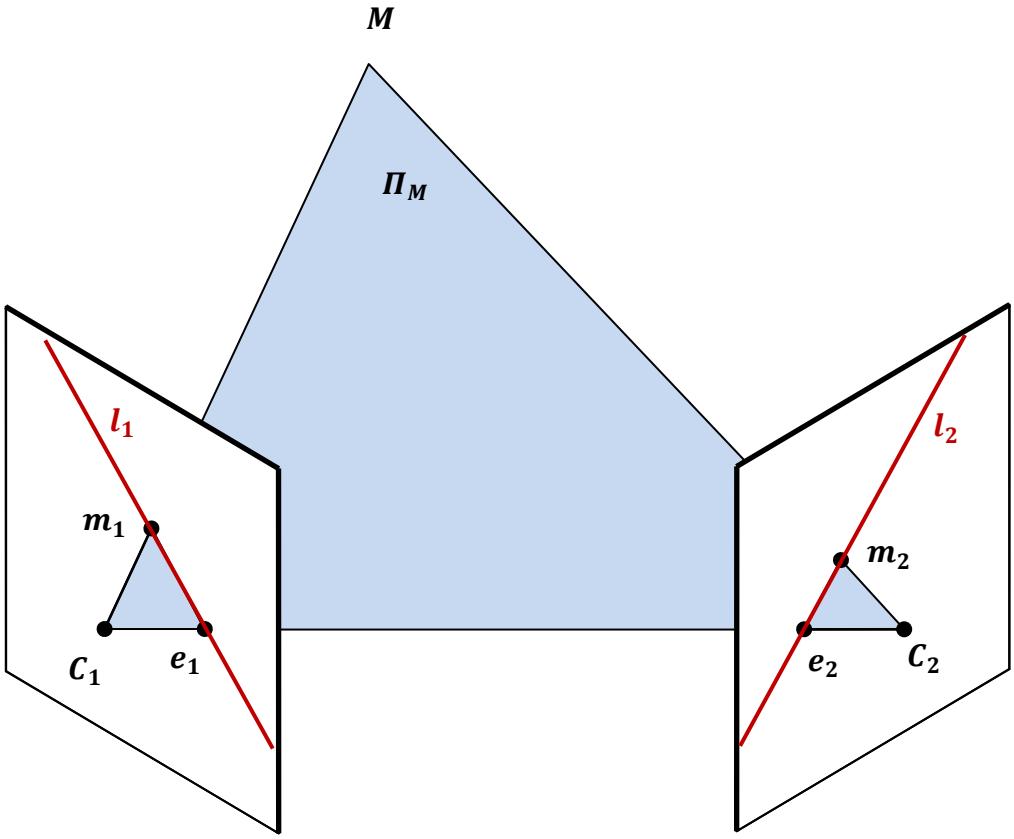


Figure 2.9: Géométrie épipolaire.

2.7. Reconstruction 3D

2.7.1. Matrice fondamentale

La matrice fondamentale est la représentation algébrique de la géométrie épipolaire. Il s'agit d'une matrice de dimension 3×3 de rang 2. Elle permet d'associer à un point m_1 de l'image $\mathcal{I}_1$ une ligne épipolaire l_2 passant par l'épipôle e_2 dans l'image $\mathcal{I}_2$:

$$l_2 \simeq F.m_1, \qquad l_2 \simeq e_2 \wedge m_2 \tag{2.30}$$

La géométrie épipolaire définit la ligne épipolaire l_2 comme étant la droite passant par le point m_2 et l'épipôle e_2. Ainsi, $m_2^t.l_2 = e_2^t.l_2 = 0$, ce qui permet de déterminer la relation fondamentale pour vérifier la correspondance deux points image :

$$m_2^t . F . m_1 = 0 \tag{2.31}$$

Les matrices F et F^t sont un rôle similaire dans le sens où la matrice F^t permet d'associer à un point m_2 de l'image $\mathcal{I}_2$ une ligne épipolaire l_2 passant par l'épipôle e_1 dans l'image $\mathcal{I}_1$. D'où :

$$m_1^t . F^t . m_2 = 0 \tag{2.32}$$

Lorsque la matrice fondamentale est connue, les épipôles e_1 et e_2 sont donnés respectivement par les noyaux de F et F^t : $F e_1 \simeq 0$ et $F^t e_2 \simeq 0$.

2.7.2. Matrice essentielle

La matrice essentielle [37] décrit la géométrie épipolaire de deux caméras calibrées. Elle permet d'associer deux points homologues dans les plans image de deux caméras C_1 et C_2. Ainsi on peut exprimer respectivement les transformations projectives P_1 et P_2 par :

$$P_1 = P_1 . [I_3, 0] \quad \text{et} \quad P_2 = P_2 . [I_3, \bar{T}] \tag{2.33}$$

A partir de l'équation (2.32) nous pouvons exprimer la relation suivante :

$$[K_2 . \widetilde{m_2}]^t . F . [K_1 . \widetilde{m_1}] = 0 \tag{2.34}$$

et :

$$\widetilde{m_2}^t . E . \widetilde{m_1} = 0 \tag{2.35}$$

avec :

$$E = K_2^t . F . K_1 . \tag{2.36}$$

La matrice essentielle E exprime la relation de déplacement entre les centres de projection de C_1 et C_2 par la relation suivante :

$$E \simeq [\bar{T}]_\times R \tag{2.37}$$

Les propriétés de la matrice essentielle sont identiques à celles de la matrice fondamentale :

- E est de rang 2 ;
- Les valeurs propres sont les mêmes que celles de la matrice fondamentale ;

2.7.3. Estimation de la matrice fondamentale

Soient $m_1^i = \left(u_1^i, v_1^i, 1\right)^t$ et $m_2 = \left(u_2^i, v_2^i, 1\right)^t$ deux points homologues, issus de la projection d'un point M de l'espace projectif sur les images $\mathcal{I}_1$ et $\mathcal{I}_2$. La relation fondamentale (cf. §2.7.1) peut s'écrire sous la forme algébrique suivante :

$$a_i^t f = 0 \tag{2.38}$$

avec :

$$f = (F_{11}, F_{12}, F_{13}, F_{21}, F_{22}, F_{23}, F_{31}, F_{32}, F_{33})^t$$
$$a_i = \left(u_2^i u_1^i, u_2^i v_1^i, u_2^i, v_2^i u_1^i, v_2^i v_1^i, v_2^i, u_1^i, v_1^i, 1\right)^t$$

et F_{ij} est l'élément de la ligne i et la colonne j de la matrice fondamentale F ;

En utilisant n couples de points homologues (m_1^i, m_2^i), on obtient l'équation matricielle suivante :

$$Af = 0 \tag{2.39}$$

avec A une matrice de dimension $n \times 9$. Afin d'éviter la solution triviale, où f est égal au vecteur nul, on ajoute la contrainte suivante :

$$\|f\| = 1 \tag{2.40}$$

La résolution de ce système permet de déterminer la matrice fondamentale. Dans le cas où $n > 8$, le système surdéterminé peut être résolu avec différentes méthodes linéaires, itératives ou robustes.

2.7.3.1. Méthodes linéaires

D'un point de vue théorique, il a été démontré que la résolution de l'équation (2.39) nécessite au moins 7 points pour sa résolution, car la matrice A est de rang 2

[41]. Dans la pratique, les données sont toujours bruitées. Le rang de la matrice est alors égal à 1. La résolution de problème nécessite donc au moins 9 points et il est préférable de le résoudre au sens des moindres carrés :

$$\min_F C(F) = \min_F \sum_{i=1}^{n} \left({m_2^i}^t F m_1^i \right)^2 = \min_f \sum_{i=1}^{n} \left(a_i^t f \right)^2 \qquad (2.41)$$

Cette méthode de résolution est connue sous le nom de la méthode des 8 points [25].

Pour la stabilité des solutions de l'équation (2.39), Hartley [25] propose une normalisation des points permettant d'améliorer le conditionnement du problème. Il propose une normalisation composée d'une translation et d'un changement d'échelle dans chaque image. Soit T_1 (respectivement T_2) la matrice de normalisation associée à l'image gauche (respectivement droite). Ces matrices sont calculées pour obtenir les conditions suivantes :

- L'origine de l'image est considérée comme étant le centre de gravité des points transformés ;
- La distance moyenne des points transformés de l'origine est égale à $\sqrt{2}$;

La méthode de résolution est détaillée dans la table Table 2.2.

Algorithme normalisé des 8 points

Objectif : Avec $n \geq 8$ correspondances de points 2D : $m_1^i(x_1^i, y_1^i) \leftrightarrow m_2^i(x_2^i, y_2^i)$, déterminer la matrice fondamentale

Algorithme :

1. Normalisation : Transformer les points m_1^i et m_2^i tels que :

$$\widehat{m_1^i} = T_1 . m_i \quad \text{et} \quad \widehat{m_2^i} = T_2 . m_2^i$$

2. Calcul de la matrice fondamentale $\hat{F}$ en utilisant les correspondances $\widehat{m_1^i} \leftrightarrow \widehat{m_2^i}$:
   - Le critère $C(\hat{F})$ est minimisé (cf. équation (2.41)), sous la contrainte (2.40), par le vecteur singulier associé à la plus petite valeur singulière de A. Pour cela, on utilise une décomposition en valeur singulière (SVD) :

$$A_{n\times 9} \rightarrow U_{n\times 9}\, D_{9\times 9}\, V^t_{9\times 9} \tag{2.42}$$

La dernière colonne de V donne $\hat{f}$;

3. La matrice $\hat{F}$ obtenue n'étant généralement pas de rang 2, on renforce la contrainte de rang par une SVD [17] de $\hat{F}$:

$$\hat{F}_{3\times 3} \rightarrow U_{3\times 3}\, \hat{D}_{3\times 3}\, V^t_{3\times 3} \tag{2.43}$$

- En annulant la plus petite des valeurs singulières :

$$D \leftarrow \begin{pmatrix} \hat{D}_{11} & & \\ & \hat{D}_{22} & \\ & & 0 \end{pmatrix} \tag{2.44}$$

- On recompose la SVD de F :

$$F \leftarrow UDV^t \tag{2.45}$$

4. Dénormalisation de la matrice F obtenue par :

$$F = T_2^t . \hat{F} . T_1$$

Table 2.2 : Algorithme normalisé des 8 points

2.7.3.2. Méthodes itératives

L'estimation de la matrice fondamentale est effectuée à partir de correspondances de primitives, en utilisant l'équation (2.39). La qualité du résultat dépend de la qualité de l'appariement. Les méthodes itératives permettent d'améliorer le résultat en effectuant plusieurs passes de calcul consistant à écarter ou ajouter des couples de primitives dans l'estimation de F.

Nous pouvons distinguer différentes approches :

- Celles basées sur la minimisation de critères de la distance entre les points et les lignes épipolaires :

$$\min_F \sum_{i=1}^{n} d^2\left(m_1^i, F^t m_2^i\right) + d^2\left(m_2^i, F m_1^i\right) \tag{2.46}$$

Cette minimisation non-linéaire est généralement effectuée avec la méthode itérative de Levenberg-Marquardt [61] ;

- Celles basées sur le gradient. Le processus de résolution consiste à minimiser :

$$\min_F \sum_{i=1}^{n} \frac{\left(m_2^i F m_1^i\right)^2}{\sqrt{l_{11}^i + l_{12}^i + {l_{21}^i}^2 + {l_{22}^i}^2}} \tag{2.47}$$

avec : $Fm_2^i = \left(l_{21}^i, l_{22}^i, l_{23}^i\right)^t$ et $F^t m_1^i = \left(l_{11}^i, l_{12}^i, l_{13}^i\right)^t$.

2.7.3.3. Méthodes robustes

Les méthodes décrites dans les paragraphes §2.7.3.2 et §2.7.3.3 considèrent l'ensemble des correspondances sans faire de distinction. Ceci rend la résolution de la fondamentale sensible aux faux appariements. Les techniques robustes permettent d'estimer plusieurs matrices fondamentales en n'utilisant qu'un échantillon de couples parmi les correspondances ou en se basant sur certains points considérés comme importants.

2.7.3.3.1. M-Estimateur

Le M-Estimateur [86] est une méthode robuste permettant de réduire l'effet des faux appariements, en pondérant chaque couple de primitives $m_1^i, m_2^i)$ par une fonction poids, notée w_i. Ainsi, l'estimation de la matrice fondamentale, via l'équation (2.41), devient :

$$\min_F \sum_{i=1}^{n} w_i . \left({m_2^i}^t F m_1^i\right)^2 \tag{2.48}$$

Chaque fonction de poids définit un M-Estimateur différent. La Table 2.3 illustre l'algorithme M-Estimateur. La Figure 2.10 présente quelques exemples de fonctions

poids. Le lecteur intéressé trouvera la définition et la particularité de chacune de ces fonctions dans [86].

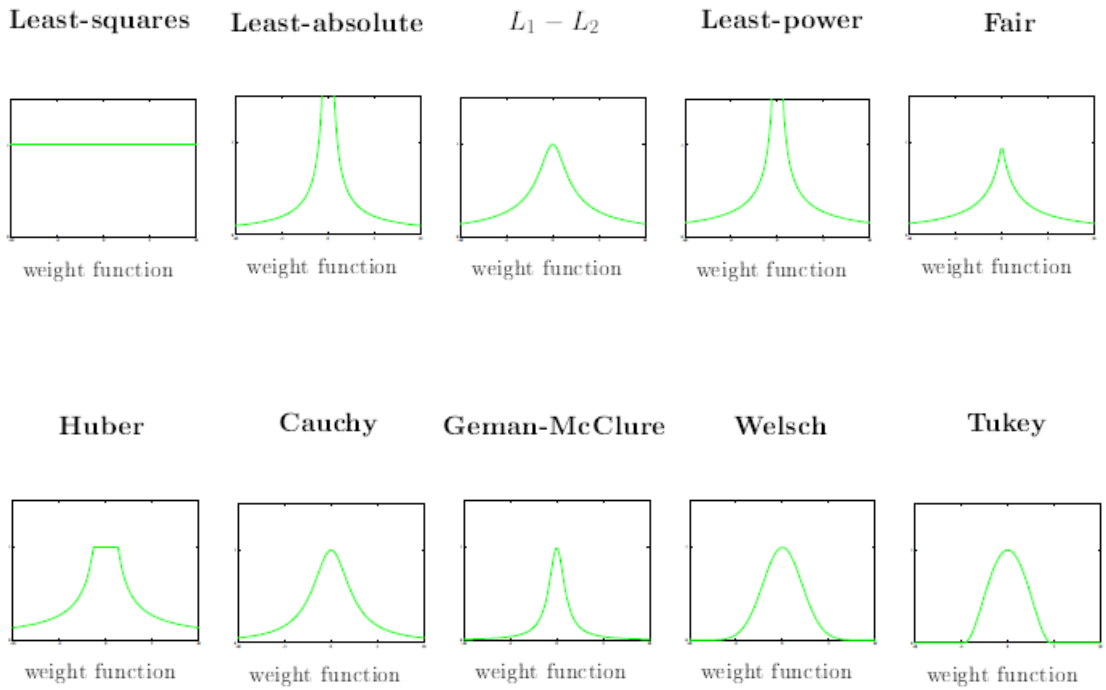


Figure 2.10 : Exemples de fonctions de poids.

Algorithme M-Estimateur

Objectif : Avec $n \geq 8$ correspondances de points 2D : $m_1^i(x_1^i, y_1^i) \leftrightarrow m_2^i(x_2^i, y_2^i)$, déterminer la matrice fondamentale F.

Algorithme :

5. Normalisation de Hartley (cf. §2.7.3.1).
6. Calcul des w_i selon le choix de la fonction de poids.
7. Résoudre le système linéaire : $w_k \; Af^k = 0$.
8. Renforcer la contrainte de rang 2 de la matrice F (cf. algorithme de la Table 2.2).
9. Si $|F^k - F^{k-1}| < \varepsilon$ ou $k > N$, aller à l'étape 7.
10. $F^{k-1} = F^k$, $k = k + 1$, aller à l'étape 3.

11. Dénormalisation de la matrice F (cf. algorithme de la Table 2.2).

l'indice k indique le numéro de l'itération et N représente le nombre maximal d'itérations.

Table 2.3 : Algorithme M-Estimateur

2.7.3.3.2. RANSAC

RANdom SAmple Consensus (RANSAC) est un estimateur robuste proposé par Fischler et Bolles [18]. Contrairement aux méthodes précédentes qui utilisent l'ensemble des données (les appariements dans notre cas), l'estimateur RANSAC permet de déterminer statistiquement si un sous-ensemble de données vérifie un modèle. La méthode de RANSAC fournit des résultats corrects même avec la présence de nombreuses données erronées. Elle utilise trois paramètres :

- Un seuil t permettant de tester si un échantillon vérifie le modèle ou non ;
- La proportion d'échantillons qui ne vérifient pas le modèle : *outliers* ;
- La probabilité d'écarter un échantillon vérifiant le modèle.

L'idée est de tirer aléatoirement le nombre minimal s de données pour estimer les paramètres du modèle. Il s'agit ensuite de compter le nombre de données vérifiant le modèle paramétré. Pour la matrice fondamentale, il est nécessaire de prendre $s = 7$ couples de points homologues. Les paramètres retenus sont ceux correspondant au modèle qui compte le plus de données.

Le paramètre t, représentant le seuil, est généralement défini de manière empirique, en fonction du modèle et du type de données à traiter. Les deux autres paramètres et le nombre minimal s, permettant de définir le nombre de tirages aléatoires, sont choisis de manière à obtenir des résultats fiables.

Etant donnés le nombre de tirages N et la probabilité δ qu'un point vérifie le modèle, la probabilité qu'un point soit un outlier est égale à $(1 - \delta)$. La probabilité d'avoir un outlier dans le sous-ensemble tiré est $\varepsilon = (1 - \delta^s)$. La probabilité de tirer un sous-ensemble contenant un outlier après N tirages est égale à $(1 - \delta^s)^N$. Le nombre de tirages N est alors donné par :

$$N = \frac{log(1-p)}{log(1-(1-\varepsilon)^s)} \tag{2.49}$$

Algorithme RANSAC

Objectif : Avec $n \geq 8$ correspondances de points 2D : $m_1^i(x_1^i, y_1^i) \leftrightarrow m_2^i(x_2^i, y_2^i)$, déterminer la matrice fondamentale F.

Algorithme :

12. Parmi l'ensemble des correspondances, choisir N sous-ensembles de taille $s = 8$ couples de points.
13. Pour chaque sous-ensemble :
    - 4.1. Calculer la matrice fondamentale F^k (cf. algorithme de la Table 2.2).
    - 4.2. Calculer les distances de Sampson[68] d_i associées à chaque correspondance de points $m_1^i \leftrightarrow m_2^i$:

$$d_i = \frac{\left(m_2^{i\,t}.F^k.m_1^i\right)^2}{\left(F^k.m_1^i\right)_x^2 + \left(F^k.m_1^i\right)_y^2 + \left(F^{k\,t}.m_2^i\right)_x^2 + \left(F^{k\,t}.m_2^i\right)_y^2} \tag{2.50}$$

    - 4.3. Calculer le seuil t [65] :

$$t = 1.96\sigma$$

$$\text{avec : } \sigma = 1.4826\left(1 + \frac{5}{n-7}\right)\sqrt{median_i|d_i|} \tag{2.51}$$

    - 4.4. Calculer le nombre de points tels que $d_i < t$.
14. Sélectionner la solution ayant le nombre le plus élevé de points.
15. Calculer la matrice fondamentale F à l'aide de la méthode des 8 points sur l'ensemble des correspondances des points retenus.

Table 2.4 : Algorithme de RANSAC

2.7.3.3.3. LMedS : Least Median of Squares

La méthode LMedS, proposée par Rousseeuw *et al.* [65], consiste à résoudre l'équation non-linéaire suivante :

$$\min \underset{i}{median}(r_i^2) \tag{2.52}$$

avec : r_i^2 l'erreur résiduelle associée à la correspondance des points 2D m_1^i et m_2^i.

Cette méthode, très similaire à la méthode de RANSAC, diffère par le critère de sélection du modèle. L'algorithme LMedS sélectionne le modèle ayant la valeur médiane des erreurs quadratiques résiduelles la plus faible. La Table 2.5 détaille le fonctionnement de l'algorithme LMedS.

Algorithme LMedS

Objectif : Avec $n \geq 8$ correspondances de points 2D : $m_1^i(x_1^i, y_1^i) \leftrightarrow m_2^i(x_2^i, y_2^i)$, déterminer la matrice fondamentale F.

Algorithme :

1. Choisir parmi l'ensemble des correspondances, choisir N sous-ensembles de taille $s = 8$ couples de points.
2. Pour chaque sous-ensemble :
   - 4.1. Calculer la matrice fondamentale F^k (cf. algorithme de la Table 2.2).
   - 4.2. Calculer la valeur médiane des erreurs quadratiques résiduelles M^k de la matrice fondamentale F^k :

$$M^k = \underset{i=1 \text{ à } n}{med} \left(d^2\left(m_1^i, {F^k}^t m_2^i\right) + d^2\left(m_2^i, F^k m_1^i\right) \right) \tag{2.53}$$

   - 4.3. La matrice F_k associée à la plus petite valeur médiane des erreurs quadratiques résiduelles est conservée.
3. Affecter à chaque correspondance un poids w_i tel que [86] :

$$w_i = \begin{cases} 1 & si\ r_i^2 \leq (2.5\hat{\sigma})^2 \\ 0 & sinon \end{cases} \quad (2.54)$$

avec : $r_i^2 = d^2\left(m_1^i, F^{k^t} m_2^i\right) \quad + d^2\left(m_2^i, F^k m_1^i\right)$

et : $\hat{\sigma} = 1.4826\left(1 + \frac{5}{n-7}\right)\sqrt{M_k}$

4. Calculer la matrice fondamentale F à l'aide de méthode des 8 points sur l'ensemble des correspondances dont le poids est non nul.

Table 2.5 : Algorithme LMedS (Least Median of Squares)

2.7.4. Estimation de la matrice essentielle

Les méthodes d'estimation de la géométrie épipolaire, et donc de la matrice fondamentale, ont été fortement développées depuis 1990. L'estimation de la matrice essentielle est généralement basée sur l'utilisation de la matrice fondamentale. Cependant, des travaux récents ont porté sur l'estimation de la matrice essentielle sans faire appel à la matrice fondamentale. Parmi ces travaux, nous pouvons citer l'algorithme des 5 points [54,53][66].

2.7.4.1. Matrice essentielle et fondamentale

La matrice essentielle peut être déterminée à partir de la matrice fondamentale (cf. équation (2.36)). Cependant, la matrice essentielle n'est généralement pas de rang 2, à cause du bruit des mesures. Comme pour la matrice fondamentale, il est nécessaire de renforcer la contrainte de rang. Pour cela, la matrice essentielle, notée E, doit avoir une valeur singulière égale à 0.

Soient U, D et V les matrices issues de la décomposition SVD [17] de la matrice essentielle E :

$$\hat{E}_{3\times3} \rightarrow U_{3\times3}\, \hat{D}_{3\times3}\, V_{3\times3}^t \quad (2.55)$$

La contrainte de rang 2 est renforcée en annulant la plus petite des valeurs singulières :

$$D \leftarrow \begin{pmatrix} \widehat{D}_{11} & & \\ & \widehat{D}_{22} & \\ & & 0 \end{pmatrix} \tag{2.56}$$

On recompose ensuite la matrice essentielle E :

$$E \leftarrow UDV^t$$

2.7.4.2. Calcul de la matrice de rotation et de translation relatives à partir de la matrice essentielle

L'objectif est de déterminer les matrices de rotation R_r et de translation T_r relatives, liant les deux vues, à partir de la matrice essentielle. Ces matrices sont données, à partir de la SVD renforcée de la matrice essentielle E, par :

$$R_r^1 = UWV^t \quad \text{ou} \quad R_r^2 = UW^tV^t \tag{2.57}$$

avec : $W = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$

$$T_r^1 = u_3 \quad ou\ T_r^2 = -u_3 \tag{2.58}$$

Avec u_3 est la troisième colonne de la matrice U.

Nous avons quatre combinaisons possibles pour les matrices de rotation et translation relatives. Pour déterminer quelle combinaison rotation/translation constitue la bonne configuration (points 3D face au stéréoscope), il faut analyser l'interprétation géométrique des solutions. La Figure 2.11 présente les configurations associées aux différentes combinaisons de solutions. Le passage de R_r^1 à R_r^2 correspond à une inversion de sens de la caméra (passage de B à B' dans la Figure 2.11). Pour trouver la bonne configuration, on procède de la manière suivante :

- On reconstruit un point 3D à partir d'un couple apparié ;
- On détermine la profondeur du point pour les deux caméras ;
- On choisit les matrices R_r et T_r de manière à obtenir une profondeur positive dans les deux vues.

La profondeur d'un point tridimensionnel M, par rapport à une caméra ayant pour matrice de projection $P = [\mathcal{H}_{3x3}\ \mathrm{p}_4]$, est définie par :

$$depth(M,P) = \frac{sign(det(\mathcal{H}))h}{H\|h_3\|} \tag{2.59}$$

avec : $h(x\ y\ 1)^t = P.M = P.(X\ Y\ Z\ H)^t$ et h_3 la 3ième ligne de $\mathcal{H}$.

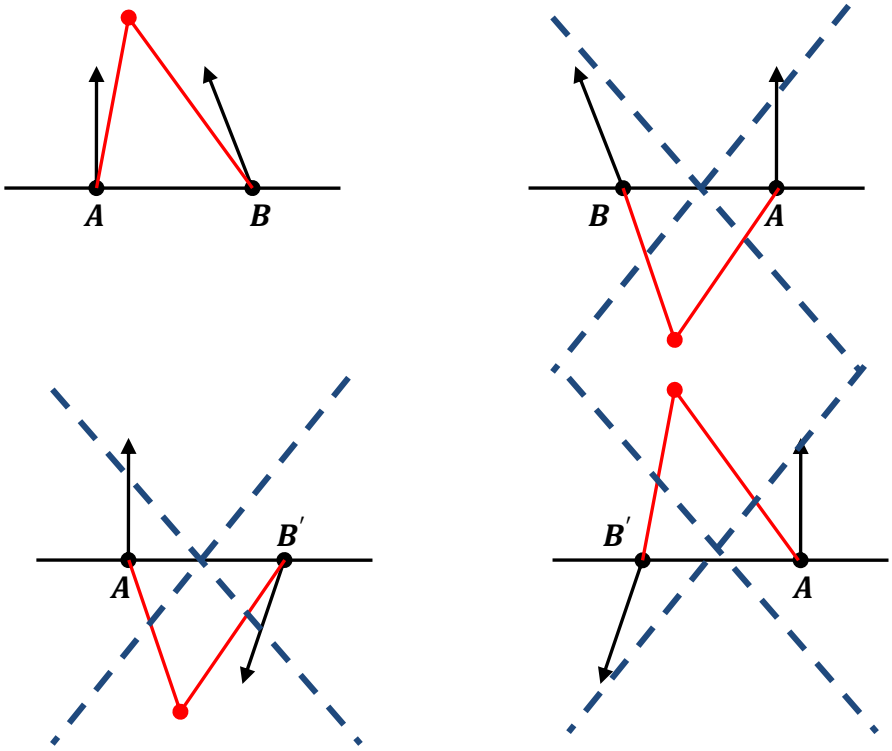


Figure 2.11 : Combinaisons possibles pour la position des caméras

2.8. Calibration d'un système de stéréovision

Le modèle sténopé (cf. §2.5) modélise une caméra idéale (simple projection perspective). Le système optique induit des distorsions géométriques qui affectent la projection des points. Ainsi, un point projeté dans une image ne correspond pas au modèle sténopé. La distorsion est d'autant plus élevée que le champ de vue de la caméra est grand. Cependant, il est possible de modéliser la distorsion en enrichissant le modèle sténopé par des termes supplémentaires (le modèle devient non linéaire).

La calibration permet de déterminer les paramètres intrinsèques et extrinsèques d'une ou plusieurs caméras en utilisant une mire dont le modèle est connu à l'avance. Les approches [74] nécessitent d'avoir un modèle de mire parfait. Des approches plus récentes [20,32] permettent d'estimer les paramètres du système de stéréovision et le modèle de la mire en même temps.

Pour réaliser la calibration, les paramètres à estimer sont :

- Les paramètres intrinsèques de chaque caméra ;
- Le déplacement entre les caméras composé d'une rotation R_r et une translation T_r ;
- Les coefficients de distorsion.

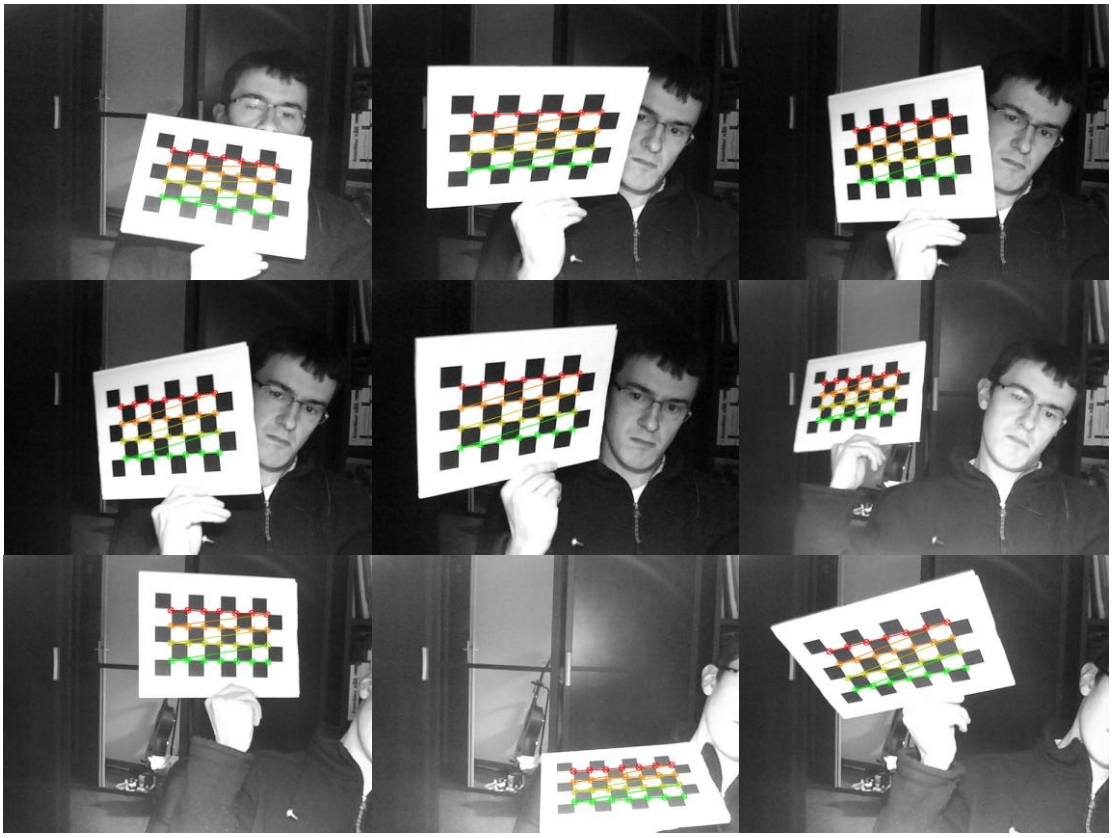


Figure 2.12 : Images de mires utilisées pour réaliser une calibration d'une caméra.

La toolbox Matlab développé par Jean-Yves Bouguet [5] permet de calibrer un système stéréoscopique en utilisant plusieurs images d'une mire de type damier (cf. Figure 2.12).

2.9. Conclusion

Par ce chapitre, nous avons pu exposer les différents outils fondamentaux pour la localisation par vision artificielle que nous allons utiliser dans les prochains chapitres. Le point SIFT est la primitive utilisée pour reconstruire un modèle 3D de l'environnement.

Les différentes méthodes mathématiques, tels que la géométrie épipolaire, calcul de la matrice fondamentale ou essentielle vont permettre de déterminer les poses relatives du stéréoscope lors de reconstruction ou localisation. Les méthodes robustes tels que RANSAC permettent d'apporter la précision aux différents algorithmes employés.

CHAPITRE 3

SIMULATEUR

3.1. Introduction

De nos jours, les systèmes d'information géographique sont de plus en plus utilisés dans de nombreuses applications telles que l'urbanisme, la réalité virtuelle, etc. Un système d'information géographique est l'intégration organisationnelle d'un logiciel et de données géographiques.

Par exemple, une reconstruction urbaine virtuelle en 3D peut être réalisée à partir d'images photographiques. Les outils permettant ce type de réalisation sont aujourd'hui très utilisés dans plusieurs applications. En effet, cela permet de présenter les projets de construction et/ou d'aménagement urbains de manière très concrète. Cette présentation peut aller de la représentation par de simples images à toute une animation virtuelle du projet. Cependant, reproduire une ville en 3D par le biais de photographies reste un problème difficile à résoudre.

Grâce à l'innovation et à l'avancée technologique, la modélisation urbaine 3D est très demandée, car elle aide à la prise de décision, à la planification et à la gestion de l'évolution des projets de construction et/ou d'aménagement. En effet, pour établir un plan d'aménagement de l'urbanisme, une étude précise est obligatoire pour permettre la coexistence des différents usagers, que ce soit du point de vue de la sécurité, de l'écologie ou de la vie en collectivité.

La simulation 3D permet d'opérer des choix. Prenons le cas de la modification de la circulation d'un quartier. Nous ne pouvons pas prévoir ce que ce changement donnera et les différents impacts qu'il aura sans passer par la modélisation et la simulation 3D. La simulation en temps réel et la visualisation des effets liés à l'opération permettent de valider certains choix en appréhendant les avantages et les inconvénients du projet. La simulation permet de se projeter sur le futur et de se rendre compte des actions à mettre en place.

La modélisation 3D est basée sur l'utilisation de plusieurs types de données : photographies au niveau du sol, photographies aériennes ou satellitaires. Les niveaux de précision dépendent ainsi du type de données utilisé.

Ce chapitre présente les étapes principales du développent d'un simulateur 3D qui est utilisé pour évaluer notre approche de localisation. Ce simulateur permet de jouer des scénarii de navigation, avec la modélisation 3D d'environnements et la planification de trajectoire d'objets mobiles (véhicules par exemple). Il permet

également de simuler les capteurs vidéo (caméras, stéréoscopes). Les images provenant de ces capteurs constituent les données des méthodes développées.

3.2. Approches de modélisation urbaine 3D

Pour réaliser un modèle urbain virtuel, plusieurs méthodes de modélisation 3D ont vu le jour. Ces méthodes diffèrent entre elles à plusieurs niveaux : les données nécessaires, le degré d'automatisation de la méthode, les types de zones supportés. Tout dépend du résultat escompté. Certaines méthodes permettent de reproduire de manière assez précise et fidèle un site urbain. D'autre se contente d'une reconstruction 3D vague et générale.

Trouver une méthode pour la reconstruction 3D revient à réaliser différentes tâches obligatoires pour obtenir le modèle voulu :

- Localiser les bâtiments sur une carte ;
- Calculer le relief du terrain ;
- Trouver la hauteur des bâtiments ;
- Trouver la forme des bâtiments (principalement les toitures) ;
- Augmenter le réalisme de la scène (ex.: utilisation de textures).

Pour réaliser ces étapes, plusieurs techniques ont été développées :

- **Plan de cadastre** : chaque ville possède des plans cadastraux. Ce sont des reproductions cartographiques à plusieurs échelles représentant différents biens immobiliers (terrains ou parcelles construites) du territoire d'une commune. Ils se composent d'un certain nombre de feuilles reliées en un atlas et comprennent le tableau d'assemblage qui donne l'image schématique du territoire de la commune et les feuilles parcellaires. Ceci permet de manière assez simple de savoir où sont situés les différents bâtiments. De plus, un plan de cadastre est toujours actualisé, c'est donc une source de données fiable et précise. Cela donne aussi une bonne définition en 2D du bâtiment, des différentes pièces, de sa forme au sol ainsi que du type d'habitation. Cependant, le plan ne donne pas d'informations sur le relief et la forme du bâtiment. Il est sous un format papier dans les archives de la ville et rarement sous format électronique.

- **Détection de bâtiments par photographies aériennes** : il s'agit d'une méthode permettant d'effectuer une modélisation 3D de sites urbains. Elle consiste à segmenter différentes photographies (images aériennes ou satellitaires à très haute résolution) afin de détecter et localiser les bâtiments.
- **Calcul du modèle numérique d'élévation (MNE)** : il est possible de calculer le MNE de deux manières :
  - Soit à partir de couples stéréoscopiques d'images aériennes ou satellitaires, en utilisant la corrélation automatique ou la restitution. Pour générer un MNE, différents algorithmes de stéréovision sont utilisés. Ils sont basés sur la juxtaposition par deux des différentes photographies en cherchant à associer chaque point d'une image à son homologue (s'il existe) dans l'autre image, puis à calculer la position 3D du point correspondant.
  - Soit à l'aide d'un capteur actif. Un laser, par exemple, permet de connaître l'élévation des points au sol. Cette technique est très appréciée, car elle permet d'obtenir un modèle d'élévation d'une très grande précision.
- **Calcul des hauteurs de bâtiments :** il est possible de calculer la hauteur des bâtiments de deux manières :
  - Par analyse des ombres.
  - A l'aide d'outils interactifs.

Parfois, selon l'environnement à modéliser, l'utilisation d'une méthode simple peut permettre à elle seule une reconstruction urbaine en 3D, en considérant le cadastre ainsi que les modèles numériques. Seules la hauteur des bâtiments et leur forme architecturale ne paraîtront pas sur le résultat de la modélisation. Le modèle obtenu peut être complété par la suite en utilisant photographies de l'environnement. Cette utilisation consiste tout simplement à superposer la photographie au modèle 3D obtenu afin de trouver de manière assez précise la hauteur des bâtiments.

3.2.1. Déterminer la forme des bâtiments

Ayant la surface et la hauteur d'un bâtiment, il faut déterminer sa forme pour obtenir une visualisation proche de la réalité.

Dans la plupart des cas, on ne s'attache pas à autant de détails, une reconstruction approximative peut suffire. La reconstruction 3D des primitives géométriques reste simplifiée, et cela suffit largement à représenter la plupart des bâtiments. Il s'agit de

paramétrer ceux-ci à l'aide de différentes caractéristiques comme la hauteur de l'immeuble, par exemple.

Pour rendre le bâtiment modélisé plus proche de la réalité, il faut trouver quelle primitive pourrait caractériser au mieux celui-ci. La segmentation d'une image aérienne permet d'extraire cette information. On peut aussi utiliser un outil interactif où l'on sélectionnera un bâtiment et l'on choisira la primitive qui correspond le mieux, en fonction des éléments visibles sur les photographies.

3.2.2. Formes complexes d'un groupe de bâtiments

S'agissant d'un groupe de bâtiments, il n'est plus possible de considérer une simplification des primitives pour caractériser le groupe de bâtiments. Ces derniers, ne peuvent plus être représentés par de simples primitives. Ils doivent être modélisés par des formes plus complexes, en effectuant une décomposition en parties élémentaires.

3.2.3. Méthode d'unification par séquences d'images

Nous pouvons utiliser des caméras non calibrées pour obtenir des images susceptibles d'être exploitées par la modélisation 3D. On associe chaque point d'une image à son correspondant dans l'autre image, puis on calcule la position 3D du point correspondant de manière automatique. Avec une suite d'images, on peut effectuer le calcul de la profondeur par stéréovision à l'aide de différentes paires d'images. On peut ainsi reconstruire la surface d'une scène complexe à l'aide de maillages de triangles. Un autre avantage de l'utilisation des images est l'augmentation du degré de réalisme par l'extraction automatique de textures lors du rendu 3D.

3.3. Le modèle virtuel de la place Stanislas

Le modèle virtuel employé dans notre simulateur a été développé par la société technomade[8]. Cette société est une start-up issue de l'INRIA Nancy. Elle a été créée en 2003 par le programme de transfert technologique de l'INRIA Lorraine pour travailler sur la visualisation interactive d'objets par Internet. Cette société produit des outils d'aide à la décision et à la communication pour des projets d'architecture et

[8] http://www.tecnomade.fr

d'urbanisme destinés aux collectivités locales. Elle remplit ce rôle par la réalisation de maquettes numériques 3D réalistes de morceaux de villes manipulables en 3D "temps-réel".

L'efficacité du procédé repose sur la technologie 3D (Représentation de la hauteur, de la largeur et de la profondeur dans l'espace pour augmenter le niveau de réalisme virtuel) "temps-réel" interactive. Cette technologie permet de visualiser instantanément n'importe quel point de vue à la demande (vue aérienne, automobile ou piéton), et de se déplacer de manière fluide dans toutes les directions. Par interactivité, on entend le fait de pouvoir se déplacer librement dans le l'environnement virtuel, dans n'importe quelle direction. Ceci permet de visualiser immédiatement telle ou telle partie de l'environnement à la demande, contrairement aux animations pré-calculées, qui défilent sans intervention possible.

Le modèle 3D de la place Stanislas (Nancy) a été mis à disposition pour la présentation de la rénovation de la place lors des journées d'expérimentation du projet PREDIT MobiVIP. Ce modèle a été conçu avec une précision centimétrique pour pouvoir mettre en correspondance des images réelles et virtuelles [7]. La figure Figure 3.1 illustre un ensemble de vue 3D de cet environnement.

Figure 3.2 : Vues du modèle 3D de la place Stanislas modélisé par la société Tecnomade. Images réalisées par le moteur graphique propriétaire de 3D temps réel de Tecnomade appelé PortEye.

Le modèle 3D est composé de plusieurs éléments :

- Les textures et les ombres[9], utilisées pour l'affichage des structures 3D du modèle. Celles-ci sont stockées dans le format JPEG ou PNG. La Figure 3.3 présente un échantillon des textures du modèle virtuel ;
- La définition des maillages 3D de la scène. Chaque maillage est constitué d'un ensemble de points 3D identifié par un nom unique et d'un type. A chaque point 3D sont également associées deux coordonnées de texture et un vecteur

[9] Technique appelé « shadow mapping » [84] permettant de pré-calculer les ombres et de les enregistrer dans une texture.

représentant la normale à la surface au point. La Figure 3.4 présente un extrait de la définition des maillages 3D ;

- La définition de l'ensemble des objets 3D composant la scène virtuelle (cf. Figure 3.5). Chaque objet est défini par un maillage et des propriétés sur l'aspect de l'objet. Ainsi, on peut configurer les textures à appliquer, des propriétés matérielles de l'objet ou encore sa transparence.

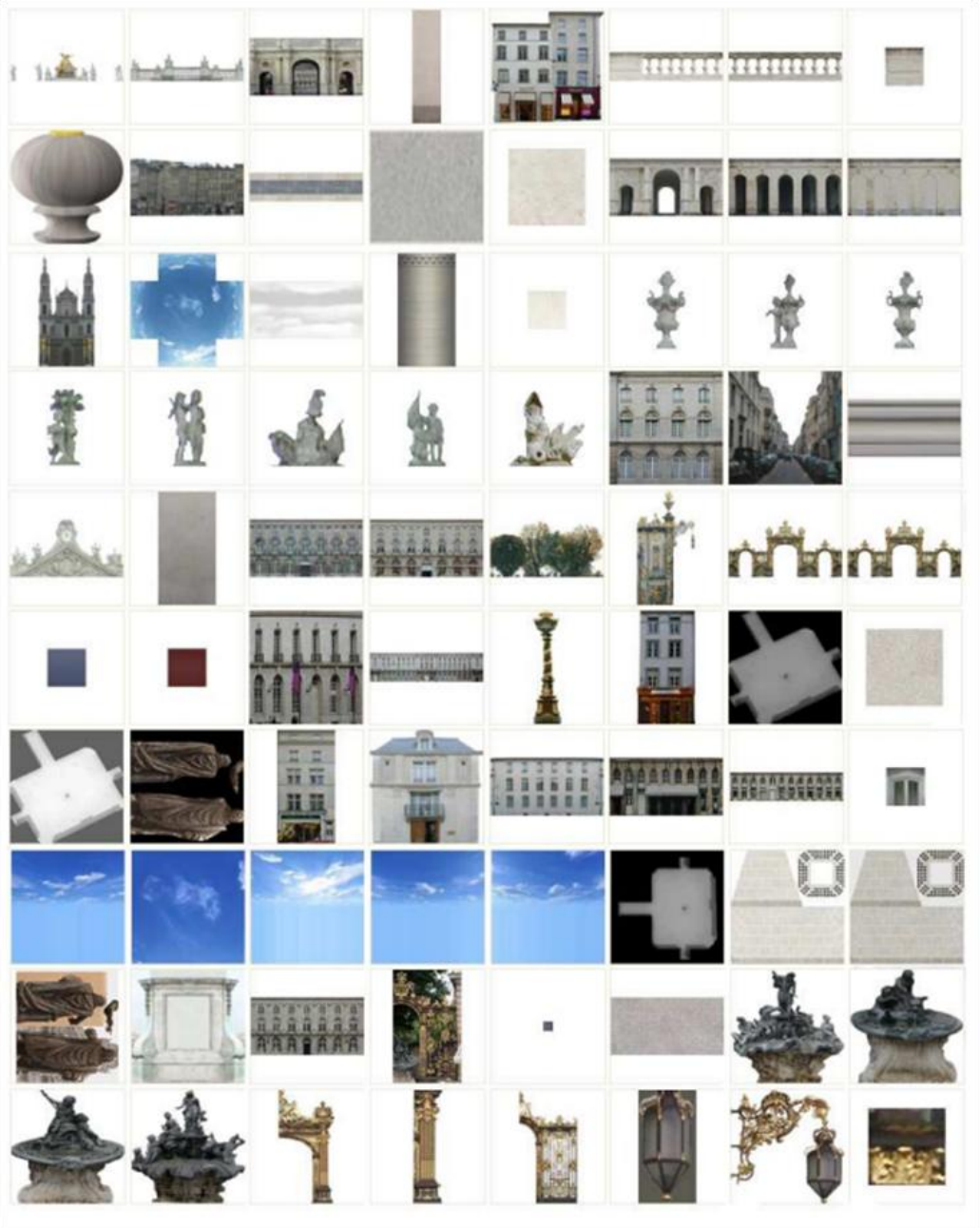


Figure 3.3 : Echantillon de textures du modèle 3D de la place Stanislas

```
<Definitions>
 <Geometrie Nom="Geom_Line41_630" Type="triangles_bruts">
 </Geometrie>
 <Geometrie Nom="Geom_Rectangle42_631" Type="triangles_bruts">
  <Point3D Vertex="-59.1352 -22.416 17.7403" Normale="0.448328 -0.893869 0.0" Tex1="0.299503 0.000499964"/>
  <Point3D Vertex="-56.4535 -21.0709 17.7403" Normale="0.448328 -0.893869 0.0" Tex1="0.995809 0.000499547"/>
  <Point3D Vertex="-56.4535 -21.0709 19.4422" Normale="0.448328 -0.893869 0.0" Tex1="0.995809 0.9995"/>
  <Point3D Vertex="-59.1352 -22.416 17.7403" Normale="0.448328 -0.893869 0.0" Tex1="0.299503 0.000499964"/>
  <Point3D Vertex="-56.4535 -21.0709 19.4422" Normale="0.448328 -0.893869 0.0" Tex1="0.995809 0.9995"/>
  <Point3D Vertex="-59.1352 -22.416 19.4422" Normale="0.448328 -0.893869 0.0" Tex1="0.299503 0.9995"/>
  <Point3D Vertex="-62.6214 -24.1645 17.7403" Normale="0.448328 -0.893869 0.0" Tex1="0.299503 0.000499964"/>
  <Point3D Vertex="-59.9398 -22.8195 17.7403" Normale="0.448328 -0.893869 0.0" Tex1="0.995809 0.000499547"/>
  <Point3D Vertex="-59.9398 -22.8195 19.4422" Normale="0.448328 -0.893869 0.0" Tex1="0.995809 0.9995"/>
  <Point3D Vertex="-72.8053 2.83156 19.4422" Normale="-0.448328 0.893869 0.0" Tex1="0.995809 0.9995"/>
  <Point3D Vertex="-78.9731 -0.261995 17.7403" Normale="-0.448328 0.893869 0.0" Tex1="0.299503 0.000499964"/>
  <Point3D Vertex="-76.2915 1.08303 19.4422" Normale="-0.448328 0.893869 0.0" Tex1="0.995809 0.9995"/>
  <Point3D Vertex="-76.2915 1.08303 17.7403" Normale="-0.448328 0.893869 0.0" Tex1="0.995809 0.000499547"/>
  <Point3D Vertex="-78.9731 -0.261995 17.7403" Normale="-0.448328 0.893869 0.0" Tex1="0.299503 0.000499964"/>
  <Point3D Vertex="-78.9731 -0.261995 19.4422" Normale="-0.448328 0.893869 0.0" Tex1="0.299503 0.9995"/>
  <Point3D Vertex="-76.2915 1.08303 19.4422" Normale="-0.448328 0.893869 0.0" Tex1="0.995809 0.9995"/>
  <Point3D Vertex="-82.459 -2.01035 17.7403" Normale="-0.448328 0.893869 0.0" Tex1="0.299503 0.000499964"/>
  <Point3D Vertex="-79.7773 -0.665323 19.4422" Normale="-0.448328 0.893869 0.0" Tex1="0.995809 0.9995"/>
  <Point3D Vertex="-79.7773 -0.665323 17.7403" Normale="-0.448328 0.893869 0.0" Tex1="0.995809 0.000499547"/>
  <Point3D Vertex="-82.459 -2.01035 17.7403" Normale="-0.448328 0.893869 0.0" Tex1="0.299503 0.000499964"/>
  <Point3D Vertex="-82.459 -2.01035 19.4422" Normale="-0.448328 0.893869 0.0" Tex1="0.299503 0.9995"/>
  <Point3D Vertex="-79.7773 -0.665323 19.4422" Normale="-0.448328 0.893869 0.0" Tex1="0.995809 0.9995"/>
    ...
 </Geometrie>
 <Geometrie Nom="Geom_Line42_632" Type="triangles_bruts">
  <Point3D Vertex="-56.1222 -19.5385 6.56499" Normale="-0.448328 0.893869 0.0" Tex1="-0.0165866 0.385341"/>
  <Point3D Vertex="-56.1222 -19.5385 7.43999" Normale="-0.448328 0.893869 0.0" Tex1="-0.0124131 0.385341"/>
    ...
 </Geometrie>
   ...
</Definitions>
```

Figure 3.4 : Extrait de la définition des maillages constituant le modèle 3D. Les données sont stockées dans le format XML[10].

[10] XML : Extensible Markup Language : langage informatique de balisage générique.

```
<SceneGraph>
 <Objet Nom="Line41">
  <InsererGeometrie Ref="Geom_Line41_630"/>
 </Objet>
 <Objet Nom="Rectangle42">
  <Culling Etat="0"/>
  <Texture Canal="1" Fichier="balustre3.png" TexMode="modulate"/>
  <TrierZ/>
  <Materiel Ambiante="1.0 1.0 1.0" Diffuse="1.0 1.0 1.0" Speculaire="1.0 1.0 1.0" Brillance="0.0"/>
  <Transparence Facteur="1.0"/>
  <InsererGeometrie Ref="Geom_Rectangle42_631"/>
 </Objet>
 <Objet Nom="Line42">
  <Texture Canal="1" Fichier="musee2.JPG" TexMode="modulate"/>
  <Materiel Ambiante="1.0 1.0 1.0" Diffuse="1.0 1.0 1.0" Speculaire="1.0 1.0 1.0" Brillance="0.0"/>
  <Transparence Facteur="1.0"/>
  <InsererGeometrie Ref="Geom_Line42_632"/>
 </Objet>
 <Objet Nom="Plane122">
  <Culling Etat="0"/>
  <Texture Canal="1" Fichier="Deco06.PNG" TexMode="modulate"/>
  <TrierZ/>
  <Materiel Ambiante="1.0 1.0 1.0" Diffuse="1.0 1.0 1.0" Speculaire="1.0 1.0 1.0" Brillance="0.0"/>
  <Transparence Facteur="1.0"/>
  <InsererGeometrie Ref="Geom_Plane122_656"/>
 </Objet>
 <Objet Nom="Box06">
  <Texture Canal="1" Fichier="Gris2.jpg" TexMode="modulate"/>
  <Materiel Ambiante="1.0 1.0 1.0" Diffuse="1.0 1.0 1.0" Speculaire="0.9 0.9 0.9" Brillance="0.25"/>
  <Transparence Facteur="1.0"/>
  <InsererGeometrie Ref="Geom_Box06_657"/>
 </Objet>

 ...
</SceneGraph>
```

Figure 3.5 : Extrait de la configuration des objets du modèle virtuel. Les données sont stockées dans le format XML

3.4. Simulateur

L'évolution de l'informatique a permis l'essor des techniques de la réalité virtuelle [63]. L'augmentation importante de la puissance intrinsèque des ordinateurs a offert de nouvelles possibilités de créer en temps réel des images de synthèse et une interaction entre l'utilisateur et le monde virtuel. Ainsi, la réalité virtuelle procure à l'homme d'être acteur d'un environnement virtuel. Autrement dit, l'apport n'est pas la création d'environnements virtuels en soi, mais le fait de pouvoir « interagir virtuellement » dans un monde artificiel.

Les systèmes de réalité virtuelle sont fondés sur différentes conditions pour fournir une « pseudo-immersion naturelle » à un utilisateur :

- Les interactions de l'utilisateur d'un système de réalité virtuelle doivent être perçues en temps réel dans l'environnement virtuel ;
- La modélisation et la numérisation d'un environnement réel doivent être proches de la réalité et permettre une bonne immersion dans le système de réalité virtuelle ;
- Une interface matérielle est nécessaire à l'utilisateur pour permettre une évolution dans l'environnement virtuel.

Les techniques de la réalité virtuelle permettent à toute personne d'agir sur un environnement virtuel par l'intermédiaire de mouvements. D'où la nécessite des dispositifs sensoriels comme une souris 2D, un clavier ou encore des gants de données.

3.4.1. Fonctionnement des modèles virtuels

La formation d'une image virtuelle se décompose en différentes étapes (cf. Figure 3.6). A partir de données brutes (points, lignes, structures 3D, …), il est possible de créer une image proche d'une image réelle.

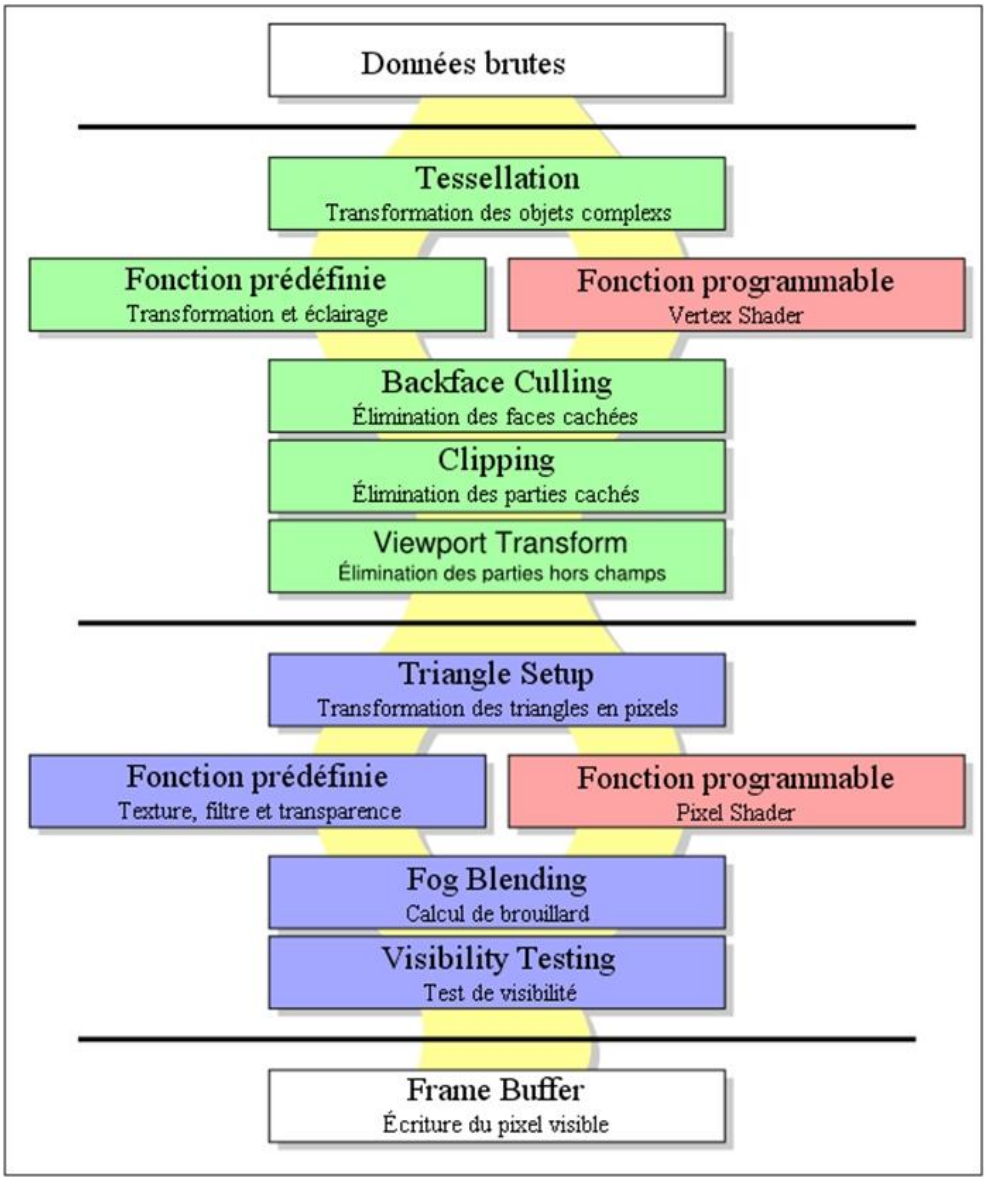


Figure 3.6 : Pipeline graphique du rendu d'une image virtuelle, permettant de transformer des données brutes en une image.

Les sections suivantes décrivent les points importants et nécessaires à la génération d'une image virtuelle.

3.4.1.1. Vertex shaders

Dans la liste des tâches graphiques séquentielles nécessaires à l'affichage d'une image 3D, la tâche du vertex shader se situe un niveau de la tâche dite de « transformations et éclairage ». Pour comprendre cette partie, il faut tout d'abord comprendre ce qu'est un shader.

Exécuté par le GPU (Graphics Processing Unit), un shader est un jeu d'instructions consacré au rendu graphique en 3D. Utilisé en image de synthèse, il paramètre une partie du processus de rendu d'une scène 3D, c'est à dire, l'affichage à l'écran réalisé par une carte graphique ou un moteur de rendu logiciel. Il effectue un traitement sur les données d'entrées, puis envoie les résultats au niveau suivant du pipeline. Les shaders permettent de décrire des effets de déformation, transparence, diffusion de la lumière, texture, réflexions et autres.

Il existe 3 types de shaders :

- **Le vertex shader** prend un sommet comme entrée. Il intervient lors des calculs de transformations. Il est utilisé une seule fois pour tous les sommets passés au GPU via des vertex buffers. Ce type de shader permet des effets de déformation de l'objet 3D, ou encore des calculs d'éclairage basiques.
- **Le geometry shader** prend une primitive comme entrée. Il est utilisé une seule fois pour toutes les primitives passées au GPU. Une primitive est un point, une ligne, ou un triangle. Il permet d'ajouter ou de supprimer des vertex à un objet 3D. Il permet de mieux détailler la géométrie de l'objet.
- **Le pixel shader** prend un pixel (parfois appelé un fragment) comme entrée et renvoie une couleur, qui est la couleur à afficher à l'écran pour ce pixel. Il est utilisé une seule fois pour chaque pixel de la primitive à afficher. Il intervient après les vertex shaders lors du calcul de l'éclairage des objets.

Les shaders fonctionnent particulièrement bien en parallèle sur des systèmes multi-GPU. Cela permet d'effectuer un traitement vectorisé, réduisant ainsi la charge de l'unité centrale et offrant un résultat de manière plus rapide. Les shaders sont flexibles et efficaces. Il est possible en effet de créer des surfaces compliquées grâce à une simple géométrie.

A travers les définitions précédentes, nous pouvons comprendre qu'un vertex shader est l'utilisation de shader sur les différents sommets des polygones. Ceci permet d'effectuer de la transformation et l'éclairage programmable (Pas clair). En entrée, nous trouverons donc des vertices (sommets du polygone) non transformés et non éclairés. Le traitement du shader permet de calculer la position des vertices, les valeurs des couleurs et, éventuellement, les coordonnées des textures.

Pour arriver au résultat souhaité, il n'existe pas de fonction par défaut. Toute la procédure de projection doit être programmée. Le but de cette programmation est en fait de l'exécuter au niveau de chaque vertex. Sans trop solliciter le processeur, de nombreux effets sont accessibles.

3.4.1.1.1. Changements d'illumination

Le pixel shader est un shader dont le but est de calculer la couleur que chaque pixel doit avoir individuellement. Le shader prend certaines entrées (position, coordonnées de texture, couleur) du pixel à colorer. La couleur calculée est ensuite renvoyée au pipeline. Les entrées peuvent provenir du geometry shader ou du vertex shader.

Le pixel shader est souvent employé pour réaliser des effets de lumière (cf. Figure 3.7). On l'utilise également pour changer l'illumination d'une scène virtuelle (cf. Figure 3.8)

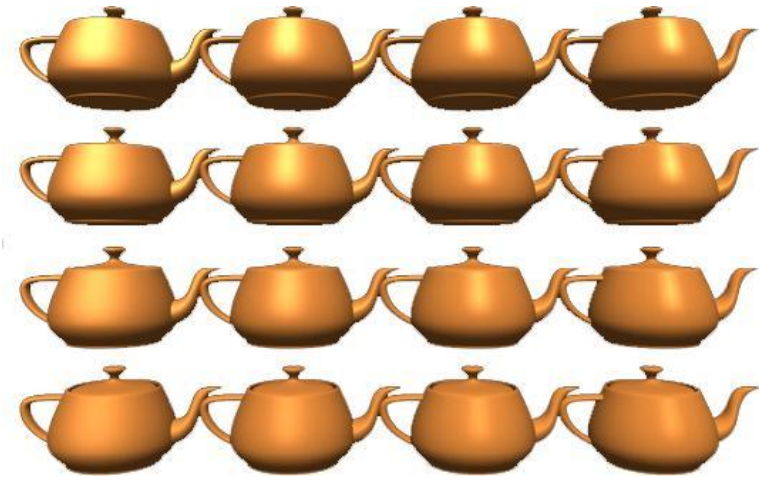


Figure 3.7 : Saturation de la lumière sur une partie de l'objet pour la simulation d'effets de réflexion spéculaire.

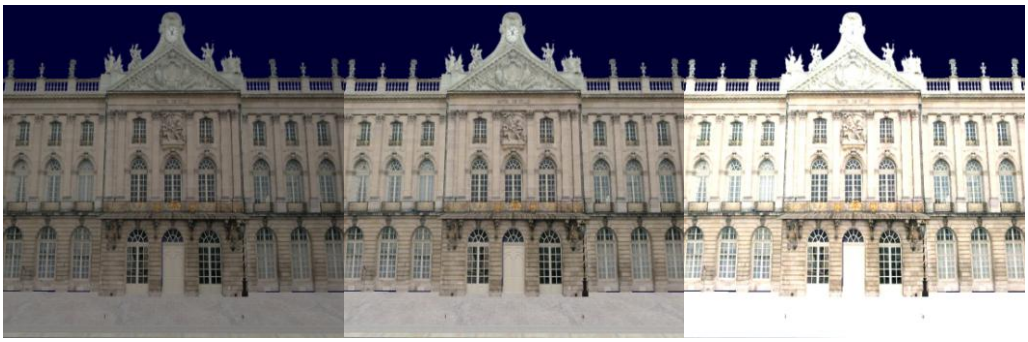


Figure 3.8 : Changement de l'éclairage par pixel shader.

3.4.1.1.2. Bruits d'images

Les sources de bruits sont diverses et peuvent être modélisées de plusieurs façons dans une image. On peut distinguer différentes sources de bruits :

- Une sur- ou sous-illumination qui réduit l'intervalle des couleurs ;
- L'étape d'échantillonnage qui provoque un phénomène de noiret ;
- La quantification ;
- La transmission des images qui peut subir des perturbations ;

Le bruit peut être dépendant (le bruit de quantification) ou indépendant (les poussières sur l'objectif) des données de l'image. La modélisation de ces bruits s'exprime par un terme additif ou multiplicatif au signal original. Généralement, les bruits sont considérés comme étant impulsionnels (bruit gaussien, exponentiel, …). On peut également avoir un bruit *de poivre et sel*[9] qui modélise les poussières sur une pellicule ou un scanner.

Les pixels shaders permettent de simuler ces différentes sources de bruits en temps réel, en appliquant à l'image les bruits souhaités. Il suffit d'appliquer le modèle associé aux bruits souhaités.

3.4.1.2. Backface culling

Pour comprendre ce qu'est le backface culling, il faut tout d'abord présenter ce qu'est le culling. En programmation 3D, le culling consiste en l'élimination, avant

l'affichage à l'écran, de parties de la scène 3D. Ces parties ne seront pas visibles à l'écran, car :

- Elles sont des parties cachées ;
- Elles sont des parties qui ne sont pas dans le champ de vision ;

Le backface culling est l'élimination des faces qui ne sont pas face à la caméra. Il s'agit d'un culling appliqué aux polygones. Pour pouvoir réaliser cette opération, il faut calculer deux vecteurs :

- Le vecteur entre la position de la caméra et le centre du polygone ;
- Le vecteur normal au polygone ;

Si le produit scalaire de ces deux vecteurs est positif, alors cela signifie que le triangle est orienté dans le même sens que la vue. L'élément n'est donc pas face à la caméra, et il peut ainsi être éliminé du rendu.

3.4.1.3. Clipping

Le clipping est une technique qui consiste à éliminer le tracé d'éléments que l'on ne souhaite pas voir à l'écran (cf. Figure 3.9). Il s'agit des éléments qui sont extérieurs à une surface bien délimitée. Ceci permet d'optimiser le temps de calcul, en évitant le calcul des objets extérieurs de la zone de clipping.

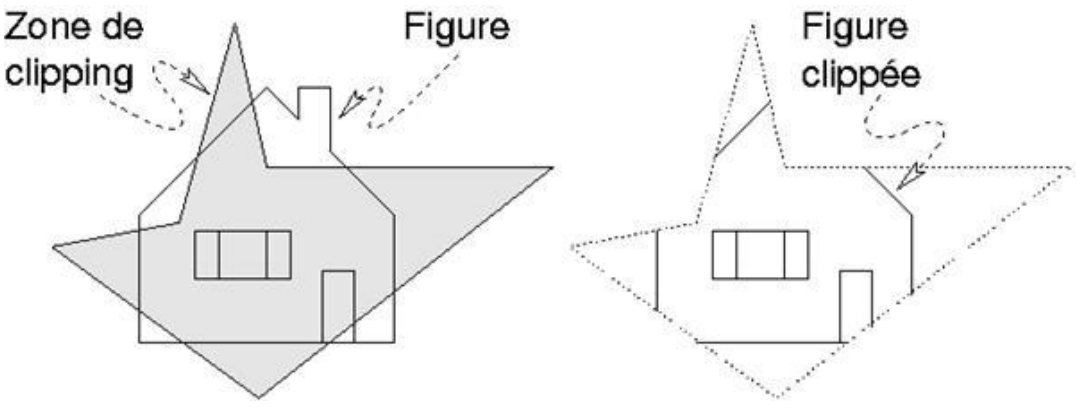


Figure 3.9 : Zone de clipping.

3.4.1.4. Rastérisation

La rastérisation est un procédé qui consiste à convertir une image vectorielle en une image matricielle (pixels), destinée à être affichée sur un écran ou imprimée par un matériel d'impression.

La première étape de la rastérisation est la projection du modèle 3D vers le plan 2D (cf. Figure 3.10). Ceci se fait en 3 sous-étapes :

- L'intégration du modèle dans le monde "Model to World" par une succession d'opérations réalisées sur un objet. Ces opérations sont la rotation, la translation et la mise à l'échelle.

  Ceci permet de produire la scène à afficher par création, placement et orientation des objets qui la composent ;
- La prise en compte de la position et de l'angle de la caméra "World to Vue" : elle permet de fixer la position et l'orientation de la caméra de visualisation ;
- La prise en compte des dimensions de l'écran "Vue to Projection" : elle permet de fixer les caractéristiques optiques de la caméra de visualisation (type de projection, ouverture). Une transformation d'affichage (Viewport) est effectuée pour fixer la taille et la position de l'image sur la fenêtre d'affichage ;

Chacune de ces étapes correspond à une matrice. Le calcul consiste en une simple multiplication des coordonnées de chaque vertex avec les trois matrices. On obtient ainsi pour chaque sommet, ses coordonnées 2D ainsi que sa profondeur.

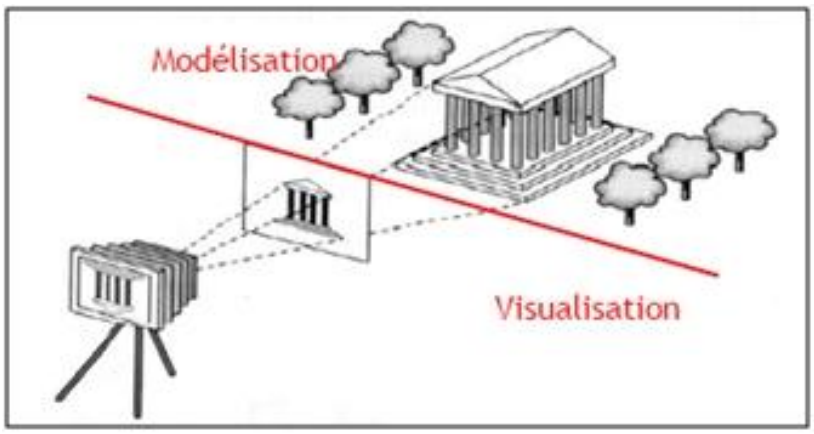


Figure 3.10 : Projection d'une scène 3D sur un plan image

3.5. Conclusion

Dans ce chapitre, nous avons présenté une plateforme de simulation permettant de générer des environnements 3D virtuels et de simuler des capteurs vidéo. Le but de ce simulateur est de créer des scenarii et de les jouer afin d'acquérir les données vidéo correspondant nécessaires à l'expérimentation et aux tests de nos travaux.

Le modèle 3D de l'environnement utilisé est celui de la place Stanislas à Nancy. Notre simulateur permet d'introduire des objets statiques et/ou dynamiques (obstacles, véhicules mobiles, etc.) dans cet environnement. Il permet également d'associer à un véhicule (véhicule prototype) un ensemble de capteurs vidéo permettent de percevoir son environnement. Le processus d'acquisition des images par le simulateur considère de nombreux paramètres tels que le changement d'éclairage, le bruit, la position des caméras, le calibrage, etc.

CHAPITRE 4

LOCALISATION PAR STEREOVISION

4.1. Introduction

Ce chapitre détaille la méthode proposée pour la localisation stéréoscopique. Celle-ci est décomposée en deux étapes. La première étape constitue une phase d'apprentissage qui permet de construire un modèle 3D de primitives de l'environnement. La deuxième étape est consacrée à la localisation du véhicule par rapport au modèle.

4.2. Reconstruction

La construction d'un modèle de points 3D est nécessaire pour calculer la position globale d'une caméra. Les points extraits du modèle de données sont issus de points caractéristiques extraits des deux images d'un stéréoscope. Ces points caractéristiques doivent être invariants aux changements d'échelle, aux rotations ou encore aux translations pour pouvoir effectuer une mise en correspondance aisée. La plupart des méthodes de reconstruction utilisent le détecteur de Harris [24]. Celui-ci étant sensible aux changements d'échelle, la construction de la base de données demande un nombre important de points et de nombreuses acquisitions. De plus, lors de la phase de localisation, le véhicule doit rester proche de sa trajectoire d'apprentissage.

La méthode proposée utilise les points SIFT (Scale Invariant Feature Transform). Introduite par Lowe [38], la technique SIFT fournit des points avec des caractéristiques permettant une meilleure mise en correspondance des points, malgré les changements de l'angle de vue et des distances, et même sous différentes conditions de luminosité. Les processus de reconstruction et de localisation sont basés sur la mise en correspondance des points SIFT.

La Figure 4.1 présente les résultats de l'extraction des points SIFT à partir d'un couple d'images stéréoscopiques virtuelles. Avec une résolution de 640x480, environ 2000 points sont extraits par image. Cette quantité de points est suffisante pour obtenir de bons résultats pour la reconstruction. Cependant, on peut augmenter ou diminuer le nombre de points en fonction du nombre d'échelles, qui constitue un paramètre principal de la méthode SIFT.

A chaque étape de la reconstruction, les points SIFT sont extraits dans chacune des images rectifiées d'un couple stéréoscopique. Ces points sont ensuite mis en correspondance pour obtenir des points 3D dans l'espace. Les appariements obtenus

sont robustes et répétitifs, ce qui permet de les utiliser comme référence pour suivre les objets et les déplacements dans l'environnement.

Figure 4.1 : Extraction des points SIFT d'un stéréoscope virtuel.

4.2.1. Mise en correspondance stéréoscopique

Afin d'établir une correspondance robuste entre les points SIFT, différentes contraintes sont appliquées. Celles-ci permettent de réduire l'espace de recherche des correspondances, mais également d'éliminer de faux appariements.

4.2.1.1. La contrainte épipolaire

La mise en correspondance est une procédure complexe qui demande un temps de calcul élevé. Sans connaissance *a priori* de la configuration des caméras, la mise en correspondance d'un point d'une image doit être réalisée sur l'ensemble des points extraits de la deuxième image. De plus, le nombre de faux appariements augmente si les motifs des images sont répétitifs (notamment les fenêtres sur les façades des bâtiments cf. Figure 4.1). Afin d'améliorer et d'accélérer les résultats de la mise en correspondance, il faut réduire l'espace de recherche des correspondances.

La géométrie épipolaire définie en §2.6.2 permet de déterminer pour un point de l'image de gauche une ligne correspondante dans l'image de droite. C'est dans cette

ligne où se situe le point correspondant s'il existe. Dans la pratique, on utilise une bande de trois lignes (cf. Figure 4.2) pour éviter de manquer certains appariements à cause de l'approximation des points projetés.

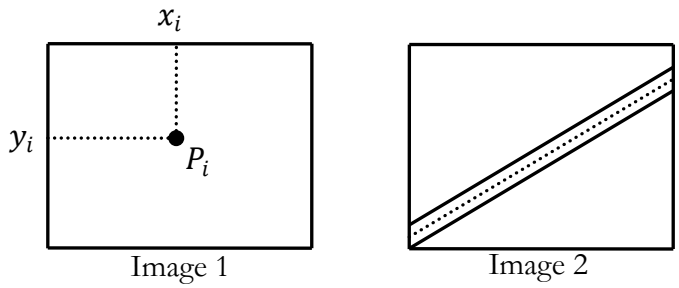


Figure 4.2 : Zones de recherche du point P_i dans l'image 2 ; cas où la matrice fondamentale est connue, on obtient une bande de recherche autour de la droite épipolaire.

Dans notre cas, la configuration du stéréoscope est particulière (cf. Figure 4.3), le système est calibré pour que les lignes épipolaires soient parallèles. Ainsi un point $P_1(x, y)$ d'une image se trouve sur la ligne y de la deuxième image (cf. Figure 4.5).

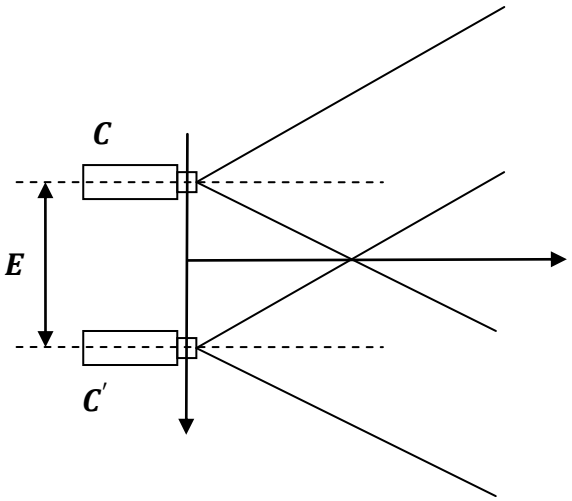


Figure 4.3 : Configuration alignée d'un stéréoscope.

4.2.1.2. La contrainte de position

La particularité de la géométrie du stéréoscope (cf. Figure 4.5 et Figure 4.1 avec un exemple d'images virtuelles rectifiées) permet d'introduire la contrainte de position.

Cette contrainte implique que l'abscisse d'un point de l'image gauche m_x soit toujours supérieure à l'abscisse du point correspondant de l'image droite m'_x. Cette contrainte permet de réduire l'espace de recherche lors de la mise en correspondance sur une ligne épipolaire (cf. Figure 4.4).

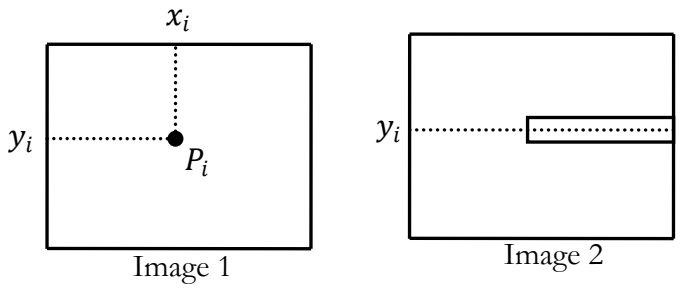


Figure 4.4 : Zones de recherche du point P_i dans l'image 2 avec un stéréoscope aligné ; la zone de recherche est une bande horizontale limitée par la contrainte de position.

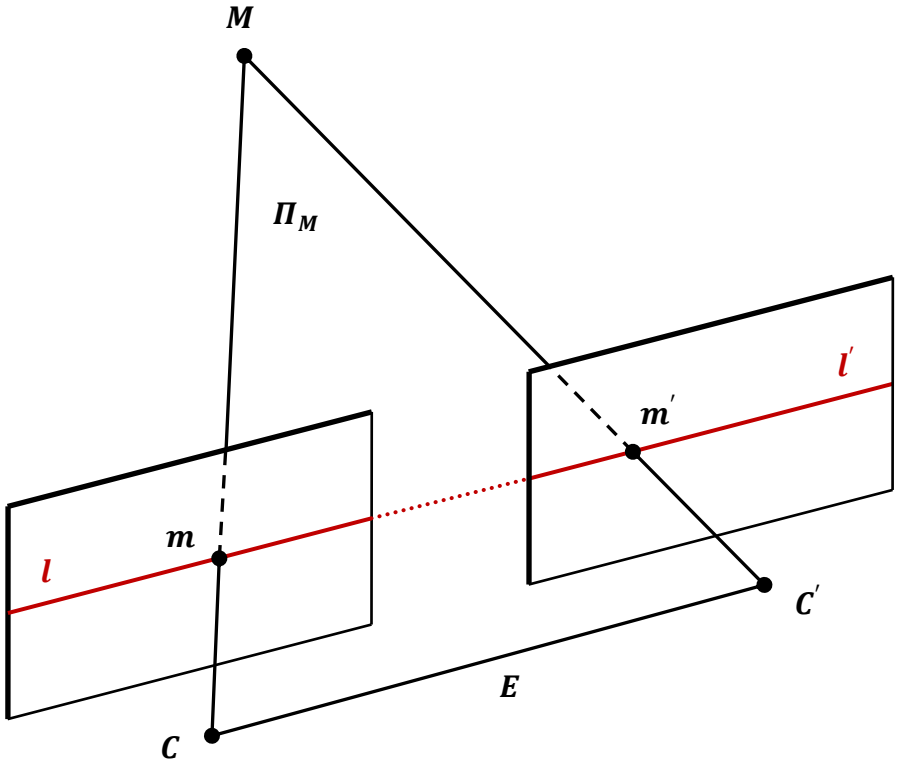


Figure 4.5 : Configuration stéréoscopique particulière. Les caméras sont alignées optiquement avec une distance inter-caméra E.

4.2.1.3. La contrainte d'orientation

La méthode d'extraction des points SIFT permet d'associer à chaque point clé une orientation $\theta(x, y)$ (voir §2.3.1.3) pour rendre ces points invariants aux rotations. Etant données deux points SIFT gauche et droit ayant respectivement les orientations θ et θ' dans la configuration d'un stéréoscope rectifiée (cf. Figure 4.5), la contrainte d'orientation impose aux couples candidats à l'appariement la condition suivante :

$$|\theta - \theta'| < \epsilon \tag{4.1}$$

avec ϵ le seuil maximum toléré entre les deux orientations. Dans la pratique, ce seuil est fixé à $\frac{\pi}{4}$.

4.2.1.4. La contrainte d'échelle

Chaque point SIFT est extrait dans une échelle définie lors de la détection des extremum (cf. 2.3.1.1). Notons qu'entre deux images stéréoscopiques, il n'y a quasiment pas de changement d'échelle (zoom). Ainsi, deux points SIFT appariés entre deux images stéréoscopiques ont quasiment la même échelle.

La contrainte d'échelle impose que deux points SIFT sont appariables si et seulement s'ils appartiennent à la même octave (cf. §2.3.1) et que leur niveau d'échelle respectif est séparé au plus d'un niveau.

4.2.1.5. La contrainte d'unicité

La contrainte d'unicité contraint chaque primitive de l'image de gauche à n'avoir qu'un seul correspondant dans l'image de droite, et réciproquement. A la suite des contraintes précédentes, si une primitive a plusieurs appariements possibles, on sélectionne le couple de points SIFT ayant le descripteur le plus proche (cf. 2.4.2).

4.2.2. Calcul du modèle

Après l'étape de mise en correspondance des points SIFT, on obtient une liste de points appariés. En utilisant la disparité horizontale et les paramètres de calibrage des caméras, on détermine la position 3D des points relativement au stéréoscope.

Soit *S* un stéréoscope avec deux caméras *C* et *C'*, alignées optiquement avec une distance inter-caméras *E*. En prenant un point quelconque $M = \{M_x, M_y, M_z\}$ de l'environnement, celui-ci se projette sur *C* et *C'* respectivement en $m = \{m_x, m_y\}$ et $m' = \{m'_x, m'_y\}$. Les projections m et m' sont définies dans le repère image de chacune des caméras. En connaissant les paramètres intrinsèques et extrinsèques des caméras, on peut établir les relations suivantes :

$$M_x = \frac{M_z . m_x}{f_c}, \qquad M_y = \frac{M_z . m_y}{f_c}, \quad M_z = \frac{f_c . E}{|m_x - m'_x|} \tag{4.2}$$

où f_c est la distance focale supposée, identique pour les deux caméras.

La Figure 4.6 illustre les étapes de la reconstruction des points SIFT, sous la forme de projection des points reconstruits sur les images stéréoscopiques.

4.2.3. Construction 3D

La reconstruction tridimensionnelle est une méthode incrémentale. Pour initialiser le processus de reconstruction, le repère d'origine du système doit être déterminé, celui-ci étant fixé par la première position du stéréoscope. Ainsi, les points reconstruits à partir de cette première position par la méthode décrite en §4.2.2 forment les points 3D initiaux pour la méthode incrémentale.

La méthode incrémentale permet de déterminer à partir de l'ensemble des positions reconstruites en $C_1, C_2, \ldots, C_N$, la position du stéréoscope en C_{N+1}. Le calcul de C_{N+1} nécessite différentes étapes :

- Estimation de la pose C_{N+1} par rapport à C_N ;
- Sélection des points du modèle potentiellement appariables avec les points SIFT de la l'image I_{N+1} ;
- Mise en correspondance robuste des points sélectionnés SIFT relatifs à la sélection précédente ;
- Calcul robuste de la pose de la caméra.

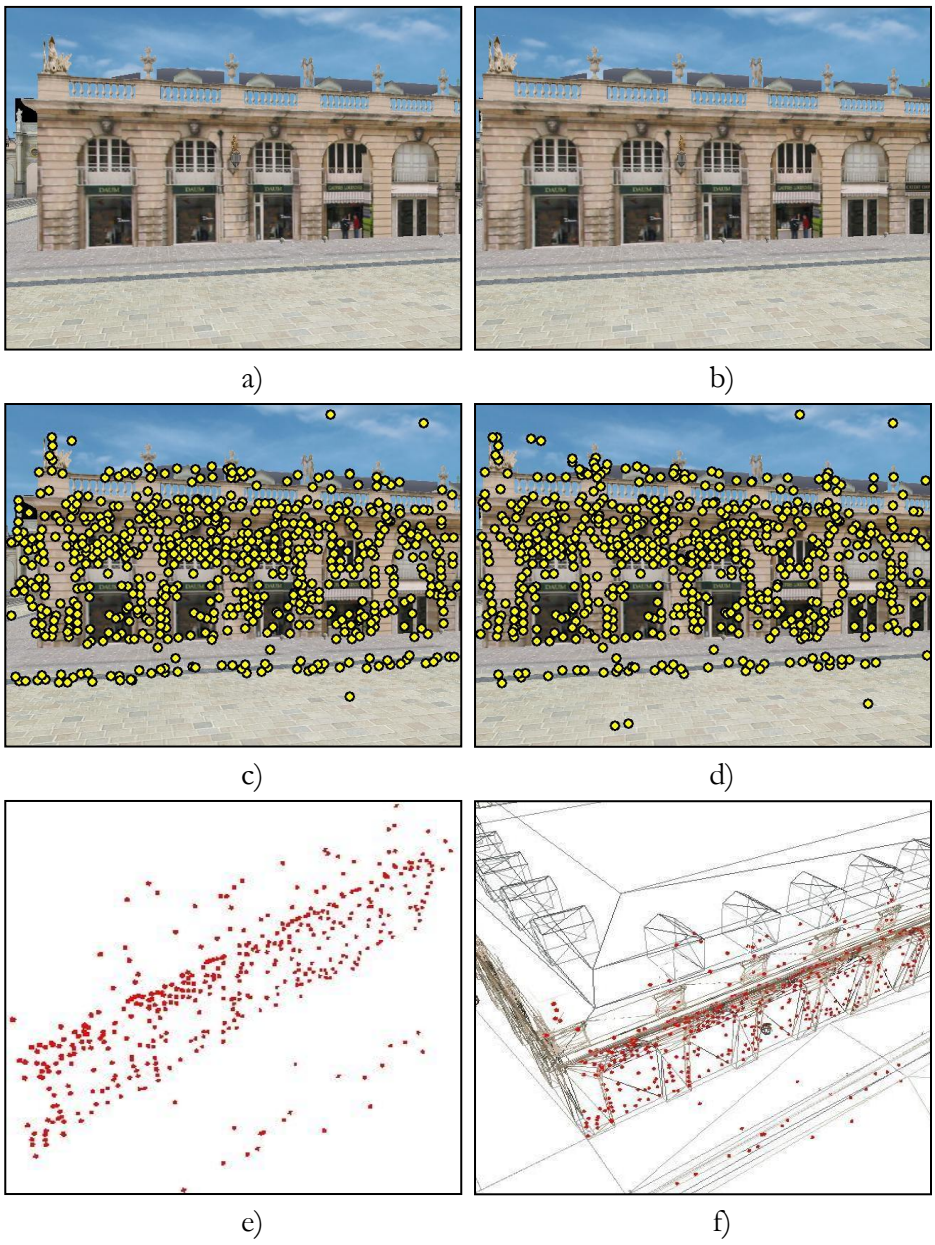


Figure 4.6 : Schéma de la reconstruction d'une paire stéréoscopique ; Les images a) et b) sont respectivement l'image gauche et droite du stéréoscope. Les images c) et d) représentent l'extraction des points SIFT respectivement de a) et b). L'image e) est la reconstruction stéréoscopique des points SIFT par rapport au stéréoscope. L'image f) montre les points SIFT placés sur le repère global.

4.2.3.1. Estimation de la pose du stéréoscope

Pour déterminer une position robuste, il est nécessaire d'initialiser le processus de calcul par la pose *a priori* du stéréoscope. L'objectif est de déterminer une estimation de C_{N+1} par rapport C_N.

A partir des points SIFT extraits de l'image gauche I_N à la position C_N et des points SIFT de l'image gauche I_{N+1} à la position C_{N+1}, on détermine les points homologues par la méthode d'appariement des points SIFT (§2.4.2). Les couples appariés sont utilisés pour calculer la matrice essentielle E (cf. §2.7.4). A partir de celle-ci les matrices de rotation R_r et de translation T_r relative entre les deux vues sont ensuite déterminées (cf. 2.7.4.2).

Avec les matrices R_r et T_r, une position approximative de C_{N+1} est obtenue. Celle-ci est utilisée pour déterminer l'ensemble des primitives appariables avec la base des points 3D. L'estimation de la précision de la pose obtenue peut être calculée en inversant la matrice $J^t J$. J est la matrice jacobienne de la fonction f des erreurs de rétroprojection ε [66] des points 3D associés aux points SIFT 2D de l'image I_N sur l'image I_{N+1}.

$$\varepsilon = f(U) \tag{4.3}$$

avec U un vecteur formé des paramètres extrinsèques de la caméra C_{N+1} et des coordonnées des points 3D associés aux points SIFT 2D de l'image I_N.

$$\varepsilon = \begin{pmatrix} \dots \\ q_i^j - P.S.Q_j \\ \dots \end{pmatrix} \tag{4.4}$$

où q_i^j est le point SIFT 2D de l'image I_{N+1}, qui correspond au point SIFT 2D associé à Q_j dans l'image I_N. S est la transformation relative entre les caméras C_N et C_{N+1} et P la matrice de projection associée à la caméra C_{N+1}.

4.2.3.2. Sélection des points SIFT appariables

A partir de l'estimation de la pose C_{N+1} et des paramètres intrinsèques caméra, il est possible de déterminer le champ de vue du stéréoscope. Le champ de vue d'une caméra (Field Of View, cf. Figure 4.7) est défini à partir de la focale f_c de la caméra :

$$FOV = 2\,arctan\frac{d}{2f} \tag{4.5}$$

avec d est la taille du capteur. Le FOV en x et y est généralement différent à cause de la résolution du capteur.

Le champ de vue est formé par quatre plans : Π_1, Π_2, Π_3 et Π_4. L'équation du plan i est donnée par :

$$A_i.x + B_i.y + C_i.z + D_i = 0 \tag{4.6}$$

Un point tridimensionnel M appartient au champ de vue si sa distance aux quatre plans est positive :

$$A_i.M_x + B_i.M_y + C_i.M_z + D_i > 0, \qquad \forall i \tag{4.7}$$

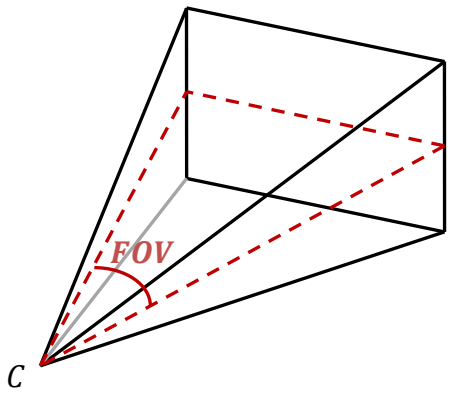


Figure 4.7 : Champ de vue (Field Of View) de la caméra défini par la focale et la dimension du capteur.

En appliquant à tous les points 3D du modèle la condition de l'équation (4.7) par rapport à la pose C_{N+1}, nous obtenons l'ensemble des points potentiellement appariables avec ceux de l'image I_{N+1}. A partir d'une pose estimée C_{N+1}, il est possible de déterminer une méthode d'appariement plus robuste (cf. 4.2.3.3).

4.2.3.3. Mise en correspondance robuste

A partir de la pose approximative C_{N+1} de la caméra et de l'ensemble des points 3D associées aux points SIFT potentiellement appariables, on peut déterminer la projection de ces points sur l'image I_{N+1}. Ces projections permettent de réduire l'espace de recherche des points à apparier à une zone rectangulaire autour des points projetés (cf. Figure 4.8).

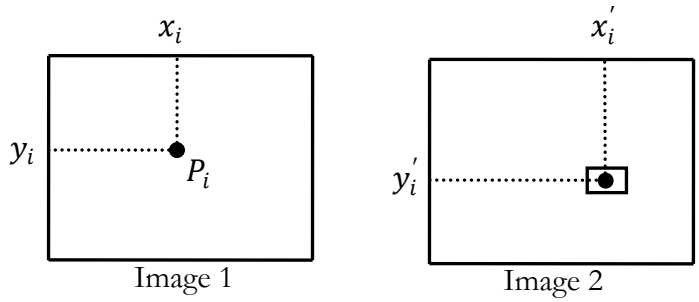


Figure 4.8 : Zones de recherche du point P_i dans l'image 2 ; cas où le point 3D reconstruit de P_i est projeté dans l'image 2 avec une incertitude connue de la pose : la zone de recherche est limitée à un rectangle dans l'image 2.

Nous pouvons alors calculer la pose robuste du stéréoscope. Cette méthode est employée également pour la localisation, elle sera décrite en §4.3.

Malgré la robustesse de cette méthode de mise en correspondance, une déviation des calculs de la pose se produit sur des distances élevées. Pour éviter cela, on utilise un ajustement de faisceaux local [51,52] qui raffine la position 3D des points calculés et celle des dernières poses du système stéréoscopique.

4.2.4. Ajustement de faisceaux local

L'algorithme d'ajustement en faisceaux [78] permet d'optimiser les paramètres extrinsèques de la caméra et les points SIFT 3D sélectionnés précédemment. L'idée de l'ajustement de faisceaux local est d'améliorer les paramètres des l dernières poses estimées du stéréoscope sur les L positions antérieures ($l < L$) (cf. Figure 4.9). L'erreur à minimiser est donnée par la fonction de coût f^i, définie par :

$$f^i(\mathcal{C}^i, \mathcal{P}^i) = \sum_{\mathcal{C}^k \in \{C_{i-M+1}, \ldots, C_i\}} \sum_{P_j \in \mathcal{P}^i} d^2\left(p_i^j, P_{proj}^i P^j\right) \tag{4.8}$$

avec : $d^2(p_i, K_i P^j)$ la distance euclidienne entre la projection du point 3D P^j via la caméra C_i et p_i^j est le point SIFT correspondant. La matrice P_{proj}^i est la matrice de projection (cf. §2.5.3) composée par les paramètres intrinsèques et extrinsèques. L'ensemble $\mathcal{C}^i$ est composé des l dernières positions de la caméra et l'ensemble $\mathcal{P}^i$ est composé des points 3D projetés sur la caméra C_i.

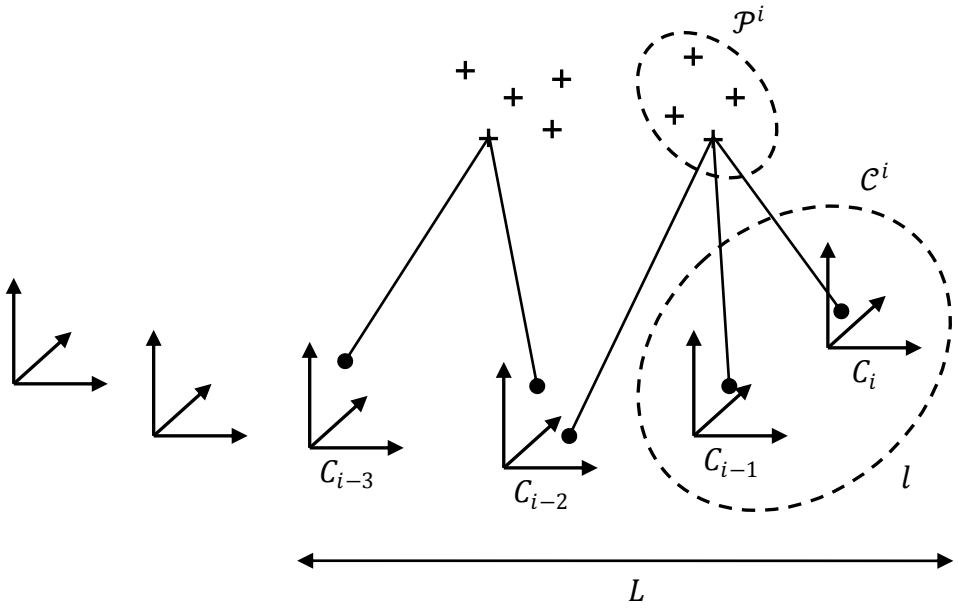


Figure 4.9 : Ajustement de faisceaux local.

La méthode d'ajustement de faisceaux local nécessite deux étapes. Premièrement, on utilise l'algorithme de Levenberg-Marquardt [61] pour minimiser la fonction de coût f^i. Deuxièmement, on ne conserve que les points dont l'erreur de rétroprojection est inférieure à un seuil défini de manière expérimentale (ce seuil est égal à 2 pixels [67]). Ces deux étapes sont répétées tant que la fonction de coût diminue.

Le choix des paramètres l et L influence la qualité de la reconstruction du modèle. Pour avoir un ajustement correct, la condition suivante $L \geq l + 2$ doit être vérifiée [51]. Lors du commencement d'une reconstruction, pour $i \leq L$, on applique

l'ajustement de faisceaux local en utilisant toutes les données. Pour que les résultats de l'ajustement de faisceaux local soient équivalents à la méthode globale [78], on doit choisir une valeur élevée pour L. Dans notre cas, $L = 10$ et $l = 3$ [15]. Récemment, une étude sur la propagation d'erreur pour l'ajustement de faisceaux local démontre que l'erreur associée est quasiment identique à la méthode globale [16].

4.2.5. Modèle reconstruit

Le résultat de reconstruction (cf. Figure 4.10) est un ensemble de points 3D. A chaque point 3D est associé à un descripteur SIFT du point gauche du couple ayant donné naissance à ce point.

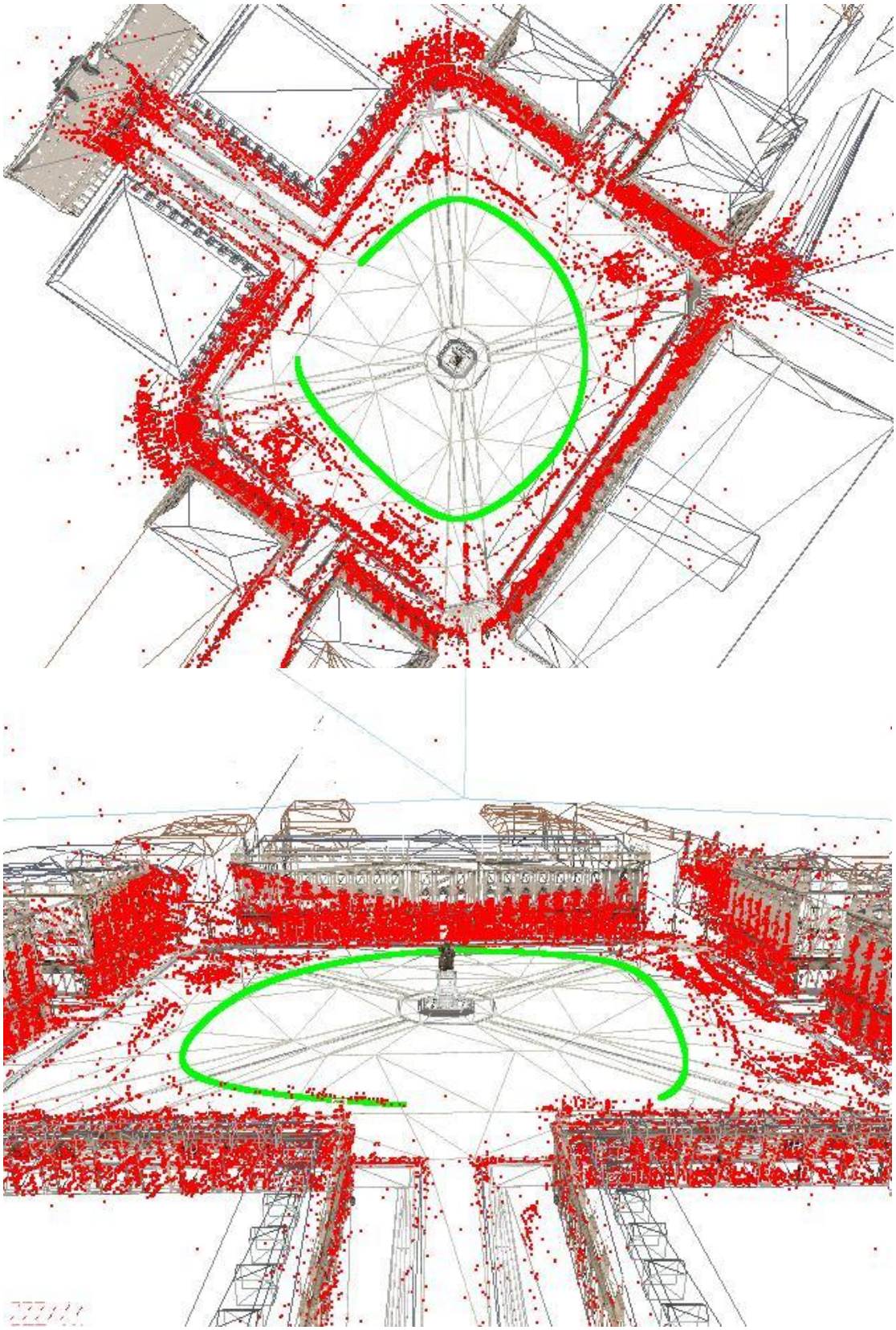


Figure 4.10 : Reconstruction des points SIFT

4.3. Localisation

A l'issue de l'étape de reconstruction, le système de localisation doit pouvoir déterminer la position des caméras, en cherchant des correspondances entre la base de données et les points caractéristiques des images. Un filtrage des points 3D doit être réalisé pour ne comparer que les points potentiellement dans le champ de vue de la caméra, en utilisant une estimation de la pose de la caméra (cf. Figure 4.11).

A l'initialisation, le système recherche les correspondances avec l'ensemble de la base de données. Ce processus, qui est long à effectuer, ne concerne que la première acquisition. Le filtre permet d'éliminer environ 90% des points, offrant ainsi une mise en correspondance rapide des points SIFT restants.

Le principe de la méthode de localisation reprend le processus de reconstruction (cf. 4.2.3) pour déterminer de manière robuste la nouvelle pose de la caméra C_{N+1}. La localisation se compose des étapes suivantes :

1. Estimation de la pose C_{N+1} par rapport à C_N (cf. 4.2.3.1) ;
2. Sélection des points SIFT du modèle potentiellement appariables avec les points SIFT correspondant à la pose C_{N+1} (cf. 4.2.3.2) ;
3. Mise en correspondance robuste des points sélectionnés avec les points SIFT correspondant à la pose C_{N+1} (cf. 4.2.3.3) ;
4. Calcul robuste de la pose de la caméra ;

Ces étapes sont communes pour la reconstruction et la localisation. Les étapes 1, 2 et 3 étant décrites précédemment (cf. 4.2.3), nous nous focalisons maintenant sur la méthode de calcul robuste de la pose de la caméra. La Table 4.1 détail l'algorithme de la localisation robuste étape par étape.

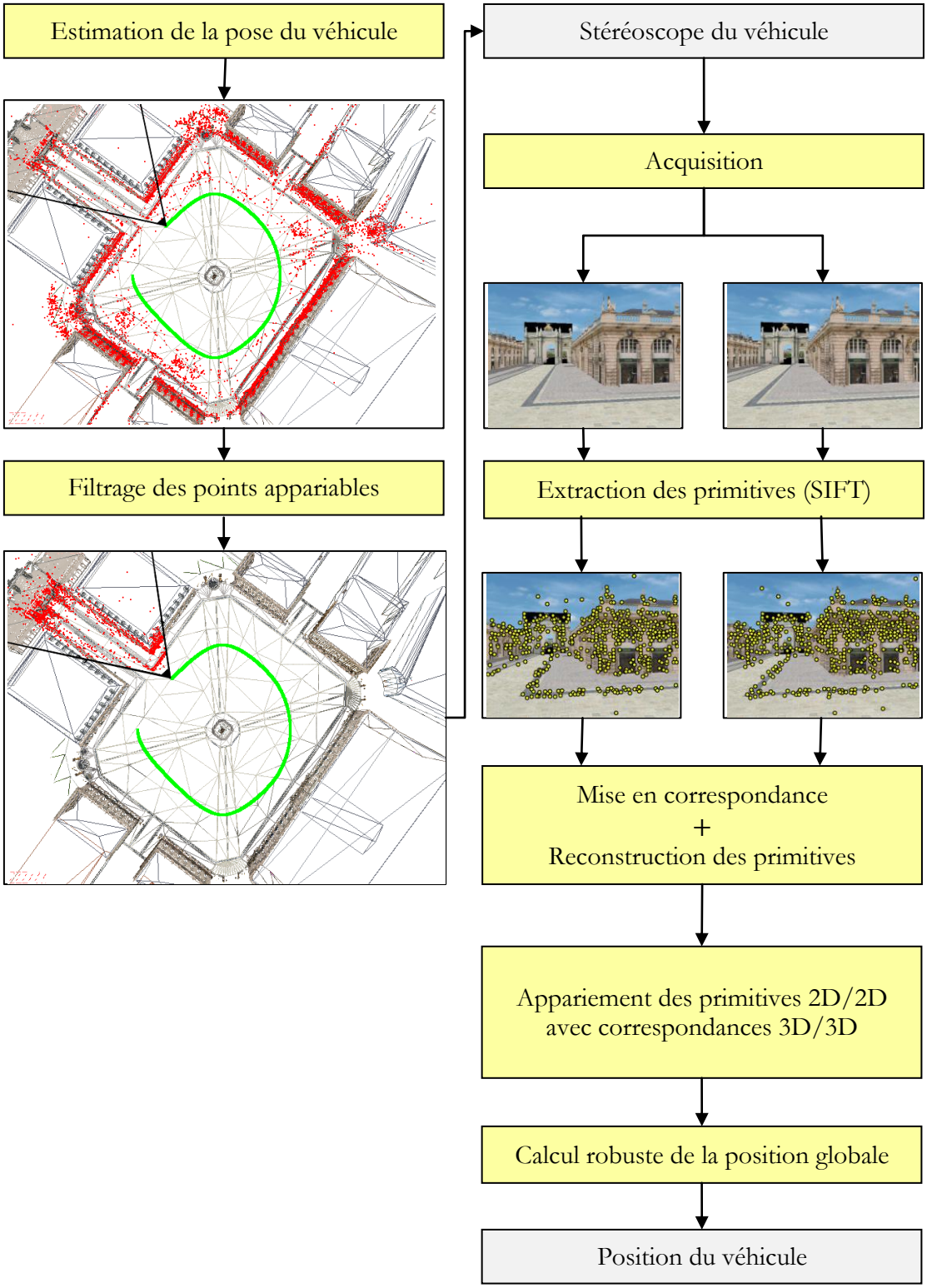


Figure 4.11 : Méthode de localisation robuste

4.3.1. Calcul robuste de la pose de la caméra

Soit $\{Q_i\}$ (avec $1 \leq i \leq n$) l'ensemble des points 3D reconstruits à partir du stéréoscope à la position C_{N+1}. Les points Q_i sont connus dans le repère caméra. Pour connaître la position globale, l'appariement des points SIFT associe à chaque point Q_i un point P_i de la base de données 3D. Notons par d_i la distance entre le stéréoscope et le point Q_i. La relation permettant d'obtenir la position dans le repère global s'écrit sous la forme d'un système d'équations :

$$\left(C_x - Q_x^i\right)^2 + \left(C_y - Q_y^i\right)^2 + \left(C_z - Q_z^i\right)^2 = d_i^2 \tag{4.9}$$

avec i variant de 0 à n.

La résolution de ce système est basée sur l'utilisation de la méthode de Newton-Gauss, qui minimise la fonction :

$$S(\beta) = \sum_{i=0}^{n} r_i^2 \tag{4.10}$$

avec $r_i = d_i - f_i(\beta), \quad \beta = \left(C_x, C_y, C_z\right)$ et :

$$f_i(\beta) = \left(C_x - Q_x^i\right)^2 + \left(C_y - Q_y^i\right)^2 + \left(C_z - Q_z^i\right)^2 \tag{4.11}$$

La méthode de Newton-Gauss est adaptée à la résolution de ce système dans le cas où β est proche de la solution, ce qui correspond à notre cas. En effet, une estimation approximative de la pose de la caméra est réalisée pour la mise en correspondance robuste (cf. 4.2.3.1). Le processus de minimisation de l'équation (4.10) est exprimé par :

$$\beta^{k+1} = \beta^k + (J^t J)^{-1} J^t r \tag{4.12}$$

avec J la matrice jacobienne de $f(\beta)$:

$$J = 2\begin{pmatrix} C_x - Q_x^0 & C_y - Q_y^0 & C_z - Q_z^0 \\ \vdots & \vdots & \vdots \\ C_x - Q_x^n & C_y - Q_y^n & C_z - Q_z^n \end{pmatrix} \tag{4.13}$$

Afin d'obtenir un résultat robuste on utilise la technique de RANSAC [18] (cf. §2.7.3.3.2). Cette technique consiste à répéter N fois la procédure suivante :

- Tirer aléatoirement trois points Q^0, Q^1 et Q^2 parmi les points 3D mis en correspondance ;
- A partir de ces points, nous calculons la position globale de la caméra C_{N+1} en utilisant la méthode détaillée dans la section suivante (cf. 4.3.2) ;
- L'erreur de rétroprojection de tous les points Q^i est ensuite évaluée, en utilisant la pose de la caméra C_{N+1}. Si l'erreur de rétroprojection d'un point Q^i est supérieur à 2 pixels, le point est considéré comme un outlier, autrement dit il s'agit d'un faux appariement ;
- Si le nombre d'appariements retenus est insuffisant (inférieur à 20 dans notre cas), la solution n'est pas conservée ;
- La pose de la caméra C_{N+1} est ensuite calculée, en minimisant la fonction $f_i(\beta)$ sur les bons appariements retenus dans l'étape précédente ;
- Nous évaluons enfin l'erreur moyenne de rétroprojection des points Q_i à partir de la pose de la caméra C_{N+1}.

La solution retenue pour la pose de la caméra C_{N+1} est celle qui minimise l'erreur moyenne de rétroprojection.

Pour obtenir un résultat précis, nous fixons la probabilité d'obtenir au moins un tirage ayant que de bons appariements à $p = 0.99$ (cf. équation (2.49)), et une proportion de faux appariements à $\varepsilon = 40\%$. La proportion de faux appariements varie en fonction de la composition des images. En prenant un taux à 40%, nous assurons une bonne précision dans la plus part des cas. Sachant qu'il faut au minimum 3 points pour évaluer la fonction $f_i(\beta)$, le nombre de tirages nécessaire est donc $N = 19$ (cf. équation (2.49)).

4.3.2. Estimation de la position de la caméra à partir de trois points 3D

A partir de trois points Q^0, Q^1 et Q^2, nous pouvons déterminer la position de la caméra C_{N+1} grâce à l'équation (4.9) :

$$\begin{cases} (C_x - Q_x^0)^2 + \left(C_y - Q_y^0\right)^2 + (C_z - Q_z^0)^2 = d_0^2 \\ (C_x - Q_x^1)^2 + \left(C_y - Q_y^1\right)^2 + (C_z - Q_z^1)^2 = d_1^2 \\ (C_x - Q_x^2)^2 + \left(C_y - Q_y^2\right)^2 + (C_z - Q_z^2)^2 = d_2^2 \end{cases} \quad (4.14)$$

Cette équation peut être réécrite par :

$$t^2.H_1 + t.H_2 + H_3 = 0 \quad (4.15)$$

où :

$$\begin{cases} H_1 = N_4^2 + N_2^2 + 1 \\ H_2 = 2(N_4 M_1 + N_2 M_2 - Q_z^0)_2 \\ H_3 = M_1^2 + M_2^2 + (Q_z^0)^2 - d_0^2 \end{cases} \quad (4.16)$$

et :

$$\begin{cases} M_1 = N_3 - Q_x^0 \\ M_2 = N_1 - Q_y^0 \end{cases} \quad (4.17)$$

avec :

$$\begin{cases} N_1 = \dfrac{A_1 D_2}{B_1 A_2} - \dfrac{D_1}{B_1} \\ N_2 = \dfrac{A_1 C_2}{B_1 A_2} - \dfrac{C_1}{B_1} \\ N_3 = -\dfrac{D_1 + B_1 P_1}{A_1} \\ N_4 = -\dfrac{B_1 P_2 + C_1}{A_1} \end{cases} \quad (4.18)$$

et :

$$A_1 = 2(Q_x^1 - Q_x^0) \text{ et } A_2 = 2(Q_x^2 - Q_x^0)$$
$$B_1 = 2\left(Q_y^1 - Q_y^0\right) \text{ et } B_2 = 2\left(Q_y^2 - Q_y^0\right)$$
$$C_1 = 2(Q_z^1 - Q_z^0) \text{ et } C_2 = 2(Q_z^2 - Q_z^0)$$

et :

$$D_1 = (Q_x^0)^2 - (Q_x^1)^2 + \left(Q_y^0\right)^2 - \left(Q_y^1\right)^2 + (Q_z^0)^2 - (Q_z^1)^2 - (d_0)^2 + (d_1)^2$$
$$D_2 = (Q_x^0)^2 - (Q_x^2)^2 + \left(Q_y^0\right)^2 - \left(Q_y^2\right)^2 + (Q_z^0)^2 - (Q_z^2)^2 - (d_0)^2 + (d_2)^2$$

Cette équation admet deux solutions :

$$t_1 = \frac{-H_2 - \sqrt{H_2^2 - 4H_1H_3}}{2H_1} \quad \text{et} \quad t_2 = \frac{-H_2 + \sqrt{H_2^2 - 4H_1H_3}}{2H_1} \tag{4.19}$$

Parmi ces solutions, une seule est valide pour la position de la caméra C_{N+1}. Elle correspond à la solution donnant la position la plus de proche de la caméra C_N.

Algorithme Localisation robuste

Objectif : A partir de la position de la caméra C_N et de l'ensemble des points reconstruits, déterminer la pose de la caméra en C_{N+1}

Algorithme :

1. Estimation de la pose de la caméra en calculant la matrice fondamentale entre C_N et C_{N+1} en utilisant la méthode robuste de RANSAC (cf. 2.7.3.3.2) ;
2. Détermination de la rotation R_r et la translation T_r relative entre C_N et C_{N+1} (cf. 2.7.4.2) ;
3. Sélection des points appariables à la position C_{N+1} en utilisant le FOV de la caméra (cf. 4.2.3.2) ;
4. Calcul de la pose robuste en utilisant RANSAC : Répéter N fois :
   - 4.1. Choisir aléatoirement trois points Q^0, Q^1 et Q^2 ;
   - 4.2. Estimer la pose de la caméra en utilisant les trois points (cf. 4.3.2) ;
   - 4.3. Vérification que le nombre d'inliers est suffisant (cf. 4.3.2) ;
   - 4.4. Calculer la pose robuste C_{N+1} en utilisant l'algorithme de Gauss-Newon sur les inliers (cf. 4.3.2) ;
   - 4.5. Estimer l'erreur moyenne de rétroprojection obtenue ;
5. La solution retenue étant celle qui minimise l'erreur moyenne de rétroprojection ;

Table 4.1 : Méthode de la localisation robuste

4.4. Conclusion

Dans ce chapitre, nous avons présenté une méthode de localisation basée sur la stéréovision. Cette méthode est composée de deux étapes principales. La première consiste à construire un modèle 3D qui peut être considéré comme un apprentissage de l'environnement de navigation. Ce modèle est constitué de points 3D issus de l'appariement de points SIFT extraits des images stéréoscopiques. Les points 3D du modèle sont localisés de manière globale. La deuxième étape réalise le processus de localisation en utilisant le modèle construit lors de la phase d'apprentissage. Le principe de ce processus consiste dans un premier à apparier les points 3D issus d'une reconstruction stéréoscopique avec les points 3D du modèle à travers les points SIFT. Il s'agit ensuite de retrouver la pose de la caméra en appliquant un processus d'optimisation.

La méthode proposée est caractérisée par l'utilisation des points SIFT comme primitives de mise en correspondance. Ce type de points a l'avantage d'être moins sensible aux changements affines. Les contraintes utilisées pour réduire l'espace de recherche des appariements permettent d'obtenir de bons résultats de mise en correspondance. Cette dernière est basée sur un descripteur de points SIFT permettant une meilleure discrimination entre les points.

La précision de localisation dépend de celle du modèle reconstruit lors de la phase d'apprentissage. Pour pouvoir augmenter les chances d'obtenir une localisation précise, notre approche fait appel à des méthodes robustes telles que la méthode de RANSAC ou encore la méthode de l'ajustement de faisceaux local.

Pour évaluer les performances de la proposée, le chapitre suivant présente les résultats expérimentaux obtenus sur la base de tests effectués en utilisant le simulateur décrit dans le chapitre 3.

CHAPITRE 5

RESULTATS EXPERIMENTAUX

5.1. Plate-forme expérimentale

Les expérimentations réelles ont été réalisées au moyen d'un véhicule électrique : le « Gem Car »[11]. Ce véhicule léger (cf. Figure 5.1) proposé par « Matra M. S. » est homologué comme Quadricycle lourd (L7e) et permet de parcourir environs 50 km à une vitesse moyenne de 45 km/h. Ce véhicule est entièrement automatisé par le laboratoire SeT[12] de l'Université de Technologie de Belfort-Montbéliard au moyen d'un système « AutoBox ».

En partant des constations suivantes :

- 80% de nos déplacements se font dans un rayon de 5Km autour du point de départ ;
- 40% de nos déplacements ne dépassent pas 2 km ;
- La vitesse moyenne de déplacement des voitures en ville est de 18 km/h ;

Le « Gem Car », dont les performances sont présentées sur Table 5.1, est un moyen de transport de proximité adapté aux centres urbains. Plusieurs déclinaisons de ce véhicule sont également proposées :

- « GEM e2 » : Véhicule 2 places ;
- « GEM e4 » : Véhicule 4 places ;
- « GEM eS » : Véhicule fourgon de 2 places avec des capacités de chargement et de traction satisfaisantes. C'est le modèle utilisé pour nos expérimentations (cf. Figure 5.1) ;
- « GEM eLXD » : Véhicule 2 places avec grande capacité de chargements et grande charge utile. Cet utilitaire électrique peut accueillir un plateau basculant électrique, une benne, des ridelles ou encore se transformer en fourgon ;

Vitesse réduite	24 km/h
Vitesse maximale	40 km/h
Autonomie à 40 km/h +/-25%	50 km
Rampe maxi en charge	20%

Table 5.1 : Performances du Gem Car

[11] http://www.gemcar.com/

[12] http://set.utbm.fr/

Figure 5.1 : Plate-forme expérimentale électrique automatisée : « Gem Car » équipée par le laboratoire SeT de l'UTBM.

5.1.1. Caractéristiques techniques du Gem Car

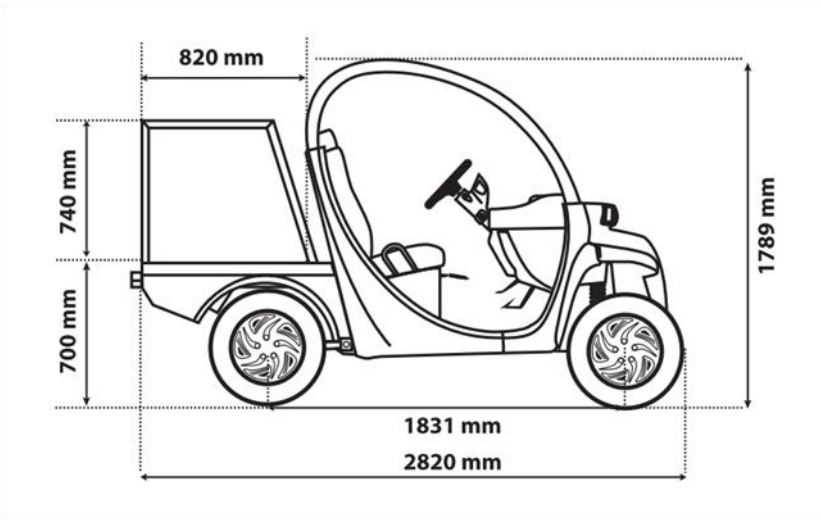


Figure 5.2 : Dimensions du Gem Car

Ce véhicule est compact (cf. Figure 5.2) et dispose d'un espace de rangement à l'arrière suffisant pour embarquer tout le matériel nécessaire pour réaliser des expérimentations. Les caractéristiques propres aux véhicules sont détaillées dans la Table 5.2. Le matériel embarqué (capteurs, PC, …) est détaillé en annexe A.6.

Moteur électrique	General Electric - 72V
Emplacement	A l'avant
Puissance maxi	3,72 kW - 9 kW en crête
Couple moteur	76 Nm
Alimentation	Courant continu
Batterie de traction	Pb Gel - 6 éléments de 12V - 65 Ah @ C1
Energie embarquée	4680 Wh @ C1
Chargeur embarqué	Alim. 220V - 16A
Transmission	Traction, différentiel
Châssis	Aluminium
Carrosserie	Sandwich ABS recyclé / PMMA et SB
Direction	À crémaillère
Freinage	Double circuit à rattrapage automatique d'usure de garnitures. Disques avant 235 mm, tambours arrière 180 mm. Frein de stationnement à commande mécanique.
Suspensions	Suspension avant à double triangle, ressorts hélicoïdaux. Suspension arrière multi-bras, ressorts hélicoïdaux.
Pneumatiques	165/70R12 Profil M+S

Table 5.2 : Caractéristiques propres du Gem Car

5.2. Les extracteurs de points

Une évaluation des opérateurs de Harris et de SIFT est effectuée pour percevoir les avantages et les limites de ces deux opérateurs. Nous nous limitons à ces deux opérateurs qui sont les plus utilisés dans les approches de SLAM. Une étude globale sur la comparaison des extracteurs de points est détaillée dans [80,47,45].

5.2.1. Critère de stabilité géométrique

Le critère de répétabilité [73] permet d'évaluer la stabilité géométrique ou robustesse d'un détecteur. Cette mesure caractérise la capacité d'un détecteur à détecter les mêmes points d'intérêt entre des images prises dans des conditions différentes (point du vue, éclairage, floue, …). On définit ce critère R par :

$$R = \frac{C}{min(n_1, n_2)} \tag{5.1}$$

avec :

- C le nombre de points mis en correspondance entre deux images ;
- n_i le nombre points extraits dans la zone en commun entre les images i et $i+1$;
- $min(n_i, n_{i+1})$ est le minimum de n_i et n_{i+1}. Ceci représente le nombre de correspondances maximales théoriques entre les images i et $i+1$ en respectant la contrainte d'unicité (cf. §4.2.1.5) ;

5.2.2. Comparaison des extracteurs dans le modèle virtuel

Les essais réalisés sur les données du modèle virtuel comparent la robustesse de l'extracteur de Harris (cf. §2.2) et l'extracteur des points SIFT (cf. §2.3). Nous ne considérons aucune réduction ou contraintes sur l'espace de recherche pour apparier les points. Ainsi, un point est apparié uniquement en maximisant sa fonction de mise en correspondance.

5.2.2.1. Evalution aux changements de points de vue

Pour évaluer la robustesse aux changements de points de vue, une série de 15 images est générée à partir d'une même scène, avec différents angles de vue et à différentes distances (cf. Figure 5.3). La mise en correspondance des différents points de vue est toujours réalisée entre la vue de face la plus proche de la scène (image 3 de la Figure 5.3) et les autres vues.

La Table 5.3 présente les résultats de l'extraction et de mise en correspondance des points par les deux extracteurs de HARIS et de SIFT. Le nombre de points

extraits par les deux méthodes est élevé et suffisant pour réaliser les tâches de reconstruction et de localisation. Les résultats de la mise en correspondance entre les différents points de vue sont très différents. L'extracteur des points SIFT permet d'avoir toujours au moins une centaine de points, ce qui est suffisant pour calculer la localisation d'une caméra. Contrairement à celui-ci, l'opérateur de Harris n'obtient pas assez d'appariements entre deux points de vue éloignés, ce qui constitue un handicape pour les étapes de reconstruction et/ou de localisation. Les résultats liés à l'image 3 correspondent à la mise en correspondance de l'image avec elle-même. Ces résultats permettent de valider le fonctionnement de la mise en correspondance sur une image composée de textures et d'objets répétitifs de la scène. La quasi-totalité des points sont correctement appariés avec eux-mêmes.

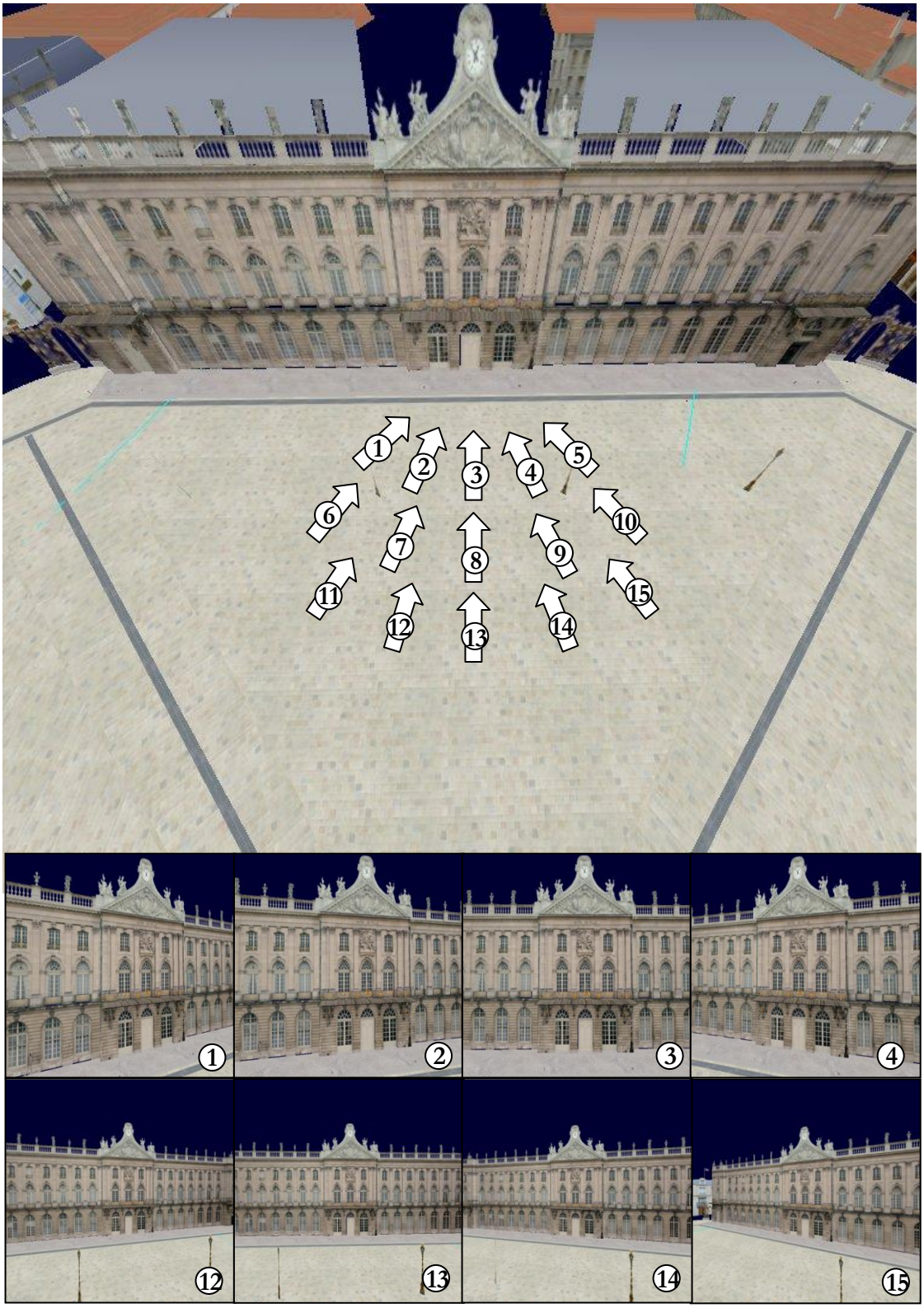


Figure 5.3: Base d'images pour la comparaison de l'extraction de primitives sur différents points de vue.

	Nombre de primitives de Harris extraites	Nombre de primitives SIFT extraites	Nombre de primitives de Harris appariés correctement	Nombre de primitives SIFT appariés correctement
Image 1	2019	1744	41	240
Image 2	1706	1801	114	612
Image 3	1671	1798	1668	1785
Image 4	2111	1699	68	387
Image 5	1856	1644	23	194
Image 6	1853	1495	65	295
Image 7	1937	1355	96	474
Image 8	1695	1274	112	583
Image 9	1970	1353	40	299
Image 10	1670	1416	22	173
Image 11	1340	1078	8	141
Image 12	1630	1060	9	205
Image 13	1505	965	13	193
Image 14	1803	1061	13	175
Image 15	1365	1066	7	134

Table 5.3 : Résultats de l'extraction et la mise en correspondance des points de Harris et de SIFT sur les différents points de vue.

Les trois figures suivantes : Figure 5.4, Figure 5.5 et Figure 5.6 présentent les résultats suivants :

- « Harris » : Critère de stabilité du détecteur de Harris ;
- « SIFT » : Critère de stabilité du détecteur SIFT.

Le critère de stabilité de l'opérateur SIFT met en évidence la grande différence entre la capacité de mettre en correspondance des points entre deux points de vue éloignés des deux opérateurs. L'opérateur SIFT obtient toujours un score supérieur à 10%. L'opérateur de Harris obtient des mauvais résultats dès que l'on s'éloigne du point de vue central. Des détails de la mise en correspondances sont disponibles en annexe A.1.

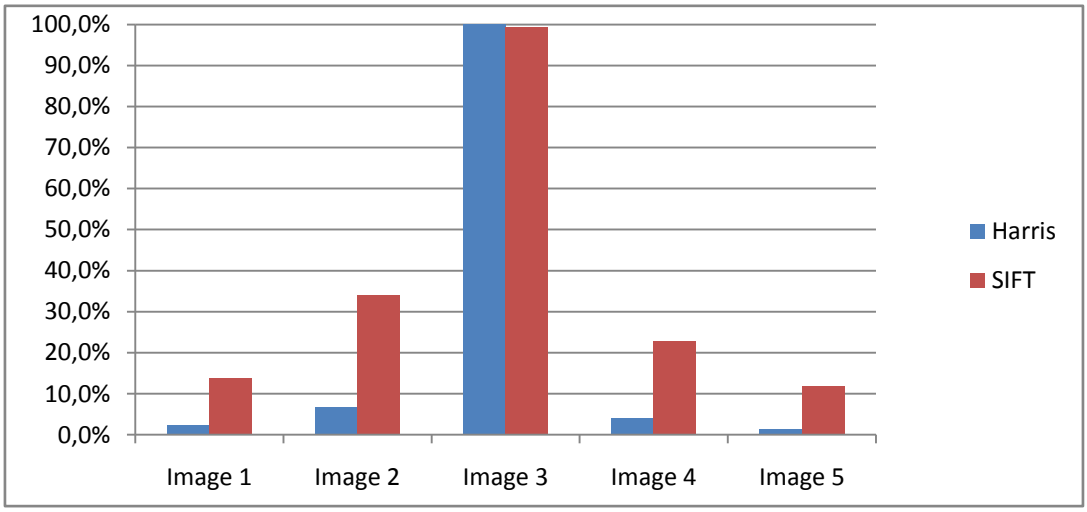


Figure 5.4 : Critère de stabilité pour les images 1 à 5 des opérateurs SIFT et Harris

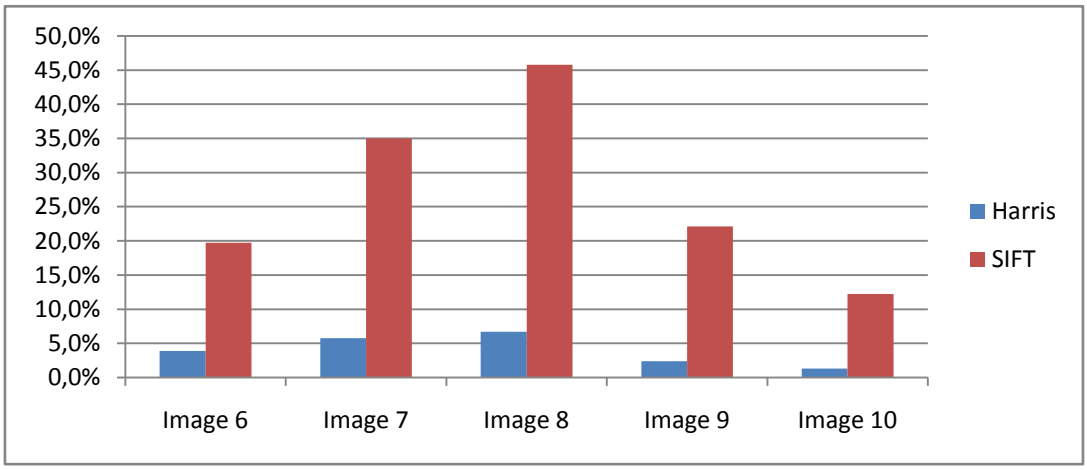


Figure 5.5 : Critère de stabilité pour les images 6 à 10 des opérateurs SIFT et Harris

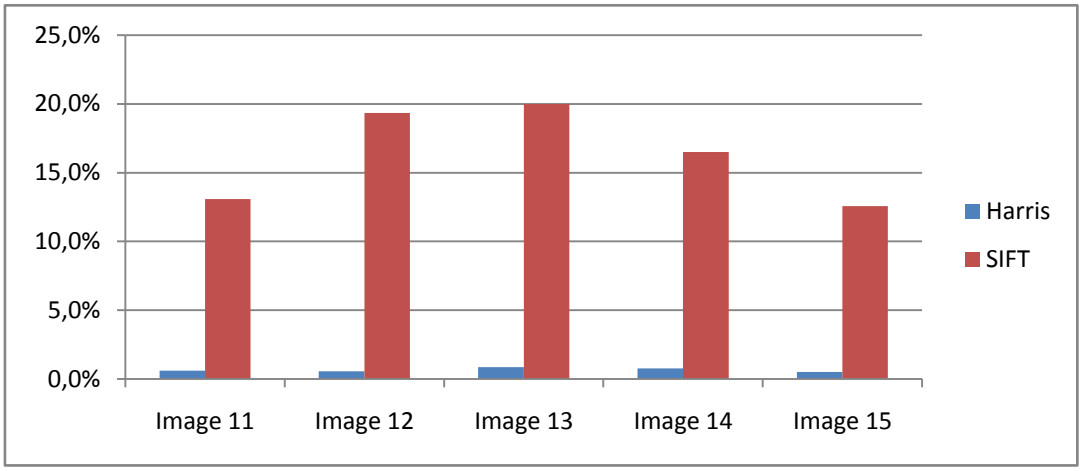


Figure 5.6 : Critère de stabilité pour les images 11 à 15 des opérateurs SIFT et Harris

5.2.2.2. Changements d'illumination avec images virtuelles

La robustesse aux changements d'illumination est évaluée par six images ayant différents niveaux d'éclairage homogènes (cf. Figure 5.7). Ces variations sont obtenues par une addition d'une composante à toute l'image.

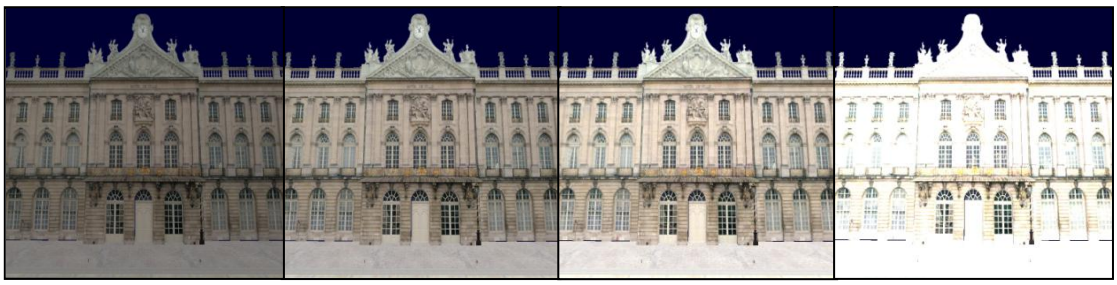


Figure 5.7 : Images virtuelles avec différents niveaux d'éclairage

La Figure 5.8 présente le critère de répétabilité des points SIFT et de Harris avec les changements d'illumination. L'image 3 correspond à la mise en correspondance de l'image avec elle-même. Ces résultats montrent que l'extracteur de SIFT est robuste aux changements d'éclairage. Malgré un nombre important de points Harris extraits (cf. Table 5.4), le nombre de points d'appariement reste faible. L'opérateur SIFT à un critère de stabilité de plus de 80%. Les détails sur l'extraction des points de SIFT et de Harris sont présentés en annexe A.2.

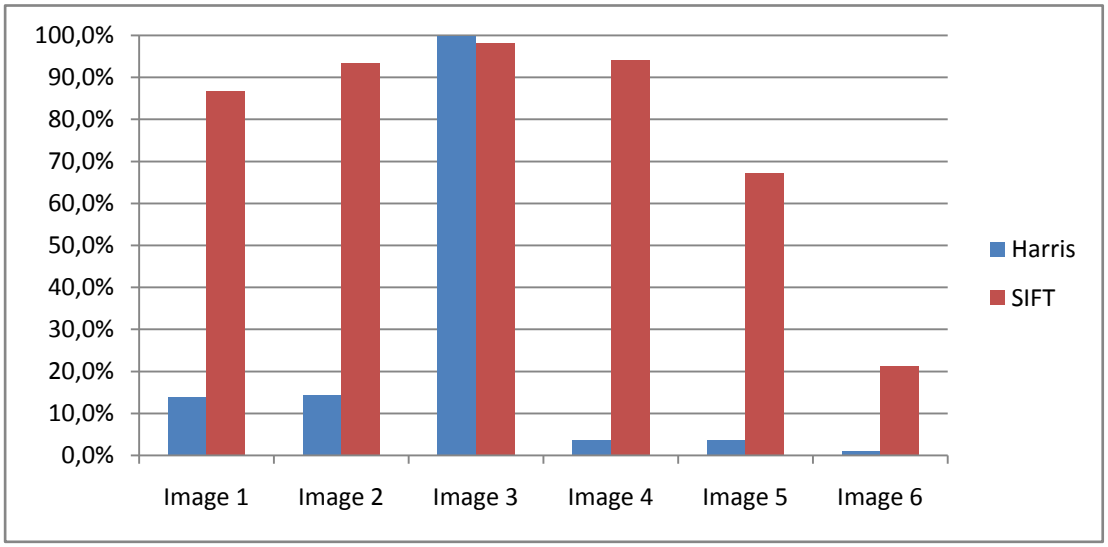


Figure 5.8 : Critère de stabilité pour les images 1 à 6 des opérateurs SIFT et Harris

	Nombre de points Harris	Nombre de points SIFT	Points Harris appariés correctement	Points SIFT appariés correctement
Image 1	2022	181	247	157
Image 2	1828	1214	253	1135
Image 3	1770	1746	1767	1713
Image 4	1736	1946	62	1641
Image 5	1662	1982	61	1172
Image 6	1607	1281	14	272

Table 5.4 : Résultats de l'extraction et la mise en correspondance des points de Harris et de SIFT sur différentes illumination.

5.2.2.3. Occultation de l'image

Quatre images sont générées pour mesurer les effets de l'occultation d'une partie de l'image sur les opérateurs SIFT et Harris. La Figure 5.9 présente trois des images utilisées.

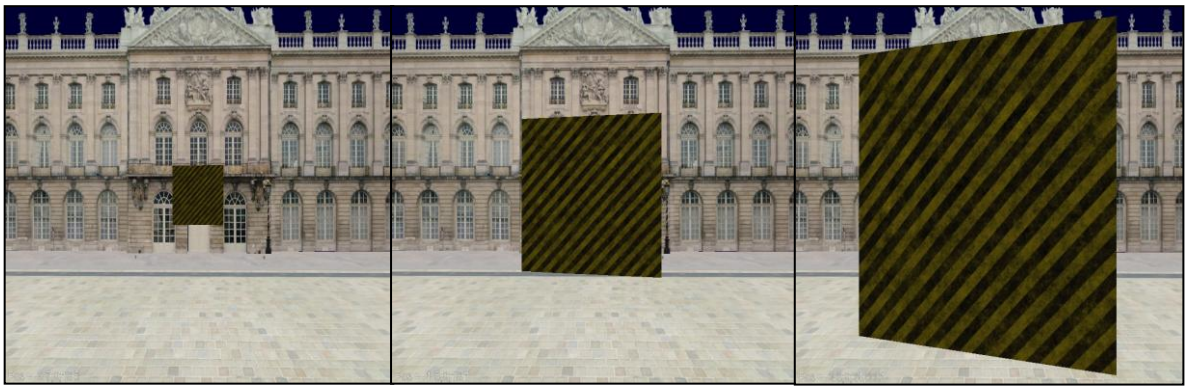


Figure 5.9 : Occultation de l'image.

La Figure 5.10 et la Table 5.5 présente les résultats issus de l'extraction et de la mise en correspondance des points SIFT et Harris. L'occultation d'une partie de l'image n'affecte que très peu les opérateurs. L'extracteur de Harris est même légèrement plus performant dans ces conditions.

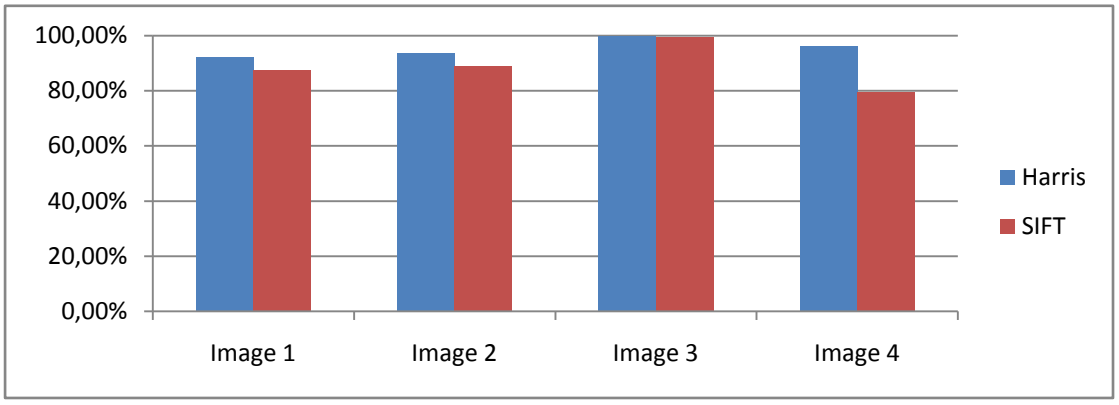


Figure 5.10 : Critère de stabilité pour les images 1 à 4 des opérateurs SIFT et Harris

	Nombre de points Harris	Nombre de points SIFT	Points Harris appariés correctement	Points SIFT appariés correctement
Image 1	1848	1077	1703	943
Image 2	1822	1060	1702	944
Image 3	1713	1030	1707	1024
Image 4	1669	1271	1603	858

Table 5.5 : Résultats de l'extraction et la mise en correspondance des points de Harris et de SIFT sur des images avec une partie de l'image occultée.

5.2.3. Comparaison des extracteurs dans le modèle réel

Pour réaliser ces tests, nous avons utilisé la base de données images[13] utilisée par les équipes de l'INRIA Grenoble Rhône-Alpes[14], CMP[15] (Center for Machine Perception), l'université de LEUVEN[16] et le Robotics Research Group[17] de l'université d'Oxford. La base de données contient différentes catégories d'images pour évaluer les différents changements possibles à l'image qui sont :

[13] http://www.robots.ox.ac.uk/~vgg/research/affine/index.html

[14] http://www.inrialpes.fr/

[15] http://cmp.felk.cvut.cz/

[16] http://www.esat.kuleuven.be/

[17] http://www.robots.ox.ac.uk/

- Flou
- Changement de point de vue
- Zoom et rotation
- Luminosité
- Compression de l'image

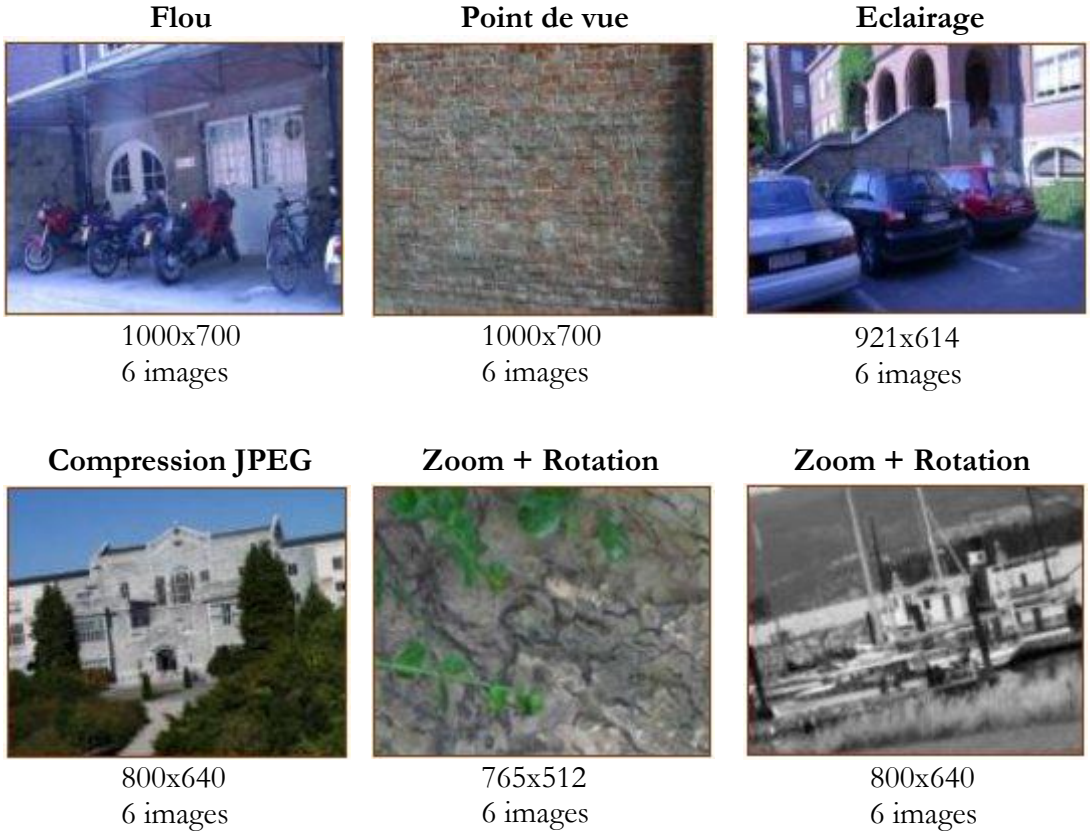


Figure 5.11 : Extrait d'images de la base de données de tests

Parmi ces catégories, la catégorie « Zoom + Rotation » n'est pas pertinente pour le système de localisation. Le mobile se déplaçant uniquement sur une surface plane les rotations autour de l'axe X (le roulis) ne constitue pas une transformation valide pour un véhicule. De plus, les caméras sont à focale fixe, et ne permettent donc pas un zoom sur l'image. La catégorie « Compression JPEG » n'est pas non plus exploitée. L'enregistrement des données se fait sans perte (au format « bmp »). Aucune compression de l'image n'est utilisée.

Les catégories restantes : « Flou », « Eclairage » et « Point de vue » représentent les changements qui peuvent se présenter dans le cadre de notre application. Dans le

changement de point de vue, la vue de face du mur est changée progressivement d'environ 10° à chaque image, jusqu'aux environs de 60°. Le changement de netteté de l'image dans la catégorie « Flou » est produit par la variation du focus de la caméra. Les changements d'éclairage sont réalisés par la fermeture du diaphragme de l'appareil.

L'objectif est d'évaluer indépendamment les effets de ces variations pour la mise en correspondance des points extraits par les détecteurs de Harris et de SIFT à partir d'images réelles.

5.2.3.1. Images réelles floues

Les images floues d'une caméra apparaissent lors de mouvements brusques de celle-ci. La Figure 5.12 illustre la différence entre une image nette et la même image floutée.

Figure 5.12 : Image d'une même scène réelle : à gauche, l'image est nette ; à droite, l'image est floutée par un changement de focus de l'appareil de prise de vue.

Les résultats de la Table 5.6 et la Figure 5.13, illustre les résultats de la mise en correspondance entre les différents niveaux de floue de l'image. De l'image 1 à 5 le niveau de floue est augmenté progressivement. Il apparaît clairement que la mise en correspondance par la méthode de Harris n'est pas applicable. L'opérateur SIFT admet des résultats corrects. Nous pouvons constater que plus le degré de floue augmente plus la mise en correspondance devient délicate. Des images détaillées sur les extractions des primitives SIFT et Harris sont présentées en annexe A.3.

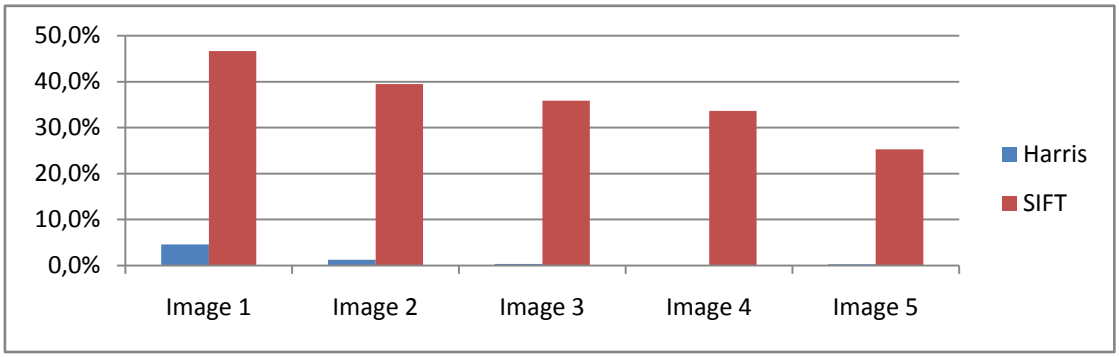


Figure 5.13 : Critère de stabilité pour les images 1 à 5 des opérateurs SIFT et Harris sur les différents niveaux de floue.

	Nombre de primitives de Harris extraites	Nombre de primitives SIFT extraites	Nombre de primitives de Harris appariés correctement	Nombre de primitives SIFT appariés correctement
Image 1	567	977	26	456
Image 2	623	1025	8	405
Image 3	898	734	3	263
Image 4	1141	523	0	176
Image 5	993	364	3	92

Table 5.6 : Résultats de l'extraction et la mise en correspondance des points de Harris et de SIFT sur les différents niveaux de floue.

5.2.3.2. Changements d'illumination avec images réelles

Dans le cas de la mise en correspondance d'images extérieures, il est difficile d'obtenir exactement les mêmes conditions d'éclairage de la scène. La Figure 5.14, illustre une même scène avec deux conditions d'illuminations différentes.

Figure 5.14 : Images réelles de la même scène avec un changement d'illumination

Les résultats de la mise en correspondance des points de Harris et de SIFT sont détaillés sur la Figure 5.15. L'extracteur de SIFT permet d'apparier un nombre correct de points. La méthode de Harris offre des résultats insuffisants. Avec moins de 50 appariements il est difficile d'avoir de bons résultats lors des étapes de reconstruction et/ou de localisation. Des détails sur l'extraction des primitives sont présentés en annexe A.4.

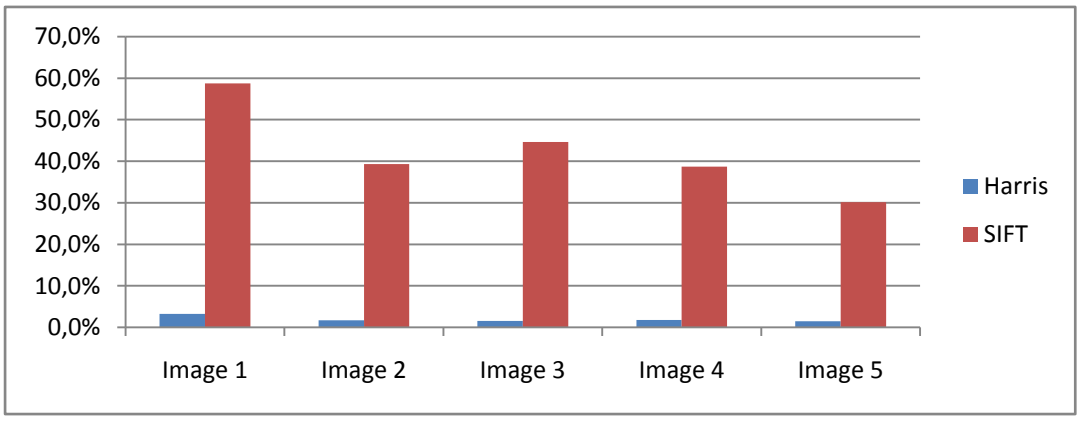


Figure 5.15 : Critère de stabilité pour les images 1 à 5 des opérateurs SIFT et Harris aux changements d'illumination sur des images réelles.

	Nombre de primitives de Harris extraites	Nombre de primitives SIFT extraites	Nombre de primitives de Harris appariés correctement	Nombre de primitives SIFT appariés correctement
Image 1	1294	562	42	330
Image 2	1158	792	20	268
Image 3	1021	437	16	195
Image 4	864	372	15	144
Image 5	614	338	9	102

Table 5.7 : Résultats de l'extraction et la mise en correspondance des points de Harris et de SIFT avec différents éclairages.

5.2.3.3. Changements de points de vue sur des images réelles

La Figure 5.16 illustre une même scène présentée sous différents points de vue. Les résultats issus de la mise en correspondances sont présentés sur la Table 5.8 et la Figure 5.16. L'opérateur de Harris ne permet pas une mise en correspondance de

points suffisante. On peut voir également la limite de l'extracteur du SIFT sur les images 4 et 5 de la séquence. Des détails sur l'extraction des points SIFT et Harris sont présentés en annexe A.5.

Figure 5.16 : Images réelles avec changements de point de vue d'une même scène.

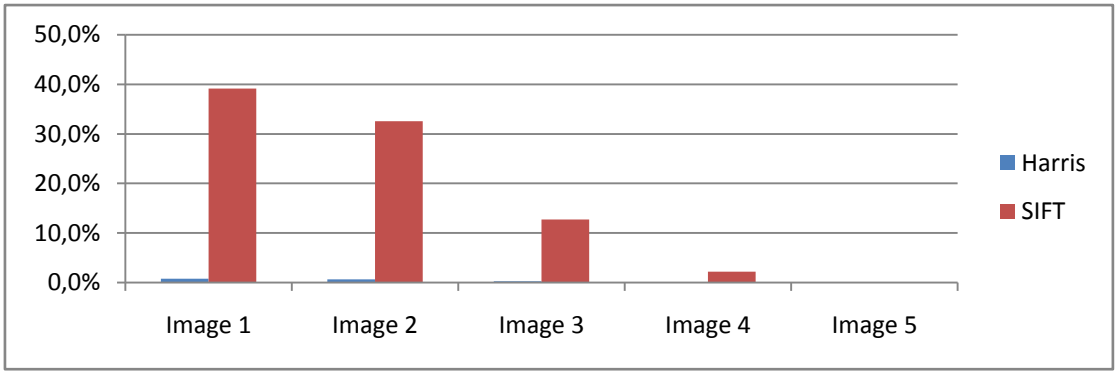


Figure 5.17 : Critère de stabilité pour les images 1 à 5 des opérateurs SIFT et Harris aux changements de points de vue sur des images réelles.

	Nombre de primitives de Harris extraites	Nombre de primitives SIFT extraites	Nombre de primitives de Harris appariés correctement	Nombre de primitives SIFT appariés correctement
Image 1	4096	1826	29	679
Image 2	4096	1764	24	564
Image 3	4096	1822	10	221
Image 4	4096	1868	0	38
Image 5	4096	1853	1	0

Table 5.8 : Résultats de l'extraction et la mise en correspondance des points de Harris et de SIFT d'images réelles avec différents points de vue. Le nombre maximal de points de Harris extraits est limité à 4096 points.

5.2.4. Conclusion des extracteurs

Les différents essais réalisés sur l'extraction et la mise en correspondances de primitives montrent que l'opérateur SIFT est robuste aux changements de points de vue, d'illumination ou encore aux occlusions de l'image. Contrairement à celui-ci, l'opérateur de Harris n'obtient pas de bons résultats.

Sans utiliser de réduction de l'espace de recherche ou de contraintes particulières, il est préférable d'utiliser la méthode de SIFT. La robustesse de cet opérateur à un coût lié au temps d'extraction des données. Les applications temps réel, avec peu de variations dans l'environnement de recherche, sont plus adaptées à l'opérateur de Harris. En effet, sa simplicité de calcul permet de traiter rapidement beaucoup de données.

5.3. Evaluation de la reconstruction

Une reconstruction 3D très précise de l'environnement n'est forcément nécessaire. L'objectif est de pouvoir se localiser précisément et la reconstruction doit fournir des données précises uniquement dans ce but. Par contre, la reconstruction doit nécessairement être précise localement.

L'évaluation de la performance de la reconstruction se mesure par la distance entre les positions réelles lors des acquisitions et les positions robustes estimées correspondantes (cf. §4.3.1). La Figure 5.18 présente la reconstruction 3D du modèle. La Figure 5.19 représente les poses déterminées par reconstruction du parcours du stéréoscope. La différence maximale est de 5cm de déviation entre la pose de la caméra estimée et la caméra réelle.

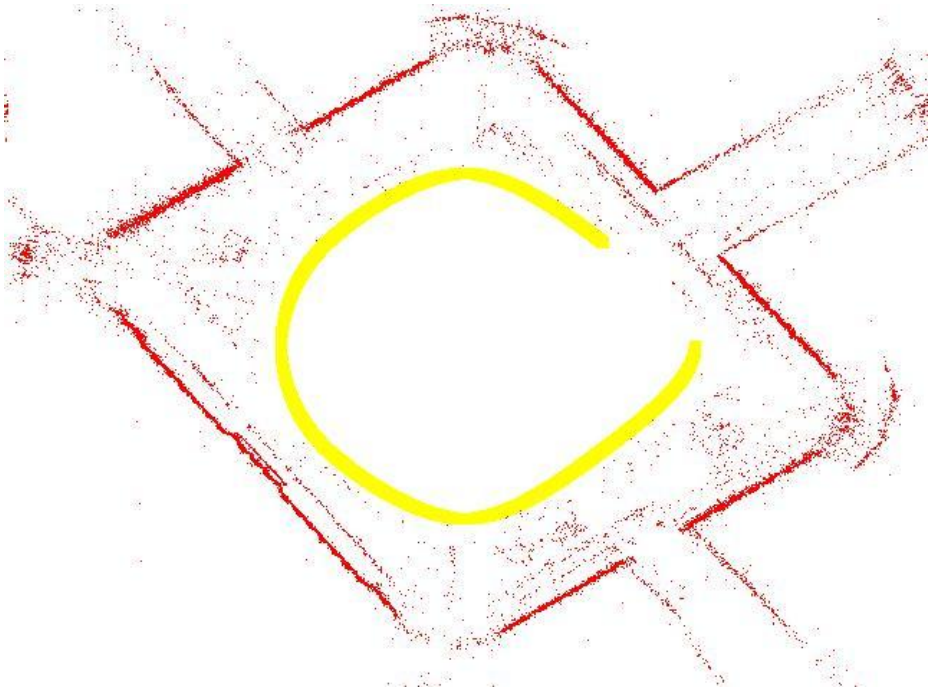


Figure 5.18 : Construction du modèle de l'environnement réalisé à partir de 80 acquisitions stéréoscopiques. Le modèle est constitué de 10488 points sur une distance d'environ 142m.

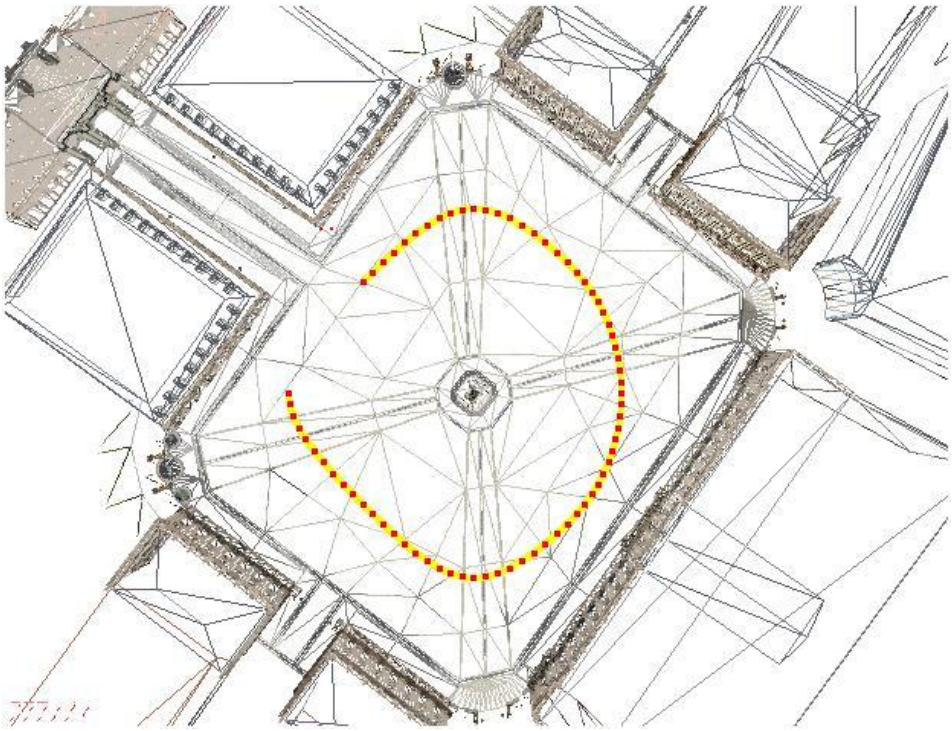


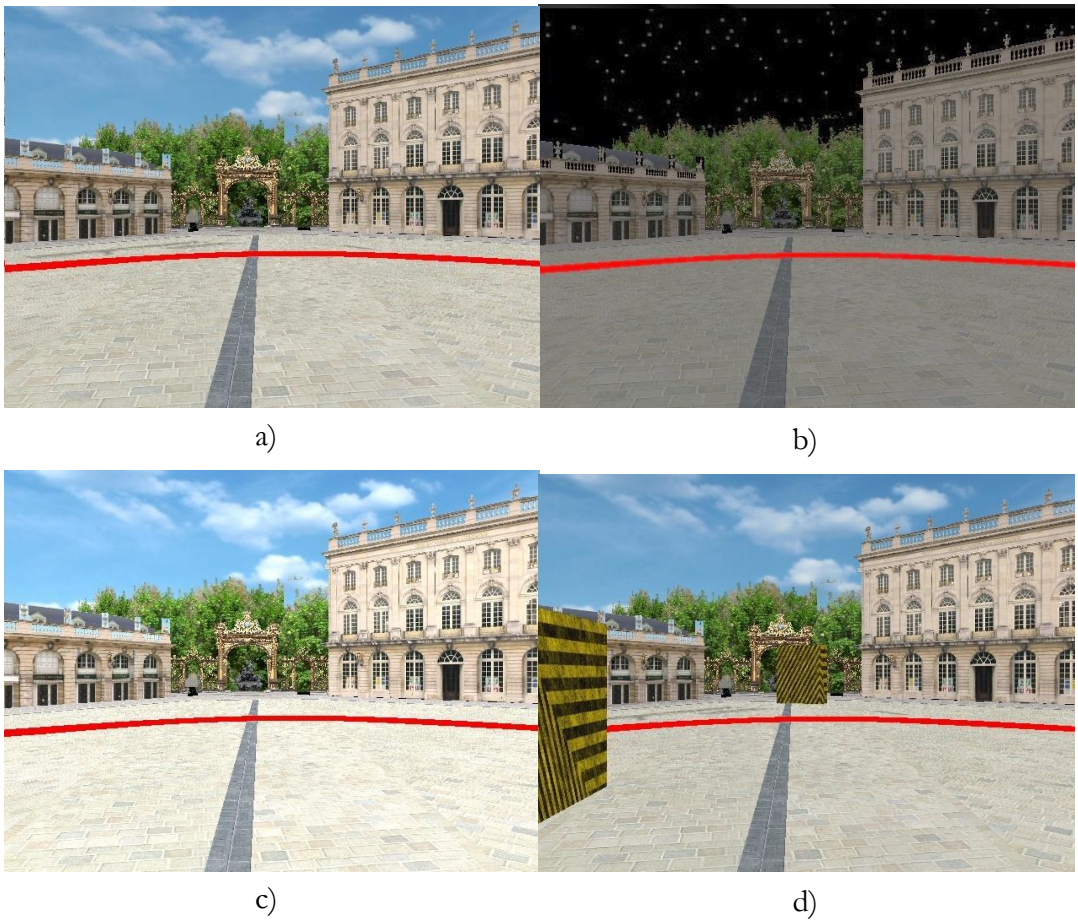
Figure 5.19 : Calcul de la position des points de la caméra reconstruit en rouge. En jaune la trajectoire effectuée.

5.4. Localisation

La méthode de localisation proposée est testée et évaluée en considérant différentes conditions de l'environnement. Les tests ont été réalisés en utilisant la plate-forme de simulation. L'évaluation compare la trajectoire effectuée avec la trajectoire calculée en estimant la déviation entre les deux.

5.4.1. Localisation dans le modèle virtuel

A partir de la trajectoire virtuelle apprise en §5.3, des tests de localisation sur différentes conditions sont réalisées (cf. Figure 5.20) :



a) b)

c) d)

Figure 5.20 : Conditions de tests pour la localisation en virtuelle. a) modèle virtuel normale ; b) sous exposée ; c) sur exposée ; d) occlusion du point de vue

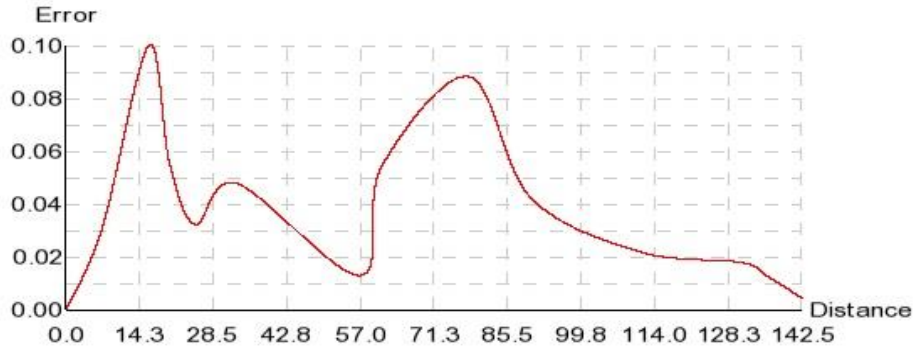


Figure 5.21 : Erreur (en mètres) dans la condition d'éclairage normale

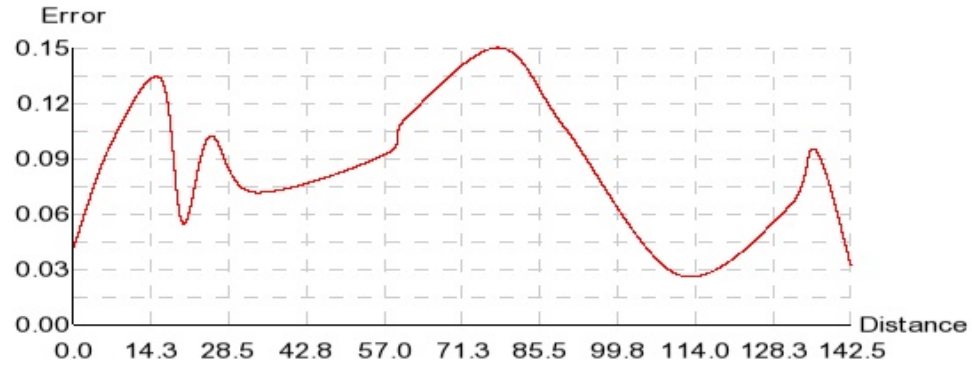


Figure 5.22 : Erreur (en mètres) dans la condition d'éclairage sous-exposée

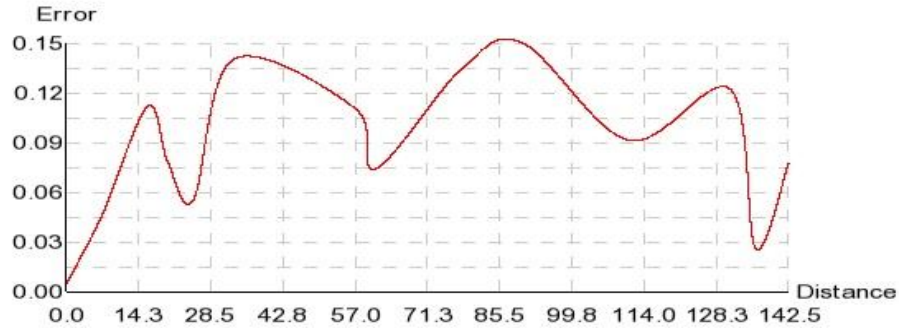


Figure 5.23 : Erreur (en mètres) dans la condition d'éclairage sur-exposée

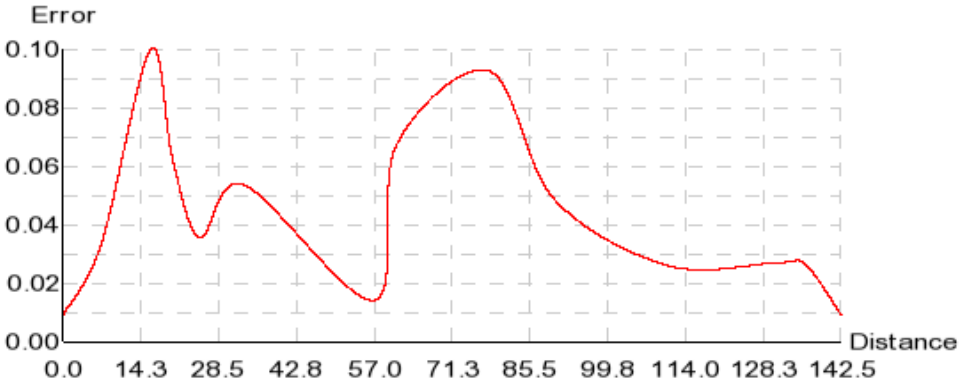


Figure 5.24 : Erreur (en mètres) dans le cas de la présence d'objets proche des objets de la trajectoire.

Les résultats obtenus lors de la localisation virtuelle sous les différentes conditions d'éclairage sont inférieurs à une déviation maximale d'environ 15cm (cf. Figure 5.21, Figure 5.22, Figure 5.23 et Figure 5.24) par rapport à la trajectoire apprise. Les performances de l'extracteur des points SIFT (cf. §5.2) permettent d'apparier les points même lorsque les conditions sont différentes.

5.4.2. Localisation dans le modèle virtuel

Les trajectoires cyan et magenta de la Figure 5.25 sont côte à côte à la trajectoire principale jaune. Les résultats (cf. Figure 5.26) issus de ce test évaluent la performance de la localisation en dehors du trajet appris. Les résultats montrent une déviation identique entre les trois trajectoires.

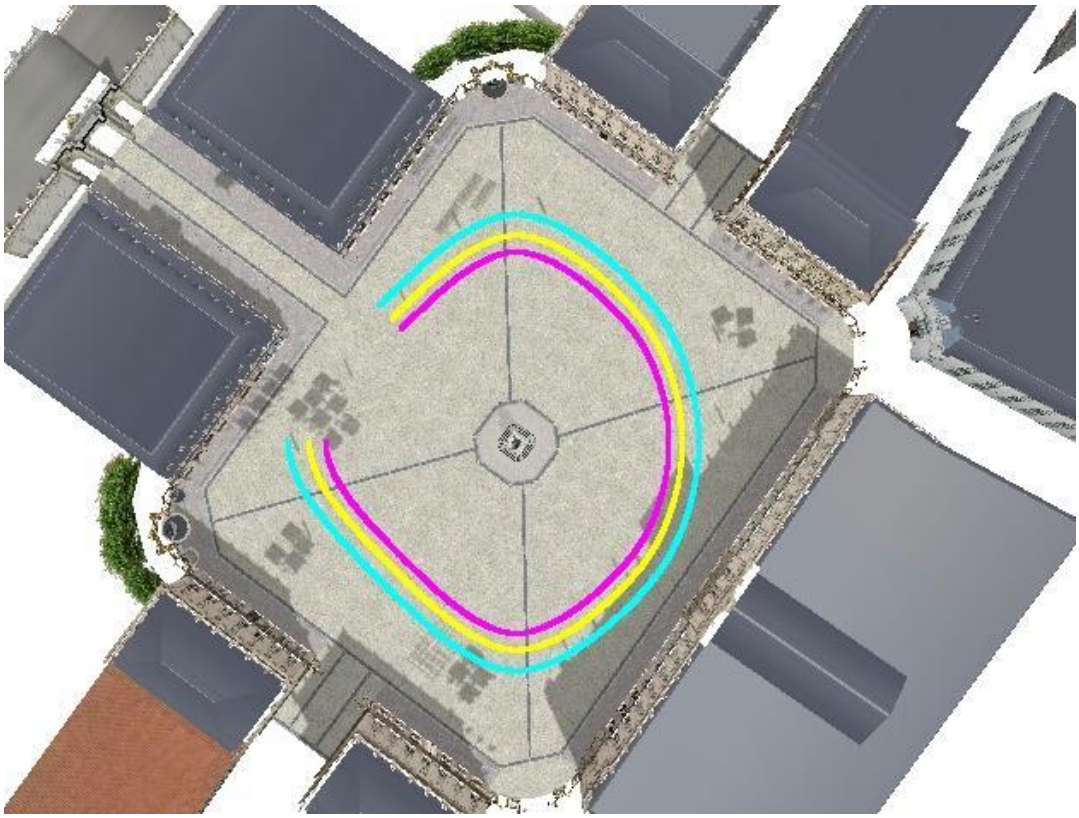


Figure 5.25 : Trajectoires parallèles autour de la trajectoire principale.

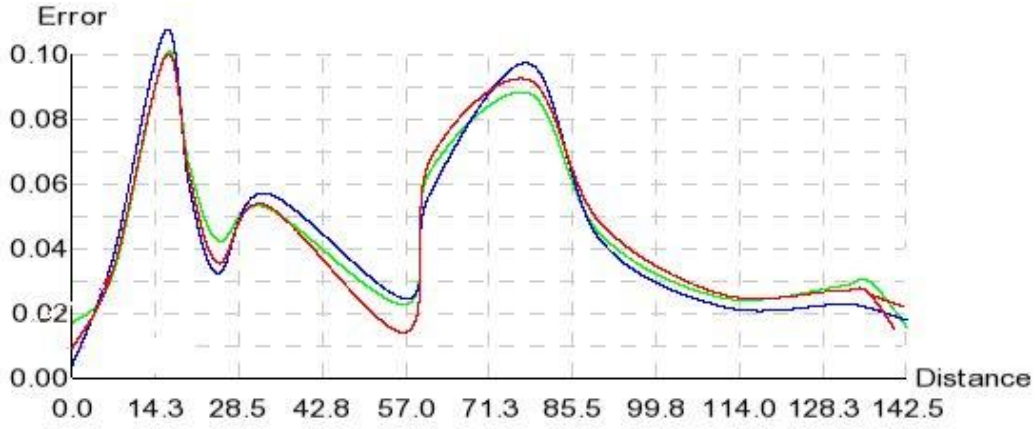


Figure 5.26 : Erreur (en mètres) des trajectoires parallèles. En rouge, déviation de la trajectoire intérieure (en magenta sur la figure des trajectoires). En vert, déviation de la trajectoire apprise (en jaune sur la figure des trajectoires) ; En bleu, la déviation de la trajectoire extérieure (en cyan sur la figure des trajectoires).

5.4.3. Localisation dans le modèle réel

5.4.3.1. Transformation WGS84 vers Lambert II étendue

Les données issues du capteur GPS sont présentées sous forme de trames de norme NMEA 0183. Parmi les données, deux informations sont essentielles pour la localisation de la longitude et la latitude du mobile. Ces informations sont exprimées dans le système géodésique associé au GPS : WGS84 (World Geodetic System 1984). Afin de pouvoir les utiliser sur un plan, il faut pouvoir convertir ceux-ci par une projection vers le système couvrant la zone à localiser. Selon la région géographique en France, on applique des projections différentes. Le modèle Lambert II s'applique sur une large zone couvrant le Territoire de Belfort. Différentes étapes sont alors nécessaires pour convertir les données en WGS84 vers le système Lambert II étendue:

- Transformation des coordonnées géographiques WGS84 (ϕ_w, λ_w) en coordonnées cartésiennes (x_w, y_w, z_w) ;
- Transformation des coordonnées cartésiennes (x_w, y_w, z_w) en coordonnées cartésiennes NTF (x_n, y_n, z_n) ;
- Transformation des coordonnées cartésiennes (x_n, y_n, z_n) en coordonnées géographiques NTF (ϕ_n, λ_n) ;
- Transformation des coordonnées géographiques NTF (ϕ_n, λ_n) en coordonnées projetées Lambert II étendue (x_{l2e}, y_{l2e}) ;

Plus de détails sur ces différentes étapes sont données dans la Table 5.9.

Algorithme Conversion WGS84 en Lambert II étendue

Objectif : Transformer les coordonnées géographiques (ϕ_w, λ_w) issues du GPS en coordonnées projetées Lambert II étendue (x_{l2e}, y_{l2e}).

Algorithme :

1. Transformation de (ϕ_w, λ_w) vers (x_w, y_w, z_w) :

$$e_w = \frac{a_w^2 - b_w^2}{a_w^2}$$

$$N = \frac{a_w}{\sqrt{(1 - e_w * sin(\phi_w)^2)}}$$
$$x_w = N \cos \phi_w \cos \lambda_w$$
$$y_w = N \cos \phi_w \sin \lambda_w$$
$$z_w = N(1 - e_w) \sin \phi_w$$

avec : $a_w = 6378137$ et $b_w = 6356752.314$

2. Transformation de (x_w, y_w, z_w) vers (x_n, y_n, z_n) par translation de vecteur $(168, 60, -32)$;

3. Transformation de (x_n, y_n, z_n) vers (ϕ_n, λ_n) :

$$e_n = \frac{a_n^2 - b_n^2}{a_n^2}$$

$$p_0 = atan\left(d_n . z_n \left(1 - \frac{a_n e_n}{\sqrt{x_n^2 + y_n^2 + z_n^2}}\right)\right) \quad (5.2)$$

$$p_1 = atan\left(d_n . z_n \left(1 - \frac{a_n e_n \cos p_0}{\sqrt{(x_n^2 + y_n^2 + z_n^2)(1 - e_2 * sin(p_0)^2}}\right)\right) \quad (5.3)$$

avec : $a_n = 6378249.2$, $b_n = 6356515$ et $d = \frac{1}{\sqrt{x_n^2 + y_n^2}}$

4.1. Tant que $|p_1 - p_0| > \epsilon$, alors $p_0 = p_1$, évaluer à nouveau p_1 (cf. équation (5.3)).

4.2. $\phi_n = p_1$ et $\lambda_n = \text{atan}\frac{y_n}{x_n}$

4. Transformation de (ϕ_n, λ_n) vers (x_{l2e}, y_{l2e}) :

$$x_{l2e} = x_s + c . exp(-nL\,) \sin\big(n(\lambda_n - \lambda_0)\big)$$
$$y_{l2e} = y_s - c . exp(-nL) \cos\big(n(\lambda_n - \lambda_0)\big)$$

avec : $n = 0.7289686274,\ c = 11745793.39,\ x_s = 600000,\ y_s = 8199695.768,$ $\lambda_0 = 0.04079234433198$ et :

$$L = log\left(tan\left(\frac{\pi}{4} + \frac{\phi_n}{2}\right)\right).\left(\frac{1 - \sqrt{e_n}.sin\,\phi_n}{1 + \sqrt{e_n}.sin\,\phi_n}\right)^{\frac{\sqrt{e_n}}{2}}$$

Table 5.9 : Conversion de données WGS84 en coordonnées projectives Lambert II étendue

5.4.3.2. Localisation sur des données réelles

Ce test a été réalisé à partir de la plate-forme expérimentale « Gem Car » avec le stéréoscope : Bumbeblee XB3 et un GPS RTK. La déviation observée entre les positions du GPS RTK et la méthode de la localisation est en moyenne de 35.4cm. La mesure la plus éloignée entre la trajectoire GPS et la méthode est de 1.18m.

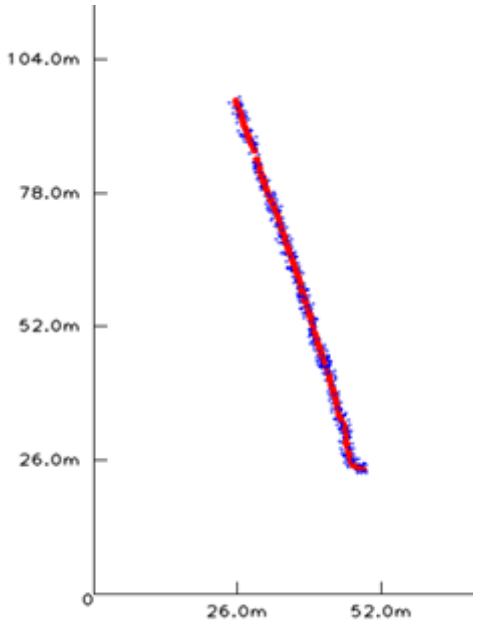


Figure 5.27 : Trajectoire réelle effectué par le GemCar. En rouge les données GPS RTK. Les points bleus correspondent à l'évaluation de la localisation par stéréovision.

5.5. Conclusion

Les expérimentations nous confirment la robustesse de la méthode proposée. Celle-ci permet une mise en correspondance aisée entre différents points de vue et différentes conditions liées à l'environnement. Lors d'un changement d'éclairage ou

bien avec une partie de l'image occultée ou bien une perte de nettteté de l'image, la méthode SIFT fournit toujours des résultats corrects. Les performances de la reconstruction permettent d'obtenir des bons résultats lors de la localisation sous le simulateur.

CONCLUSION ET PERSPECTIVES

CONCLUSION

Les travaux présentés dans cette thèse contribuent aux systèmes de localisation de véhicules. En proposant un système de reconstruction 3D en utilisant la stéréoscopie, nous avons vu qu'il était possible de se localiser précisément dans un environnement urbain.

L'approche proposée est décomposée en deux étapes. La première étape constitue une phase d'apprentissage. Elle permet de construire un modèle 3D de l'environnement de navigation par des acquisitions stéréoscopiques. Pour construire un modèle 3D, les primitives SIFT extraites des caméras gauche et droite sont reconstruites localement. Par une méthode robuste, ces primitives sont appariées avec les primitives du modèle. Ainsi on peut déterminer avec des appariements 3D/3D une position précise du capteur stéréoscopique. Un ajustement de faisceaux local est également effectué sur les données afin d'éviter les erreurs de dérives.

La deuxième étape utilise le modèle précédemment créé. Les méthodes utilisées pour la localisation sont identiques à celles de la reconstruction. On détermine sur une trajectoire proche de la trajectoire apprise une pose précise du stéréoscope.

La méthode développée se différencie essentiellement des méthodes monoculaires par sa méthode de localisation basée sur des appariements 3D/3D. La localisation fonctionne directement avec des points 3D du modèle et non pas en utilisant des images clés. Il est pour l'instant difficile de savoir si la méthode stéréoscopique est plus précise que la méthode monoculaire. La méthode de reconstruction, les primitives choisies, les algorithmes de localisation, sont différents ce qui rend la comparaison difficile.

PERSPECTIVES

Les travaux issus de cette thèse ont permis de montrer la réalisation d'une méthode de localisation par stéréovision. Cependant, de nombreux tests restent à réaliser en conditions réelles d'utilisation. En particulier la comparaison avec les méthodes monoculaires.

Lors de reconstruction, il se peut que des doublons de point SIFT 3D soit présent dans le modèle reconstruit. Par exemple, un point SIFT reconstruit à partir de deux

points de vue suffisamment éloignés pour que la méthode de mise en correspondance considère ce même point comme différent. Dans le cas idéal, un point 3D reconstruit devrait pouvoir être mis en correspondance depuis tous les points de vue. Récemment, une nouvelle approche appelée Affine-SIFT [50] permet de s'approcher d'un extracteur complètement invariant. Cette approche pourrait être une très bonne alternative aux points SIFT pour optimiser la base de points 3D.

Comme nous avons pu le voir lors des expérimentations, les points SIFT ne permettent pas d'être complètement invariants aux changements de vue. L'utilisation d'autres extracteurs invariants tels que le SURF, MSER, Hessian-Affine, etc. pourrait apporter des améliorations des résultats de reconstruction et de localisation.

L'implémentation des différents algorithmes de localisation et de reconstruction ont été développés en utilisant le langage C++, mais ne sont pas optimisés pour fonctionner en temps réel. Le temps de calcul est pénalisé lourdement par l'extraction des points SIFT qui nécessite un temps de traitement d'environ 1s. Il est donc nécessaire d'utiliser une version s'appuyant sur la puissance des GPU et des technologies SIMD (Simple Instruction Multiple Data) des récents ordinateurs. D'après les travaux de Wu sur le projet SiftGPU[18], qui utilise entre autre la technologie CUDA[19], l'utilisation du GPU permettrait d'atteindre une fréquence 12.5Hz pour extraire les points SIFT pour une résolution identique à la nôtre de 1280x960.

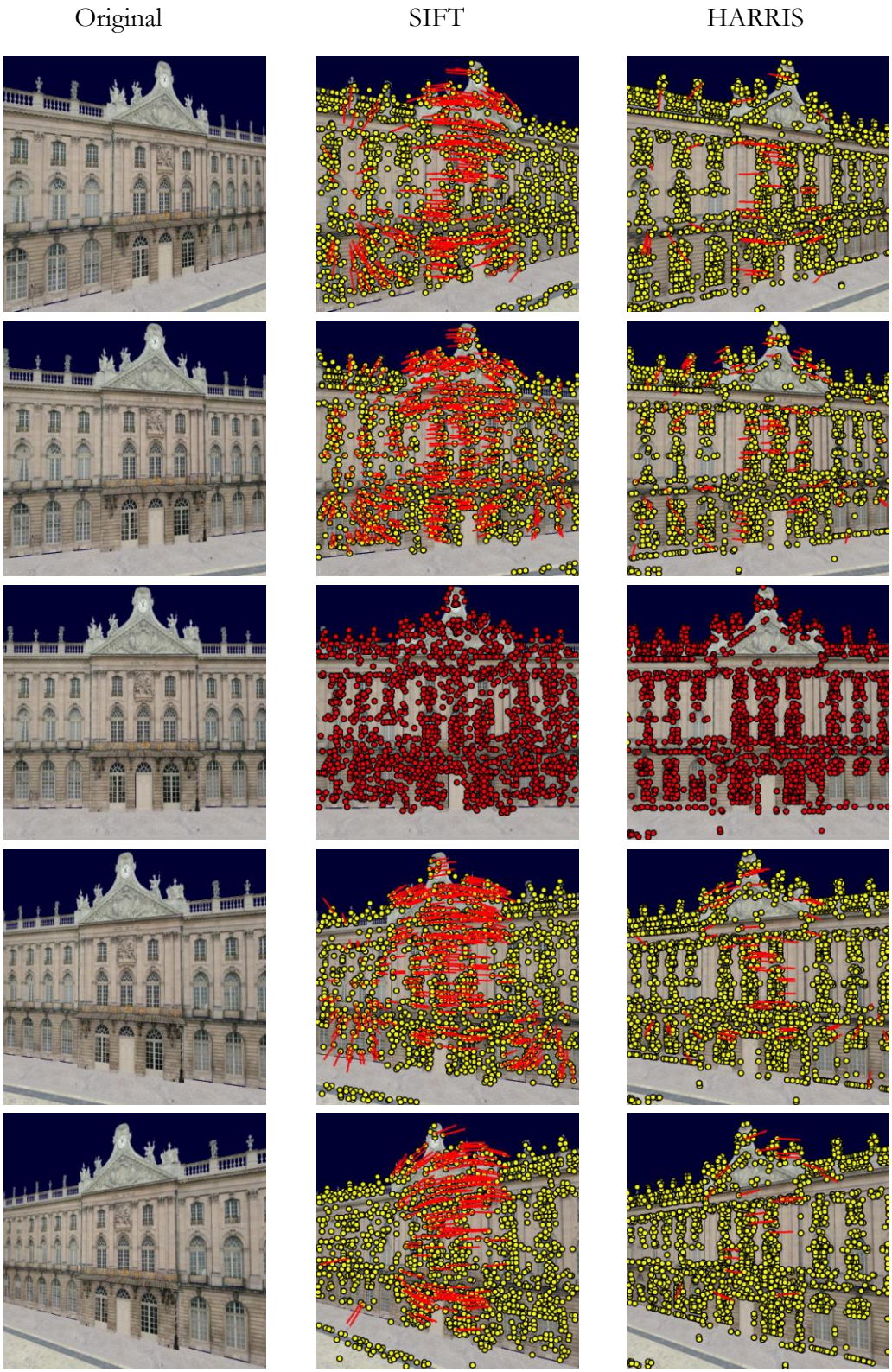
La méthode de localisation stéréoscopique présentée n'est qu'une brique pour le mur de la navigation autonome. De nombreuses tâches sont nécessaires à un véhicule afin de pouvoir naviguer automatiquement dans un milieu urbain. La préparation d'un démonstrateur en temps réel est en court de développement.

[18] http://www.cs.unc.edu/~ccwu/siftgpu/

[19] http://www.nvidia.fr/object/cuda_what_is_fr.html

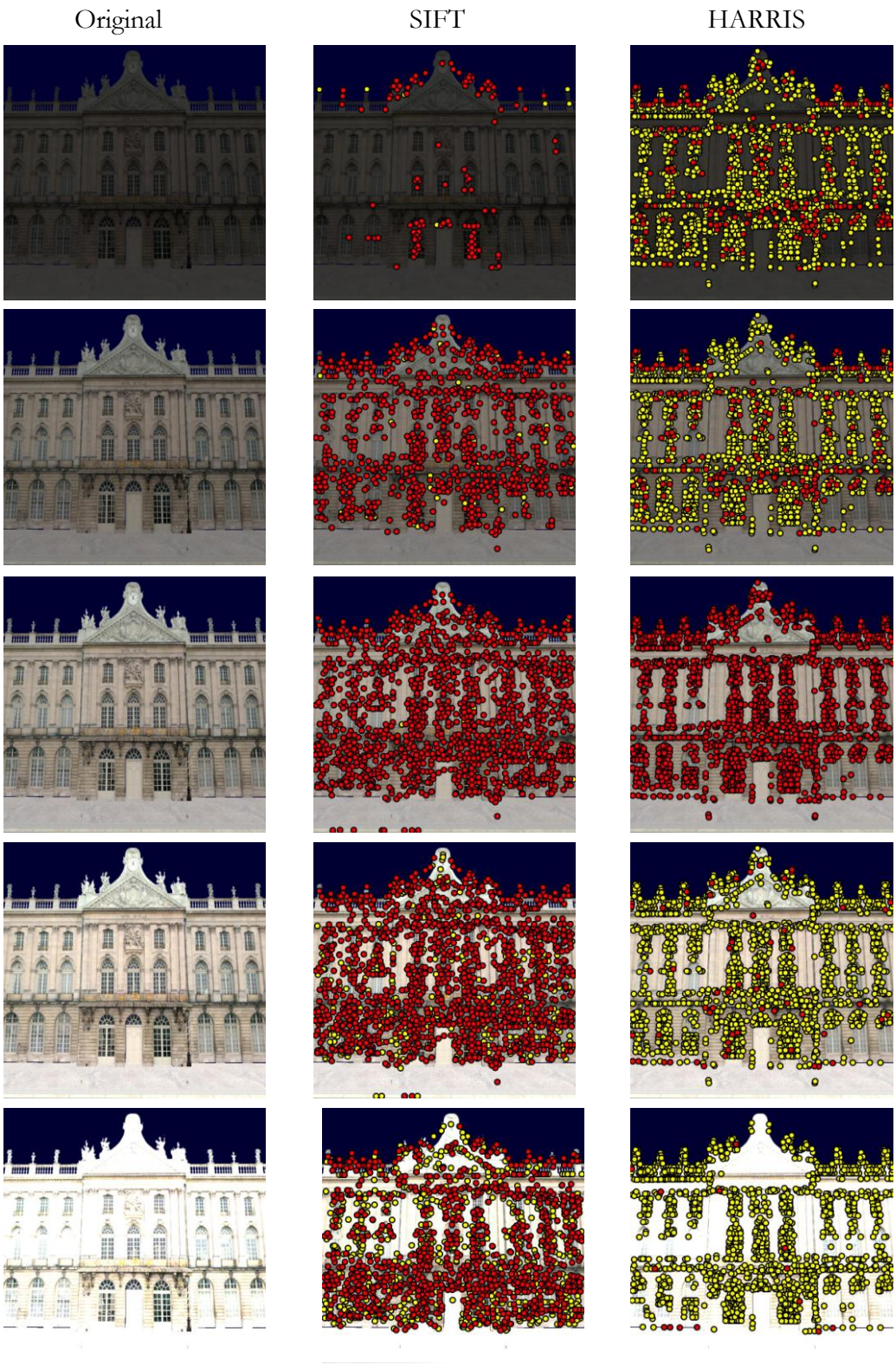
ANNEXES

A.1. Changements de point de vue avec des images virtuelles



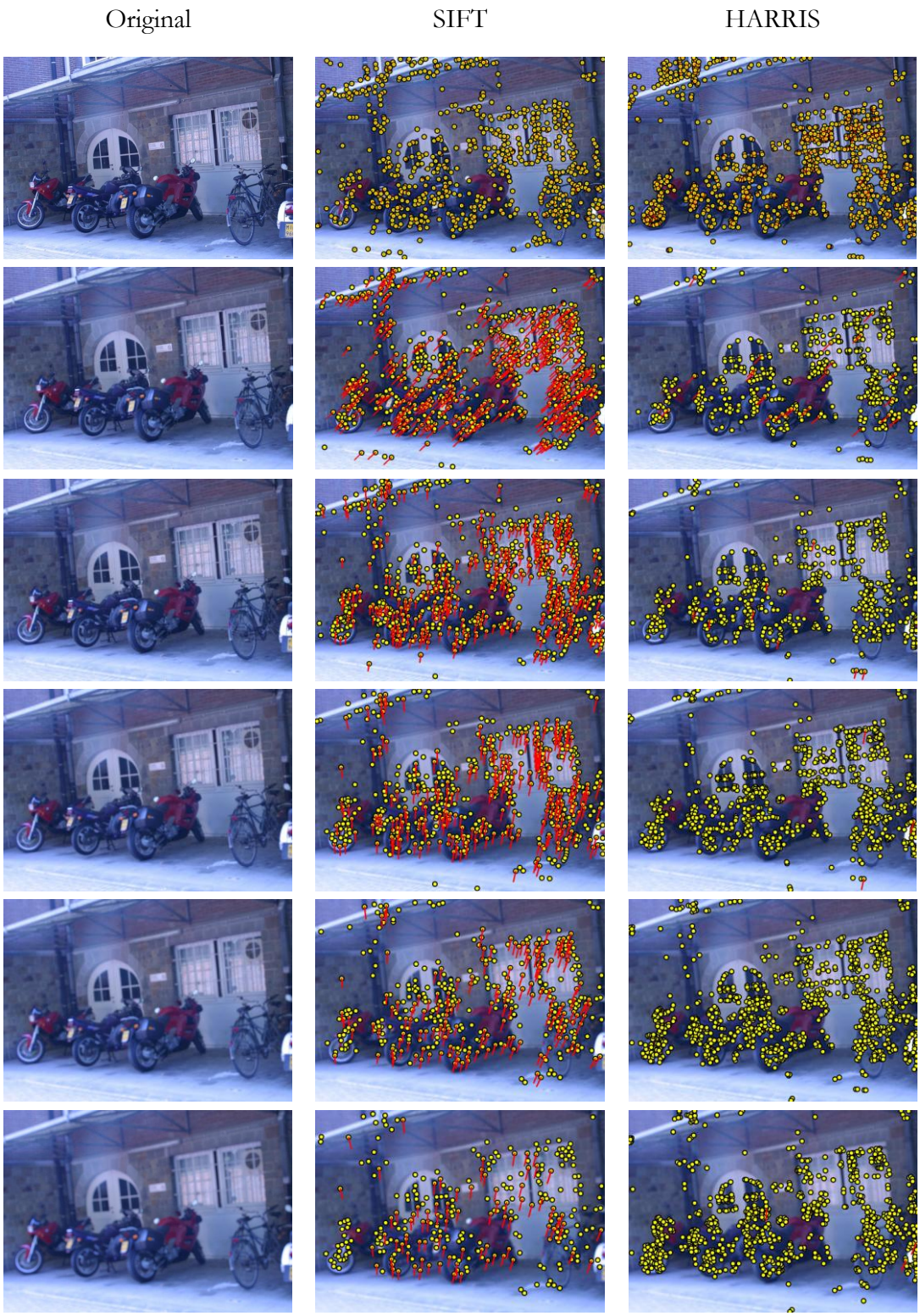
Les points en jaunes correspondent aux points extraits. Les traits ou points rouges correspondent aux points appariés.

A.2. Changements d'illumination avec des images virtuelles



Les points en jaunes correspondent aux points extraits. Les points en rouges correspondent aux points appariés.

A.3. Images avec différents niveaux de flous

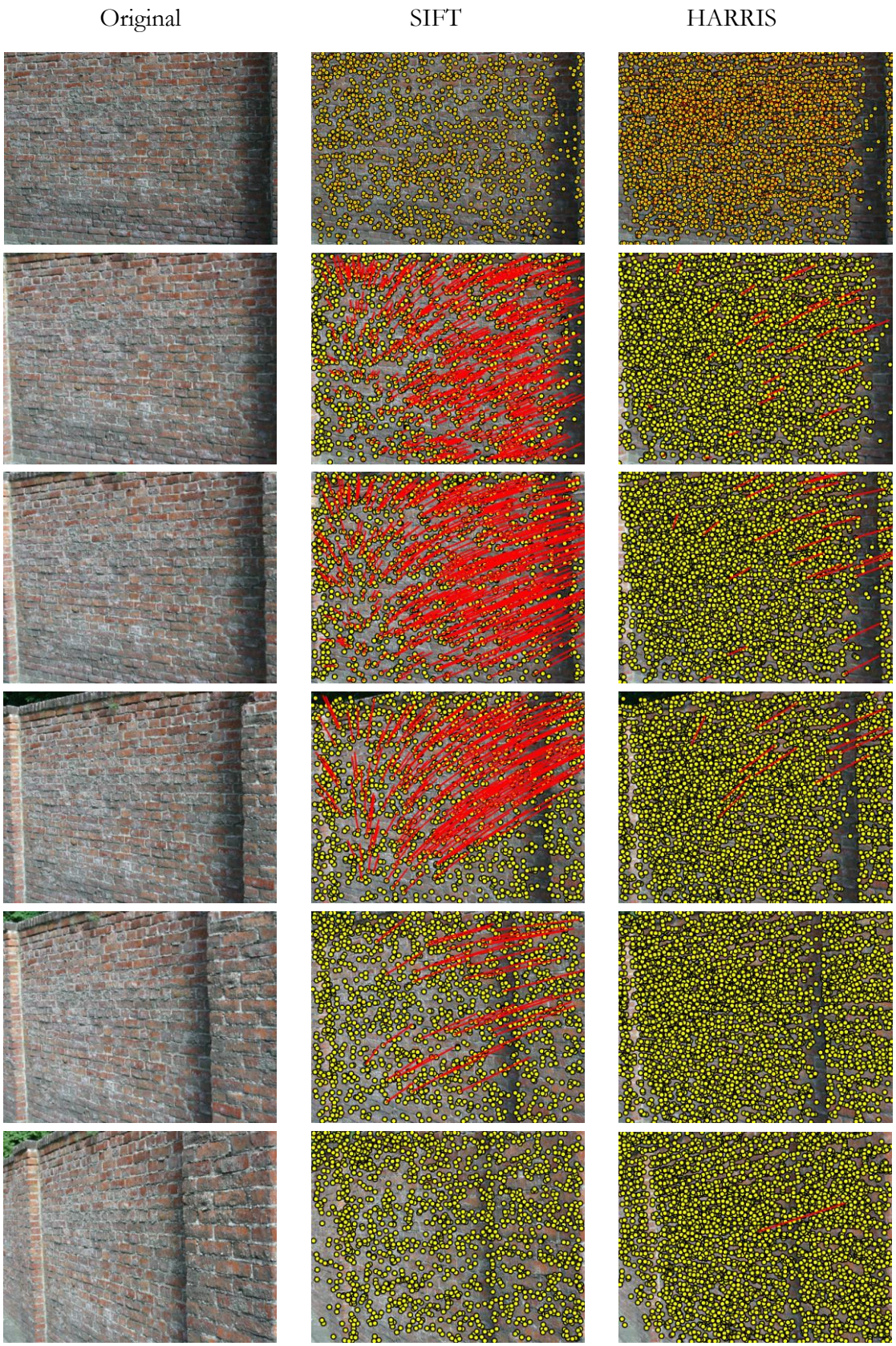


Les points en jaunes correspondent aux points extraits. Les traits rouges correspondent aux points appariés.

A.4. Changements d'illumination avec des images réelles

Les points en jaunes correspondent aux points extraits. Les traits rouges correspondent aux points appariés

A.5. Changements de points de vue avec des images réelles



A.6. Capteurs et ordinateur embarqués

MircoAutobox

Ce système temps réel est destiné à la réalisation du prototypage rapide de fonction de contrôle de véhicule. Elle fonctionne sans l'intervention de l'utilisateur, simplement comme un calculateur. Ce système offre des interfaces vers tous les principaux bus automobiles : CAN, LIN, K/L-Line et FlexRay.

LMS 221

Le LMS 221 est un Télémètre laser mobile. Il réalise des mesures avec un balayage en deux dimensions : il scrute son environnement et récupère les coordonnées polaires de celui-ci. C'est une combinaison d'un télémètre à temps de vol avec un système de rotation du faisceau de mesure, grâce à cette technique on obtient une vision sous forme de radar.

BEO LUX

C'est un lidar qui fournit des données de haute qualité sur les objets environnants et l'environnement. Il permet la réalisation rapide et simultanée de toutes les applications frontales, comme le CAC Stop & Go, la protection des piétons, ANB, PreCrash, Traffic Jam adjoint et adjoint aux intersections.

Caractéristiques principales :

- Fréquence de balayage: 12,5 Hz (25Hz)
- Angle horizontal: 100 °
- Champ de vision à portée: 0,3 m à 200 m
- Champ vertical de vision : 3,2°
- Jusqu'à trois mesures en réflexion par impulsion laser
- Taille: H85 x long 128 x Larg 83
- Ethernet et CAN-Interface

MTi

Le MTi est une centrale inertielle miniature de haute précision, qui permet de mesurer en temps réel les mouvements (accélération, vitesse de rotation) d'un objet et de calculer son orientation. La centrale MTi regroupe 9 capteurs de type MEMS : 3 gyroscopes, 3 accéléromètres, et 3 magnétomètres.

Caractéristiques particulières :

- Données brutes et orientation calculée à une cadence max de 512Hz/120Hz ;
- Précision en dynamique : 2°RMS ;
- Extrêmement légère et compacte (50 g) ;
- Logiciel intégré Magnetic Field Mapper permettant de compenser les perturbations électro-magnétiques ;

Bumblebee

Le Bumblebee XB3 est un stéréoscope équipé de 3 capteurs multi-baseline IEEE-1394b (800Mb/s). C'est une caméra stéréo conçue pour améliorer la flexibilité et la précision. Le XB3 est doté de capteurs 1,3 méga-pixels et dispose de deux lignes de base disponibles pour le traitement stéréo. La ligne de base élargie et la haute résolution fournissent plus de précision à plus longue portée, tandis que la ligne de base étroite améliore ce qui est à faible portée

Caractéristiques techniques :

- Capteur : Trois Sony 1 / 3 "CCD à balayage progressif, Color / BW ;
- Résolution et FPS : 1280x960 à 15fps ;
- Objectifs : 3.8mm (70 ° HFOV) ou 6mm (50 ° HFOV) Focale ;
- Calibration : pré-calibré à 0,05 pixel erreur RMS ;

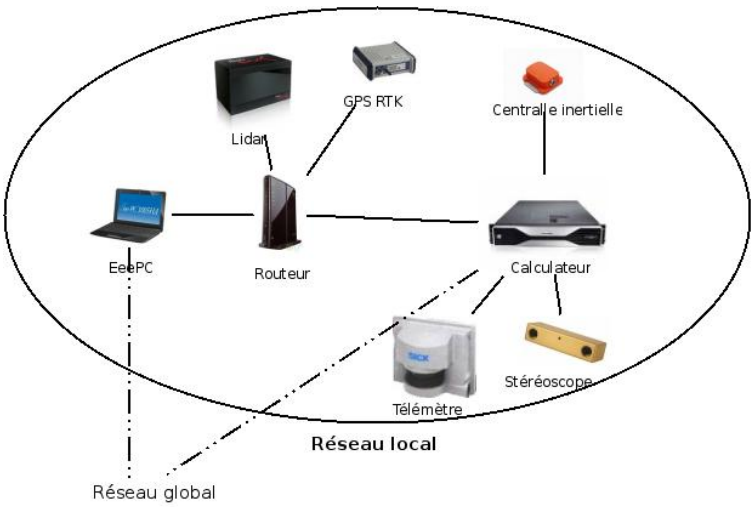
ProFlex 500

Ce récepteur GNSS (Global Navigation Satellite System) bi-fréquence, multi-applications et multi-constellations permet d'utiliser les réseaux GPS et Glonass pour une initialisation rapide et une précision centimétrique sur longue distance.

Caractéristiques principales :

- Multi-constellations : GPS + Glonass + SBAS, et bientôt Galileo
- 75 canaux, sortie données brutes et position à 10Hz (20Hz en option)

Architecture hardware



BIBLIOGRAPHIE

[1] A Baumberg, "Reliable feature matching across widely separated views," in *Conference on Computer Vision and Pattern Recognition*, Hilton Head, South Carolina, 2000, pp. 774-781.

[2] P. Beaudet, "Rotationally invariant image operators," *International Joint Conference on Pattern Recognition*, pp. 579-583, 1978.

[3] J Borenstein, H. R Everett, L Feng, and D Wehe, "Mobile Robot Positionning : Sensors and Techniques," *Journal of Robotic Systems*, pp. 14(4): 231-249, 1997.

[4] J. Borenstein and L. Feng, "Measuerment and correction of systematic odometry errors in mobile robots," *IEEE Transactions on Robotics and Automation*, (12), 1996.

[5] J.-Y. Bouguet. Camera Calibration Toolbox for Matlab. [Online]. http://www.vision.caltech.edu/bouguetj/calib_doc/

[6] M Brown and D.G Lowe, "Invariant features from interest point groups," in *British Machine Vision Conference*, Cardiff, Wales, 656-665, p. 2002.

[7] C. Capelle, "Localisation de véhicules et détection d'obstacles. Apport d'un modèle virtuel 3D urbain," Université des Sciences et Technologies de Lille, Lille, Thèse de doctorat 2008.

[8] C. Capelle, "Localisation de véhicules et détection d'obstacles. Apport d'un modèle virtuel 3D urbain," Thèse de doctorat 2008.

[9] R. H. Chan, C.-W. Ho, and M. Nikolova, "Salt-and-Pepper Noise Removal by Median-Type Noise Detectors and Detail-Preserving Regularization," *IEEE Transactions on Image Processing*, vol. 14, no. 10, pp. 1479-1485, 2005.

[10] J.L Crowley and A.C Parker, "A representation for shape based on peaks and ridges in the difference of low-pass transform," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, pp. 6(2):156-170.

[11] T. Cui and S. S. Ge, "Autonomous vehicle positioning with GPS in urban canyon environments," *IEEE Transaction on Robotics and Automation*, vol. 19, no. 1.

[12] R Deriche and Giraudon G., "Acurate corner detection: An analytic study," INRIA, Sophia-Antipolis, France, Technical Report 1420 1991.

[13] R. O. Duda and P. E. Hart, "Use of the Hough Transformation to Detect Lines and Curves in Pictures," *Comm. ACM*, vol. 15, pp. 11-15, 1972.

[14] Y Dufournaud, C Schmid, and R Horaud, "Matching images with different resolutions," in *Proceedings of Conference Computer Vision and Pattern Recognition*, 2000, pp. 612-618.

[15] A. Eudes and M. Lhuillier, "Propagations d'erreur pour l'ajustement de faisceaux local," in *ORASIS - Congrès des jeunes chercheurs en vision par ordinateur*, 2009.

[16] A. Eudes and M. Lhuillier, "Propagations d'erreur pour l'ajustement de faisceaux local," in *ORASIS - Congrès des jeunes chercheurs en vision par ordinateur*, 2009.

[17] O. Faugeras, *Three-Dimensional Computer Vision - A Geometric ViewPoint*.: MIT Press, 1993.

[18] M. A. Fischler and R. C. Bolles, "Random sample consensus : A paradigm for model fitting with applications to image analysis and automated cartography," *Comm. Of the ACM*, pp. 381-395, June 1981.

[19] Morel and G.Y, "ASIFT: A New Framework for Fully Affine Invariant Image Comparison,".

[20] D. Garcia, Orteu J. J., and M. Devy, "Accurate calibration of a stereovision sensor: comparison of different approaches," in *5th Fall Workshop on Vision, Modeling and Visualisation*, 2000.

[21] V Gouet, "Mise en correspondance d'Images en Couleur," in *PhD thesis*, Université Montpellier II, 2000.

[22] B. B Hansen and Morse B. S S, "Multiscale image registration using scale trace correlation," in *Proceedings of Conference on Computer Vision and Pattern Recognition*, 1999, pp. 202-208.

[23] C Harris, "Geometry from visual motion," in *Action Vision*.: MIT Press, 1992, pp. 263-284.

[24] C Harris and M Stephens, "A combined corner and edge detector," *Fourth Alvey Vision Conference*, pp. 147-151, 1988.

[25] R. I. Hartley, "In defense of the eight-point algorithm," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 580-593, 1997.

[26] C. T. Hong, T. H. Rasmussen, and C. E. Shneier, "Road detection and tracking for autonomous mobile robots," in *PIE Aerosense Conference*, Orlando, 2002.

[27] T. Kadir, A. Zisserman, and M. Brady, "An affine invariant salient region detector," in *European Conference on Computer Vision*, 2004, pp. 404-416.

[28] R.E Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35-45, 1960.

[29] A. King, "Inertial navigation : forty years of evolution," , 1998.

[30] K. Kluge and S Laskshmanan, "Lane boundary detection using deformable templates : effects of image subsampling on detected lane edges," in *Second Asian Conference on Computer Vision (ACCV'95)*, Singapore, 1995, pp. 329-339.

[31] W. Krasprzak, H. Niemann, and D. Wetzel, "Adaptive road parameter estimation in monocular image sequences," in *5th British Machine Vision Conference (BMVC'94)*, 1994, pp. 691-700.

[32] D. C. Lavest and M. Viala, "Une mire d'étalonnage: à quelle précision?," *Actes du 10ème congrés AFCET de Reconnaissance des Formes et Intelligence Artificielle*, vol. 3, pp. 35-45, 1998.

[33] T Lendeberg, "Detecting salient blob-like image structures and their scales with a scale-space primal sketch: a method for focus-of-attention," *Internation Journal of Computer Vision*, pp. 11(3):283-318, 1993.

[34] T Lendeberg, "Scale-space theory: A basic tool for analysing structures at different scales," *Journal of Applied Statistics*, pp. 21(2):224-270, 1994.

[35] T Linderberg, "Feature detection with automatic scale selection," *International Journal of Computer Vision*, pp. 30(2):79-116, 1998.

[36] T Linderberg, "Scale-space theory: A basic tool for analysing structures at different scales," *Journal of Applied Statistics*, pp. 21(2):224-270, 1994.

[37] H. C. Longuet-Higgins, "A computer algorithm for reconstructing a scene from two projections," in *Nature*, 1981, pp. 293:133-135.

[38] D. G Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, pp. 60(2):91-110, 2004.

[39] D. G Lowe, "Object recognition from local scale-invariant features," in *International conference on Computer Vision*, Corfu, Greece, 1999, pp. 1150-1157.

[40] D. G Lowe, "Object recognition from local scale-invariant features," in *International Conference on Computer Vision*, Corfu, Greece, 1999, pp. 1150-1157.

[41] Q. T. Luong, "Matrice fondamentale et Autocaliration en vision par ordinateur," Université de Paris Sud, Orsay, Thèse de doctorat 1992.

[42] David Marr, *Vision*.: Fremann and Company, 1982.

[43] J. Mattas, O. Chum, M. Urban, and T. Pajdla, "Robust wide baseline stereo from maximally stable extremal regions," in *Proceedings 13th British Machine Vision Conference*, 2002, pp. 384-393.

[44] K Mikolajczyk and Schmid C, "Scale & affine invariant interest point detectors," *International Journal of Computer Science*, pp. 60(1):63-86, 2004.

[45] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1615-1630, 2005.

[46] K Mikolajczyk and C Schmid, "An affine invariant interest point detector," in *European Conference on Computer Vision (ECCV)*, Copenhagen, Denmark, 2002, pp. 128-142.

[47] K Milolajczyk et al., "A comparison of affine region detectors," *International Journal of Computer Vision*, pp. 65(1/2):43-72, 2005.

[48] P Montesinos, V Gouet, and R Deriche, "Differential invariants for color images," in *Proceedings of 14th International Conference on Pattern Recognition*, Australia, 1998.

[49] H Moravec, "Rover visual obstacle avoidance," *International Joint Conference on Artificial Intelligence*, pp. 785-790, 1981.

[50] J.M. Morel and G. Yu, "ASIFT: A New Framework for Fully Affine Invariant Image Comparison," *SIAM Journal on Imaging Sciences*, vol. 2, no. 2, pp. 438-469, 2009.

[51] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, and P. Sayd, "Reconstruction 3D générique et temps réel," in *Congrès francophone AFRIF-AFIA de reconnaissance des formes et d'intelligence artificielle*, Amiens, 2008.

[52] E. Mouragnon, M. Lhuilllier, M. Dhome, F. Dekeyser, and P. Sayd, "Real Time localization and 3D reconstruction," in *IEEE Conference on Computer Vision and Pattern Recognition*, New-York, USA, 2006.

[53] D. Nistér, "An efficient solution to the five-point relative pose problem," in *Transactions on Pattern Analysis and Machine Intelligence*, 2004, pp. 26(6):756-770.

[54] D. Nistér, "An efficient solution to the five-point relative pose problem.," in *In Conference on Computer Vision and Pattern Recognition*, 2003, pp. 147-151.

[55] S. Nogueira, "Localisation d'un véhicule dans un environnement urbain par analyse de scènes prises par des caméras embarquées," *Prix A'Doc de la jeune recherche en Franche-Comté*, pp. 75-88, 2007.

[56] S. Nogueira, F. Gechter, Y. Ruichek, A. Koukam, and F. Charpillet, "Environment perception for vehicle autonomous navigation in urban areas," in *Biennal on DSP for In-Vehicle and Mobile Systems.*: Springer, 2006.

[57] S Nogueira, Y Ruichek, and F. Carpillet, "A learning based global localization method using stereovision," in *IEEE Intelligent Transportation on Vehicular Electronics and Safety*, Pune, India, 2009.

[58] S. Nogueira, Y. Ruichek, and F. Charpillet, "A Self Navigation Technique using Stereovision Analysis," in *Stereo Vision*, Asim Bhatti, Ed.: InTech Education and Publishing , 2008, pp. 287-298.

[59] S. Nogueira, Y. Ruichek, and F. Charpillet, "Localisation de mobiles par

apprentissage en utilisant la stéréovision," in *5ième Colloque Interdisciplinaire en Instrumentation*, Le Mans, France, 2010.

[60] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*.: McGraw Hill, 1965.

[61] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes in C: The Art of Scientific Computing*, Second Edition ed.: Cambridge University Press, 1992.

[62] C. Rasmussen, "Grouping dominant orientations for ill-structured road following," in *IEEE Conference on Computer vision and Pattern Recognition (CVPR'04)*, Washington, 2004.

[63] H. Rheinglod, *La Réalité Virtuelle*.: Dunod, 1993.

[64] Azriel Rosenfeld, "From image analysis to computer vision : An annotated bibliography, 1955-1979," *Computer Vision and Image Understanding*, pp. 84:298-324, 2001.

[65] P. J. Rousseeuw and A. M. Leroy, *Robust Regression and Outlier Detection*. New York: Wiley Series, 1987.

[66] E. Royer, "Cartographie 3D et localisation par vision monoculaire pour la navigation autonome d'un robot mobile," Thèse de doctorat 2006.

[67] E. Royer, M. Lhuillier, M. Dhome, and J.-M. Lavest, "Localisation par vision monoculaire pour la navigation autonome : précision et stabilité de la méthode," in *15e congrès francophone AFRIF-AFIA Reconnaissance des Formes et Intelligence Artificielle*, Tours, France, 2006.

[68] P. Sampson, "Fitting conic sections to "very scattered" data: An iterative refinement of the bookstein algorithm," in *Computer Graphics and Image Processing*, 1982, pp. 18:97-108.

[69] F Schaffalitzky and A. Zisserman, "Multi-view matching for unordered image sets, or 'How do I organize my holidays snaps?'," in *European Conference on Computer Vision*, Copenhagen Denmark, 2002, pp. 414-431.

[70] F Schaffalitzky and A Zisserman, "Multi-view matching for unordrer image sets," in *Proceedings 7th European Conference on Computer Vision*, 2002, pp. 414-431.

[71] C. Schmid and R Mohr, "Local grayvalue invariants for image retrieval," *IEEE Trans. on Parttern Analysis and Machine Intelligence*, pp. 19(5):530-534.

[72] C Schmid and R Mohr, "Local grayvalue invariants for image retrieval," *PAMI*, pp. 19(5):530-534, 1997.

[73] C. Schmid, R. Mohr, and C. Bauckhage, "Evaluation of Interest Point Detectors," *International Journal of Computer Vision*, vol. 37, no. 2, pp. 151-172, 2000.

[74] S. W. Shih, Y. P. Hung, and W. S. Lin, "When should consider lens distorsion in camera calibration," *Pattern Recognition*, vol. 28, no. 3, pp. 447-461, 1995.

[75] A Shokoufandeh, Marsic, I, and D.J Dickinson, "View-based object recognition using saliency maps," *Image and Vision Computing*, no. 17:445-460, 1999.

[76] S Thrun, D Fox, W Burgard, and F Dellaert, "Robust monte carlo localization for mobile robots.," *Artificial Intelligence Journal*, pp. 99-141, 2001.

[77] P Torr, "Motion Segmentation and Outlier Dectection," Dept. of Engineering Science, University of Oxford, UK, Ph.D Thesis 1995.

[78] B. Triggs, P. McLauchlan, R. I. Hartley, and A. Fitzgibbon, "Bundle Adjustment a Modern Synthesis," *Vision Algorithms: Theory and Pracctice*, vol. 1883, pp. 298-372, 2000.

[79] R. Turchetto and R. Manduchi, "Visual curb localization for autonomous navigation," in *IEEE RSJ/International conference on Intelligent Robot and System (IROS'03)*, Las Vegas, 2003, pp. 1336-1342.

[80] T. Tuytelaars and K. Mikolajczyk, *Local Invariant Feature Detectors: A Survey*, Publishers Inc., Ed.: Foundations and Trends in Computer Graphics and Vision, 2008.

[81] T Tuytelaars and L Van Gool, "Matching widely seperated views based on affine invariant regions," *International Journal of Computer Vision*, pp. 59(1):61-85, 2004.

[82] T Tuytelaars and L Van Gool, "Wide baseline stereo based on local, affinely invariant regions," in *British Machine Vision Conference*, Bristol, UK, 2000, pp. 412-422.

[83] R. Wang, Y. Xu, and Y. Zhao, "A vision-based road edge detection algorithm," in *Sixth IEEE Workshop on Applications of Computer Vision (WASC'02)*, Orlando, 2002, pp. 237-241.

[84] L. Williams, "Casting curved shadows on curved surfaces," in *ACM SIGGRAPH Computer Graphics*, New York, 1978, pp. 270-274.

[85] O. J. Woodman, "An introduction to inertial navigation," University Of Cambridge, Computer Laboratory, Technical Report UCAM-CL-TR-696 2007.

[86] Z. Zhang, "Parameter Estimation Techniques: A Tutorial with Application to Conic Fitting," *Image and Vision Computing Journal*, vol. 15, no. 1, pp. 59-76, 1997.

[87] Z Zhang, R Deriche, O Faugeras, and Q.T Luong, "A robust technique for matching two uncalibrated images through the recovery of the unknow epipolar geometry," in *Artificial Intelligence*., 1995, pp. 78:87-119.