Reconstruction 3D de l'environnement dynamique d'un véhicule à l'aide d'un système multi-caméras hétérogène en stéréo wide-baseline

Laurent Mennillo

Université Clermont Auvergne
École Doctorale des Sciences Pour l'Ingénieur

Thèse présentée par :

Laurent MENNILLO
Ingénieur ESIL Informatique

Pour obtenir le grade de :

Docteur d'Université
Spécialité : Vision pour la Robotique

Reconstruction 3D de l'environnement dynamique d'un véhicule à l'aide d'un système multi-caméras hétérogène en stéréo *wide-baseline*

Soutenue publiquement le 5 juin 2019 devant le jury :

M.	Michel DHOME	Directeur de thèse
Mme.	Ouiddad LABBANI-IGBIDA	Rapporteuse et examinatrice
M.	Frédéric MONDOT	Examinateur
M.	El Mustapha MOUADDIB	Président du jury
M.	Eric ROYER	Examinateur
Mme.	Sylvie TREUILLET	Rapporteuse et examinatrice

Remerciements

Je tiens tout d'abord à remercier l'Institut Pascal ainsi que le Groupe Renault, qui sont à l'initiative de ces travaux de thèse et ont permis de leur donner deux dimensions complémentaires, en les inscrivant à la fois dans les domaines académique et industriel. Au sein de ces deux entités, je remercie tout particulièrement les équipes avec lesquelles j'ai eu le plaisir de travailler, pour leur support scientifique, logistique et humain.

Je remercie également les membres du jury de thèse – avec qui j'ai pris grand plaisir à partager – pour l'intérêt qu'ils ont porté à ces travaux, que j'ai pu apprécier de part la qualité de leurs rapports ainsi que lors de leurs remarques et questions pertinentes durant la soutenance.

Je remercie mon directeur et mon encadrant de thèse au sein de l'Institut Pascal, ainsi que mes encadrants de thèse au sein de Renault, qui, malgré les inévitables difficultés scientifiques, logistiques et humaines que j'ai pu traverser, m'ont apporté confiance et conseils durant ces travaux.

Enfin, je tiens à remercier tout particulièrement mes parents, ma famille et mes proches, ami·e·s et collègues, pour leur intérêt, leurs conseils, leur soutien, leur patience (et leur amour, pour certain·e·s), durant ces années de thèse. Nommer chacun·e m'étant très difficile, je leur laisserai le soin de se reconnaître dans ces quelques lignes. Beaucoup méritent de partager cette réussite, tant leur implication s'est révélée essentielle. Je leur en suis profondément reconnaissant.

Résumé

Cette thèse a été réalisée dans le secteur de l'industrie automobile, en collaboration avec le Groupe Renault et concerne en particulier le développement de systèmes d'aide à la conduite avancés et de véhicules autonomes. Les progrès réalisés par la communauté scientifique durant les dernières décennies, dans les domaines de l'informatique et de la robotique notamment, ont été si importants qu'ils permettent aujourd'hui la mise en application de systèmes complexes au sein des véhicules. Ces systèmes visent dans un premier temps à réduire les risques inhérents à la conduite en assistant les conducteurs, puis dans un second temps à offrir des moyens de transport entièrement autonomes. Les méthodes de SLAM multi-objets actuellement intégrées au sein de ces véhicules reposent pour majeure partie sur l'utilisation de capteurs embarqués très performants tels que des télémètres laser, au coût relativement élevé. Les caméras numériques en revanche, de par leur coût largement inférieur, commencent à se démocratiser sur certains véhicules de grande série et assurent généralement des fonctions d'assistance à la conduite, pour l'aide au parking ou le freinage d'urgence, par exemple. En outre, cette implantation plus courante permet également d'envisager leur utilisation afin de reconstruire l'environnement dynamique proche des véhicules en trois dimensions. D'un point de vue scientifique, les techniques de SLAM visuel multi-objets existantes peuvent être regroupées en deux catégories de méthodes. La première catégorie et plus ancienne historiquement concerne les méthodes stéréo, faisant usage de plusieurs caméras à champs recouvrants afin de reconstruire la scène dynamique observée. La plupart reposent en général sur l'utilisation de paires stéréo identiques et placées à faible distance l'une de l'autre, ce qui permet un appariement dense des points d'intérêt dans les images et l'estimation de cartes de disparités utilisées lors de la segmentation du mouvement des points reconstruits. L'autre catégorie de méthodes, dites monoculaires, ne font usage que d'une unique caméra lors du processus de reconstruction. Cela implique la compensation du mouvement propre du système d'acquisition lors de l'estimation du mouvement des autres objets mobiles de la scène de manière indépendante. Plus difficiles, ces méthodes posent plusieurs problèmes, notamment le partitionnement de l'espace de départ en plusieurs sous-espaces représentant les mouvements individuels de chaque objet mobile, mais aussi le problème

d'estimation de l'échelle relative de reconstruction de ces objets lors de leur agrégation au sein de la scène statique. La problématique industrielle de cette thèse, consistant en la réutilisation des systèmes multi-caméras déjà implantés au sein des véhicules, majoritairement composés d'un caméra frontale et de caméras surround équipées d'objectifs très grand angle, a donné lieu au développement d'une méthode de reconstruction multi-objets adaptée aux systèmes multi-caméras hétérogènes en stéréo *wide-baseline*. Cette méthode est incrémentale et permet la reconstruction de points mobiles éparses, grâce notamment à plusieurs contraintes géométriques de segmentation des points reconstruits ainsi que de leur trajectoire. Enfin, une évaluation quantitative et qualitative des performances de la méthode a été menée sur deux jeux de données distincts, dont un a été développé durant ces travaux afin de présenter des caractéristiques similaires aux systèmes hétérogènes existants.

Mots-clés : Reconstruction 3D, SLAM visuel multi-objets, Systèmes multi-caméras hétérogènes, Stéréo *wide-baseline*, Ajustement de faisceaux.

Abstract

This Ph.D. thesis, which has been carried out in the automotive industry in association with Renault Group, mainly focuses on the development of advanced driver-assistance systems and autonomous vehicles. The progress made by the scientific community during the last decades in the fields of computer science and robotics has been so important that it now enables the implementation of complex embedded systems in vehicles. These systems, primarily designed to provide assistance in simple driving scenarios and emergencies, now aim to offer fully autonomous transport. Multibody SLAM methods currently used in autonomous vehicles often rely on high-performance and expensive onboard sensors such as LIDAR systems. On the other hand, digital video cameras are much cheaper, which has led to their increased use in newer vehicles to provide driving assistance functions, such as parking assistance or emergency braking. Furthermore, this relatively common implementation now allows to consider their use in order to reconstruct the dynamic environment surrounding a vehicle in three dimensions. From a scientific point of view, existing multibody visual SLAM techniques can be divided into two categories of methods. The first and oldest category concerns stereo methods, which use several cameras with overlapping fields of view in order to reconstruct the observed dynamic scene. Most of these methods use identical stereo pairs in short baseline, which allows for the dense matching of feature points to estimate disparity maps that are then used to compute the motions of the scene. The other category concerns monocular methods, which only use one camera during the reconstruction process, meaning that they have to compensate for the ego-motion of the acquisition system in order to estimate the motion of other objects. These methods are more difficult in that they have to address several additional problems, such as motion segmentation, which consists in clustering the initial data into separate subspaces representing the individual movement of each object, but also the problem of the relative scale estimation of these objects before their aggregation within the static scene. The industrial motive for this work lies in the use of existing multi-camera systems already present in actual vehicles to perform dynamic scene reconstruction. These systems, being mostly composed of a front camera accompanied by several surround fisheye cameras in wide-baseline stereo, has led to the

development of a multibody reconstruction method dedicated to such heterogeneous systems. The proposed method is incremental and allows for the reconstruction of sparse mobile points as well as their trajectory using several geometric constraints. Finally, a quantitative and qualitative evaluation conducted on two separate datasets, one of which was developed during this thesis in order to present characteristics similar to existing multi-camera systems, is provided.

Keywords : 3D reconstruction, Multibody VSLAM, Heterogeneous multi-camera systems, Wide-baseline stereo, Bundle-adjustment.

Table des matières

Table des figures

Liste des tableaux

Liste des Algorithmes

Notations

$\mathbb{R}^n$	Espace vectoriel de dimension n sur le corps des réels
$\mathbb{P}^n$	Espace projectif de dimension n sur le corps des réels
$A \in \mathbb{R}^{m \times n}$	Matrice à $m \in \mathbb{N}^*$ lignes et $n \in \mathbb{N}^*$ colonnes
$\mathbf{x}$	Vecteur
$\|\mathbf{x}\|$	Norme euclidienne du vecteur $\mathbf{x}$
P	Point 3D
$\bar{P}$	Coordonnées homogènes du point 3D P
p	Point 2D
$\bar{p}$	Coordonnées homogènes du point 2D p
T_1^2	Matrice de passage homogène de la base $\mathcal{B}_1$ vers la base $\mathcal{B}_2$
R	Matrice de rotation
$\mathbf{t}$	Vecteur de translation
f	Distance focale
(u_0, v_0)	Point principal
K	Matrice des paramètres intrinsèques d'une caméra
$[P]$	Matrice de projection d'une caméra
F	Matrice fondamentale
E	Matrice essentielle
Π	Plan (plan épipolaire dans la sous-section 1.1.4)
e	Épipole
l	Droite épipolaire
N	Nombre de points 3D
J	Nombre de caméras du système multi-caméras
T	Nombre de temporalités de la séquence
S_1	Ensemble des points d'intérêt détectés lors du processus d'extraction
S_2	Ensemble des points d'intérêt suivis lors du processus d'extraction

$C_{j,t}$	Pose de la caméra j à la temporalité t
P_n	n^{e} point 3D
P_n^t	n^{e} point 3D à la temporalité t
$\mathtt{p}_{j,t}$	Ensemble de points d'intérêt associés à la caméra j à la temporalité t
$p_{j,t}^n$	Point d'intérêt associé au point 3D P_n appartenant à $\mathtt{p}_{j,t}$
$\mathtt{m}_{j,j'}^{t,t'}$	Ensemble des appariements entre $\mathtt{p}_{j,t}$ et $\mathtt{p}_{j',t'}$
$m_\phi(p,p')$	Appariement potentiel entre $p \in \mathtt{p}_{j,t}$ et $p' \in \mathtt{p}_{j',t'}$
$m_\Gamma(p,p')$	Appariement définitif entre $p \in \mathtt{p}_{j,t}$ et $p' \in \mathtt{p}_{j',t'}$
$\mathtt{o}_n$	Ensemble des observations du point P_n
$\mathtt{o}_n^t$	Ensemble des observations du point P_n à la temporalité t
$o_{j,t}^n$	Observation de P_n par la caméra j à la temporalité t, assimilable à $p_{j,t}^n$
C_L	Contrainte de localité d'appariement
C_E	Contrainte épipolaire d'appariement
C_{E_θ}	Contrainte épipolaire d'appariement appliquée aux caméras *fisheye*
C_{In}	Contrainte d'évaluation des inliers
C_{In_θ}	Contrainte d'évaluation angulaire des inliers
C_C	Contrainte de consistance
C_{C_θ}	Contrainte de consistance angulaire
C_{S1}	1^{e} contrainte d'immobilisme - 2 temporalités différentes
C_{S2}	2^{e} contrainte d'immobilisme - 3 observations différentes
C_{M1}	1^{e} contrainte de mobilité - Consistance stéréo
C_{M1_θ}	1^{e} contrainte angulaire de mobilité - Consistance angulaire stéréo
C_{M2}	2^{e} contrainte de mobilité - 2 observations par temporalité
C_{M3}	3^{e} contrainte de mobilité - 2 temporalités différentes
C_{T1}	1^{e} contrainte de trajectoire - Vitesse
C_{T2}	2^{e} contrainte de trajectoire - Changement d'élévation
C_{T3}	3^{e} contrainte de trajectoire - Changement de direction
T_L	Seuil associé à C_L, en pixels
T_E	Seuil associé à C_E, en pixels
T_{E_θ}	Seuil associé à C_{E_θ}, en degrés
T_{In}	Seuil associé à C_{In}, en pixels
T_{In_θ}	Seuil associé à C_{In_θ}, en degrés
T_C	Seuil associé à C_C, en pixels
T_{C_θ}	Seuil associé à C_{C_θ}, en degrés
T_{M1}	Seuil associé à C_{M1}, en pixels

T_{M1_θ}	Seuil associé à C_{M1_θ}, en degrés
$\mathsf{T}_{T1_{min}}$	Seuil associé à C_{T1}, en mètres
$\mathsf{T}_{T1_{max}}$	Seuil associé à C_{T1}, en mètres
T_{T2}	Seuil associé à C_{T2}, en mètres
T_{T3}	Seuil associé à C_{T3}, en degrés

Introduction

Cette thèse a été réalisée dans le secteur de l'industrie automobile, en collaboration avec le groupe Renault et concerne en particulier le développement de systèmes d'aide à la conduite avancés ou ADAS (*Advanced Driver-Assistance Systems*, en anglais) et de véhicules autonomes. Les progrès réalisés par la communauté scientifique durant les dernières décennies, dans les domaines de l'informatique et de la robotique notamment, ont été si importants qu'ils permettent aujourd'hui la mise en application de systèmes complexes au sein des véhicules. Ces systèmes visent dans un premier temps à réduire les risques inhérents à la conduite en assistant les conducteurs, puis dans un second temps à offrir des moyens de transport entièrement autonomes. Aujourd'hui encore à l'état de prototypes à accès restreint, pour les plus avancés, ces véhicules intelligents apparaissent progressivement sur les routes dans un cadre très contrôlé, alors que leur commercialisation de masse accompagnée de la législation adéquate est planifiée pour les années à venir.

Contexte scientifique et industriel

Beaucoup de laboratoires, constructeurs et équipementiers développent activement les technologies associées aux transports intelligents de demain. Bien que les débuts des recherches scientifiques et applications industrielles portant sur les voitures autonomes datent des années 20, les premières expérimentations réelles aux résultats significatifs sont apparues dans les années 80, avec des projets tels que le Navlab de l'Université Carnegie Mellon ou encore l'Eureka Prometheus Project, piloté conjointement par Mercedes-Benz et l'Université Bundeswehr de Munich. Depuis lors, plusieurs laboratoires et entreprises ont contribué à la recherche et mise en application de ces véhicules autonomes, notamment, parmi les plus connus, l'Université de Carnegie Mellon, l'Université d'Oxford ou encore le MIT pour les acteurs académiques, alors que dans le secteur industriel, Mercedes-Benz, BMW, Google (maintenant Waymo) ou plus récemment Tesla et Nvidia sont actuellement très avancés sur le sujet. Ces différents acteurs de la recherche et de l'industrie utilisent divers capteurs, données et algorithmes afin de produire des

véhicules capables de percevoir et d'analyser leur environnement. Parmi ces capteurs, une première catégorie concerne les capteurs proprioceptifs, tels que les accéléromètres, gyroscopes, odomètres, ou encore les capteurs d'angle de braquage des roues. D'autres appartiennent à la catégorie des capteurs extéroceptifs, comme les récepteurs GPS ou encore les capteurs basés sur la mesure d'ondes lumineuses, d'ondes sonores ou d'ondes électromagnétiques. Pour majeure partie, les informations recueillies par ces capteurs permettent notamment la mise en place de systèmes ayant pour objectif la perception en trois dimensions de l'environnement situé autour du véhicule ainsi que l'estimation précise de sa localisation au sein de cet environnement. Parmi les plus utilisés à ces fins, les télémètres laser occupent une large place, car ils offrent de très bonnes performances, au prix cependant de relatives difficultés d'intégration et de coûts élevés. Les caméras numériques font également partie des équipements ayant profité d'avancées technologiques importantes, tant d'un point de vue technique, de part leur miniaturisation, que d'un point de vue commercial, avec des prix devenus plus abordables, ce qui a conduit à leur introduction courante dans la plupart des véhicules disponibles sur le marché. Pour majeure partie actuellement, ces caméras assurent une fonction d'aide visuelle au conducteur, en proposant des images de l'environnement proche du véhicule à la visibilité difficile, comme les caméras de recul ou les systèmes à 360° de type AVM (*Around View Monitor*, en anglais) pour l'aide au parking, par exemple. Plus récemment, d'autres tâches plus spécifiques impliquant l'utilisation de caméras (surveillance des angles morts, détection de franchissement de ligne, de panneaux de circulation ou de piétons, etc.), commencent à se démocratiser. Les fonctionnalités offertes par ces capteurs ne se limitent cependant pas qu'à ces quelques exemples d'applications. L'estimation précise de leurs paramètres de calibration permet également de les utiliser afin de reconstruire le modèle de la scène observée en trois dimensions, offrant ainsi un vaste potentiel d'applications supplémentaires. Alors que certaines de ces applications sont assez évidentes et déjà disponibles sur certains véhicules, telles que l'estimation visuelle de leur odométrie ou encore la détection d'obstacles statiques, d'autres, plus complexes, devraient permettre l'évaluation dynamique de la topologie de l'espace navigable, des règles de circulation ainsi que du contexte routier dans lequel se trouve le véhicule en rapport aux autres usagers, afin de détecter toute situation dangereuse et de s'y adapter. C'est précisément dans le développement de telles applications que s'inscrit le sujet de cette thèse, qui se concentre sur la détection et la reconstruction en trois dimensions par vision des éléments fixes et mobiles situés autour d'un véhicule.

Problématique et objectifs

L'une des problématiques industrielles de ces travaux s'inscrit dans une logique de réduction du coût des systèmes actuellement utilisés dans le but de percevoir l'environnement dynamique d'un véhicule en trois dimensions. Comme évoqué dans la précédente section, bien que l'utilisation de télémètres laser réponde exactement à ce problème, leur coût relativement élevé empêche pour le moment leur adoption de masse, alors que les systèmes de caméras actuellement intégrés aux véhicules n'imposeraient certainement qu'un investissement marginal afin d'offrir des prestations approchantes. Ces systèmes multi-caméras consistent généralement en une caméra frontale, placée derrière le rétroviseur central, ainsi que de quatre caméras *surround*, équipées d'objectifs très grand angle (également appelés *fisheye*, en anglais), placées sur la calandre, le hayon de coffre et intégrées aux rétroviseurs gauche et droit afin de permettre une observation de l'environnement du véhicule à 360°. Partant de cette observation, la problématique industrielle de ces travaux se situe dans l'utilisation de tels systèmes multi-caméras à champs recouvrants, présentant des focales et points d'observation très différents, pour la reconstruction de scènes dynamiques.

D'un point de vue scientifique, les méthodes basées sur l'utilisation de caméras et permettant la reconstruction 3D de la scène observée ainsi que la localisation du robot au sein de cette scène, font partie des méthodes de SLAM visuel ou VSLAM (*Visual Simultaneous Localization And Mapping*). Parmi ces méthodes, une importante partie se concentre sur la reconstruction d'environnements statiques, c'est à dire que la composante dynamique, donc mobile, de la scène observée (voitures, motos, cyclistes, piétons, etc.), n'est pas prise en compte lors du processus de reconstruction et en est simplement éliminée. Ce type d'approche n'est relativement pas adapté au contexte d'aide à la conduite ou de conduite autonome, dans la mesure où la plupart des situations routières impliquent d'autres usagers. Une autre approche, développée par la suite et visant à étendre les méthodes classiques de reconstruction, a consisté à introduire au sein des méthodes de SLAM existantes un processus de détection et de reconstruction de ces composantes dynamiques. Il s'agit des approches dites de SLAM multi-objets (*Multibody VSLAM*, en anglais). Plusieurs méthodes de SLAM multi-objets existent dans la littérature et peuvent être regroupées en deux catégories. La première catégorie et plus ancienne historiquement regroupe les méthodes faisant usage de plusieurs caméras à champs visuels recouvrants, c'est à dire en stéréo-vision, afin de reconstruire la scène dynamique observée. La plupart de ces méthodes reposent en général sur l'utilisation de paires de caméras identiques et placées à faible distance l'une de l'autre, ce qui permet un appariement dense des points d'intérêt dans les images et l'estimation de cartes de disparités utilisées lors de la segmentation du mouvement des points reconstruits. L'autre catégorie

de méthodes, les méthodes monoculaires, ne font usage que d'une unique caméra lors du processus de reconstruction. Cela implique la compensation du mouvement propre du système d'acquisition lors de l'estimation indépendante du mouvement des autres objets mobiles de la scène. Plus difficiles, ces méthodes posent plusieurs problèmes, notamment le partitionnement de l'espace de départ en plusieurs sous-espaces représentant les mouvements individuels de chaque objet mobile, mais aussi le problème d'estimation de l'échelle relative de reconstruction de ces objets lors de leur agrégation au sein de la scène statique.

Compte tenu de cette problématique industrielle et scientifique, le principal objectif de ces travaux de thèse se situe enfin dans l'adaptation des méthodes de SLAM multi-objets existantes au cas particulier de systèmes multi-caméras à focales hétérogènes et points d'observations éloignés, c'est à dire en stéréo *wide-baseline*.

Contributions

Durant ces travaux de thèse, une méthode de détection et reconstruction incrémentale de points mobiles a été proposée. Sa conception permet un fonctionnement sur plusieurs types de systèmes multi-caméras à large champs recouvrants, notamment dans le cas particulier et relativement difficile de systèmes hétérogènes en stéréo *wide-baseline*. Au sein de cette méthode, deux mécanismes ont permis cette flexibilité d'utilisation. Le premier concerne le processus d'extraction et d'appariement de points d'intérêt, qui permet le regroupement de toutes les observations temporelles et stéréo associées à chaque point 3D observé par le système multi-caméras de manière récursive. Le deuxième mécanisme se rapporte à la méthode de classification des points reconstruits, reposant d'une part sur l'évaluation de l'ensemble de leurs observations grâce à plusieurs contraintes géométriques visant à détecter leur mouvement, et d'autre part sur l'évaluation de leur trajectoire afin de filtrer ceux présentant un mouvement erratique ou dégénéré. Une autre contribution concerne la création d'un jeu de données représentatif de tels systèmes multi-caméras, répondant notamment aux contraintes industrielles évoquées et permettant d'envisager l'utilisation éventuelle de la méthode proposée en conditions réelles. Enfin, les travaux présentés dans ce manuscrit ont donné lieu à une première publication [109], alors qu'une version plus complète est à soummettre [110].

Plan du manuscrit

Ce manuscrit se compose de trois chapitres. Le premier chapitre est un état de l'art, dans lequel sont notamment abordées les méthodes de reconstruction par vision existantes de scènes statiques et dynamiques, accompagnées d'une présentation de leurs

principaux concepts théoriques. Le deuxième chapitre présente la méthode développée dans le cadre de ces travaux de thèse afin d'étendre les techniques de reconstruction multi-objets existantes au cas particulier de systèmes multi-caméras à focales hétérogènes en stéréo *wide-baseline*. Enfin, dans le dernier chapitre sont présentés les résultats de la méthode obtenus sur deux jeux de données distincts, dont celui développé durant ces travaux et présentant des caractéristiques similaires aux systèmes actuellement implantés sur certains véhicules de grande série.

Chapitre 1

État de l'art et fondements scientifiques

1.1 SLAM visuel

1.1.1 Problématique, définitions et travaux antérieurs

1.1.1.1 Méthodes de reconstruction par vision

Le SLAM (*Simultaneous Localization And Mapping*, en anglais), ou localisation et cartographie simultanées, est un processus permettant de reconstruire en trois dimensions la carte de l'environnement à portée d'un capteur et de s'y localiser simultanément. Ces capteurs peuvent être de différents types, actifs ou passifs et mesurer différentes grandeurs physiques en vue d'estimer une distance. Bien que chacun permette en théorie la réalisation d'un SLAM, leurs performances en matière de précision, de vitesse d'acquisition, de difficulté de mise en œuvre et de coût de revient peuvent fortement varier, ce qui implique le choix d'un capteur adapté à la tâche et aux conditions de fonctionnement du robot sur lequel il sera installé. Comme mentionné en introduction de ce manuscrit, l'utilisation de caméras numériques a été motivée par le cahier des charges défini dans le cadre de cette thèse, et bien que ces travaux ne se concentrent que sur ce type de capteur, de plus amples détails sur le problème du SLAM en général, exprimé indépendamment du type de capteur utilisé, ont été présentés dans un tutoriel en deux parties proposé par Durrant-Whyte et Bailey [31, 6]. Les méthodes faisant usage de capteurs photosensibles et permettant de reconstruire la scène observée en trois dimensions peuvent être classées dans quatre grandes catégories. Deux de ces catégories concernent les méthodes actives et les méthodes passives. La différence entre les deux réside dans l'usage de sources spécifiques de lumière émise de manière contrôlée lors de la mesure dans les méthodes actives, alors que les méthodes passives ne font usage que de sources

naturelles de lumière. Les deux autres catégories concernent les méthodes basées sur l'exploitation d'un unique point de vue sur la scène (uni-vue) et celles nécessitant des points de vue multiples (multi-vues). Parmi les méthodes actives uni-vue, on peut-citer celles de type *Time-of-Flight* [67, 5, 47], où l'on mesure le délai entre une impulsion lumineuse et sa réflexion sur la surface à reconstruire (principe utilisé notamment par les LIDARs et plus récemment, les caméras de type *Time-of-Flight*), et celles de type *Shape-from-Shading* basées sur la mesure de l'ombrage induit par une source lumineuse contrôlée sur une surface aux propriétés de réflectance connues [195, 30]. Dans la catégorie des méthodes actives multi-vues se trouvent notamment celles de type *Lumière Structurée* [151, 144], où l'on analyse la projection sur la surface à reconstruire d'un motif lumineux particulier, les méthodes de type *Stéréo Active* [73, 5], où l'on triangule à l'aide d'une caméra la position d'un point projeté par un laser sur la surface à reconstruire, et enfin les méthodes de type *Photometric Stereo* qui sont une généralisation multi-vues du principe de *Shape-from-Shading* [187, 38]. Les méthodes passives uni-vue sont basées sur les principes de *Shape-from-Texture* ou *Shape-from-Template* [3, 8], où l'on analyse la déformation d'une texture aux propriétés géométriques connues sur une surface (par exemple, son homogénéité), de *Shape-from-Contour* [19], similaire au *Shape-from-Texture*, en utilisant le contour de formes géométriques planaires (cercles, rectangles, etc.), et de *Shape-from-Defocus*, qui utilise le point de netteté optique de la caméra afin d'en estimer la distance par rapport à la surface à reconstruire [43]. Enfin, parmi les méthodes passives multi-vues, on retrouve celles basées sur le principe de *Shape-from-Silhouettes*, calculant à intervalles de rotation réguliers l'intersection de la forme à reconstruire avec le cône de vision de la caméra [107], de *Stéréo Passive*, où l'on triangule la position d'un point de la scène observé simultanément par deux caméras à différents points de vue [61, 40, 128, 154], ainsi qu'une extension de ce principe utilisée pour la reconstruction de scènes statiques et appelée *Structure-from-Motion* [28, 120], lorsque les prises de vue sont effectuées par une unique caméra à chaque instant et point de vue. La figure 1.1 illustre ces deux derniers principes de reconstruction. Dans ce chapitre seront exposées, entre autres, les différentes méthodes de la littérature permettant de réaliser un SLAM visuel hybride tel que celui utilisé dans le chapitre 2, reposant sur les principes de *Structure-from-Motion* et de *Stéréo Passive* et nécessitant l'utilisation de plusieurs caméras. Ce type de SLAM implique certaines notions spécifiques qui font depuis quelques années l'objet d'articles et d'ouvrages de référence [61, 146, 50] et dont les fondements scientifiques seront réintroduits de manière analogue dans cette section.

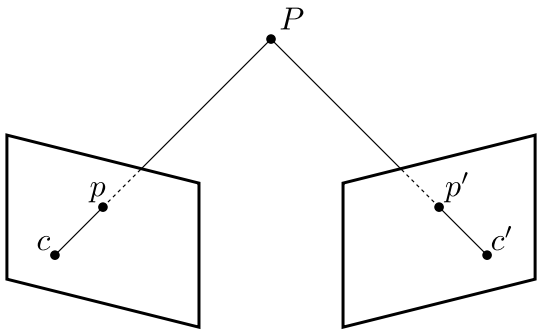

FIGURE 1.1 – *Stéréo Passive* et *Structure-from-Motion*. Dans le cas du principe de *Stéréo Passive*, les deux vues sont prises simultanément à deux points d'observation différents, alors que dans le cas du principe de *Structure-from-Motion*, il y a un décalage temporel entre la première et la seconde vue afin de passer du premier au second point d'observation. Dans cette illustration, les caméras de centre c et c' observent le point 3D P qui est reprojeté sur le plan image de chaque caméra en p et p', respectivement.

1.1.1.2 Fonctionnement d'un SLAM visuel incrémental

Les techniques permettant la réalisation d'un SLAM visuel incrémental présentées dans cette section visent à extraire la structure tridimensionnelle d'une scène puis à s'y localiser à partir d'observations acquises de manière séquentielle à différents points de vue et en deux dimensions, généralement sous la forme d'images ou de flux vidéo. Elles reposent sur les principes de *Stéréo Passive* et de *Structure-from-Motion* ou *SfM*, évoqués dans le paragraphe précédent, qui reposent à leur tour sur le concept de triangulation, dont l'explication est la suivante. Considérant l'image acquise par une caméra comme la projection perspective d'une scène en trois dimensions, la position dans l'espace d'un point de cette scène observée depuis deux points de vue différents se trouve à l'intersection des deux rayons passant chacun, pour chaque caméra, par son centre optique et la projection de ce point sur l'image associée. Afin de trianguler les points d'une scène pour retrouver leur position dans l'espace, il faut donc passer par deux étapes fondamentales, qui sont la détection et l'appariement dans les images de points d'intérêt homologues, correspondant aux mêmes points de la scène, et connaître précisément les caractéristiques optiques ainsi que la position dans l'espace des caméras. Ces caractéristiques optiques sont modélisées grâce à des modèles géométriques comportant des paramètres intrinsèques, tandis que la position et l'orientation d'une caméra, regroupées sous le terme de pose, correspondent à ses paramètres extrinsèques. À noter que les paramètres extrinsèques sont systématiquement définis dans un référentiel qu'il convient de préciser. Afin de lever toute ambiguïté d'écriture, dans ce manuscrit seront désignées par *paramètres extrinsèques de pose*, ou plus simplement *pose*, la position et l'orientation d'une caméra définies relativement au repère monde, c'est à dire l'espace en trois dimensions

dans lequel le robot évolue. De manière moins spécifique, seront désignées par *paramètres extrinsèques* la position et l'orientation d'une caméra définies relativement à un repère propre au robot qui l'utilise. Dans le cadre d'un système multi-caméras par exemple, ce repère propre désigne généralement le repère de l'une des caméras du système. Alors que les paramètres intrinsèques se calculent bien souvent grâce à une mire d'étalonnage aux caractéristiques physiques connues, ce que l'on appelle le processus de calibration intrinsèque d'une caméra ou l'étalonnage intrinsèque d'une caméra, le calcul de pose et la calibration extrinsèque font appel à la géométrie épipolaire notamment, qui définit les relations géométriques entre deux images en se basant sur un certain nombre de correspondances de points d'intérêt. Une fois les poses des caméras estimées, il est ensuite possible de reconstruire en trois dimensions les points d'intérêt appareillés par triangulation. Ces points ainsi que les poses des caméras sont enfin optimisés afin de minimiser les erreurs de reconstruction éventuelles. La figure 1.2 illustre les différentes étapes d'un SLAM visuel incrémental.

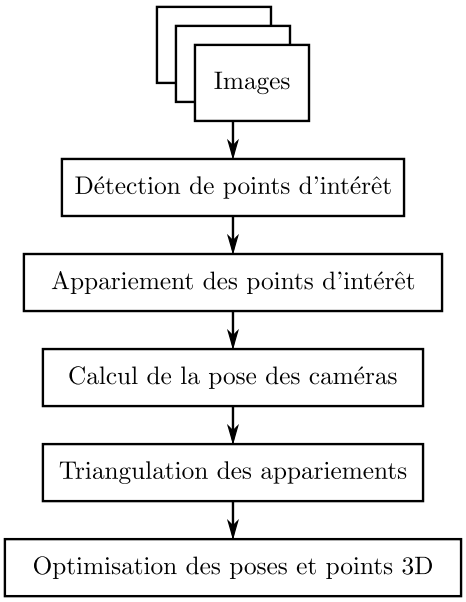

FIGURE 1.2 – Fonctionnement d'un SLAM visuel incrémental.

1.1.1.3 Distinction entre SLAM visuel et odométrie visuelle

Bien que basées sur les mêmes principes (en l'occurrence, *SfM* et *Stéréo Passive*), les méthodes de SLAM et d'odométrie visuels n'ont pas tout à fait les mêmes philosophies. Les méthodes de SLAM ont pour objectif l'estimation précise et globale de la géométrie de la scène et de la trajectoire du robot, tandis que les méthodes d'odométrie ne s'attachent qu'à les estimer de manière locale. En conséquence, les méthodes d'odométrie

visuelle peuvent constituer une première étape dans la réalisation d'un SLAM global, plus lourd et complexe à implémenter afin de garantir que la dérive de la trajectoire du robot reste minimale en globalité. En particulier, les méthodes de SLAM implémentent souvent une étape permettant la fermeture de boucles, mettant en correspondance les points déjà reconstruits à d'autres points d'intérêt observés plus tard dans la séquence, ce qui se produit généralement lorsque le robot repasse sur une trajectoire déjà empruntée. L'idée principale réside dans la mise en correspondance de ces points d'intérêt lorsque les poses de leurs caméras respectives sont proches. Chaque correspondance représente alors une contrainte supplémentaire qui, lors de l'étape d'optimisation globale, permet une réduction parfois importante des dérives de trajectoire et de facteur d'échelle. Cette technique, bien qu'efficace dans un contexte de SLAM, augmente sensiblement le temps de calcul, ce qui ne justifie pas forcément son utilisation dans les applications d'odométrie visuelle qui, à l'inverse, ont profité d'autres techniques spécifiques, comme l'optimisation par ajustement de faisceaux local, par exemple. Le choix entre SLAM visuel et odométrie visuelle dépend donc des besoin de l'application considérée en matière de performance et de précision, alors que d'un point de vue scientifique, ces deux philosophies ont mutuellement bénéficié de leurs avancées respectives.

1.1.1.4 Travaux antérieurs

Premières méthodes. Deux catégories distinctes de méthodes d'odométrie et de SLAM visuels, basées l'une sur le principe de *Structure-from-Motion* et l'autre sur celui de *Stéréo Passive*, ont été développées dans la littérature. L'une des premières méthodes développées reposait sur le concept de *Slider Stereo*, introduit dans les travaux de Moravec [119] afin de guider les Rovers utilisés lors des premières missions d'exploration de la planète Mars par la NASA. Le principe de cette méthode consistait, pour chaque position du véhicule, à prendre différentes vues de la scène en effectuant entre chaque prise une translation précisément contrôlée de la caméra du Rover (montée sur rail, d'où le terme *Slider* employé). Bien qu'intuitivement proche du principe de *SfM*, car n'utilisant qu'une unique caméra, la méthode était en réalité équivalente au principe de *Stéréo Passive*, du fait de la connaissance à priori des poses relatives des caméras pour toutes les vues associées à une position donnée du véhicule. Ces travaux précurseurs ont permis d'établir plusieurs des étapes fondamentales des méthodes d'odométrie et de SLAM visuels actuels. Moravec a en premier lieu introduit son détecteur de points d'intérêt puis utilisé les contraintes épipolaires associées à chaque paire d'images afin de réduire la fenêtre de recherche pour chaque correspondance. La sélection de ces correspondances était par la suite réalisée par corrélations croisées normalisées à plusieurs échelles afin de compenser les différences de taille éventuelles en pixels de chaque point d'intérêt. Enfin, l'élimination d'outliers reposait sur le filtrage des points reconstruits présentant des in-

consistances géométriques, alors que la recherche de la transformation rigide permettant de recaler les points triangulés entre chaque nouvelle position du Rover se faisait grâce à une méthode des moindres carrés. Bien que les premiers travaux d'odométrie et de SLAM visuels se soient appuyés sur le principe de *Stéréo Passive*, d'autres méthodes ont été développées par la suite, lorsque s'est posé le problème de la reconstruction d'objets très éloignés, pour lesquels la parallaxe entre les deux caméras d'une paire stéréo n'est plus assez importante pour permettre une triangulation directe des correspondances et se ramène ainsi au cas monoculaire, c'est à dire au principe de *SfM*. De manière plus générale, les méthodes suivant ces travaux se sont particulièrement concentrées sur le développement de deux des étapes introduites par Moravec, qui sont la détection, l'extraction et l'appariement de points d'intérêt ainsi que la manière dont est estimée et optimisée la position du véhicule grâce à ces appariements.

Méthodes stéréo. Matthies *et al.* [108] ont réutilisé la méthode de détection et d'appariement de points proposée par Moravec sur une paire stéréo classique, en incorporant cette fois à l'étape d'estimation du mouvement du véhicule la matrice de covariance représentant l'incertitude associée à la position des points reconstruits. Lacroix *et al.* [82] ont par la suite réalisé un appariement stéréo dense et sélectionné les points d'intérêt en étudiant les extrémas locaux de la fonction de corrélation utilisée pour créer la carte de disparité entre les deux images. Leurs travaux, plus tard repris par [114, 69], ont mis en évidence une corrélation forte entre la courbure de la fonction de corrélation et la déviation standard de la profondeur des points reconstruits. Cheng *et al.* [24, 105] ont finalement repris ces avancées en les améliorant de nouveau, utilisant cette fois le détecteur de coins de Harris [60] et la courbure de la fonction de corrélation autour des points d'intérêt afin de définir la matrice de covariance représentant l'incertitude leur étant associée, puis ont utilisé l'algorithme RANSAC (*RANdom SAmple Consensus*, en anglais) [44] afin d'éliminer les outliers lors de l'étape d'estimation par moindres carrés du mouvement du véhicule. Une approche différente a par la suite été proposée par Milella *et al.* [114], qui ont utilisé le détecteur de coins développé par Shi et Tomasi [155] ainsi qu'un algorithme de type ICP (*Iterative Closest Point*, en anglais) [12] après l'étape de minimisation par moindres carrés afin de recaler plus précisément les points reconstruits d'une position du véhicule à une autre. Toutes ces méthodes ont la particularité d'estimer la position courante du véhicule en recalant les points 3D nouvellement triangulés sur ceux triangulés à la position précédente, ce que l'on appelle une estimation de pose *3D-to-3D*. Une autre approche, encore très utilisée actuellement, a été proposée par Nistér *et al.* dans leur article *Visual Odometry* [128]. Les auteurs ont d'une part réalisé la détection indépendante de points d'intérêt dans chaque image afin de les appareiller directement, permettant ainsi de réduire les erreurs de dérive engendrées lors du suivi

par corrélation réalisé jusqu'alors. D'autre part, ces travaux ont également proposé une méthode alternative d'estimation de pose robuste, de type *3D-to-2D*, faisant usage de la technique PnP (*Perspective-n-Points*, en anglais) [59], qui consiste à recaler les reprojections 2D des points préalablement reconstruits en stéréo sur ceux de l'image suivante, conjointement à l'algorithme RANSAC afin d'éliminer les éventuels outliers. Enfin, il s'agit des premiers travaux à avoir proposé une méthode d'odométrie visuelle fonctionnant en temps réel sur les machines de l'époque. Les méthodes présentées jusqu'à présent ont la particularité d'êtres éparses, reposant sur la détection et l'appariement de points d'intérêts. Parallèlement à ces méthodes, des méthodes denses ont été introduites, reposant pour la plupart sur le calcul du flot optique entre les deux images du système stéréo afin d'obtenir une carte de profondeur de la scène observée. Ce calcul repose en général sur la minimisation d'une fonctionnelle d'énergie, modélisant les propriétés stables des images dans le temps grâce à un terme d'attache aux données ainsi que la régularité du flot optique via un terme de lissage. Les travaux originels présentés par Horn *et al.* [68] sont à l'origine de variantes plus robustes au bruit, changements d'illumination et occultations, notamment les travaux introduits par Brox *et al.* [20], Kim *et al.* [76], Mileva *et al.* [115], Steinbrücker *et al.* [157], Werlberger *et al.* [186] ou Geiger *et al.* [54]. Par ailleurs, afin de respecter les discontinuités du flot optique, plusieurs méthodes prenant en compte les bords de l'image ou contours du flot optique ont été introduites [198]. Deux types de techniques pour le calcul du flot optique sont généralement utilisés [150]. D'une part, l'optimisation discrète, consistant à modéliser l'image et les vecteurs de son déplacement par des champs de Markov aléatoires afin de rechercher les déplacements les plus probables, avec une minimisation ensuite effectuée par coupe de graphe (*Graph Cuts*, en anglais) [79], propagation des convictions (*Belief Propagation*, en anglais) [77] ou programmation dynamique [88]. L'optimisation continue d'autre part, reposant sur l'utilisation de méthodes variationnelles afin d'optimiser une fonctionnelle d'énergie modélisant les données de l'image, produisant ainsi des résultats plus lisses en contraignant les valeurs de profondeur de certaines régions de pixels. Ces méthodes variationnelles décomposent le flot optique directement le long des lignes épipolaires [4, 156], après rectification de l'image [11], ou résolvent directement la profondeur de chaque pixel [171].

Méthodes monoculaires. Contrairement aux méthodes stéréo, pour lesquelles les poses relatives d'au moins deux caméras sont connues à chaque instant et permettent une triangulation directe des points de la scène simultanément observés, les méthodes monoculaires reposent sur le principe de *SfM*, c'est à dire que le déplacement d'une caméra ne peut être estimé qu'à partir de correspondances visuelles au sein d'images prises à différents instants et positions dans l'espace. Ce type d'estimation de pose n'est en outre possible qu'à un facteur d'échelle près, sans connaissance à priori des dimensions

réelles de la scène à reconstruire. Trois catégories de méthodes ont été développées dans la littérature. Les méthodes de type points d'intérêt, celles de type apparence et les méthodes hybrides. Les méthodes de type points d'intérêt sont basées sur la détection et l'appariement de points, celles de type apparence utilisent l'intensité des pixels de tout ou partie de l'image alors que les méthodes hybrides sont une combinaison des deux. La première méthode d'odométrie visuelle monoculaire à grande échelle utilisant des points d'intérêt a été proposée par Nistér *et al.* dans [128], dont la partie stéréo est présentée dans le paragraphe précédent. L'aspect monoculaire est quant à lui divisé en deux étapes, dont la première consiste à utiliser les propriétés de la géométrie épipolaire conjointement à l'algorithme RANSAC afin d'estimer le déplacement initial de la caméra et la structure géométrique de la scène par triangulation. La deuxième étape repose ensuite, comme dans le cas stéréo, sur une technique PnP pour l'estimation des poses suivantes à partir des points préalablement reconstruits et de leurs correspondances 2D. L'une des nouveautés majeures introduites par les auteurs se situe dans leur méthode de calcul de la matrice essentielle, définie dans la section 1.1.4 de ce manuscrit, couplée à l'algorithme RANSAC afin d'estimer de manière robuste et efficiente le déplacement de la caméra lors de la première étape [127]. Cette technique d'estimation de pose a été très largement utilisée par la suite [25, 91, 120]. Corke *et al.* [25] ont notamment proposé une méthode d'odométrie visuelle pour laquelle les correspondances de points ont été réalisées par le calcul du flot optique épars sur des images omnidirectionnelles obtenues grâce à une caméra catadioptrique. Par la suite, Lhuillier [91] et Mouragnon *et al.* [120] ont introduit un algorithme d'optimisation par ajustement de faisceaux local dans un système de SLAM, n'affectant que les dernières images de la séquence et réduisant ainsi le nombre de poses et de points 3D à optimiser afin de proposer un algorithme temps réel vidéo. Parmi les méthodes récentes de SLAM monoculaire de type points d'intérêt, on peut notamment citer les travaux de Davison *et al.* [28], qui ont utilisé un filtre de Kalman étendu afin d'optimiser poses et points 3D, ceux de Klein et Murray [78], qui ont séparé dans leur méthode PTAM les processus de suivi et de reconstruction afin d'obtenir un algorithme temps-réel, ou encore ceux de Mur-Artal *et al.* [121, 122], avec leur système ORB-SLAM très robuste, précis et fonctionnant en temps réel, reprenant la plupart des techniques les plus avancées dans le domaine. Parmi les méthodes de type apparence, on peut notamment citer les travaux de Goecke *et al.* [56], qui ont utilisé la transformation de Fourrier-Mellin afin de recaler les images perspectives de la surface du sol autour de la voiture, ceux de Milford and Wyeth [117, 116], qui ont présenté une méthode permettant d'extraire de manière approximative les vitesses de rotation et de translation d'une caméra montée sur véhicule grâce à un suivi de type *Template Matching* du centre de l'image, intégrée à leur système de SLAM bio-inspiré RatSLAM [118], ou encore la méthode hybride proposée par Scaramuzza *et al.* [149],

plus robuste aux occultations partielles, en couplant une méthode de type apparence afin d'estimer la rotation de la caméra à une méthode plus conventionnelle de type points d'intérêt sur la surface du sol, permettant l'estimation de la translation ainsi que le facteur d'échelle de la reconstruction. Du côté des méthodes denses et de manière similaire aux méthodes stéréo, le calcul du flot optique a également été utilisé pour les méthodes monoculaires, notamment la méthode proposée par Valgaerts *et al.* [169], reposant sur un modèle variationnel permettant de calculer simultanément le flot optique de deux images ainsi que la matrice fondamentale associée par minimisation d'une unique fonctionnelle d'énergie.

Méthodes multi-caméras. Les systèmes multi-caméras permettent la reconstruction d'une scène à partir de différentes vues prises simultanément. Lorsque les champs de vue de chaque caméra sont recouvrants, on peut considérer ces systèmes comme de multiples paires stéréo permettant de reconstruire la scène grâce aux techniques classiques évoquées précédemment. À l'inverse, lorsqu'il n'y a pas de champ recouvrant entre les vues de chaque caméra, plusieurs approches pour la reconstruction sont envisageables. L'une de ces méthodes consiste à reconstruire séparément les scènes associées à chacune des caméras, puis à fusionner leurs structures géométriques respectives, pour enfin terminer par une optimisation de type ajustement de faisceaux des poses et points 3D, comme proposé par Carrera dans [22] ou encore Lébraly dans [86]. Une autre approche consiste à reconstruire directement la scène à partir de toutes les vues de manière simultanée, en utilisant la contrainte épipolaire généralisée (*Generalized Epipolar Constraint* ou GEC, en anglais) introduite par Pless dans son article *Using Many Cameras as One* [132]. Cette contrainte est exprimée grâce à la matrice essentielle généralisée (*Generalized Essential matrix*, en anglais) qui est une matrice de taille six par six représentant les six degrés de libertés du système multi-caméras. Dans l'article original, l'idée principale réside dans l'utilisation d'un modèle de caméra généralisé (*Generalized Camera Model* ou GCM, en anglais) afin de remplacer les pixels des différentes images par un ensemble de rayons, ou raxels, représentés chacun par un vecteur de coordonnées plückeriennes. Ce modèle implique en théorie l'utilisation de dix-sept rayons afin de retrouver la pose du système multi-caméras dans son ensemble, ce qui a été démontré impossible en pratique par Li *et al.* [94], qui ont néanmoins proposé un algorithme linéaire pour résoudre les ambiguïtés posées par la précédente méthode. D'autres approches utilisant des algorithmes d'optimisation non-linéaires ont également été proposées, notamment dans [92, 86].

Réduction des degrés de libertés grâce aux contraintes de mouvement. Alors que la plupart des méthodes d'odométrie visuelle permettent l'estimation de la pose d'une caméra sur six degrés de liberté, d'autres méthodes ont été spécifiquement déve-

loppées pour estimer la pose de caméras montées sur des robots possédant leurs propres contraintes de mouvements. Cela permet de réduire les degrés de liberté de la pose à estimer et engendre généralement une diminution du temps de calcul ainsi qu'une augmentation de la précision des estimations. Le calcul d'homographies afin d'estimer le mouvement propre du véhicule sur une surface plane a notamment été utilisé dans [96, 75]. Une autre méthode a été proposée par Scaramuzza *et al.*, qui ont utilisé les contraintes de mouvement non-holonomes des véhicules traditionnels afin de réduire le test RANSAC de rejet d'outliers à un seul point [148, 145], quand la méthode de Nistér en demande au minimum cinq, puis ont montré qu'il était possible d'obtenir le facteur d'échelle de la reconstruction lorsque le véhicule effectue au moins un virage [147]. Plus récemment, Lee *et al.* [87], ont réutilisé ces contraintes afin de calculer la matrice essentielle généralisée d'un système multi-caméras grâce à deux correspondances de points seulement.

Méthodes directes. Les travaux présentés jusqu'ici entrent dans la catégorie des méthodes indirectes, c'est à dire qu'elles intègrent une étape de transformation des pixels de l'image vers une représentation résolvant déjà une partie du problème géométrique sous-jacent. Ces représentations sont généralement calculées sous la forme de points d'intérêt, raxels, flot optique ou formes paramétriques de primitives géométriques (segments linéaires ou courbes, par exemple), intégrées à leur tour à un modèle probabiliste ou non linéaire afin de minimiser une erreur géométrique. À l'inverse, les méthodes dites directes, apparues beaucoup plus récemment car pour la plupart dépendantes du gain de performances offert par le calcul sur GPU, utilisent directement les valeurs d'intensité des pixels de l'image obtenues sur une période donnée et les intègrent à des modèles probabilistes qui s'attachent à minimiser une erreur photométrique. Parmi ces méthodes directes, nous pouvons citer notamment les travaux de Stühmer *et al.* [159], qui se sont inspirés de l'approche variationnelle introduite dans [192], permettant l'extraction en temps réel de cartes de profondeurs denses par le calcul de flot optique, la méthode DTAM, inspirée des travaux de Stühmer *et al.* et introduite par Newcombe *et al.* [125], permettant d'obtenir le modèle 3D dense de la scène par minimisation d'une unique fonctionnelle d'énergie conjointement aux poses des caméras par recalage d'image global, la méthode REMODE introduite par Pizzoli *et al.* [131], basée sur les travaux des mêmes auteurs [48] et utilisant une approche probabiliste bayésienne [181] afin de créer une carte de profondeur dense couplée à une méthode de lissage prenant en compte l'incertitude de la mesure, ou encore celle de Engel *et al.* [36, 34], semi-dense et temps réel par le calcul du flot optique des pixels à fortes variations de contraste, adaptée plus tard au cas stéréo [35] et au cas épars [33].

Calibration. La calibration ou l'étalonnage d'une caméra permet une reconstruction euclidienne de la scène observée. Cette procédure se compose en général d'une étape de calibration intrinsèque et, dans le cas de systèmes multi-caméras par exemple, d'une étape de calibration extrinsèque. La calibration intrinsèque d'une caméra consiste à retrouver de manière totalement ou semi-automatique les paramètres internes de la caméra, c'est à dire la longueur focale, le point principal, un paramètre de non orthogonalité des cellules du capteur ainsi que les coefficients de distorsion permettant la rectification géométrique des images brutes. La calibration extrinsèque, en revanche, consiste à retrouver les transformations géométriques permettant de passer du repère de la caméra considérée à un autre repère propre au robot, celui d'une autre caméra par exemple, dans le cas d'un système multi-caméras. Parmi les méthodes de calibration intrinsèques existantes, la plupart utilisent des mires planes sur lesquelles un motif aux caractéristiques connues permet l'estimation des paramètres. C'est le cas notamment des méthodes populaires proposées dans la librairie OpenCV [18] ou la toolbox Matlab [15]. Ces techniques utilisent des mires à damiers facilement reconnaissables automatiquement, alors que d'autres utilisent des mires noires avec pastilles réfléchissantes [84] ou encore des cibles à cercles concentriques [85], et reposent pour la plupart sur l'utilisation d'un algorithme de type transformation linéaire directe pour le calcul des paramètres, tel que celui proposé par Zhang [197]. Les méthodes de calibration extrinsèques reposent quant à elles sur les techniques de reconstruction classiques évoquées précédemment, lorsque les champs des caméras sont recouvrants [103]. A l'inverse, lorsque les champs sont non-recouvrants, plusieurs techniques ont été envisagées, notamment la création de scènes panoramiques dans lesquelles les points d'intérêt observables simultanément par plusieurs caméras ont été préalablement définis dans un même repère [130], l'observation successive de points d'intérêt par des caméras différentes, où la vitesse du système doit être préalablement connue [83], ou encore l'estimation des paramètres extrinsèques grâce aux mouvements et à l'exploitation de la contrainte de rigidité du système multi-caméras [37, 86].

1.1.1.5 Structure de la section

En premier lieu seront définis dans la section 1.1.2 les modèles de caméra permettant de simuler le processus physique de formation d'une image ainsi que les techniques de correction des distorsions géométriques générées par ce processus. Seront ensuite abordés la détection, l'appariement et le suivi de points d'intérêt dans la section 1.1.3. Dans les sections 1.1.4 et 1.1.5, les principes de la géométrie épipolaire, de l'estimation de la pose d'une caméra et de la triangulation d'un point à partir de plusieurs vues seront exposés, alors que la section 1.1.6 se concentrera plus particulièrement sur l'estimation de la pose d'une caméra à partir de points 3D déjà reconstruits. Enfin, dans la section 1.1.7 seront discutées les techniques d'optimisation conjointe de poses et points 3D.

1.1.2 Modèles intrinsèques de caméra et correction des distorsions

Quand la plupart des caméras numériques aujourd'hui se composent des deux éléments principaux que sont le système optique (l'objectif, comprenant le plus souvent plusieurs lentilles) et le capteur photosensible (typiquement basé sur la technologie CCD ou CMOS), auxquels viennent s'ajouter beaucoup d'éléments mécaniques (obturateur, diaphragme, prisme pour les appareils reflex, etc.) et électroniques (puces de traitement d'image spécialisées, capteurs de luminosité, stabilisateur d'image, etc.), les modèles intrinsèques de caméra utilisés en vision par ordinateur se révèlent généralement plus synthétiques. Les deux fonctions principales offertes par ces modèles géométriques sont la projection d'entités en trois dimensions (points, droites, etc.) sur le plan image ainsi que la projection inverse, c'est à dire le lancer de rayons partant du plan image vers l'espace 3D. Beaucoup de modèles sont aujourd'hui utilisés en vision par ordinateur, dont la plupart sont présentés dans un état de l'art très complet réalisé par Sturm *et al.* [161]. Brièvement, trois catégories de modèles de caméra existent. Les modèles globaux en premier lieu, pour lesquels chaque paramètre parmi l'ensemble des paramètres du modèle influence la fonction de projection sur tout le champ de vue de la caméra. Parmi ces modèles se trouve notamment le modèle sténopé, très largement utilisé, ainsi que d'autres modèles comme le modèle unifié. Viennent ensuite les modèles locaux, définis là encore par un ensemble de paramètres, à la différence que ceux-ci n'influent chacun que de manière locale sur la fonction de projection et n'affectent donc qu'une partie du champ de vue. Enfin, les modèles discrets sont définis par des ensembles de paramètres associés chacun à un unique pixel de l'image. Ces modèles sont généralement complétés par des fonctions d'interpolation qui permettent de travailler sur plus de points que la résolution de l'image ne le permet, les rapprochant ainsi des modèles locaux. Parmi les modèles discrets, nous pouvons citer notamment ceux basés sur les raxels, qui sont des rayons de projection associés un à un à chaque pixel. Dans le cas de systèmes multi-caméras correctement calibrés, l'utilisation de raxels offre notamment la possibilité de travailler sur une image générique unique et non sur les images de chaque caméra individuellement [132]. Dans cette section, nous décrirons deux des modèles les plus utilisés en vision par ordinateur, car adaptés à la configuration expérimentale définie en introduction de ce manuscrit, c'est à dire le modèle sténopé (1.1.2.1) et le modèle unifié (1.1.2.5), ainsi que les techniques de correction de distorsion leur étant associées (1.1.2.4 et 1.1.2.8).

1.1.2.1 Modèle sténopé

Le modèle sténopé simule le processus physique de la formation d'une image par projection perspective de la scène observée sur un plan. C'est une projection centrale dans laquelle chaque rayon optique partant d'un point P de la scène en trois dimensions passe

par le centre optique de la caméra (ou centre de projection) et intersecte ensuite le plan image situé à distance f du centre optique en un point p. Les concepts de centre optique, plan image et distance f se retrouvent dans le monde physique et correspondent respectivement à l'objectif, au capteur photosensible et à la focale de la caméra. Une illustration de chambre noire, ou *camera obscura*, ancêtre des premiers appareils photographiques et première application physique du modèle sténopé, est visible figure 1.3.

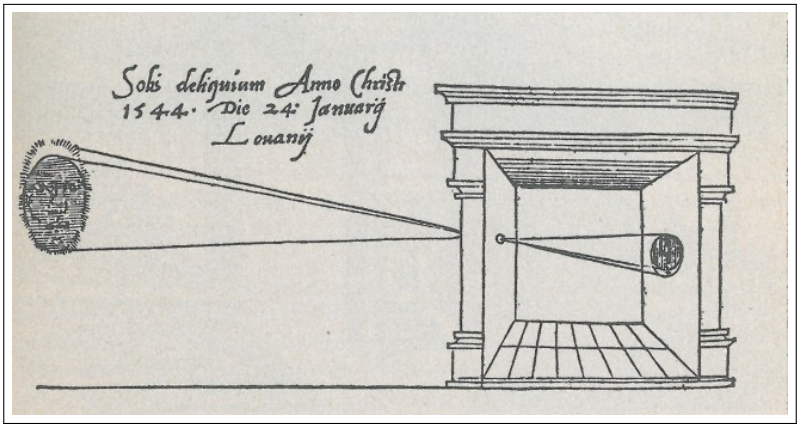

FIGURE 1.3 – Première illustration de camera obscura, extraite du livre *De Radio Astronomica et Geometrica*, écrit par Gemma Frisius en 1545.

Le modèle présenté dans cette section est illustré figure 1.4. Dans ce modèle, l'origine du repère caméra $(c; \mathbf{X}_c, \mathbf{Y}_c, \mathbf{Z}_c)$ se situe au niveau du centre optique, alors que les axes de ce repère sont définis tels que $\mathbf{X}_c$ pointe vers la droite, $\mathbf{Y}_c$ pointe vers le bas et $\mathbf{Z}_c$ pointe vers l'avant, dans la direction de l'axe optique.

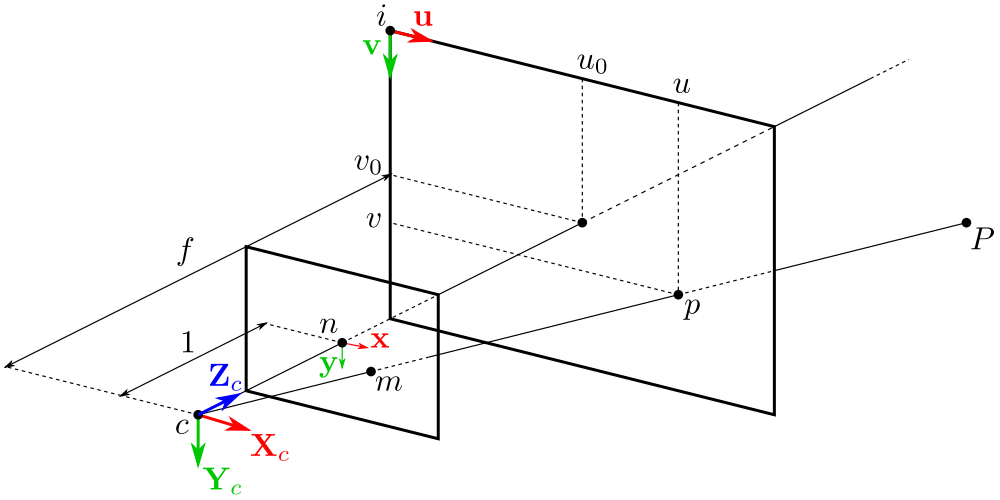

FIGURE 1.4 – Modèle sténopé.

Le plan image, à l'inverse d'un appareil photo classique, est placé devant le centre

optique à distance $Z_c = f$, alors que l'origine de son repère $(i; \mathbf{u}, \mathbf{v})$ se situe au niveau du coin supérieur gauche de l'image, avec $\mathbf{u}$ pointant vers la droite et $\mathbf{v}$ vers le bas. Ce plan forme la base des points 2D $p = (u, v)$ qui correspondent aux points de l'image et dont les coordonnées sont exprimées en pixels. On note $\bar{p} = (u, v, 1)^T \equiv (X_i, Y_i, Z_i)^T$ les coordonnées homogènes d'un point 2D dans le repère image.

Nous définissons un deuxième plan, le plan normalisé, situé à distance unitaire $Z_c = 1$ du centre optique et dont l'origine du repère $(n; \mathbf{x}, \mathbf{y})$ se situe à l'intersection de l'axe optique, avec $\mathbf{x}$ pointant vers la droite et $\mathbf{y}$ vers le bas. Ce plan forme la base des points 2D normalisés $m = (x, y)$. On note $\bar{m} = (x, y, 1)^T \equiv (X_c, Y_c, Z_c)^T$ les coordonnées homogènes d'un point 2D normalisé dans le repère caméra.

Enfin, nous définissons un quatrième repère, le repère monde $(w; \mathbf{X}_w, \mathbf{Y}_w, \mathbf{Z}_w)$ dans lequel sont exprimés la position et l'orientation du repère caméra c ainsi que les points 3D $P_w = (X_w, Y_w, Z_w)^T$ dont les coordonnées homogènes sont notées $\bar{P}_w = (X_w, Y_w, Z_w, 1)^T$.

1.1.2.2 Projection d'un point 3D sur le plan image du modèle sténopé

Comme dit précédemment, la première fonction assurée par un modèle de caméra doit être de permettre la projection d'entités de l'espace en trois dimensions sur le plan image de la caméra. Le cas qui nous intéresse plus précisément ici est la projection d'un point 3D.

Soit $\bar{P}_w = (X_w, Y_w, Z_w, 1)^T$ les coordonnées homogènes d'un point 3D de l'espace exprimé dans le repère monde. La projection du point $\bar{P}_w$ en un point 2D p sur le plan image se décompose en quatre étapes explicitant la transformation projective T_i^w suivante :

$$\bar{P}_w = \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix} \xmapsto{Projection} p = \begin{pmatrix} u \\ v \end{pmatrix} \tag{1.1}$$

$$Projection \rightarrow p = T_i^w \bar{P}_w \tag{1.2}$$

Étape 1. La première étape consiste à effectuer un changement de base afin d'obtenir les coordonnées homogènes du point 3D $\bar{P}_c$ dans le repère caméra à partir de ses coordonnées homogènes $\bar{P}_w$ exprimées dans le repère monde.

Transformations homogènes et changement de base

Soit $\mathbf{x} \in \mathbb{R}^n$ un vecteur uni-colonne de dimension n. Soit $\bar{\mathbf{x}} = (\mathbf{x}, h)^T \in \mathbb{P}^n$ les coordonnées homogènes de ce vecteur. On définit $\mathcal{B}_1$ et $\mathcal{B}_2$ deux bases de $\mathbb{R}^n$. Les vecteurs $\bar{\mathbf{x}}_1$ et $\bar{\mathbf{x}}_2$ expriment le vecteur $\bar{\mathbf{x}}$ en coordonnées homogènes dans les bases $\mathcal{B}_1$ et $\mathcal{B}_2$, respectivement.

Transformation homogène. La matrice de passage homogène ou transformation homogène T_1^2 de la base $\mathcal{B}_1$ vers la base $\mathcal{B}_2$ est une matrice de taille $(n+1) \times (n+1)$ vérifiant :

$$\bar{\mathbf{x}}_1 = T_1^2 \bar{\mathbf{x}}_2 \tag{1.3}$$

Transformation euclidienne. On définit R la matrice de rotation entre les bases $\mathcal{B}_1$ et $\mathcal{B}_2$ et $\mathbf{t}$ le vecteur de translation entre les bases $\mathcal{B}_1$ et $\mathcal{B}_2$. Une transformation euclidienne entre les deux bases se présente sous la forme :

$$\bar{\mathbf{x}}_1 = \begin{bmatrix} R & \mathbf{t} \\ 0_{1\times n} & 1 \end{bmatrix} \bar{\mathbf{x}}_2 \tag{1.4}$$

On définit R_w^c la matrice de rotation du repère caméra exprimée dans le repère monde et $\mathbf{t}_w^c$ le vecteur translation du repère caméra exprimé dans le repère monde. Ces deux valeurs représentent les paramètres extérieurs de la caméra, c'est à dire sa position et son orientation dans le repère monde. Dans le cas qui nous intéresse ici, le changement de repère du point $\bar{P}$ s'obtient grâce à la transformation euclidienne T_c^w suivante :

$$\bar{P}_w = \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix} \overset{1.}{\longmapsto} \bar{P}_c = \begin{pmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{pmatrix} \tag{1.5}$$

$$1. \rightarrow \bar{P}_c = T_c^w \bar{P}_w \tag{1.6}$$

L'expression de cette transformation en fonction de R_w^c et $\mathbf{t}_w^c$ est de la forme :

$$T_c^w = T_w^{c-1} = \begin{bmatrix} R_w^c & \mathbf{t}_w^c \\ 0_{1\times 3} & 1 \end{bmatrix}^{-1} = \begin{bmatrix} R_w^{cT} & -R_w^{cT}\mathbf{t}_w^c \\ 0_{1\times 3} & 1 \end{bmatrix} \tag{1.7}$$

L'équation (1.6) se réécrit donc :

$$\bar{P}_c = \begin{pmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{pmatrix} = \begin{bmatrix} R_w^{cT} & -R_w^{cT}\mathbf{t}_w^c \\ 0_{1\times 3} & 1 \end{bmatrix} \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix} \tag{1.8}$$

Pour des raisons de lisibilité, les paramètres extrinsèques de pose de la caméra, ou plus simplement la pose de la caméra, notée C et composée d'une matrice de rotation R et d'un vecteur translation $\mathbf{t}$, sont introduits :

$$R = R_w^{cT} \qquad \mathbf{t} = -R_w^{cT}\mathbf{t}_w^c \qquad C = \begin{bmatrix} R & \mathbf{t} \\ 0_{1\times 3} & 1 \end{bmatrix} \tag{1.9}$$

Tenant compte des notations introduites dans les équations (1.9), l'équation (1.6) s'écrit enfin :

$$\bar{P}_c = \begin{pmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{pmatrix} = C\bar{P}_w = \begin{bmatrix} R & \mathbf{t} \\ 0_{1\times 3} & 1 \end{bmatrix} \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix} \tag{1.10}$$

Étape 2. La deuxième étape consiste ensuite à projeter le point $\bar{P}_c$ en un point 2D $\bar{m}$ sur le plan normalisé :

$$\bar{P}_c = \begin{pmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{pmatrix} \overset{2.}{\mapsto} \bar{m} = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \equiv \begin{pmatrix} X_c \\ Y_c \\ Z_c \end{pmatrix} \tag{1.11}$$

$$2. \rightarrow \bar{m} = [I_3|0_{3\times 1}]\,\bar{P}_c \tag{1.12}$$

Étape 3. La troisième étape permet d'obtenir les coordonnées homogènes du point $\bar{p}$ dans le repère image à partir du point normalisé $\bar{m}$. Ces coordonnées sont obtenues par transformation affine avec la matrice K, que l'on appelle aussi matrice de calibration intrinsèque de la caméra :

$$\bar{m} = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \overset{3.}{\mapsto} \bar{p} = \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \equiv \begin{pmatrix} X_i \\ Y_i \\ Z_i \end{pmatrix} \tag{1.13}$$

$$3. \rightarrow \bar{p} = K\bar{m} \tag{1.14}$$

La matrice K se compose de cinq paramètres, comprenant la distance focale de la caméra f, les dimensions élémentaires d_x et d_y de la matrice CCD (les dimensions des photodiodes du capteur de la caméra), ainsi que les coordonnées (u_0, v_0) du point principal de la caméra (le point d'intersection entre l'axe optique et le plan image). Enfin, pour des raisons de lisibilité, on pose $f_x = (f/d_x)$ et $f_y = (f/d_y)$:

$$K = \begin{bmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{1.15}$$

On peut maintenant réécrire l'équation (1.14) :

$$\bar{p} = \begin{pmatrix} X_i \\ Y_i \\ Z_i \end{pmatrix} = \begin{bmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} X_c \\ Y_c \\ Z_c \end{pmatrix} = \begin{pmatrix} f_x X_c + u_0 Z_c \\ f_y Y_c + v_0 Z_c \\ Z_c \end{pmatrix} \tag{1.16}$$

Étape 4. La dernière étape permet enfin de retrouver les coordonnées cartésiennes de $p = (u, v)$ à partir de ses coordonnées homogènes $\bar{p} = (X_i, Y_i, Z_i)$.

$$\bar{p} = \begin{pmatrix} X_i \\ Y_i \\ Z_i \end{pmatrix} \overset{4.}{\mapsto} p = \begin{pmatrix} u \\ v \end{pmatrix} \tag{1.17}$$

Pour cela, on introduit la fonction $\pi : \mathbb{P}^2 \to \mathbb{R}^2$:

$$\bar{p} = \begin{pmatrix} X_i \\ Y_i \\ Z_i \end{pmatrix} \mapsto \pi(\bar{p}) = \frac{1}{Z_i} \begin{pmatrix} X_i \\ Y_i \end{pmatrix} = \begin{pmatrix} u \\ v \end{pmatrix} = p \tag{1.18}$$

En appliquant π à l'équation (1.16), on trouve :

$$4. \to p = \begin{pmatrix} u \\ v \end{pmatrix} = \pi(\bar{p}) = \frac{1}{Z_c} \begin{pmatrix} f_x X_c + u_0 Z_c \\ f_y Y_c + v_0 Z_c \end{pmatrix} = \begin{pmatrix} f_x \frac{X_c}{Z_c} + u_0 \\ f_y \frac{Y_c}{Z_c} + v_0 \end{pmatrix} = \begin{pmatrix} f_x x + u_0 \\ f_y y + v_0 \end{pmatrix} \tag{1.19}$$

Matrice de projection d'une caméra avec le modèle sténopé

La concaténation des quatre étapes décrites ci-avant permet enfin de réécrire la transformation projective T_i^w, appelée matrice de projection de la caméra :

$$p = T_i^w \bar{P}_w = \pi \left(K \left[I_3 | 0_{3\times 1} \right] C \bar{P}_w \right) \tag{1.20}$$

Pour des raisons de lisibilité, cette matrice de projection T_i^w est ensuite renommée $[P]$ dans la suite du manuscrit.

1.1.2.3 Lancer de rayon dans le modèle sténopé

Le lancer de rayon est l'étape permettant de retrouver la droite passant par un point de l'image p et le point de l'espace 3D P lui étant associé. Dans le modèle sténopé, cette étape est directe, le rayon étant la droite ayant pour vecteur directeur $\mathbf{r} = (c, p)^T$, avec c le centre optique de la caméra.

1.1.2.4 Correction des distorsions géométriques dans le modèle sténopé

Plusieurs types d'aberrations optiques sont engendrées par l'objectif de la caméra lors de la prise de vue (monochromatiques et chromatiques principalement). Ces aberrations ont toutes un impact sur les images brutes générées par la caméra. Dans le cadre de la géométrie multi-vues qui nous intéresse dans ce manuscrit, certaines sont plus gênantes que d'autres, en particulier les distorsions géométriques de l'image qui dévient de la projection rectilinéaire assumée par le modèle sténopé présenté ci-avant. Ces distorsions géométriques doivent être corrigées en ajoutant au modèle sténopé un autre modèle de correction des distorsions. En pratique, même s'il existe plusieurs types de distorsions géométriques (radiales et tangentielles) [185], seules les distorsions radiales sont corrigées dans la vaste majorité des travaux de la littérature car ce sont généralement les plus gênantes. Dans la figure 1.5, on peut observer la même image, avant et après correction des distorsions. Dans le modèle sténopé, ces distorsions radiales peuvent être modélisées grâce à un polynôme. Deux modèles de correction sont aujourd'hui utilisés dans la littérature pour le modèle sténopé. Le modèle direct permet de passer des coordonnées non-distordues m aux coordonnées distordues m_d [15], alors que le modèle indirect, qualifié de standard par [161], permet de passer des coordonnées distordues m_d aux coordonnées non-distordues m.

Modèle direct. Le modèle direct permet de générer les distorsions induites par le modèle physique en projetant le point 3D P dans le plan image en un point distordu p_d.

FIGURE 1.5 – Correction des distorsions. En haut, la vue distordue, en bas, la vue corrigée. La distorsion en barillet est visible sur le poteau à droite de la première vue.

C'est un modèle utilisé notamment lorsque l'on veut simuler plus fidèlement les caractéristiques physiques d'un objectif au sein d'une caméra virtuelle. Pour cela, on rajoute une étape $Dist$ entre les étapes de projection 2. et 3. du modèle sténopé standard. Cette étape est une transformation polynomiale du point normalisé $\bar{m}$ en un point normalisé distordu $\bar{m}_d$, que l'on utilise ensuite comme dans le modèle sténopé standard pour obtenir le point image distordu p_d :

$$\bar{P}_w \overset{1.}{\mapsto} \bar{P}_c \overset{2.}{\mapsto} \bar{m} \overset{Dist}{\longmapsto} \bar{m}_d \overset{3.}{\mapsto} \bar{p}_d \overset{4.}{\mapsto} p_d \tag{1.21}$$

On note D le polynôme de degré n ayant pour coefficients a_k, avec $k \in 1 \dots n$, permettant de passer du point normalisé $\bar{m}$ au point normalisé distordu $\bar{m}_d$:

$$Dist \rightarrow \bar{m}_d = (1 + D(r))\bar{m} \tag{1.22}$$

Avec $r = \sqrt{x^2 + y^2} = \|\bar{m}\|$ la distance radiale entre le point $\bar{m}$ et l'axe optique, et $D(r)$ le polynôme de la forme :

$$D(r) = \sum_{k=1}^{n} a_k r^{2k} \tag{1.23}$$

Modèle indirect. Le modèle indirect, à l'inverse du modèle direct, permet la correction des distorsions optiques présentes sur les images brutes de la caméra afin de se ramener au cas idéal de projection rectilinéaire des points de l'espace sur le plan image. Son principe de fonctionnement est malgré tout très similaire au modèle direct, car il effectue l'opération inverse de $Dist$, notée $Undist$ ici, et permettant de calculer les coordonnées du point non-distordu $\bar{m}$ à partir de celles du point distordu $\bar{m}_d$. Pour retrouver le point image non-distordu $\bar{p}$ à partir du point image distordu $\bar{p}_d$, il est donc nécessaire d'effectuer les opérations suivantes :

$$\bar{p}_d \xmapsto{K^{-1}} \bar{m}_d \xmapsto{Undist} \bar{m} \xmapsto{K} \bar{p} \tag{1.24}$$

On note I le polynôme de degré n ayant pour coefficients b_k, avec $k \in 1 \dots n$, permettant de passer du point normalisé distordu $\bar{m}_d$ au point normalisé non-distordu $\bar{m}$:

$$Undist \rightarrow \bar{m} = (1 + I(r_d))\bar{m}_d \tag{1.25}$$

Avec $r_d = \sqrt{x^2 + y^2} = \|\bar{m}_d\|$ la distance radiale entre le point distordu $\bar{m}_d$ et l'axe optique, et $I(r_d)$ le polynôme de la forme :

$$I(r_d) = \sum_{k=1}^{n} b_k r_d^{2k} \tag{1.26}$$

En pratique, pour les modèles direct et indirect, les valeurs de n utilisées dans les polynômes D et I sont comprises entre $3 \leq n \leq 6$.

1.1.2.5 Modèle unifié

Le modèle unifié, initialement présenté par Geyer *et al.* [55], a été développé afin de modéliser des caméras catadioptriques ou à très grand angle de vue, comme les caméras *fisheye*. La différence avec le modèle sténopé classique consiste en l'introduction d'une demi-sphère unité de centre s situé à distance $Z_c = \xi \geq 0$ du centre optique. Les équations de projection et de lancer de rayon sont différentes dans ce modèle qui comporte certains avantages, en particulier lors de la correction des distorsions. Brièvement, dans le modèle unifié, chaque point 3D P de la scène est d'abord projeté sur la demi-sphère unité en un point 3D Q, avant d'être à nouveau projeté sur le plan normalisé dans le repère caméra en un point 2D m et enfin sur le plan image en un point 2D p. Une illustration du modèle unifié est visible figure 1.6.

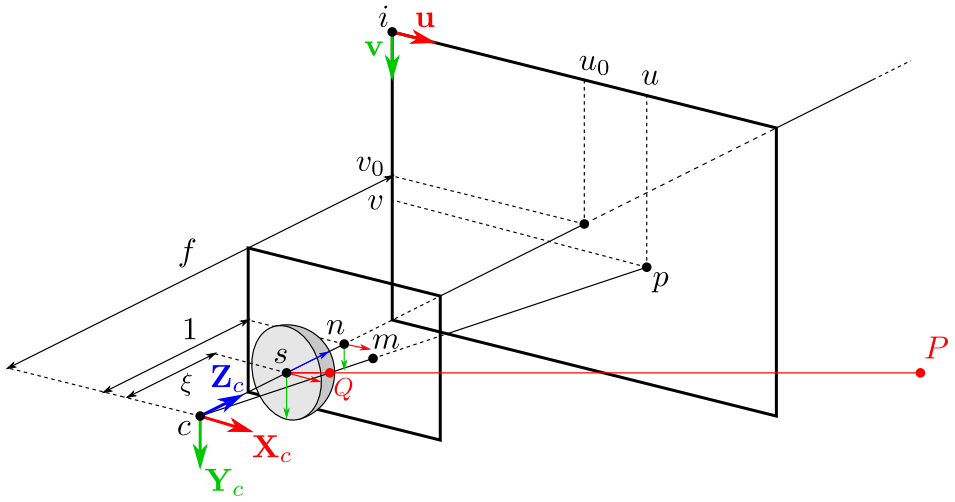

FIGURE 1.6 – Modèle unifié.

1.1.2.6 Projection d'un point 3D sur le plan image du modèle unifié

La projection d'un point 3D P_w exprimé dans le repère monde en un point p dans le plan image du modèle unifié s'effectue en trois étapes particulières à ce modèle, avant de se ramener aux étapes de projection 3. et 4. du modèle sténopé.

Étape 1. La première étape consiste à effectuer un changement de repère du point 3D homogène $\bar{P}_w$ dont les coordonnées sont exprimées dans le repère monde vers le point 3D homogène $\bar{P}_s$ aux coordonnées exprimées dans le repère s de la demi-sphère. La transformation utilisée ici est de la forme :

$$\bar{P}_w = \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix} \overset{1.}{\mapsto} \bar{P}_s = \begin{pmatrix} X_s \\ Y_s \\ Z_s \\ 1 \end{pmatrix} \tag{1.27}$$

$$1. \rightarrow \bar{P}_s = \begin{pmatrix} X_s \\ Y_s \\ Z_s \\ 1 \end{pmatrix} = T_s^c T_c^w \bar{P}_w = \begin{bmatrix} I_3 & -\mathbf{t}_\xi \\ 0_{1\times 3} & 1 \end{bmatrix} C \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix} \tag{1.28}$$

Avec $-\mathbf{t}_\xi = (0, 0, -\xi)^T$ le vecteur translation permettant de passer du repère caméra au repère de la demi-sphère et C la matrice de passage permettant de passer du repère monde au repère caméra.

Étape 2. La deuxième étape effectue la projection du point 3D P_s en un point 3D Q_s sur la demi-sphère :

$$P_s = \begin{pmatrix} X_s \\ Y_s \\ Z_s \end{pmatrix} \overset{2.}{\mapsto} Q_s \tag{1.29}$$

$$2. \rightarrow Q_s = \frac{P_s}{\rho} = \frac{1}{\rho} \begin{pmatrix} X_s \\ Y_s \\ Z_s \end{pmatrix} \tag{1.30}$$

Avec $\rho = \sqrt{X_s^2 + Y_s^2 + Z_s^2}$.

Étape 3. Dans la troisième étape, on projette le point Q_s de la demi-sphère sur le plan normalisé dans le repère caméra en un point $\bar{m}$:

$$Q_s = \frac{1}{\rho} \begin{pmatrix} X_s \\ Y_s \\ Z_s \end{pmatrix} \overset{3.}{\mapsto} \bar{m} = \begin{pmatrix} X_c \\ Y_c \\ 1 \end{pmatrix} \tag{1.31}$$

$$3. \rightarrow \bar{m} = \begin{pmatrix} X_c \\ Y_c \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{X_s}{Z_s + \rho\xi} \\ \frac{Y_s}{Z_s + \rho\xi} \\ 1 \end{pmatrix} \equiv \begin{pmatrix} \frac{X_s}{\rho} \\ \frac{Y_s}{\rho} \\ \frac{Z_s}{\rho} + \xi \end{pmatrix} \tag{1.32}$$

Étapes suivantes. Les étapes de projection suivantes sont les mêmes que les étapes 3. et 4. du modèle sténopé, c'est à dire permettant de passer du point normalisé $\bar{m}$ au point image p :

$$\bar{m} \overset{K}{\mapsto} \bar{p} \overset{\pi}{\mapsto} p \tag{1.33}$$

1.1.2.7 Lancer de rayon dans le modèle unifié

Chaque rayon partant de la caméra vers un point 3D de l'espace passe par le centre de la sphère s dans le modèle unifié. Cette étape consiste à obtenir les coordonnées du point Q_s de la sphère correspondant à un point p du plan image afin de retrouver le rayon QP ayant pour vecteur directeur $\mathbf{r} = (s, Q_s)^T$.

Étape 1. La première étape consiste à obtenir les coordonnées du point normalisé homogène $\bar{m}$ à partir de $\bar{p}$:

$$\bar{m} = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = K^{-1}\bar{p} \tag{1.34}$$

Étape 2. On calcule enfin les coordonnées du point Q_s sur la demi-sphère à partir de $\bar{m}$:

$$Q_s = \eta \begin{pmatrix} x \\ y \\ 1 - \eta^{-1}\xi \end{pmatrix} \tag{1.35}$$

Avec $\eta = \frac{\xi + \sqrt{1+(x^2+y^2)(1-\xi^2)}}{x^2+y^2+1}$

1.1.2.8 Correction des distorsions dans le modèle unifié

Le modèle unifié, contrairement au modèle sténopé classique, intègre directement la modélisation des distorsions. En effet, il a été démontré, notamment dans [27], que la non-linéarité induite par la projection sur la demi-sphère unité permet une modélisation suffisante des distorsions. Par conséquent, c'est le paramètre ξ, c'est à dire la distance à laquelle se trouve la demi-sphère unité du centre optique, qui permet de modéliser correctement les distorsions. D'autre part, le modèle unifié est par définition inversible. Les coordonnées d'un point non-distordu s'obtiennent donc directement à partir de celles d'un point distordu dans ce modèle, et inversement. Dans le cas de la correction d'images distordues, il est possible de définir un autre plan image, non-distordu et parallèle au plan focal. Pour cela, on utilise une nouvelle matrice de calibration intrinsèque K', dont on peut choisir le paramètre f' définissant la taille de l'image corrigée :

$$K' = \begin{bmatrix} f' & 0 & u_0 \\ 0 & f' & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{1.36}$$

Pour passer d'un point distordu $\bar{p}$ à un point non-distordu $\bar{p}'_s$, on utilisera la relation :

$$\bar{p} \overset{K^{-1}}{\longmapsto} \bar{m} \overset{(1.35)}{\longmapsto} Q_s \overset{\pi}{\mapsto} \bar{m}'_s \overset{K'}{\longmapsto} \bar{p}'_s \tag{1.37}$$

Inversement, pour passer d'un point non-distordu $\bar{p}'_s$ à un point distordu $\bar{p}$:

$$\bar{p}'_s \overset{K'^{-1}}{\longmapsto} \bar{m}'_s \overset{(1.30)}{\longmapsto} Q_s \overset{(1.32)}{\longmapsto} \bar{m} \overset{K}{\longmapsto} \bar{p} \tag{1.38}$$

1.1.3 Détection, description et appariement de points d'intérêt

Un point d'intérêt est un ensemble plus ou moins large de pixels connexes d'une image qui présente des propriétés invariantes permettant sa détection dans de multiples vues d'une même scène. C'est un élément fondamental de l'analyse d'image, utilisé par nombre d'algorithmes de suivi en vision par ordinateur. En outre, la détection, la description et l'appariement de points d'intérêt sont des étapes cruciales d'une méthode de SLAM visuel, car les points appareillés appartenant chacun à des images distinctes permettent de calculer la pose des caméras ayant servi aux prises de vues et sont ensuite triangulés. Ces points appareillés assurent donc un rôle important, tant sur le plan de l'odométrie visuelle du système que par rapport au modèle 3D reconstruit. Bien que certains algorithmes se basent sur le calcul du flot optique afin de réaliser un suivi dense, c'est à dire en associant un à un chaque pixel de deux images comparables [7], cette section se concentrera plus particulièrement sur le suivi de points d'intérêt épars, c'est à dire n'associant seulement que quelques régions spécifiques d'une image à une autre. Ce choix est motivé par trois raisons. Premièrement, le cahier des charges énoncé en introduction du manuscrit a été défini tel que le système multi-caméras envisagé pour le SLAM est hétérogène et *wide-baseline*. Ce type de configuration rend difficile le suivi dense de tous les pixels étant donné leur fortes variations d'une prise de vue à une autre, en particulier lorsque ces prises de vues ont été produites par des caméras différentes. Deuxièmement, le suivi dense de tous les pixels de l'image n'améliore pas forcément la précision du SLAM lorsqu'il n'est qu'approximatif, mais peut au contraire la détériorer lorsque les pixels ne sont pas précisément associés d'une image à une autre (dans les zones uniformes de pixels, typiquement) et ne sont donc pas utilisables pour le calcul de pose ni la triangulation de point. Enfin, le calcul du flot optique reste coûteux et lent, même si l'introduction récente du calcul sur GPU (*Graphics Processing Unit*, en anglais) a considérablement amélioré les performances de ce type d'algorithmes. La figure 1.7 illustre le processus de détection, description et appariement de points d'intérêts entre deux vues d'une même scène. La première section 1.1.3.1 de cette partie du manuscrit introduira différents types de points d'intérêt ainsi que certains des détecteurs leur étant associés, la section 1.1.3.2 présentera plusieurs techniques permettant de décrire ces points d'intérêt, dans la section 1.1.3.3 seront exposées quelques méthodes permettant de comparer ces descripteurs, la quatrième section 1.1.3.4 discutera des combinaisons de détecteurs et descripteurs, alors que dans la dernière section 1.1.3.5 seront abordées les différentes stratégies permettant l'appariement de points d'intérêt dans plusieurs vues.

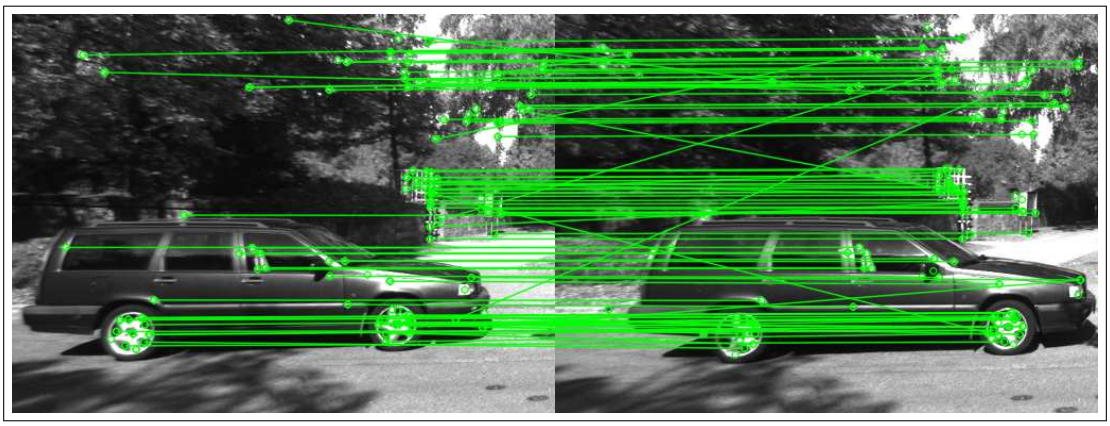

FIGURE 1.7 – Détection, description et appariement de points d'intérêts. On peut remarquer la présence d'appariements erronés entre couples de points d'intérêts ne correspondant pas au même point de la scène.

1.1.3.1 Types de points d'intérêt et détecteurs associés

Chaque détecteur de point d'intérêt est associé à une réponse particulière de l'image qui présente des propriétés invariantes. Plusieurs détecteurs existent, chacun associés à un type de réponse plus ou moins complexe pouvant engendrer des résultats très différents (robustesse, rapidité, etc.). De manière générale, la qualité de l'image à analyser (résolution, contraste, bruit, etc.) est un critère important lors de la détection de points d'intérêt. Plusieurs catégories de détecteurs existent, dont en particulier les détecteurs de coins et les détecteurs de régions.

Détecteurs de coins. Dans une image se trouvent généralement un certain nombre de zones dont la luminance change brusquement. Ces zones forment ce que l'on appelle les contours de l'image. Les coins peuvent être définis comme le changement de direction brusque d'un contour et font partie des premiers types de points d'intérêt utilisés en analyse d'image. Basés sur l'étude du gradient de l'image, ils ont d'abord été introduits par Moravec [119] puis développés par Harris [60], Förstner [49] ou Shi et Tomasi [155]. Mikolajczyk et Schmid [112] ont ensuite proposé une approche permettant au détecteur de Harris d'être invariant au facteur d'échelle, alors que pour éviter certaines opérations de fenêtrage et de filtrage coûteuses, Trajkovic et Hedley [165] ont développé un détecteur ne se basant pas sur les dérivées discrètes de l'image mais sur la comparaison du pixel associé au point d'intérêt évalué à la valeur des pixels d'un cercle discrétisé entourant ce point. Rosten et Drummond [139] ont par la suite amélioré la rapidité de l'algorithme en réduisant le nombre de tests par pixel grâce à des techniques d'apprentissage pour proposer le détecteur de coins FAST (*Features from Accelerated Segment Test*, en anglais). Une liste plus exhaustive des détecteurs basés sur l'intensité et les contours se trouve

dans l'évaluation de Schmid *et al.* [153].

Détecteurs de régions. Au delà de la détection de coins, il est également possible d'utiliser les extrémas locaux de certains filtres utilisés en analyse d'image afin de détecter des régions d'intérêt. Beaucoup d'approches consistent à approximer le Laplacien d'une Gaussienne, dont on a démontré l'invariance au facteur d'échelle [97]. Lowe [100] a proposé avec le détecteur SIFT (*Scale-Invariant Feature Transform*, en anglais), très populaire, de sélectionner les extrémas locaux des différences de Gaussiennes de l'image à différents facteurs d'échelle, plus rapides à calculer que le Laplacien, alors que Bay *et al.* a par la suite introduit le détecteur SURF (*Speeded-Up Robust Features*, en anglais) [10], qui se base sur une approche de calcul efficiente de la matrice Hessienne de l'image à différents facteurs d'échelle.

1.1.3.2 Descripteurs de points d'intérêt

Les descripteurs de points d'intérêt assurent une fonction discriminante et permettent d'associer avec un maximum de robustesse les points projetés sur les images représentant le même point 3D de la scène observée. Les techniques utilisées pour décrire un point d'intérêt sont variées et ne présentent pas les mêmes performances dans tous les domaines (rapidité, robustesse, etc.), il est donc nécessaire de choisir une description adaptée aux conditions expérimentales dans lesquelles leur utilisation est envisagée.

Premiers descripteurs. Le bloc de pixels connexes au point d'intérêt a été l'un des premiers descripteurs utilisé dans la littérature. Bien que basique et peu robuste aux transformations géométriques de tous types, c'est un descripteur rapide à calculer et encore très utilisé dans le contexte du suivi visuel, notamment lorsque la fréquence d'image est élevée et que les points changent peu d'une image à la suivante (pas de déformation importante). Beaucoup de caractéristiques dérivées des intensités locales autour du point d'intérêt considéré ont été proposées afin d'obtenir des descripteurs robustes, dont la plupart ont été évalués par Mikolajczyk et Schmid dans [113].

Descripteurs invariants au facteur d'échelle et aux rotations. Le descripteur SIFT est probablement le plus connu et le plus utilisé des descripteurs locaux et est à l'origine du développement de plusieurs autres descripteurs invariants au facteur d'échelle et aux rotations. Brièvement, ce descripteur opère dans une fenêtre de référence locale présentant un facteur d'échelle et un angle de rotation dominants par rapport au référentiel global de l'image. Le descripteur est basé sur le calcul d'histogrammes de gradients distribués sur une grille autour du point d'intérêt. SURF est une alternative plus rapide à SIFT qui adopte des approches similaires permettant l'invariance au facteur d'échelle

et aux rotations, combinées à des approximations efficientes. Le descripteur se base sur les réponses de filtres de type *box-filters*, plus rapides à calculer que les filtres Gaussiens. BRIEF, introduit plus tard par Calonder *et al.* [21] est encore plus rapide à calculer que SURF pour des performances équivalentes, sans être toutefois robuste aux rotations. C'est un descripteur basé sur l'utilisation de chaînes binaires représentant les intensités d'un patch lissé autour du point d'intérêt. Plus récemment, le descripteur ORB (*Oriented FAST and Rotated BRIEF*, en anglais) à été introduit par Rublee *et al.* [141] afin de remédier à la sensibilité de BRIEF aux rotations, alors que les descripteurs KAZE et AKAZE, introduits par Alcantarilla *et al.* [2, 1], effectuent la détection de points d'intérêt dans un espace d'échelle non linéaire au moyen de filtres de diffusion non-linéaires afin d'obtenir un floutage localement adaptatif permettant de préserver le contour des objets de la scène. Une autre variation, cette fois ci orientée vers l'appariement dense d'images stéréo en *wide-baseline* à été proposée par Tola *et al.* [164] avec le descripteur DAISY et se base comme SIFT sur des histogrammes de gradients, à la différence que ceux-ci sont pondérés par des Gaussiennes et calculés de manière concentrique autour du point d'intérêt.

1.1.3.3 Comparaison de descripteurs

La comparaison de deux descripteurs afin d'évaluer leur ressemblance est dépendante du type de descripteur utilisé. Les descripteurs de type bloc sont le plus souvent comparés par corrélation croisée de type ZNCC (*Zero Mean Normalized Cross-Correlation*, en anglais). Les normes L_1 ou L_2 sont utilisées pour la plupart des descripteurs, alors que la distance de Hamming (H_1 ou H_2) est utilisée dans le cas plus spécifique de descripteurs binaires (BRIEF, ORB, etc.).

1.1.3.4 Combinaisons de détecteurs et descripteurs

Certaines méthodes comprennent la détection et la description de points d'intérêt, comme SIFT ou SURF, par exemple, alors que d'autres doivent être combinées pour permettre leur appariement. C'est le cas du détecteur de Harris, qui est souvent utilisé conjointement à un descripteur de type bloc. Récemment, Gauglitz [52] a proposé un benchmark évaluant la plupart des détecteurs et descripteurs existants dans le contexte du suivi visuel. L'analyse des résultats obtenus permet de conclure que chaque combinaison de détecteur et descripteur présente un comportement différent et sensible aux conditions expérimentales. Plusieurs paramètres tels que la qualité des images, la distance entre chaque pose de caméra, les distorsions de perspective ou le flou de mouvement ont un impact fort sur les performances obtenues.

1.1.3.5 Appariement de points d'intérêt dans plusieurs vues

Il existe plusieurs stratégies permettant l'appariement de points d'intérêt correspondant au même point 3D de la scène observée dans plusieurs vues. Certaines visent à réduire la zone de recherche d'une image à une autre, d'autres à s'assurer que les descripteurs sont les plus similaires possibles. Bien souvent, il s'agit d'une combinaison des deux.

Réduction de la zone de recherche. Parmi les méthodes permettant de réduire la zone de recherche, on peut citer l'utilisation de fenêtres de recherche centrées autour des points d'intérêt de la première image dans la suivante, très utilisées dans le suivi vidéo, lorsque les points restent relativement proches d'une image à l'autre. Une autre méthode, adaptée aux systèmes multi-caméras de type paire stéréo rigide, consiste à utiliser les paramètres extrinsèques des caméras conjointement à leur géométrie épipolaire (une présentation de la géométrie épipolaire est proposée dans la section 1.1.4) afin de réduire la recherche de point correspondant dans la seconde image au parcours d'une droite, appelée droite épipolaire, ou d'une bande de pixels.

Similarité des descripteurs. Les méthodes permettant de s'assurer que les descripteurs appareillés sont les plus similaires possibles sont généralement basées sur l'utilisation de ratios permettant d'éliminer les appariements ambigus. Lowe a proposé avec SIFT de calculer pour chaque descripteur le score de ressemblance avec le plus proche descripteur et le meilleur second. La correspondance est ensuite établie si le ratio de leurs distances respectives est inférieur à 0.8, c'est à dire que le plus proche descripteur est significativement meilleur que les autres correspondances. Une autre technique, appelée *cross-check* en anglais, permet d'assurer un appariement mutuel des descripteurs de la première image vers la deuxième et inversement, afin d'être certain que les descripteurs sont effectivement les plus proches. En effet, l'appariement itératif de tous les descripteurs de la première image vers la seconde peut conduire à la mise en correspondance de plusieurs descripteurs de la première image vers le même descripteur de la seconde. Ces correspondances multiples sont alors filtrées en réalisant l'opération inverse, c'est à dire de la deuxième image vers la première, afin de ne retenir que les paires de descripteurs ayant été mutuellement mises en correspondance dans les deux sens.

1.1.4 Géométrie épipolaire

La géométrie épipolaire est un modèle mathématique permettant de décrire les relations géométriques entre deux vues d'une même scène et dont un bref historique est proposé par Sturm dans [160]. Ce modèle est indépendant de la structure géométrique

de la scène observée. Le principe sur lequel repose la géométrie épipolaire consiste à considérer que chaque point 3D P de la scène, ses projections 2D p et p' sur chaque plan image ainsi que les centres optiques c et c' des caméras sont coplanaires. Le plan Π auquel appartiennent ces points est appelé plan épipolaire. Chaque triplet (P, p, p') forme ainsi un plan épipolaire particulier parmi l'ensemble des plans épipolaires possibles concourants à la droite passant par les centres optiques des deux caméras. L'intersection de ces plans épipolaires avec les plans image forment des droites épipolaires, alors que l'intersection de la droite passant par chaque centre optique et les plans image se fait en deux points appelés épipoles. Dans le cas où les paramètres intrinsèques et extrinsèques des caméras sont connus, dans les systèmes stéréo typiquement, cette propriété permet de réduire la recherche d'un point d'intérêt à appareiller dans l'autre vue à un parcours de la droite épipolaire formée par l'intersection du plan image et du plan épipolaire auquel le point de la première vue et les centres optiques des caméras appartiennent. La figure 1.8 illustre le principe de la géométrie épipolaire et la recherche de point d'intérêt correspondant le long d'une droite épipolaire.

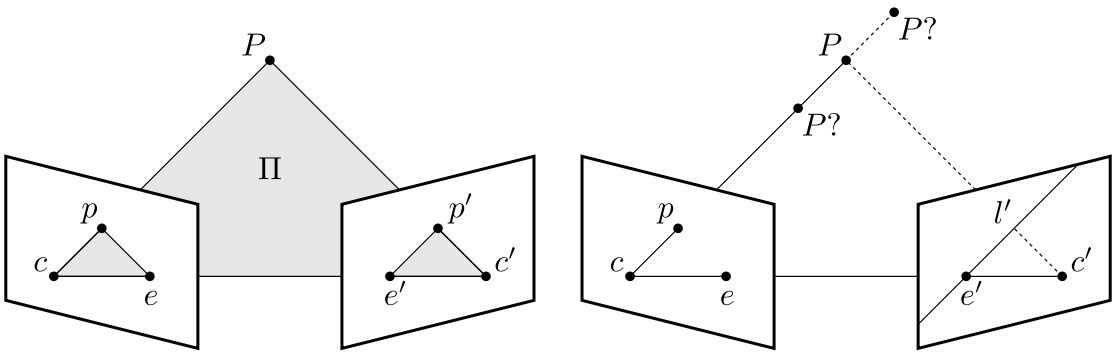

FIGURE 1.8 – Géométrie épipolaire. À gauche, le plan épipolaire Π formé par les points c, c', p, p' et P. La droite joignant les centres des caméras c et c' intersecte les images aux épipoles e et e', alors que les intersections entre le plan épipolaire et les images forment les droites épipolaires l et l'. À droite, la recherche de correspondance du point p sur la droite épipolaire l'.

1.1.4.1 Matrice Fondamentale F et matrice Essentielle E

La matrice fondamentale a été introduite en 1992 par Luong [102], Faugeras [42, 41] et Hartley [62, 64] alors que la matrice essentielle a été introduite par Longuet-Higgins en 1981 [99]. Ces matrices sont les représentations algébriques de la géométrie épipolaire. Elles permettent d'exprimer la relation entre points d'intérêt de deux vues correspondants puis de retrouver la transformation permettant de passer d'un repère caméra à l'autre, c'est à dire les poses de chaque caméra, composées d'une rotation R et d'un vecteur translation $\mathbf{t}$. Leurs propriétés sont plus largement détaillées dans [61].

Pour toute paire de deux points d'intérêt $\bar{p}$ et $\bar{p}'$, exprimés en coordonnées homogènes dans les repères image i et i' et représentant le même point P de la scène, il existe une matrice fondamentale $F_{3\times3}$ satisfaisant l'équation :

$$\bar{p}'^T F \bar{p} = 0 \tag{1.39}$$

De manière similaire, pour toute paire de deux points d'intérêt $\bar{m}$ et $\bar{m}'$, exprimés en coordonnées homogènes dans les repères normalisés n et n' et représentant le même point P de la scène, il existe une matrice essentielle $E_{3\times3}$ satisfaisant l'équation :

$$\bar{m}'^T E \bar{m} = 0 \tag{1.40}$$

L'unique différence entre les relations (1.39) et (1.40) est l'utilisation de points appartenant au plan image dans le cas de la matrice fondamentale alors que la matrice essentielle repose sur l'utilisation de points appartenant au plan normalisé, ce qui implique la connaissance à priori des paramètres de calibration intrinsèques des caméras (les matrices K et K'). Lorsque l'on remplace les points $\bar{m}$ et $\bar{m}'$ de l'équation 1.40 par le résultat de l'équation 1.14, on obtient $\bar{p}'^T K'^{-T} E K^{-1} \bar{p} = 0$, ce qui conduit aux relations entre matrice fondamentale et matrice essentielle :

$$F = K'^{-T} E K^{-1} \qquad\qquad E = K'^T F K \tag{1.41}$$

En conséquence, la matrice essentielle peut être considérée comme une forme particulière de la matrice fondamentale. En l'absence de paramètres intrinsèques, cette dernière ne permet de reconstruire la scène observée qu'à une transformation projective près, c'est à dire ne respectant pas les angles, les rapports de distance à l'exception du birapport, ni le facteur d'échelle, quand la matrice essentielle permet une reconstruction euclidienne de la scène à un facteur d'échelle près. En outre, la matrice fondamentale possède sept degrés de liberté. Celle-ci est composée de neuf éléments, dont un facteur d'échelle non significatif, alors que son déterminant doit être nul ($det(F) = 0$), ce qui permet de retirer au final deux degrés de liberté. La matrice essentielle pour sa part ne possède que cinq degrés de liberté, six degrés de liberté pour exprimer une rotation R et un vecteur translation $\mathbf{t}$, alors que le facteur d'échelle n'est là encore pas significatif, ce qui retire un degré de liberté. D'autres propriétés permettent d'obtenir ces cinq degrés de liberté, à savoir un déterminant nul ($det(E) = 0$) et le respect de la propriété $2EE^TE - tr(EE^T)E = 0$, c'est à dire que deux des valeurs singulières de cette matrice doivent être égales et la troisième nulle pour qu'elle soit considérée comme matrice essentielle.

Calcul de F et E. Le principe de calcul des matrices fondamentale et essentielle repose sur la contrainte de leurs degrés de libertés par des appariements de points d'in-

térêts satisfaisant les relations 1.39 ou (1.40), en plus de contraintes liées directement à leur propriétés. Il faut donc sept points appareillés au minimum pour déterminer une matrice fondamentale quand la matrice essentielle n'en nécessite que cinq. Plusieurs algorithmes ont été introduits afin de calculer la matrice fondamentale, dont entre autres l'algorithme des huit-points normalisé introduit par [63], consistant à retrouver F à partir des équations linéaires homogènes basées sur huit appariements de points préalablement normalisés. Parmi ceux permettant de calculer la matrice essentielle, on peut citer notamment l'algorithme des cinq-points proposé par Nistér [127], basé sur une procédure efficiente de type élimination de Gauss-Jordan, puis d'une version améliorée de cet algorithme introduite par Li *et al.* [93], qui se base sur l'utilisation de variables cachées. Dans le contexte de cette thèse, le système multi-caméras utilisé est calibré à l'avance et les paramètres intrinsèques des caméras sont connus. L'utilisation de la matrice essentielle afin de déterminer la géométrie initiale de la scène a donc été privilégiée.

Estimation robuste de E. Comme dit précédemment, le calcul de la matrice essentielle repose sur la contrainte de ses cinq degrés de libertés par des appariements de points d'intérêt satisfaisant la relation 1.40. En pratique, ces appariements ne sont pas toujours exacts ou corrects, c'est à dire qu'ils sont soit sujets au bruit, soit ne représentent tout simplement pas le même point 3D de la scène. Afin de réduire l'impact de ces mauvais appariements sur le calcul de la matrice essentielle, l'algorithme des cinq-points est le plus souvent réalisé conjointement à un filtrage statistique robuste de type RANSAC. L'objectif de l'algorithme RANSAC est de maximiser le nombre d'inliers (les paires de points d'intérêt correspondants satisfaisant la contrainte épipolaire parmi les paires disponibles) et donc de minimiser le nombre d'outliers (les paires ne satisfaisant pas la contrainte) lors de l'estimation de la matrice E. Son principe de fonctionnement consiste à estimer itérativement la matrice essentielle à partir de paires de points appareillés sélectionnés au hasard parmi les paires disponibles pour chaque itération (cinq paires, dans le cas de l'algorithme des cinq-points), puis de tester l'hypothèse sur l'ensemble des paires restantes, l'hypothèse retenue parmi celles testées étant celle permettant d'obtenir le maximum d'inliers. Ainsi estimée, la matrice E s'accorde avec un maximum de paires de points dans les deux vues, et peut-être considérée comme une représentation correcte du déplacement des deux caméras.

À noter cependant que dans le cas d'un algorithme de SLAM visuel de type *Structure-from-Motion*, basé sur le déplacement du point de vue pour estimer la géométrie de la scène, l'estimation de E et donc du déplacement des caméras n'est valable que lorsque la scène à reconstruire est statique et rigide (ne se déplace ni ne se déforme entre deux vues). Le cas extrême dans lequel une majorité de points appareillés est associée à divers objets en mouvement (traversée d'une foule de piétons, etc.) peut fausser l'estimation

du déplacement réel des caméras dans le repère monde. Ce type d'estimation présente donc une limitation forte, à considérer notamment lorsque l'application s'inscrit dans un contexte d'odométrie visuelle sensible. En pratique, l'utilisation de caméras grand-angle, couvrant un maximum de champ visuel, permet généralement de corriger l'impact de ces mauvais appariements en augmentant le rapport entre quantité de points de la scène rigide et quantité de points ambigus.

1.1.4.2 Décomposition de E pour retrouver les poses des caméras

Une fois la matrice essentielle estimée, il est nécessaire de la décomposer afin d'obtenir la pose C' de la deuxième caméra relativement à la pose C, c'est à dire une matrice de rotation R' et un vecteur translation $\mathbf{t}'$. On effectue pour cela une décomposition en valeurs singulières de E (*Singular Value Decomposition*, ou SVD, en anglais), comme proposé par [39]. La matrice essentielle E est définie telle que :

$$E = [\mathbf{t}']_\times R' = SR' \tag{1.42}$$

Où $[\mathbf{t}']_\times = S$ est la matrice antisymétrique associée au vecteur translation $\mathbf{t}'$ de C' et R' est la rotation de C'. La décomposition en valeurs singulières de E est de la forme :

$$\text{SVD}(E) = U \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} V^T \tag{1.43}$$

Avec U et V des matrices unitaires. On définit les deux matrices W et Z :

$$W = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad Z = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{1.44}$$

Si l'on se réfère à l'équation (1.43), on peut réécrire (1.42) :

$$E = [\mathbf{t}']_\times R' = SR' = (UZU^T)(UXV^T) \tag{1.45}$$

Avec $X = W$ ou $X = W^T$. Au final, on a donc :

$$S = UZU^T \qquad R' = UWV^T \quad \text{ou} \quad UW^TV^T \tag{1.46}$$

Enfin, si l'on considère que la pose de la première caméra est de la forme $C = [I_3|0_{3\times1}]$, ces équations nous donnent quatre solutions $[R'|\mathbf{t}']$ pour la pose C' de la deuxième caméra :

$$[R'|\mathbf{t}'] = [UWV^T|\mathbf{u}_3] \text{ ou } [UWV^T|-\mathbf{u}_3] \text{ ou } [UW^TV^T|\mathbf{u}_3] \text{ ou } [UW^TV^T|-\mathbf{u}_3] \quad (1.47)$$

L'interprétation de ces quatre solutions est la suivante. Les deux solutions pour $R' = UWV^T$ et $R' = UW^TV^T$ représentent une rotation de 180° autour de la droite joignant les centres optiques des caméras c et c', alors que les deux solutions pour $\mathbf{t}' = \mathbf{u}_3$ et $\mathbf{t}' = -\mathbf{u}_3$ représentent une translation inverse le long de la droite joignant les centres des caméras c et c'. Une illustration de ces quatre solutions est donnée figure 1.9. De ces quatre solutions, une seule est valide, celle où les points reconstruits le sont devant les deux caméras. Il suffit alors de projeter les points 3D reconstruits (la triangulation de points 3D est abordée dans la section 1.1.5) dans chaque repère caméra afin de s'assurer qu'ils sont bien en avant de leur axe optique :

$$\forall \bar{P}_w \begin{cases} C\bar{P}_w = (X_c, Y_c, Z_c, 1)^T \text{ avec } Z_c > 0 \\ C'\bar{P}_w = (X_{c'}, Y_{c'}, Z_{c'}, 1)^T \text{ avec } Z_{c'} > 0 \end{cases} \quad (1.48)$$

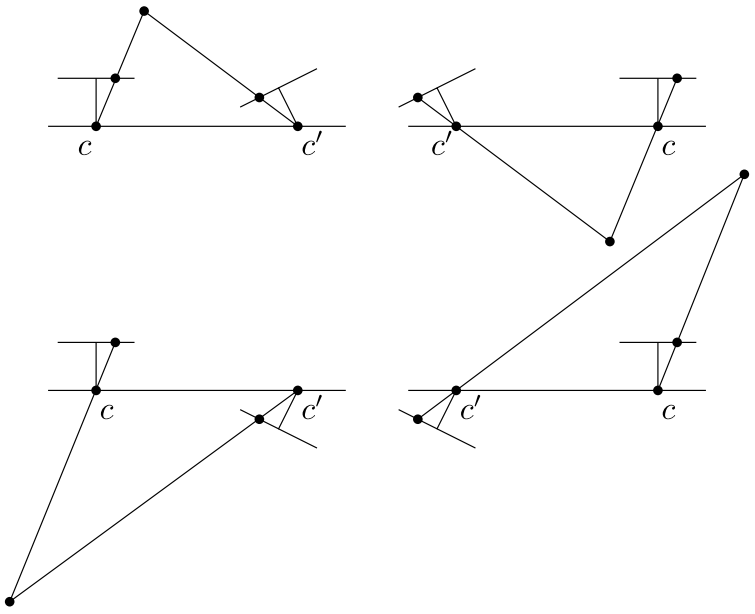

FIGURE 1.9 – Les quatre solutions R et $\mathbf{t}$ du calcul de pose à partir de E. De gauche à droite, la translation est inversée alors que de haut en bas, la caméra de centre c' effectue une rotation de 180° autour de la droite joignant les centres optiques des caméras. Seule la première solution, en haut à gauche, est valide.

1.1.4.3 Épipoles et droites épipolaires

À partir de F ou E, il est possible de calculer la position des épipoles et de tracer les droites épipolaires correspondant à un point d'intérêt sur chaque plan image. Les coordonnées des épipoles e et e' s'expriment :

$$Fe = 0 \qquad F^T e' = 0 \tag{1.49}$$

Les droites épipolaires l' et l, définies comme les intersections du plan épipolaire relatif aux points $\bar{p}$ et $\bar{p}'$ avec les plans image i' et i, respectivement, s'expriment :

$$l' = F\bar{p} \qquad l = F^T\bar{p}' \tag{1.50}$$

1.1.5 Triangulation

Une fois obtenues les poses C et C' de deux caméras (estimées grâce à la géométrie épipolaire ou à partir de points 3D déjà reconstruits, voir section 1.1.6), il est par la suite possible de trianguler deux points d'intérêt appareillés p et p' afin de retrouver les coordonnées du point 3D P leur étant associé. Le principe de la triangulation consiste à calculer l'intersection d'un premier rayon passant par le centre optique c de la première caméra et le point d'intérêt p de son plan image et d'un deuxième rayon passant de manière analogue par c' et p'. En pratique, la présence de bruit dans les images ainsi que les approximations numériques générées par l'ordinateur ne permettent pas l'obtention de deux rayons qui s'intersectent parfaitement. Il faut donc approximer cette intersection. Une méthode assez largement utilisée est la méthode du point milieu, qui consiste à estimer les coordonnées 3D du point P comme le milieu du plus court segment joignant les deux rayons.

Brièvement, on considère les centres optiques des caméras c et c' ainsi que les observations p et p' du point 3D P sur chaque vue. Le principe de la méthode consiste à calculer la perpendiculaire commune aux droites (cp) et $(c'p')$, dirigée par le vecteur $\overrightarrow{v} = \overrightarrow{cp} \wedge \overrightarrow{c'p'}$, avec $\wedge$ le produit vectoriel. On définit le plan Π, passant par c et formé par les vecteurs $\overrightarrow{cp}$ et $\overrightarrow{v}$ et le plan Π', passant par c' et formé par les vecteurs $\overrightarrow{c'p'}$ et $\overrightarrow{v}$. On définit ensuite le point $Q = c + \alpha\overrightarrow{cp}$ tel que l'intersection entre la droite (cp) et le plan Π', avec $\alpha = \frac{\overrightarrow{cc'}\cdot\overrightarrow{c'p'}\overrightarrow{cp}\cdot\overrightarrow{c'p'} - \overrightarrow{cc'}\cdot\overrightarrow{cp}\left\|\overrightarrow{c'p'}\right\|^2}{\left(\overrightarrow{cp}\cdot\overrightarrow{c'p'}\right)^2 - \left\|\overrightarrow{cp}\right\|\left\|\overrightarrow{c'p'}\right\|}$. De la même manière, nous définissons le point Q' tel que l'intersection entre la droite $(c'p')$ et le plan Π. Enfin, le point 3D P triangulé est le milieu du segment $[QQ']$.

Cette méthode n'étant pas invariante aux transformations affines ni projectives, son seul avantage est sa rapidité de calcul [65]. En revanche, cette solution peut se révéler

valable lorsqu'elle est effectuée conjointement à une optimisation globale des poses et points 3D (voir section 1.1.7 pour plus de détails), qui permet de rectifier sa faible précision et qui est la raison pour laquelle elle a été utilisée dans ces travaux. La figure 1.10 illustre la méthode du point milieu.

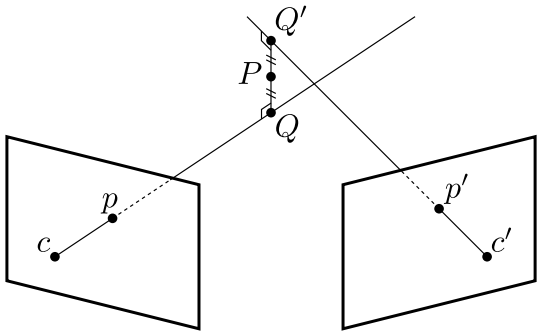

FIGURE 1.10 – Méthode du point milieu.

La solution optimale au problème de triangulation a été présentée par Hartley et Sturm dans [65]. Elle consiste à trouver les deux rayons s'intersectant parfaitement et minimisant pour chacun la distance entre son point d'intersection avec le plan image et le point d'intérêt.

1.1.6 Calcul de pose à partir de points 3D

Alors que la géométrie épipolaire pour estimer la pose de chaque caméra à partir d'appariements de points d'intérêt est appropriée lorsque l'on ne dispose pas encore de la structure géométrique de la scène, il est ensuite plus rapide d'estimer ces poses à partir des points 3D déjà reconstruits et de leurs projections 2D associées. Ce type d'estimation de pose, appelé *Perspective-n-Points* ou PnP, en anglais, est généralement utilisé lorsque l'on souhaite reconstruire une scène de manière incrémentale après en avoir estimé la géométrie initiale, ce qui est le cas des applications de type SLAM basées sur des séquences vidéo, par exemple. Parmi les méthodes existantes, nous pouvons citer en particulier la méthode originale de Grunert [58], qui permet à partir de trois points 3D observés dans l'image de résoudre un système d'équations exprimant la lois des cosinus dans l'espace euclidien.

Brièvement, on considère le centre optique c de la caméra ainsi que les points 3D observés P_1 P_2 et P_3. On définit ensuite $L_1 = |cP_1|$, $L_2 = |cP_2|$, $L_3 = |cP_3|$, $\alpha = \widehat{P_2cP_3}$, $\beta = \widehat{P_1cP_3}$, $\gamma = \widehat{P_1cP_2}$, $a_1 = 2\cos\alpha$, $a_2 = 2\cos\beta$, $a_3 = 2\cos\gamma$, $l_1 = |P_2P_3|$, $l_2 = |P_1P_3|$ et $l_3 = |P_1P_2|$. Les triangles cP_2P_3, cP_1P_3 et cP_1P_2 permettent d'obtenir le système d'équations *P3P* à résoudre suivant :

$$\begin{cases} {L_2}^2 + {L_3}^2 - L_2 L_3 a_1 - {l_1}^2 = 0 \\ {L_3}^2 + {L_1}^2 - L_3 L_1 a_2 - {l_2}^2 = 0 \\ {L_1}^2 + {L_2}^2 - L_1 L_2 a_3 - {l_3}^2 = 0 \end{cases} \tag{1.51}$$

L'estimation de pose à partir de trois points produit plusieurs solutions, dont au plus quatre vraisemblables. L'utilisation d'une méthode robuste de type RANSAC ou l'introduction d'un quatrième point 3D permet en général de lever l'ambiguïté. La figure 1.11 illustre le problème *P3P*.

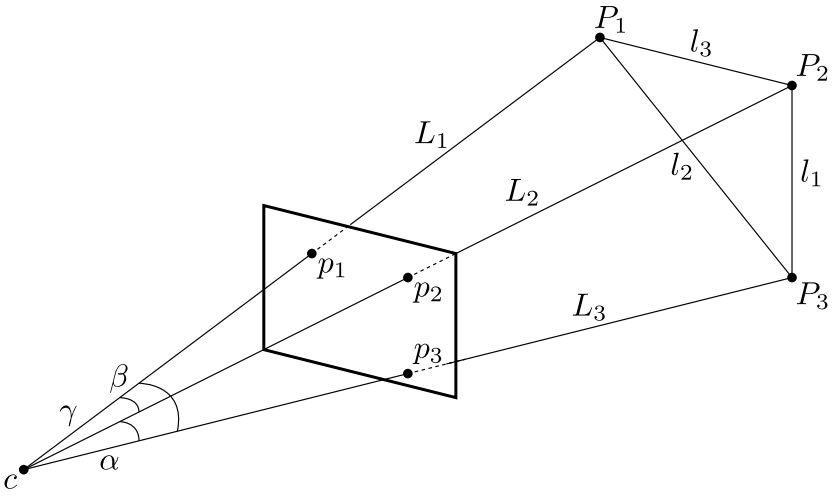

FIGURE 1.11 – Perspective-3-Points.

Un état de l'art sur l'estimation de pose à partir de trois correspondances a notamment été proposé par Haralick *et al.* dans [59], alors qu'une étude plus récente, accompagnée d'une formalisation algébrique et géométrique du problème a été proposée par Gao *et al.* dans [51]. Plus récemment, Lepetit *et al.* [90] ont proposé une nouvelle méthode à la complexité linéaire pour $n >= 4$ points 3D et donnant d'excellents résultats. Brièvement, le principe de la méthode consiste à exprimer l'ensemble des points 3D comme la somme pondérée de quatre points de contrôle. Le problème se réduit ensuite à l'estimation des coordonnées de ces quatre points de contrôle dans le repère caméra. Ces algorithmes sont généralement réalisés conjointement à une méthode de type RANSAC afin d'en assurer la robustesse.

1.1.7 Optimisation conjointe de poses et points 3D

Les méthodes présentées dans les sections précédentes permettent effectivement de reconstruire en trois dimensions la scène observée à partir de plusieurs vues, mais sont cependant susceptibles de générer des erreurs géométriques, pour la plupart engendrées

par diverses approximations (bruit numérique, calibration, estimation de pose, triangulation, approximations numériques, etc.). Ces erreurs géométriques peuvent en outre devenir importantes lorsque plusieurs sources d'approximations se cumulent, mais aussi lorsqu'elles se propagent d'observations en observations, dans le cas d'un algorithme de SLAM incrémental par exemple, pour finalement avoir une influence conséquente sur les éléments reconstruits par la suite (dérives de la trajectoire et du facteur d'échelle, déformations géométriques de plus en plus importantes, etc.). Par ailleurs, le problème du SLAM est sur-contraint, c'est à dire qu'un seul point 3D de la scène peut être observé plusieurs fois, ce qui amène à résoudre un système avec plus d'équations que de variables. Dans un contexte simulé, lorsque les approximations sont négligeables, cela ne pose pas forcément problème et la solution pour les variables d'une équation particulière convient pour les autres équations partageant ces variables. En pratique, pourtant, la présence d'approximations ne garantit pas que toutes les équations du système partagent la même solution. Il est donc nécessaire de corriger ces approximations afin de garder une précision acceptable pendant la reconstruction. L'objectif des méthodes présentées dans cette section consiste à minimiser de manière globale les erreurs d'approximation générées par les précédentes étapes du SLAM afin de trouver une solution convenable pour toutes les équations du système, par approche purement probabiliste dans les méthodes de type filtrage et ajustement par moindres carrés non linéaires dans les méthodes de type ajustement de faisceaux. Une étude comparative des performances de chacune des deux méthodes a été proposée par [158]. Les conclusions de l'article penchent définitivement en faveur des méthodes de type ajustement de faisceaux dans le contexte du SLAM visuel incrémental, plus efficientes et à la précision meilleure.

1.1.7.1 Méthodes de type filtrage

Les méthodes de type filtrage utilisent en général un filtre de Kalman étendu (*Extended Kalman Filter*, en anglais), introduit par Kalman en 1960 [74] et sont généralement composées de trois étapes basées sur les principes de prédiction et de correction d'un vecteur d'états représentant la position des caméras et des points reconstruits. La première étape est une étape de prédiction qui consiste à estimer de manière probabiliste le déplacement de la caméra ainsi que la position des points observés précédemment par rapport à ce déplacement. La deuxième étape est une étape de mesure qui intègre les nouvelles observations des points déjà observés afin de corriger les prédictions précédentes. Enfin, la dernière étape consiste à mettre à jour le vecteur d'états en ajoutant les nouveaux points observés. Plusieurs méthodes basées sur l'utilisation d'un filtre de Kalman ont été proposées, notamment [28].

1.1.7.2 Méthodes de type ajustement de faisceaux

Les méthodes de type ajustement de faisceaux (*Bundle-Adjustment*, en anglais) sont basées sur l'utilisation d'algorithmes d'ajustement par moindres carrés non linéaires. Plusieurs algorithmes non linéaires existent, dont en particulier la technique d'optimisation développée par Levenberg et Marquardt [106] ou encore celle proposée par Powell [134]. L'objectif de ces méthodes consiste à minimiser de manière itérative l'erreur de reprojection ou l'erreur angulaire des points 3D reconstruits par rapport à leurs observations. Dans le cas de l'erreur de reprojection, il s'agit de la distance entre la projection sur le plan image du point 3D reconstruit et le point d'intérêt correspondant pour chaque observation, alors que dans le cas de l'erreur angulaire, introduite plus tard par Lhuillier [92], il s'agit de l'angle entre les deux rayons partant chacun du centre optique de la caméra, puis passant l'un par le point 3D reconstruit et l'autre par le point d'intérêt correspondant, pour chaque observation. Un état de l'art très complet sur ce type d'optimisation, utilisant l'erreur de reprojection, a été réalisé par Triggs *et al.* dans [166]. L'algorithme de Levenberg-Marquardt à été privilégié dans ces travaux car plus généralement utilisé dans la littérature.

Moindres carrés non linéaires. Soit un vecteur de paramètres $\mathbf{x}$, un vecteur de mesures $\mathbf{y}$ et une fonction de modélisation non-linéaire f exprimant la relation entre le vecteur de paramètres et le vecteur de mesures, suivant :

$$\mathbf{y} = f(\mathbf{x}) \tag{1.52}$$

Dans cette relation, chaque mesure $\mathbf{y}_k$ est égale à $f_k(\mathbf{x})$. En pratique, lorsque le calcul de f est approché ou que les mesures $\mathbf{y}$ sont bruitées, cette égalité n'est pas garantie, ce qui implique que la différence entre les deux produise une erreur ϵ_k :

$$\epsilon_k = \mathbf{y}_k - f_k(\mathbf{x}) \tag{1.53}$$

On définit le vecteur des résidus ϵ contenant les erreurs ϵ_k tel que :

$$\epsilon = \mathbf{y} - f(\mathbf{x}) \tag{1.54}$$

L'objectif des méthodes de moindres carrés non linéaires consiste à minimiser le vecteur de résidus ϵ, c'est à dire la fonction de coût suivante :

$$\|\epsilon\|^2 = \epsilon^T \epsilon = \sum_k \|\epsilon_k\|^2 = \sum_k \|\mathbf{y}_k - f_k(\mathbf{x})\|^2 \tag{1.55}$$

Erreur de reprojection. Soit $[P]_{j,t}$ la matrice de projection de la caméra j à l'instant t (cf. équation 1.20) avec $j \in 0 \dots J$ et $t \in 0 \dots T$. Pour rappel, cette matrice de projection se compose de la pose de la caméra $C_{j,t}$ ainsi que de ses paramètres intrinsèques K_j. Soit l'ensemble des points 3D P_n avec $n \in 0 \dots N$ et l'ensemble des points 2D $p^n_{j,t}$, désignant les points d'intérêt vus par la caméra j au temps t, associés au point 3D P_n et respectant la géométrie épipolaire de la scène, c'est à dire les inliers. On définit la fonction de projection f telle que :

$$f(C_{j,t}, P_n) = [P]_{j,t}\bar{P}_n \tag{1.56}$$

Idéalement, $f(C_{j,t}, P_n) = p^n_{j,t}$. Cependant, comme expliqué précédemment, plusieurs sources d'approximations ne permettent pas toujours de vérifier l'égalité. L'erreur de reprojection du point 3D P_n dans l'image produite par la caméra j à l'instant t est donc définie telle que :

$$\epsilon^n_{j,t} = \left\| p^n_{j,t} - f(C_{j,t}, P_n) \right\| \tag{1.57}$$

Où $\epsilon^n_{j,t}$ représente la distance euclidienne entre la reprojection du point 3D P_n et son observation $p^n_{j,t}$ dans l'image produite par la caméra j à l'instant t. La figure 1.12 illustre l'erreur de reprojection.

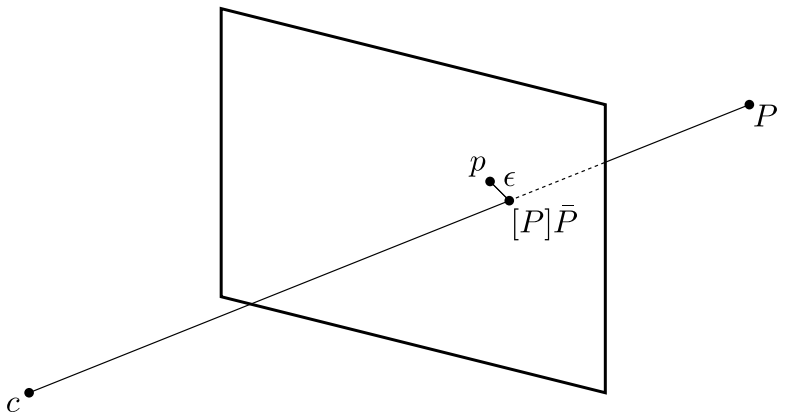

FIGURE 1.12 – Erreur de reprojection. Dans cette illustration, l'erreur de reprojection $\epsilon = \left\| p - [P]\bar{P} \right\|$.

Erreur angulaire. Soit $[P]_{j,t}$ la matrice de projection de la caméra étalonnée j au temps t et de centre optique $c_{j,t}$. Soit maintenant $p^n_{j,t}$ l'observation sur le plan image du point 3D P_n. Comme abordé dans la sous-section 1.1.2.3, le rayon passant par le centre optique $c_{j,t}$ et le point d'intérêt $p^n_{j,t}$ a pour vecteur directeur $\mathbf{r}_{p^n_{j,t}} = (c_{j,t}, p^n_{j,t})^T$. De manière analogue, le rayon passant par $c_{j,t}$ et le point 3D P_n a pour vecteur directeur

$\mathbf{r}_{P^n_{j,t}} = (c_{j,t}, [P]_{j,t}\bar{P}_n)^T$. Enfin, on définit les deux vecteurs directeurs normalisés $\mathbf{d}^n_{j,t}$ et $\mathbf{D}^n_{j,t}$, tels que :

$$\mathbf{d}^n_{j,t} = \frac{\mathbf{r}_{p^n_{j,t}}}{\left\|\mathbf{r}_{p^n_{j,t}}\right\|} \qquad \mathbf{D}^n_{j,t} = \frac{\mathbf{r}_{P^n_{j,t}}}{\left\|\mathbf{r}_{P^n_{j,t}}\right\|} \tag{1.58}$$

Ces deux vecteurs normalisés doivent idéalement être confondus, ce qui traduit la colinéarité du centre de la caméra $c_{j,t}$, de l'observation $p^n_{j,t}$ sur l'image et du point 3D P_n. En pratique, ce n'est que rarement le cas. On introduit donc l'erreur $\epsilon^n_{j,t}$ correspondant à l'angle entre les vecteurs $\mathbf{d}^n_{j,t}$ et $\mathbf{D}^n_{j,t}$, traditionnellement calculé suivant la formule : $\epsilon^n_{j,t} = \arccos(\mathbf{d}^n_{j,t} \cdot \mathbf{D}^n_{j,t})$. Lhuillier a cependant démontré dans [92] la faible convergence de cette expression dans le cadre d'un ajustement de faisceaux avec la méthode de Levenberg-Marquardt, du fait de sa classe de régularité C^0 à l'exacte solution $\epsilon^n_{j,t} = 0$. Cette méthode imposant l'usage d'une fonction de calcul des résidus de classe C^2 à minima, l'auteur a proposé une formulation alternative, définie telle que :

$$\epsilon^n_{j,t} = \pi(R^n_{j,t}\mathbf{D}^n_{j,t}) \tag{1.59}$$

Avec $R^n_{j,t}$ une matrice de rotation définie telle que $R^n_{j,t}\mathbf{d}^n_{j,t} = [0\,0\,1]^T$ et la fonction $\pi : \mathbb{R}^3 \to \mathbb{R}^2$ définie telle que $\pi([x\,y\,z]^T) = \left[\frac{x}{z}\ \frac{y}{z}\right]^T$. À noter que selon cette formule, l'erreur $\epsilon^n_{j,t}$ ne représente alors plus un angle mais un vecteur 2D dont la norme euclidienne correspond à la tangente entre les vecteurs $\mathbf{d}^n_{j,t}$ et $\mathbf{D}^n_{j,t}$, tel que :

$$\left\|\epsilon^n_{j,t}\right\| = tan(\mathbf{d}^n_{j,t}, \mathbf{D}^n_{j,t}) \tag{1.60}$$

On introduit finalement la fonction de calcul de l'erreur angulaire g, prenant en paramètres l'observation $p^n_{j,t}$, la pose $C_{j,t}$ de la caméra ainsi que le point 3D associé P_n :

$$g(p^n_{j,t}, C_{j,t}, P_n) = tan(\mathbf{d}^n_{j,t}, \mathbf{D}^n_{j,t}) = \left\|\epsilon^n_{j,t}\right\| \tag{1.61}$$

La figure 1.13 illustre l'erreur angulaire.

Ajustement de faisceaux. Soit maintenant l'ensemble de toutes les poses des caméras $\mathcal{C} = (C_{0,0},\ \ldots\ , C_{J,T})$, l'ensemble de tous les points 3D $\mathcal{P} = (P_0,\ \ldots\ , P_N)$ et le vecteur de paramètres $\mathbf{x} = (\mathcal{C}, \mathcal{P})$. De même, on définit le vecteur de mesure $\mathbf{y}$, composé de tous les inliers $p^n_{j,t}$.

Dans un premier cas, le principe des méthodes de type ajustement de faisceaux consiste à trouver le vecteur de paramètres $\hat{\mathbf{x}}$ minimisant le carré des erreurs de reprojection des points 3D par rapport à leurs observations, défini par la fonction de coût suivante :

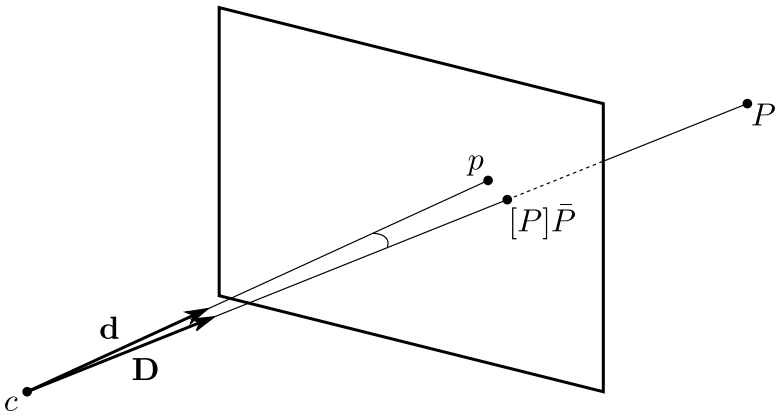

FIGURE 1.13 – Erreur angulaire. Dans cette illustration, l'erreur angulaire $\|\epsilon\| = tan(\mathbf{d}, \mathbf{D})$.

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \sum_{j=0}^{J} \sum_{t=0}^{T} \sum_{n=0}^{N} \|\mathbf{y}_{j,t,n} - f_{j,t,n}(\mathbf{x})\|^2 \tag{1.62}$$

Avec $\mathbf{y}_{j,t,n} = p^n_{j,t}$ et $f_{j,t,n}(\mathbf{x}) = f(C_{j,t}, P_n)$.

Dans le cas d'un ajustement de faisceaux reposant sur le calcul de l'erreur angulaire, la fonction de coût associée permet la minimisation du carré des erreurs angulaires, tel que :

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \sum_{j=0}^{J} \sum_{t=0}^{T} \sum_{n=0}^{N} g_{j,t,n}(\mathbf{y}, \mathbf{x})^2 \tag{1.63}$$

Avec $g_{j,t,n}(\mathbf{y}, \mathbf{x}) = g(p^n_{j,t}, C_{j,t}, P_n)$.

Attention cependant, chaque point 3D P_n n'est pas automatiquement associé à une observation dans toutes les vues (pas de $p^n_{j,t}$ associé, ou alors $p^n_{j,t}$ n'est pas un inlier), ce qui implique que son erreur de reprojection ou son erreur angulaire ne soit pas définie pour toutes les poses $C_{j,t}$.

1.1.7.3 Ajustement de faisceaux avec l'algorithme de Levenberg-Marquardt

L'algorithme de minimisation de Levenberg-Marquardt combine les avantages d'une méthode du premier ordre de type descente de gradient, permettant de converger vers une solution minimale même si la solution initiale en est éloignée et d'une méthode du second ordre de type Gauss-Newton, permettant de converger rapidement vers un minimum lorsque la solution s'en approche.

Erreur de reprojection. Dans un premier cas, le principe de la minimisation décrite ici consiste à trouver pour chaque itération le pas $\delta_{\mathbf{x}}$ appliqué au vecteur $\mathbf{x}$ et réduisant l'erreur de reprojection globale $\|\epsilon\|$, c'est à dire $\|\mathbf{y} - f(\mathbf{x} + \delta_{\mathbf{x}})\| < \|\mathbf{y} - f(\mathbf{x})\|$. L'approximation de Taylor au premier ordre de f est de la forme :

$$f(\mathbf{x} + \delta_{\mathbf{x}}) \approx f(\mathbf{x}) + J\delta_{\mathbf{x}} \tag{1.64}$$

Où J est la Jacobienne de f en $\mathbf{x}$. Après réécriture de la fonction de coût, on cherche le pas $\delta_{\mathbf{x}}$ minimisant $\|\mathbf{y} - (f(\mathbf{x}) + J\delta_{\mathbf{x}})\| = \|\epsilon - J\delta_{\mathbf{x}}\|$, c'est à dire vérifiant les équations normales :

$$J^T J\delta_{\mathbf{x}} = J^T \epsilon \tag{1.65}$$

Cela revient à effectuer une itération de type Gauss-Newton, dont la convergence n'est pas garantie mais rapide lorsque la solution initiale est proche d'un minimum. Pour garantir la convergence de la méthode, l'algorithme de Levenberg-Marquardt utilise les équations normales augmentées :

$$N\delta_{\mathbf{x}} = J^T \epsilon \tag{1.66}$$

Avec N une matrice égale à $J^T J$, excepté sur sa diagonale où ses valeurs sont multipliées par un facteur d'amortissement positif λ, tel que :

$$N = J^T J + \lambda \operatorname{diag}(J^T J) \tag{1.67}$$

Ce facteur d'amortissement permet de modifier le comportement de l'algorithme. Lorsque la valeur de λ est faible, la méthode de Levenberg-Marquardt se rapproche d'une minimisation de type Gauss-Newton alors que lorsque la valeur de λ est forte, l'algorithme se rapproche d'une minimisation de type descente de gradient. Le paramètre λ est ajusté après chaque itération. Lorsque l'erreur globale augmente, c'est à dire que l'algorithme ne converge pas, on augmente la valeur de λ et on réitère. À l'inverse, lorsque l'erreur globale diminue, on réduit la valeur de λ afin de converger plus rapidement.

Erreur angulaire. Dans le cas d'un ajustement de faisceaux reposant sur l'erreur angulaire, le principe de la minimisation consiste à trouver pour chaque itération le pas $\delta_{\mathbf{x}}$ appliqué au vecteur $\mathbf{x}$ et réduisant l'erreur angulaire globale $\|\epsilon\|$, c'est à dire $g(\mathbf{y}, \mathbf{x} + \delta_{\mathbf{x}}) < g(\mathbf{y}, \mathbf{x})$. L'approximation de Taylor au premier ordre de g en $\mathbf{y}$ et $\mathbf{x}$ est de la forme :

$$g(\mathbf{y} + \delta_{\mathbf{y}}, \mathbf{x} + \delta_{\mathbf{x}}) \approx g(\mathbf{y}, \mathbf{x}) + J_{\mathbf{y}}\delta_{\mathbf{y}} + J_{\mathbf{x}}\delta_{\mathbf{x}} + \frac{\delta_{\mathbf{y}}\delta_{\mathbf{x}}}{2} J_{y,x} \tag{1.68}$$

Avec $J_{\mathbf{y}}$ la Jacobienne de g en $\mathbf{y}$ et $J_{\mathbf{x}}$ la Jacobienne de g en $\mathbf{x}$. L'optimisation n'affectant que le vecteur de mesures $\mathbf{x}$ (le vecteur d'observations $\mathbf{y}$ étant constant), l'équation précédente se réécrit alors :

$$g(\mathbf{y}, \mathbf{x} + \delta_{\mathbf{x}}) \approx g(\mathbf{y}, \mathbf{x}) + J_{\mathbf{x}}\delta_{\mathbf{x}} \tag{1.69}$$

Lorsque rapporté à la fonction de coût, l'optimisation consiste enfin à chercher le pas $\delta_{\mathbf{x}}$ minimisant $g(\mathbf{y}, \mathbf{x}) + J_{\mathbf{x}}\delta_{\mathbf{x}} = \epsilon + J_{\mathbf{x}}\delta_{\mathbf{x}}$, c'est à dire vérifiant les équations normales :

$$-J_{\mathbf{x}}^T J_{\mathbf{x}}\delta_{\mathbf{x}} = J_{\mathbf{x}}^T \epsilon \tag{1.70}$$

Ces equations sont alors de même forme que celles déjà introduites dans l'équation (1.65).

1.1.7.4 Stratégies d'optimisation

Plusieurs stratégies on été mises en place dans la littérature afin d'optimiser les poses et points 3D d'un système de SLAM par ajustement de faisceaux. Certaines sont des stratégies d'optimisation globales, c'est à dire que l'on optimise toute la séquence, de manière incrémentale ou hiérarchique [45, 140], ce qui s'avère généralement coûteux en temps de calcul et impose dans le cas hiérarchique d'avoir à disposition la séquence complète de toutes les images avant de procéder à la reconstruction. D'autres encore sont locales, c'est à dire que l'on optimise seulement les dernières occurrences de certaines temporalités de référence, que l'on appelle des images clés, chacune associée à un ensemble de points 3D [120, 194]. Dans les travaux de cette thèse, c'est cette dernière approche qui a été retenue, étant donné que la précision de la reconstruction dans sa globalité importe peu, seule la précision locale étant critique afin de reconstruire au mieux les objets mobiles dans les images.

1.2 SLAM visuel multi-objets

1.2.1 Problématique

Les méthodes de SLAM visuel présentées dans la section précédente, bien que satisfaisantes sur le plan applicatif, présentent certaines limitations fonctionnelles dans le contexte de cette thèse. En particulier, ces méthodes ne se concentrent uniquement que sur la reconstruction de scènes statiques, c'est à dire qu'elles n'assurent pas correctement la détection, la reconstruction ni le suivi des éventuels objets mobiles de la scène observée. En effet, quand bien même les méthodes basées sur le principe de *Stéréo Passive* et faisant usage de paires stéréo calibrées permettent en théorie la reconstruction à chaque instant

de n'importe quel point de l'espace en trois dimensions, elles n'intègrent cependant pas de mécanisme leur permettant de suivre temporellement les points mobiles, ce qui empêche toute détection de leur mouvement. Dans ces méthodes, les points mobiles peuvent alors être interprétés de deux manières. Une première consiste à les considérer à tort comme statiques pour chaque paire d'images de la séquence, ce qui entraîne la reconstruction de trajectoires flottantes, alors qu'une autre interprétation consiste tout simplement à les rejeter. L'approche *Multi-Objets*, introduite plus récemment, vise à étendre les méthodes de SLAM visuel classiques afin de permettre la reconstruction de scènes dynamiques présentant un ou plusieurs objets mobiles rigides, c'est à dire permettant de segmenter les points de la scène en plusieurs groupes correspondant chacun à un mouvement particulier. La figure 1.14 montre un exemple de segmentation multi-objets. Si les approches basées sur le principe de *Stéréo Passive* permettent notamment de calculer précisément le flot d'une scène (*Scene Flow*, en anglais), le problème de détection, reconstruction et suivi d'objets mobiles est plus difficile lorsque l'on passe au cas monoculaire, basé sur le principe de *Structure-from-Motion*, car il s'agit alors d'estimer simultanément la pose de la caméra dans le repère monde ainsi que le mouvement des points mobiles à chaque nouvelle image, ce qui implique le calcul de plusieurs contraintes épipolaires supplémentaires correspondant à chaque mouvement particulier. À cette difficulté s'ajoute le traitement des mouvements dégénérés se produisant dans l'un des plans épipolaires, pour lesquels les techniques évoquées jusqu'à présent ne sont plus assez robustes pour estimer le mouvement relatif des points par rapport à la caméra qui les observe, ou encore le problème d'estimation de l'échelle relative de chaque groupe de points par rapport aux autres lorsque leur reconstruction est réalisée de manière indépendante. Cette partie du manuscrit présentera certaines des méthodes de SLAM multi-objets de la littérature. En particulier, les sections 1.2.2 et 1.2.3 couvriront respectivement les méthodes monoculaires basées sur le principe de *SfM* et les méthodes stéréo basées sur le principe de *Stéréo Passive*. Enfin, il est important de rappeler que les méthodes présentées dans cette section ne concernent que la reconstruction d'objets rigides ou rigides par morceaux. Les méthodes de reconstruction d'objets déformables appartiennent elles à un ensemble de techniques très différentes qui ne seront pas abordées dans ce manuscrit.

1.2.2 Méthodes monoculaires

1.2.2.1 Segmentation de mouvement

Dans le contexte de la vision par ordinateur, la segmentation de mouvement consiste à regrouper les pixels ou points d'intérêt au sein d'images spatio-temporellement liées sur la base du mouvement particulier des objets auxquels ils appartiennent. De manière plus générale, ce problème peut également être considéré comme un problème de parti-

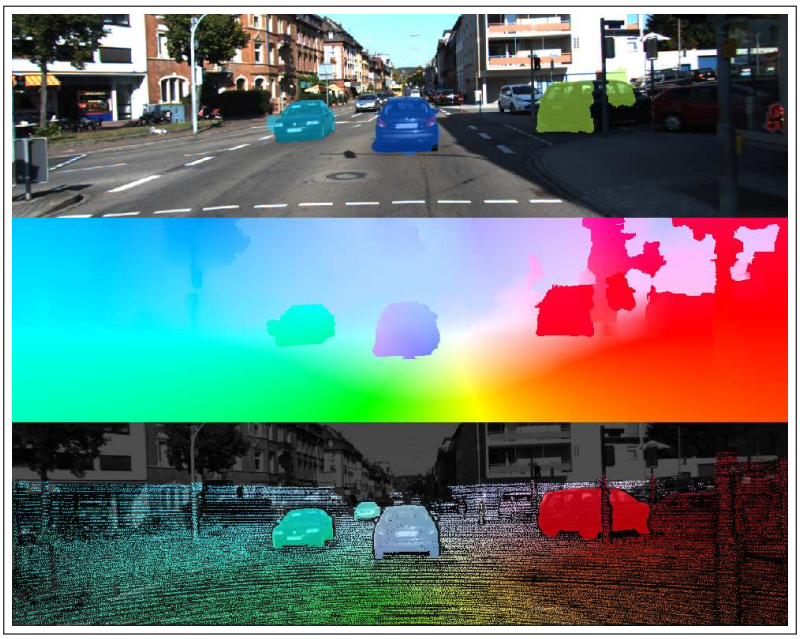

FIGURE 1.14 – Exemple de segmentation multi-objets extrait de l'article publié par Menze *et al.* en 2015 [111]. De haut en bas, le résultat de leur méthode de segmentation multi-objets, le flot de scène et enfin la vérité de terrain acquise grâce à un télémètre laser.

tionnement des données en sous-espaces de dimensions inférieures (*Subspace Clustering Problem*, en anglais), pour lequel l'objectif consiste à diviser les données en différents groupes homogènes appartenant chacun à un sous-espace dans lequel elles partagent des caractéristiques communes. En outre, la principale difficulté de ce type de problème provient du fait que le nombre de groupes ainsi que la dimension de leur sous-espace sont inconnus au départ. Bien que le cas général du problème de partitionnement vise à regrouper les données en sous-espaces de dimensions potentiellement différentes, le cas du SLAM visuel multi-objets en est une version simplifiée, étant donné que le problème se réduit au regroupement de trajectoires de points appartenant à de multiples sous-espaces affines. Plusieurs approches ont été proposées dans la littérature, dont la plupart ont été évaluées et comparées par Vidal dans [174]. On trouvera notamment des méthodes algébriques, itératives, statistiques ou encore basées sur le partitionnement spectral. D'un point de vue applicatif, plusieurs de ces méthodes n'effectuent qu'une segmentation globale des mouvements de la scène observée et ne fonctionnent ainsi que lorsque toutes les données sont disponibles avant de lancer la procédure de segmentation, ce qui limite leur utilisation dans le contexte de cette thèse, qui se concentre plus particulièrement sur la segmentation de mouvement incrémentale.

Méthodes algébriques. Plusieurs méthodes algébriques permettant de résoudre le problème de partitionnement en sous-espaces pour la segmentation de mouvement ont été proposées dans la littérature. Bien que généralement adaptées au traitement de données non-bruitées, des solutions modérément robustes au bruit existent. Parmi ces algorithmes, nous pouvons citer celui proposé par Costeira *et al.* [26], basé sur le principe de factorisation matricielle mais seulement adapté aux modèles de projection parallèle, plus tard amélioré par Zappella *et al.* [193] afin de le rendre robuste au suivi partiel des points d'intérêts et par Li *et al.* qui ont adapté la méthode aux caméras perspectives [95], ou encore la méthode d'analyse en composantes principales généralisée, ou GPCA (*Generalized Principal Component Analysis*, en anglais), proposée par Vidal *et al.* [176]. D'une part, les algorithmes basés sur la factorisation matricielle permettent la segmentation de mouvement par décomposition de la matrice des données en plusieurs matrices de rang faible. Ils se rapprochent par conséquent des méthodes d'analyse en composantes principales étendues à plusieurs sous-espaces linéaires indépendants. D'autre part, l'analyse en composantes principales généralisée permet de décomposer les données en plusieurs sous-espaces linéaires non-nécessairement indépendants. L'idée principale derrière la GPCA est de représenter les sous-espaces par un ensemble de polynômes dont le degré est égal aux nombre de sous-espaces et dont les dérivées en chaque point donnent les vecteurs normaux du sous-espace passant par ce point. La segmentation est ensuite obtenue par regroupement de ces vecteurs normaux. Bien que la méthode de base implique la connaissance à priori du nombre de sous-espaces à déterminer, les auteurs proposent également une extension permettant de résoudre le partitionnement lorsque ce n'est pas le cas. En pratique, la GPCA présente de bons résultats lorsque la scène présente plusieurs mouvements de points dépendants, ce qui est le cas des séquences routières ou articulées, mais montre rapidement ses limites lorsque plusieurs mouvements indépendants apparaissent, comme constaté dans [174]. Vidal *et al.* ont par la suite proposé une méthode géométrique de segmentation pour deux vues perspectives basée sur l'utilisation de la GPCA et l'introduction d'une généralisation multi-objets de la matrice fondamentale (*Multibody Fundamental Matrix*, en anglais) [177], étendue plus tard par Vidal et Hartley pour trois vues perspectives avec l'introduction du tenseur trifocal multi-objets (*Multibody Trifocal Tensor*, en anglais) [175]. Ces méthodes purement algébriques ne sont cependant pas robustes au bruit, ce qui limite fortement leur utilisation sur données réelles.

Méthodes itératives. Les méthodes itératives ont été introduites afin d'améliorer la robustesse au bruit des méthodes algébriques présentées dans le paragraphe précédent. Leur principe consiste à l'obtention d'une segmentation initiale, par factorisation matricielle ou GPCA dans un premier temps, puis à l'utilisation d'une ACP classique afin d'affiner chaque sous-espace. Les points sont ensuite assignés au sous-espace le plus proche.

L'itération de ces deux dernières étapes jusqu'à convergence produit une meilleure estimation de chaque sous-espace et ainsi une meilleure segmentation. Parmi les méthodes itératives existantes, nous pouvons citer notamment les algorithmes des K-plans [17] et K-sous-espaces [168], qui sont des généralisations de l'algorithme des K-moyennes [29] à des données appartenant à des hyperplans et sous-espaces affines, respectivement.

Méthodes statistiques. Les méthodes algébriques et itératives présentées dans les précédents paragraphes permettent la segmentation de mouvement dans le cas de données modérément bruitées, mais ne modélisent ni ne prennent en compte la distribution des données au sein des sous-espaces, ni la distribution du bruit leur étant associé. Les estimations générées par ces méthodes ne sont donc pas optimales. À l'inverse, des méthodes statistiques telles que la MPPCA (*Mixtures of Probabilistic PCA*, en anglais) [163], l'ALC (*Aglomerative Lossy Compression*, en anglais) [104], la méthode par sélection de modèles minimisant la taille de la description [138] introduite par Schindler *et al.* [152] ou encore RANSAC sont des modèles génératifs qui intègrent ces informations de distribution. En pratique, l'utilisation de RANSAC afin d'estimer successivement plusieurs matrices fondamentales correspondant aux différents mouvements de la scène est très rapide et comparable qualitativement à la GPCA pour la détection de quelques mouvements indépendants, mais n'est plus assez efficace lorsque leur nombre augmente, comme montré dans [174]. Une autre approche probabiliste a été proposée par Kundu *et al.* [81, 80] et permet la détection de mouvements indépendants, y compris dans les plans épipolaires, grâce à l'utilisation de la contrainte FVB (*Flow Vector Bound*, en anglais), qui consiste à assigner pour chaque point d'intérêt de la scène des seuils de profondeur minimum et maximum, puis à calculer leur distance relative par rapport aux lignes épipolaires. Cette contrainte a plus tard été reprise dans la méthode dense proposée par Namdev *et al.* [124], en l'utilisant conjointement au calcul de flot optique afin de construire un modèle probabiliste de la segmentation, minimisé ensuite par coupe de graphe. Une autre méthode [46], inspirée des travaux de Schindler *et al.*, propose une segmentation rapide intégrant une dernière étape basée sur le partitionnement spectral, mais nécessite cependant la connaissance à priori du nombre de mouvements indépendants au sein de la scène. Plus récemment, Ranftl *et al.* [136] ont introduit une méthode dense basée sur la minimisation d'une fonction énergie permettant le recalage de modèles géométriques multiples inspirée des travaux de Isack et Boykov [71] alors que Sabzevari et Scaramuzza [142, 143] ont proposé une méthode basée sur la factorisation d'une matrice de trajectoires dont les hypothèses sont générées grâce à une approche multi-RANSAC [199] permettant la détection d'homographies planaires correspondant aux contraintes de mouvements planaires et planaires circulaires appliquées aux objets mobiles.

Méthodes basées sur le partitionnement spectral. Les méthodes basées sur le partitionnement spectral, pour lesquelles une évaluation a été proposée dans [182], sont particulièrement populaires lorsqu'il s'agit de segmenter des données appartenant à un espace de dimension élevée. L'idée principale de ces algorithmes consiste à construire plusieurs matrices d'affinité permettant de mesurer la similarité entre deux points, formant ainsi les arrêtes d'un graphe, complet ou non. Lors de l'étape suivante, les vecteurs propres de ces matrices d'affinité sont ensuite utilisés afin de construire une autre matrice représentant la projection des données de l'espace original vers un espace de dimension inférieure. La segmentation est enfin obtenue en appliquant la méthode des K-moyennes sur cette dernière matrice. Parmi les méthodes basées sur le partitionnement spectral, on peut citer notamment les méthodes dont le calcul des matrices d'affinités se base sur la factorisation matricielle [16], GPCA, LSA (*Local Subspace Affinity*, en anglais) [191], SLBF (*Spectral Local Best-fit Flats*, en anglais) [196], LLMC (*Locally Linear Manifold Clustering*, en anglais) [57], SSC (*Sparse Subspace Clustering*, en anglais) [32], LRR (*Low-Rank Representation*, en anglais) [98] ou encore SCC (*Spectral Curvature Clustering*, en anglais) [23]. Ces méthodes se révèlent performantes sur le plan de la segmentation des mouvements, notamment la méthode SSC ayant obtenu les meilleurs résultats [174] sur le jeu de données Hopkins 155 introduit par Tron et Vidal [167], au prix cependant de temps de calcul assez longs. Plus récemment, Ji *et al.* [72] ont proposé une méthode réalisant la segmentation conjointement à l'appariement de points d'intérêt en formulant le problème en termes de matrices de permutation partielles (*Partial Permutation Matrices*, en anglais) afin d'appareiller les points selon leurs descripteurs et dont la trajectoire satisfait les contraintes du sous-espace correspondant.

1.2.2.2 Reconstruction multi-objets

La reconstruction multi-objets comporte quelques spécificités supplémentaires afin de parvenir à un résultat cohérent. Deux des étapes supplémentaires à prendre en compte après l'étape de segmentation concernent notamment le suivi des objets mobiles ainsi que l'estimation de leur échelle relative par rapport à celle de la scène.

Suivi d'objets mobiles. Le processus de reconstruction multi-objets basé sur le principe de *SfM* se compose d'une première étape, décrite dans la section précédente et relativement bien étudiée dans la littérature, qui consiste à segmenter les points d'intérêt sur la base de leur mouvement propre au sein de la scène. À l'issue de cette étape, le résultat obtenu consiste alors en plusieurs groupes de points d'intérêt correspondant chacun à une contrainte épipolaire. Une des limitations engendrées par ce résultat provient cependant du fait que les points d'intérêt au sein d'un groupe n'appartiennent pas forcément tous au même objet physique, alors qu'en pratique les différents mouvements

de la scène sont chacun produits par un objet rigide ou rigide par morceaux. En outre, ces mouvements sont susceptibles d'apparaître, disparaître, fusionner et se fractionner lorsque l'on raisonne en termes épipolaires. Pour l'exemple, les points d'intérêt appartenant à deux voitures roulant sur la même voie à même allure satisfont naturellement la même contrainte épipolaire, ce qui conduit à une représentation correcte de deux voitures par un seul groupe de points si l'on ne se base que sur leur mouvement propre. La scène peut ensuite évoluer de plusieurs manières. Si l'on imagine que l'une des voitures tourne brusquement, l'unique groupe de points se scindera en deux afin de représenter les deux mouvements. Si l'une des voitures s'arrête, le groupe se scindera à nouveau en deux puis fusionnera avec le mouvement nul du repère monde, satisfaisant alors la contrainte épipolaire générale de la scène. Une deuxième étape à considérer concerne donc le suivi à proprement parler de ces objets rigides, et non le suivi individualisé des points d'intérêt qui les composent. Ozden *et al.* [129] ont montré qu'un suivi correct de ces groupes de points d'intérêt permet non seulement une reconstruction plus rapide et de meilleure qualité mais également une meilleure robustesse aux occultations et au problème d'échelle relative. Plus précisément, la prise en considération correcte de la fusion de deux groupes de points peut permettre la réduction du nombre de paramètres à estimer pour la reconstruction (lorsqu'un objet s'immobilise et fusionne avec la scène rigide du repère monde, par exemple, il est plus rapide et précis d'utiliser la géométrie épipolaire de la scène statique que de ré-estimer la pose relative de l'objet de manière spécifique). Il en est de même pour le cas du fractionnement d'un groupe de points, pour lequel il est alors possible d'utiliser l'arrangement 3D des points rigides déjà reconstruits afin de ré-estimer la pose relative des nouveaux objets par PnP. Enfin, comme abordé dans le paragraphe précédent, le suivi d'objets permet également de résoudre le problème de l'échelle relative de chaque objet par rapport à la scène, étant donné que la fusion et le fractionnement sont transitifs, c'est à dire que le facteur d'échelle se propage lorsque les objets interagissent. L'algorithme proposé par Ozden *et al.* consiste donc en partie à réaliser un graphe représentant l'évolution conjointe des différents objets de la scène, alors que l'aspect segmentation utilise un modèle probabiliste basé sur la sélection de modèles introduite par Schindler *et al.* [152]. Le suivi d'objets réalisé par Kundu *et al.* [80] fait quant à lui usage de filtres particulaires de type *BOT* (*Bearing-Only Tracking*, en anglais) associés à chaque objet afin d'en estimer la vitesse et la trajectoire.

Estimation de l'échelle relative des objets mobiles. Le problème d'estimation de l'échelle relative des objets mobiles provient du fait que chaque objet mobile est reconstruit de manière indépendante après l'étape de segmentation dans la plupart des méthodes. En effet, ces objets mobiles sont chacun reconstruits en rapport à leur contrainte épipolaire particulière, qui définit arbitrairement et de manière indépendante l'échelle

de chaque objet. Plusieurs solutions à ce problème ont été proposées, notamment par [129, 80, 123, 136]. Alors qu'Ozden *et al.* estiment l'échelle de chaque objet par rapport aux autres grâce à l'utilisation d'un graphe de composantes connectées transitif défini par la fusion ou division des différents objets de la scène, Kundu *et al.* choisissent le facteur d'échelle de chaque objet tel que la valeur moyenne de l'échelle des particules des filtres particulaires associés un à un à chaque objet. Namdev *et al.* ont ensuite proposé une solution au problème pour les mouvements non-holonomes planaires ou localement linéaires. Enfin, la méthode dense proposée par Ranftl *et al.* [136] repose sur l'utilisation de deux contraintes, l'une d'ordre hiérarchique, c'est à dire que les objets mobiles occultent l'environnement statique, et l'autre assurant une certaine régularité de la scène, c'est à dire que lorsque l'on considère que les objets mobiles reposent sur l'environnement statique, la différence de profondeur au niveau de la jonction entre les plans appartenant à l'objet et les plans appartenant à l'environnement, c'est à dire le sol par exemple, doit être minimale.

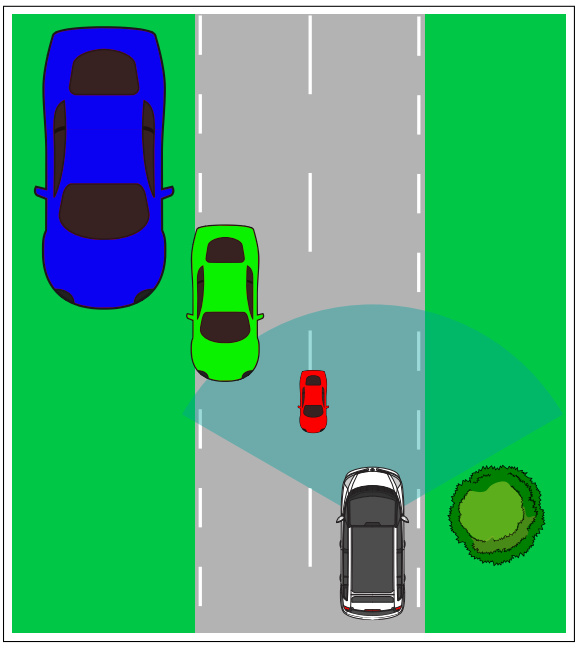

FIGURE 1.15 – Estimation de l'échelle relative des objets mobiles reconstruits de manière indépendante. Dans cet exemple, trois échelles de reconstruction du même objet sont illustrées. Le problème d'estimation de l'échelle relative consiste à retrouver la taille réelle de l'objet dans le référentiel de la scène.

1.2.2.3 Discussion sur les méthodes monoculaires

Parmi les méthodes monoculaires présentées dans cette sous-section, seules quelques unes s'approchent du cadre expérimental dans lequel s'inscrivent les travaux de cette

thèse. En premier lieu, la plupart de ces méthodes ont une approche globale et non incrémentale de la segmentation multi-objets, c'est à dire que leur point de départ impose la connaissance à priori de toutes les données de la séquence à segmenter (c'est à dire toutes les images), à l'exception de celles proposées par Kundu *et al.* [80], Namdev *et al.* [123] ou encore Sabzevari *et al.* [143], qui sont donc potentiellement adaptées au contexte automobile qui nous intéresse. Ces trois méthodes présentent néanmoins quelques limitations. En effet, étendre l'approche monoculaire proposée aux systèmes multi-caméras requiert le suivi 2D et le recalage 3D de chaque objet mobile reconstruit lorsque celui-ci est à nouveau ou simultanément observé par une autre caméra. Ce type d'architecture n'est pas idéalement adapté aux systèmes multi-caméras actuellement intégrés dans la plupart des véhicules intelligents, présentant pour majorité des champs recouvrants, car il impose ces deux étapes supplémentaires de suivi et de recalage, nécessaires à la fusion de tous les champs de vue, qu'un système multi-caméra stéréo intègre directement dès l'étape d'appariement de points d'intérêt. Une autre limitation levée par les systèmes stéréo concerne l'estimation correcte du facteur d'échelle relatif des objets mobiles reconstruits. Dans les méthodes [129], [80] et [123] ce facteur d'échelle relative nécessite l'ajout de plusieurs contraintes supplémentaires à la contrainte épipolaire (voir paragraphe précédent).

1.2.3 Méthodes stéréo

Le SLAM multi-objets basé sur le principe de *Stéréo Passive* a été abordé de plusieurs manières dans la littérature. Il existe des méthodes denses, semi-denses et des méthodes éparses. En premier lieu, les méthodes denses reposent sur les avancées récentes pour le calcul de flot optique dense généralisé à trois dimensions. Ces techniques sont regroupées sous le terme de flot de scène (*Scene flow*, en anglais) et consistent à conjointement estimer la géométrie et le mouvement 3D de tous les pixels d'une séquence d'images. En second lieu, les méthodes éparses développées plus tard ne se concentrent que sur la reconstruction de points d'intérêt et usent, pour la majorité, des techniques de reconstruction classiques vues dans la section précédente auxquelles sont ajoutées une étape de segmentation de mouvement, bien souvent basée sur le calcul du flot de scène particulier de chaque point d'intérêt. De manière plus générale, bien qu'il existe des méthodes éparses [89, 137], les méthodes denses sont beaucoup plus représentées dans la littérature, en raison de l'utilisation quasi exclusive de paires stéréo homogènes et calibrées qui permettent le calcul de flot optique dense de manière efficace [54].

1.2.3.1 Méthodes denses

Les travaux précuseurs de Vedula *et al.* [172, 173], premiers à avoir introduit le concept d'estimation de flot de scène, reposent sur deux étapes qui consistent à calculer de manière indépendante le flot optique pour toutes les vues de la scène puis à recaler la carte des disparités 3D sur les flots 2D de chaque vue afin d'obtenir le flot de scène. Plusieurs méthodes ont adopté une approche variationnelle et consistent à optimiser de manière incrémentale la régularité locale de la profondeur et du mouvement de chaque pixel. Wedel *et al.* [184, 183], dont les travaux ont inspiré ceux de Rabe *et al.* [135], ont proposé une méthode permettant d'estimer le flot optique d'une image de référence et la différence de disparité pour l'autre vue. Huguet *et al.* [70] ont proposé une estimation conjointe de la géométrie et du flot optique pour deux vues, plus tard généralisé par Valgaerts *et al.* [170] lorsque les paramètres extrinsèques des caméras ne sont pas connus. Basha *et al.* [9] ont pour leur part utilisé une paramétrisation 3D de la profondeur et du mouvement de chaque pixel par rapport à chaque vue de référence afin d'estimer tous les paramètres conjointement. La méthode introduite par Pons *et al.* [133] minimise l'erreur de prédiction de ces estimations par une mesure globale de similarité entre l'image en entrée et l'image prédite. D'autres approches, inspirées des développements récents pour le calcul de flot optique [126, 162, 189, 188] et de stéréo dense [14, 13, 190], ont été proposées par Vogel *et al.* [179, 178, 180] qui ont représenté la scène telle qu'une collection de plusieurs plans rigides mobiles et dont la méthode a été plus tard améliorée par Menze *et al.* [111].

1.2.3.2 Méthodes éparses

Lenz *et al.* [89] ont proposé une méthode de segmentation calculant dans un premier temps le flot de scène de points d'intérêts suivant la méthode proposée par Geiger *et al.* [54]. Ces points d'intérêt sont ensuite triangulés grâce à la méthode de Delaunay afin de former un graphe de composantes connectées, à son tour segmenté par élimination des arrêtes reliant les points dont les disparités sont trop éloignées. Ces travaux ont plus tard été repris par Reddy *et al.* [137] qui ont proposé une méthode probabiliste basée sur l'utilisation de graphes de poses (*Pose Graphs*, en anglais) afin d'améliorer la précision de la reconstruction des objets mobiles.

Chapitre 2

SLAM visuel multi-objets pour systèmes multi-caméras hétérogènes en stéréo *wide-baseline*

2.1 Problématique industrielle

La problématique industrielle de cette thèse vise pour rappel à étendre de manière fonctionnelle les systèmes multi-caméras déjà implantés au sein de certains véhicules. Le choix de l'application envisagée, c'est à dire la reconstruction incrémentale de l'environnement dynamique du véhicule, a été dicté par les possibilités offertes par le type et l'arrangement des capteurs. Leurs paramètres intrinsèques et extrinsèques ont ainsi constitué le point de départ de la réflexion sur la méthode à aborder. Considérant que ce système multi-caméras est à champs recouvrants, l'aspect stéréo a été exploité pour la détection des objets mobiles de la scène, afin notamment d'éviter l'estimation complexe de leur échelle relative, comme abordé dans le paragraphe correspondant de la sous-section 1.2.2.2. Cependant, l'utilisation efficace des méthodes de SLAM visuel multi-objets existantes, basées sur le principe de *Stéréo Passive* et présentées dans la sous-section 1.2.3 a été compromise par l'arrangement wide-baseline et multi-focales du système considéré. En effet, la plupart des méthodes stéréo de la littérature reposent sur l'utilisation de caméras identiques, positionnées à faible distance l'une de l'autre, ce que l'on appelle *short-baseline*, et pointant dans le même sens. Ce type de configuration génère des images relativement proches, permettant ainsi d'envisager un appariement de points dense. Seulement, le problème d'appariement est beaucoup plus difficile lorsque les images présentent de fortes disparités, ce qui est le cas de la configuration expérimentale définie dans ces travaux et qui a orienté les recherches vers le développement d'une approche éparse et purement géométrique (contrairement aux approches variationnelles) de détec-

tion et reconstruction de points mobiles. Par ailleurs, l'aspect incrémental de la méthode envisagée a également exclu les approches globales de certaines méthodes monoculaires, basées sur l'utilisation d'un nombre élevé de vues successives afin de détecter le mouvement des points mobiles, ce qui a conduit à la mise en place d'une limitation à trois temporalités consécutives dans la méthode proposée.

2.2 Vue d'ensemble

La méthode proposée comporte plusieurs modules et étapes, décrits dans cette section. La première étape, qui n'est effectuée qu'une seule fois pour chaque configuration expérimentale, consiste à calibrer les paramètres intrinsèques et extrinsèques du système multi-caméras. A cette fin, la méthode d'étalonnage multi-caméras proposée par Lébraly *et al.* [86] a été utilisée. Les étapes suivantes, appartenant à deux modules indépendants, sont ensuite réitérées pour chaque nouvelle temporalité, c'est à dire chaque nouvel ensemble d'images acquises par le système multi-caméras. Le premier module présenté dans la section 2.3 est une méthode de SLAM visuel permettant d'estimer précisément le déplacement du système multi-caméras. Vient ensuite le module de détection et reconstruction de points mobiles, présenté dans la section 2.4 et dont la première étape, détaillée dans la sous-section 2.4.1, consiste à extraire, suivre et appareiller une nouvelle fois les points d'intérêt dans les images, indépendamment des points utilisés dans le module de SLAM visuel. Cette décision a été motivée par trois raisons. D'une part, la mesure indépendante de l'odométrie du véhicule autorise l'utilisation de techniques d'estimation différentes à l'odométrie visuelle, reposant éventuellement sur d'autres sources pour la mesure. D'autre part, le module de SLAM visuel implémenté fonctionne de manière monoculaire, sur le principe de *SfM* et ne gère donc pas les appariements stéréo. Enfin, cela permet de mesurer l'impact de l'étape d'extraction et d'appariement de points d'intérêt sur les résultats produits par le module de détection et reconstruction de points mobiles uniquement. L'odométrie du système multi-caméras ainsi que les appariements spécifiques au module de détection et reconstruction de points mobiles permettent alors le calcul des contraintes multi-vues du processus de segmentation présenté dans la sous-section 2.4.3. Enfin, les procédures de SLAM visuel et de segmentation comportent chacune une étape d'optimisation. L'optimisation associée au module de SLAM visuel affecte les poses des caméras ainsi que les points 3D reconstruits par ajustement de faisceaux local minimisant l'erreur de reprojection, alors que celle associée au module de détection et reconstruction de points mobiles n'affecte que les points 3D reconstruits par ajustement de faisceaux local minimisant l'erreur de reprojection ou l'erreur angulaire. Une vue d'ensemble de la méthode proposée est présentée dans la figure 2.1.

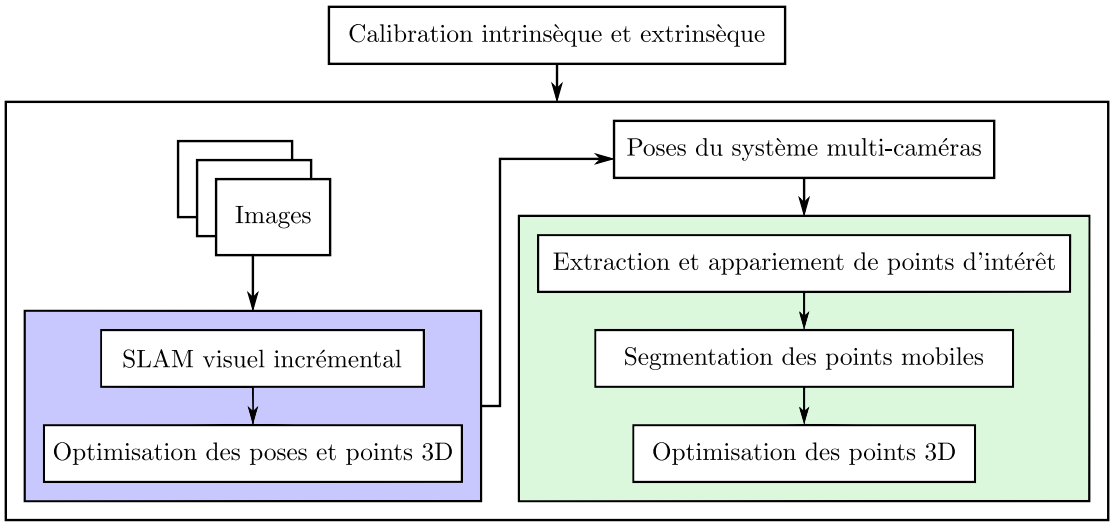

FIGURE 2.1 – Vue d'ensemble de la méthode proposée, avec en bleu le module de SLAM visuel incrémental et en vert le module de détection et reconstruction de points mobiles.

2.3 Module de SLAM visuel multi-caméras incrémental

Plusieurs stratégies sont envisageables lors de l'implémentation d'un système de SLAM visuel multi-caméras incrémental, dont en particulier celle adoptée afin d'initialiser le système, celle utilisée ensuite lorsque l'on dispose déjà de la géométrie initiale de la scène à reconstruire, puis enfin la stratégie d'optimisation multi-caméras. Dans ces travaux, nous avons largement réutilisé la méthode proposée par Mouragnon *et al.* [120], ainsi qu'une extension multi-caméras proposée par Lébraly *et al.* [86], pour sa robustesse, précision et flexibilité. Plus précisément, l'approche retenue consiste à estimer de manière indépendante les structures géométriques associées à chaque caméra par *SfM*, puis à fusionner ces structures par optimisation de type ajustement de faisceaux multi-caméras des poses et points 3D.

2.3.1 Modèles de caméras retenus

Dans la sous-section 1.1.2, deux modèles sont présentés, chacun offrant avantages et inconvénients. En pratique, les travaux réalisés dans cette thèse font usage des deux modèles. Le modèle unifié a été utilisé lors de la calibration intrinsèque et extrinsèque du système multi-caméras présenté dans le chapitre 3, ainsi que pour la correction de points 2D distordus. Les équations de projection et de lancer de rayon du modèle sténopé classique ont en revanche été privilégiées en présence de points corrigés.

2.3.2 Extraction et appariement de points d'intérêt

Dans la sous-section 1.1.3, plusieurs détecteurs et descripteurs de points d'intérêt sont présentés. En pratique, ceux retenus dans ces travaux pour la mise en œuvre du module de SLAM visuel sont les détecteurs de coins de Harris ainsi que des descripteurs de type bloc. Lors de l'étape d'appariement de points, le descripteur d'une première image est appareillé au meilleur descripteur de l'image suivante parmi ceux présents dans la fenêtre de recherche centrée sur les coordonnées du point de la première image, alors que l'évaluation du score de ressemblance est calculé par corrélation croisée normalisée (ZNCC). Lorsque ce score est trop faible pour le meilleur appariement, c'est à dire en dessous d'un seuil fixé au préalable, l'appariement n'est pas retenu. Le processus d'extraction de points d'intérêt produit, pour chaque image acquise par la caméra j au temps t, l'ensemble de points d'intérêt $\mathtt{p}_{j,t}$, alors que les appariements correspondants à deux ensembles $\mathtt{p}_{j,t}$ et $\mathtt{p}_{j',t'}$ sont désignés par $\mathtt{m}_{j,j'}^{t,t'}$. À noter qu'alors que les descripteurs sont directement calculés sur l'image distordue, les ensembles de points d'intérêt $\mathtt{p}_{j,t}$ de chaque vue sont ensuite corrigés selon la technique de correction abordée dans la sous-section 1.1.2.8. Ce sont ces points corrigés qui sont ensuite utilisés dans le reste de la section.

2.3.3 Estimation robuste de la géométrie initiale

L'estimation robuste de la géométrie initiale d'une séquence d'images grâce aux contraintes épipolaires ne peut se faire directement sur chacune des toutes premières images de la séquence, mais nécessite la sélection de temporalités de référence appelées images clés, présentées dans la dernière partie de la sous-section 1.1.7. Comme indiqué dans cette partie, ces images clés permettent la mise en œuvre d'algorithmes d'optimisation par ajustement de faisceaux locaux, mais permettent également d'assurer que le déplacement de la caméra ne soit trop faible, évitant ainsi d'engendrer un calcul mal conditionné de la matrice essentielle. La stratégie employée par la plupart des systèmes de la littérature existants [140, 120] consiste à sélectionner l'image clé suivante selon un intervalle maximisant le déplacement de la caméra et pour lequel le nombre de points appareillés avec l'image clé courante ne se situe pas en dessous d'un certain seuil $\mathtt{T}_K$. La géométrie initiale de la scène est ensuite calculée sur le premier triplet d'images clés selon une méthode RANSAC itérative en trois temps. La première étape consiste, à partir d'un échantillon de cinq points appareillés dans les trois images, à estimer la matrice essentielle entre la première et la troisième image grâce à l'algorithme des cinq points proposé par Nistér [127] afin de trianguler un nuage de cinq points 3D (sous-section 1.1.4). Dans la deuxième étape est ensuite calculée la pose de la deuxième caméra correspondant aux cinq points 3D triangulés via la technique PnP proposée par Grunert

[58] (sous-section 1.1.6). La dernière étape consiste enfin à trianguler grâce aux poses estimées la totalité des appariements entre les trois images via la méthode du point-milieu (sous-section 1.1.5). Le triplet de poses retenu est celui maximisant le nombre de points 3D cohérents, c'est à dire se reprojetant correctement dans chacune des trois images. La figure 2.2 illustre l'estimation robuste de la géométrie initiale grâce au premier triplet d'images clés.

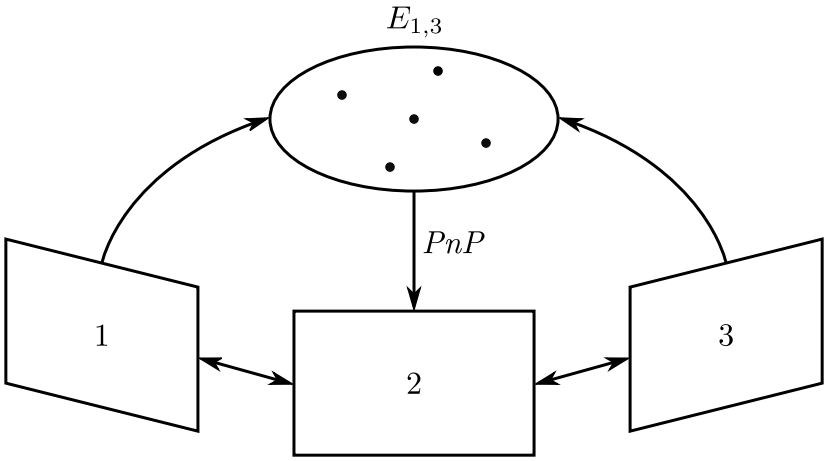

FIGURE 2.2 – Estimation robuste de la géométrie initiale. Dans cette illustration, $E_{1,3}$ est la matrice essentielle entre les images 1 et 3.

2.3.4 Calcul de pose grâce à la géométrie existante

Après estimation robuste de la géométrie initiale grâce au premier triplet d'images, les poses suivantes des caméras sont estimées par PnP grâce aux appariements entre les points d'intérêt de l'image courante et ceux déjà triangulés de l'image clé précédente, comme expliqué dans la sous-section 1.1.6. Ces nouvelles poses permettent ensuite la triangulation des appariements de points nouvellement observés à la dernière temporalité $t = T$ et satisfaisant la contrainte épipolaire de la scène.

2.3.5 Optimisation par ajustement de faisceaux multi-caméras

La méthode d'optimisation par ajustement de faisceaux du système multi-caméras utilisée dans le module de SLAM visuel s'inspire de celle développée par Lébraly *et al.* [86]. L'approche consiste en premier lieu à considérer que le système multi-caméras se compose d'une caméra maître ainsi que d'une ou plusieurs caméras supplémentaires. L'optimisation n'intervenant qu'après avoir estimé de manière indépendante la géométrie de la scène correspondante à chaque caméra du système par *SfM*, son principe réside ensuite dans l'exploitation de la contrainte de rigidité du système multi-caméras, qui

consiste à redéfinir la pose de chaque caméra relativement à la pose de la caméra maître et permet ainsi l'optimisation d'une unique pose lors de l'ajustement de faisceaux. Cette redéfinition implique l'utilisation des paramètres extrinsèques propres à chaque caméra du système, invariants car calculés en amont lors de l'étape de calibration. Plus en détail, on considère la pose $C_{M,t}$ de la caméra maître M ainsi que la pose $C_{j,t}$ de la caméra j à l'instant t. Les paramètres extrinsèques de la caméra j se composent d'une matrice de rotation R_M^j et d'un vecteur translation $\mathbf{t}_M^j$ formant la transformation homogène T_M^j définie telle que :

$$T_M^j = \begin{bmatrix} R_M^j & \mathbf{t}_M^j \\ 0_{1\times 3} & 1 \end{bmatrix} \tag{2.1}$$

Chaque transformation homogène T_M^j permet de passer du repère de la caméra maître M à celui de la caméra j et inversement selon :

$$C_{M,t} = T_M^j C_{j,t} \qquad\qquad C_{j,t} = T_M^{j-1} C_{M,t} \tag{2.2}$$

La réécriture des poses de chaque caméra du système relativement à la pose de la caméra maître est reportée dans le calcul de l'erreur de reprojection utilisée lors de l'étape d'optimisation par ajustement de faisceaux présentée dans la sous-section 1.1.7. Pour rappel, l'équation (1.56) de reprojection classique d'un point 3D P_n sur le plan image de la caméra j à l'instant t est définie telle que :

$$f_{j,t,n}(C_{j,t}, P_n) = [P]_{j,t} P_n = \pi \left(K_j \left[I_3 | 0_{3\times 1} \right] C_{j,t} P_n \right)$$

Lors de l'optimisation multi-caméras, la pose $C_{j,t}$ passée en paramètre de la fonction de projection $f_{j,t,n}$ est ainsi remplacée par la pose $C_{M,t}$, tel que :

$$f_{j,t,n}(C_{M,t}, P_n) = [P]_{j,t} P_n = \pi \left(K_j \left[I_3 | 0_{3\times 1} \right] T_M^{j-1} C_{M,t} P_n \right) \tag{2.3}$$

L'ajustement de faisceaux achevé, la pose de chaque caméra du système est enfin retrouvée grâce à la nouvelle pose de la caméra maître suivant la relation (2.2).

2.3.6 Implémentation

L'algorithme 1 présente l'implémentation du module de SLAM visuel incrémental. Pour rappel, cet algorithme permet principalement l'estimation visuelle de l'odométrie du système multi-caméras. Afin d'expliciter quelque peu le pseudo-code présenté ci-après, sont introduites quelques notations et définitions. Le système multi-caméras se compose de plusieurs caméras synchronisées j produisant chacune une image à chaque temporalité t. Si les points d'intérêt associés à chaque image forment l'ensemble $\mathtt{p}_{j,t}$,

les points 3D associés à ces points d'intérêt, s'ils existent, sont désignés par $\mathtt{P}_{j,t}$. La variable k désigne la dernière occurrence de temporalité clé, avec $k \in 0 \dots T$ et enfin, la fonction $Estimation\_Robuste\_Geometrie\_Initiale()$ est détaillée dans la sous-section 2.3.3, alors que la fonction $Ajustement\_Faisceaux\_MultiCameras()$ est détaillée dans la sous-section 2.3.5.

Algorithme 1 : Module de SLAM visuel incrémental multi-caméras.

Données : Images acquises par les caméras j à chaque temporalité t.
Résultat : Poses $C_{j,t}$ des caméras et points 3D associés $\mathtt{P}_{j,t}$.
début
  pour chaque t **faire**
    pour chaque j **faire**
      si $Geometrie\_Existante()$ **alors**
        $\mathtt{p}_{j,t} = Extraction\_Points\_Interet(j,t)$;
        $\mathtt{m}_{j,j}^{k,t} = Appariement(\mathtt{p}_{j,k}, \mathtt{p}_{j,t})$;
        $C_{j,t} = RANSAC\_PnP(\mathtt{P}_{j,k}, \mathtt{m}_{j,j}^{k,t}, \mathtt{p}_{j,t})$;
        $\mathtt{P}_{j,t} = Triangulation(C_{j,k}, C_{j,t}, \mathtt{p}_{j,k}, \mathtt{p}_{j,t})$;
        si $\mathtt{m}_{j,j}^{k,t} < \mathtt{T}_K$ **alors**
          $k = t$;
        fin
      sinon
        $\mathtt{p}_{j,t} = Extraction\_Points\_Interet(j,t)$;
        $\mathtt{m}_{j,j}^{k,t} = Appariement(\mathtt{p}_{j,k}, \mathtt{p}_{j,t})$;
        si $\mathtt{m}_{j,j}^{k,t} < \mathtt{T}_K$ **alors**
          $k = t$;
        fin
        si $Nombre\_Images\_Cle == 3$ **alors**
          $Estimation\_Robuste\_Geometrie\_Initiale()$;
        fin
      fin
    fin
    $Ajustement\_Faisceaux\_Multi\_Cameras()$;
  fin
fin

2.4 Module de détection et reconstruction de points mobiles

2.4.1 Extraction et appariement de points d'intérêt

L'appariement dense de points d'intérêt en stéréo a été relativement bien étudié dans le cas de la reconstruction de scènes dynamiques. Comme évoqué dans le premier

chapitre, la plupart des techniques de la littérature reposent sur le calcul du flot de scène afin de segmenter les mouvements rigides de la scène observée [111]. En revanche, bien qu'il soit en théorie possible de calculer le flot optique dense de deux images obtenues à partir de caméras stéréo relativement éloignées [164], les méthodes le permettant sont actuellement coûteuses en temps de calcul. L'approche retenue dans ces travaux, bien que conventionnelle, produit des appariements de points d'intérêt relativement robustes dans le cas qui nous intéresse ici, c'est à dire en wide-baseline et stéréo multi-focale.

2.4.1.1 Extraction de points d'intérêt

Plusieurs détecteurs et descripteurs de points d'intérêt on été utilisés dans ces travaux, dont en particulier les détecteurs et descripteurs SIFT [100], ORB [141], AKAZE [1] ou encore DAISY [164], qui produisent un nombre élevé de points relativement stables et dont l'influence est comparée dans le chapitre 3. Le résultat produit par le processus d'extraction est un ensemble de points d'intérêt $\mathtt{p}_{j,t}$ associé à chaque image. Ce processus d'extraction de points d'intérêt procède en trois étapes pour chaque nouvelle vue, dans l'espace image distordu.

1. Extraction de l'ensemble de points d'intérêt S_1. Dans cette étape, une première détection et description des points d'intérêt de l'image est réalisée. Cette détection est d'abord effectuée sur toute l'image afin d'assurer la détection des points d'intérêt les plus significatifs. L'image est ensuite divisée selon une grille n par n, formant ainsi plusieurs groupes de points correspondant chacun à une case de la grille. Les points d'intérêt présentant la meilleure réponse dans chaque case sont retenus à l'issue de cette étape et forment l'ensemble S_1. Cette étape permet d'assurer une répartition relativement uniforme des points d'intérêt dans l'image, mais évite également la détection d'un nombre trop élevé de points dans les zones très texturées.

2. Extraction de l'ensemble de points d'intérêt S_2. Cette étape permet d'améliorer le suivi temporel des points d'intérêt déjà triangulés grâce à la méthode proposée par Lucas et Kanade [101], qui, avec les travaux de Horn et Schunck [68], font partie des premières méthodes de calcul du flot optique entre deux images. Brièvement, l'ensemble des points d'intérêt observés par la même caméra et détectés à la temporalité précédente sont le point de départ d'un problème de recalage dans l'image suivante, c'est à dire que l'on cherche à trouver le vecteur exprimant le déplacement de chaque point d'intérêt d'une temporalité à la suivante au sein d'une zone de recherche prédéfinie.

3. Fusion des ensembles S_1 et S_2. La dernière étape procède enfin à la fusion des ensembles S_1 et S_2, avec en premier lieu une étape d'élimination des doublons, c'est à

dire les points dont la distance euclidienne en pixels est très faible. Sur les points restants, ceux de S_2 sont retenus et complétés par les meilleurs points de S_1, classés par réponse pour chaque case de la grille, afin d'obtenir un ensemble de points $\mathtt{p}_{j,t}$ dont le nombre total est seuillé par un maximum.

2.4.1.2 Correction directe des distorsions

Les points retenus pour chaque image $\mathtt{p}_{j,t}$ sont ensuite corrigés selon la méthode de correction des distorsions dans le modèle de caméra unifié, détaillée dans la sous-section 1.1.2.8. Ce sont ces points qui sont ensuite utilisés dans le reste de la section.

2.4.1.3 Appariement de points d'intérêt entre deux vues

Suite à l'extraction des points d'intérêt de chaque image, le processus d'appariement entre deux ensembles de points d'intérêt $\mathtt{p}_{j,t}$ et $\mathtt{p}_{j',t'}$ repose sur deux contraintes géométriques, une contrainte de localité C_L et la contrainte épipolaire C_E.

Contrainte de localité d'appariement. La contrainte de localité d'appariement C_L est utilisée pour l'appariement temporel de points d'intérêt, c'est à dire observés par la même caméra j aux temps t et t'. Cette contrainte permet l'appariement potentiel de deux points d'intérêt $p \in \mathtt{p}_{j,t}$ et $p' \in \mathtt{p}_{j,t'}$ lorsque leur distance euclidienne en pixels d_{L_2} d'une image à l'autre est inférieure au seuil T_L. Chaque appariement potentiel $m_\phi(p, p')$ satisfait ainsi la relation :

$$\mathsf{C}_L(p, p') \iff d_{L_2}(p, p') < \mathsf{T}_L \tag{2.4}$$

Cette contrainte est appliquée aux coordonnées distordues des points d'intérêt, c'est à dire que la recherche d'appariement potentiel se fait directement dans l'espace pixellique original des images non corrigées. En effet, dans le cas particulier des caméras *fisheye*, les images acquises présentent des distorsions géométriques très importantes. La correction de ces distorsions provoque alors l'augmentation sensible des coordonnées pixelliques des points d'intérêt à mesure que ceux-ci s'éloignent du centre de l'image, ce qui impose la distorsion de la zone de recherche en conséquence. Le travail sur les coordonnées originales des points d'intérêt permet de fixer la taille de la zone de recherche et ainsi d'éviter cette étape supplémentaire de distorsion.

Contrainte épipolaire d'appariement standard. La contrainte épipolaire d'appariement C_E est utilisée pour l'appariement stéréo de points d'intérêt simultanément observés par les caméras j et j' au temps t, à champs recouvrants et dont les paramètres extrinsèques sont connus. Cette contrainte permet l'appariement potentiel de deux points

d'intérêt $p \in \mathtt{p}_{j,t}$ et $p' \in \mathtt{p}_{j',t}$ si les distances euclidiennes d_{L_2} à leurs droites épipolaires respectives l et l' sont inférieures au seuil T_E. Chaque appariement potentiel $m_\phi(p, p')$ satisfait ainsi la relation :

$$\mathsf{C}_E(p, p') \iff \begin{cases} d_{L_2}(p, l) < \mathsf{T}_E \\ d_{L_2}(p', l') < \mathsf{T}_E \end{cases} \tag{2.5}$$

Avec $l = F_{jj'}^T \bar{p}'$, $l' = F_{jj'} \bar{p}$ et $F_{jj'}$ la matrice fondamentale entre les caméras j et j', déterminée lors de l'étape de calibration extrinsèque du système multi-caméras.

Contrainte épipolaire d'appariement appliquée aux caméras *fisheye*. Bien qu'en théorie correcte dans le cas général, la contrainte épipolaire d'appariement présentée dans le paragraphe précédent n'est pas directement applicable au cas particulier des caméras *fisheye*, notamment en ce qui concerne le calcul de la distance des appariements potentiels à leur droite épipolaire associée. En effet, comme indiqué dans le paragraphe relatif à la contrainte de localité d'appariement C_L, ce type de caméra génère des distorsions importantes de l'image, et leur correction ne permet alors plus la définition d'un seuil unique de distance T_E approprié pour tous les points de l'image à leurs droites épipolaires respectives. La méthode alternative proposée consiste à calculer non pas une distance à cette droite mais l'angle formé par les deux vecteurs partant du centre optique de la caméra et passant l'un par le point d'intérêt et l'autre par la projection perpendiculaire de ce point sur la droite. Cette contrainte est alors désignée par C_{E_θ}. Plus en détail, on considère le vecteur normalisé $\mathbf{d}_p$ partant du centre optique $c_{j,t}$ de la première caméra vers le point d'intérêt $p \in \mathtt{p}_{j,t}$, de vecteur directeur $\mathbf{r}_p = (c_{j,t}, p)^T$, et le vecteur normalisé $\mathbf{d}_l$ partant du centre optique de la caméra vers la projection orthogonale $proj_{(l)}(p)$ de p sur la droite épipolaire $l = F_{jj'}^T \bar{p}'$, de vecteur directeur $\mathbf{r}_l = (c_{j,t}, proj_{(l)}(p))^T$, tel que :

$$\mathbf{d}_p = \frac{\mathbf{r}_p}{\|\mathbf{r}_p\|} \qquad \mathbf{d}_l = \frac{\mathbf{r}_l}{\|\mathbf{r}_l\|} \tag{2.6}$$

On définit de manière similaire les vecteurs normalisés $\mathbf{d}_{p'}$ et $\mathbf{d}_{l'}$, puis on introduit enfin la contrainte épipolaire d'appariement appliquée aux caméras *fisheye*, permettant l'appariement potentiel $m_\phi(p, p')$ du point $p \in \mathtt{p}_{j,t}$ au point $p' \in \mathtt{p}_{j',t}$ si l'angle $\theta_{p,l} = \arccos(\mathbf{d}_p \cdot \mathbf{d}_l)$ et l'angle $\theta_{p',l'} = \arccos(\mathbf{d}_{p'} \cdot \mathbf{d}_{l'})$ ont un degré inférieur au seuil T_{E_θ}, tel que :

$$\mathsf{C}_{E_\theta}(p, p') \iff \begin{cases} \theta_{p,l} < \mathsf{T}_{E_\theta} \\ \theta_{p',l'} < \mathsf{T}_{E_\theta} \end{cases} \tag{2.7}$$

À noter que cette contrainte fonctionne également dans le cas de caméras à focales

plus longues, son utilisation n'est donc pas exclusivement limitée aux caméras *fisheye*. Une illustration de cette contrainte d'appariement est visible figure 2.3.

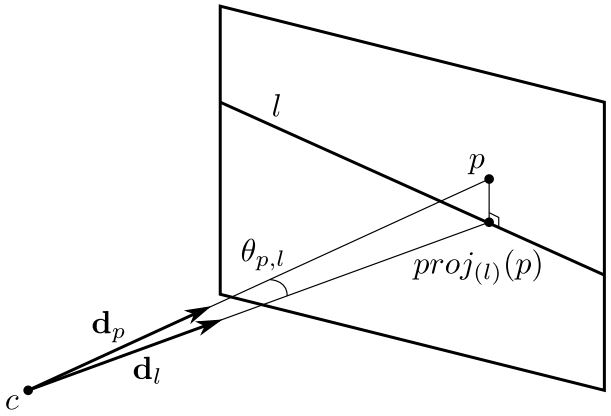

FIGURE 2.3 – Contrainte épipolaire d'appariement appliquée aux caméras *fisheye*.

Discussion sur les contraintes d'appariement. En pratique, bien qu'il soit possible d'utiliser la contrainte épipolaire d'appariement pour effectuer les appariements temporels, étant donné que l'on possède déjà les poses des caméras du système précédemment calculées par le module de SLAM visuel, cette solution ne conviendrait pas pour le suivi de points mobiles qui nous intéresse ici. En effet, ces points ne respectent pas par définition la contrainte épipolaire, sauf cas ambigu où leur mouvement se produit sur un plan épipolaire. Un appariement temporel basé sur l'utilisation de la contrainte épipolaire d'appariement empêcherai donc tout suivi de points mobiles, ce qui justifie l'utilisation de la contrainte de localité d'appariement.

Sélection des appariements définitifs $\mathtt{m}_\Gamma$ parmi les appariements potentiels $\mathtt{m}_\phi$. Après avoir déterminé les appariements potentiels relatifs à deux ensembles de points d'intérêt $\mathtt{p}_{j,t}$ et $\mathtt{p}_{j',t'}$, suivant la contrainte de localité C_L ou les contraintes épipolaires C_E et C_{E_θ}, l'étape finale d'appariement de points consiste à sélectionner parmi ces appariements potentiels les appariements définitifs $m_\Gamma(p,p')$, avec $p \in \mathtt{p}_{j,t}$, $p' \in \mathtt{p}_{j',t'}$ et $m_\Gamma(p,p') \in \mathtt{m}_{j,j'}^{t,t'}$. Dans cette étape, le sens d'appariement est pris en compte, c'est à dire que les points de la première image sont considérés comme points de référence, alors que les points de la deuxième image sont considérés comme points candidats. En conséquence, l'appariement s'effectue pour chaque point de la deuxième image vers ceux de la première, ce qui ne garantit donc pas l'appariement de tous les points de la première image vers ceux de la seconde et peut également engendrer l'appariement d'un point de la seconde image vers de multiples points de la première. Lorsqu'il n'existe qu'un seul appariement potentiel d'un point de la seconde image vers un point de la première, le

score de ressemblance entre les descripteurs de ces points doit dépasser un certain seuil pour que l'appariement soit retenu. En revanche, lorsqu'il existe plus d'un appariement potentiel liant le point $p' \in \mathtt{p}_{j',t'}$ de la deuxième image à l'ensemble de points d'intérêt $\mathtt{p}_{j,t}$ associés à la première image, l'appariement définitif retenu est celui pour lequel la distance entre les descripteurs des deux points est minimale. Comme abordé dans la sous-section 1.1.3.3, le calcul de cette distance est alors dépendant du type de descripteur utilisé.

2.4.1.4 Lois d'appariement multi-caméras

L'appariement temporel entre deux ensembles de points d'intérêt $\mathtt{p}_{j,t}$ et $\mathtt{p}_{j,t'}$ n'est possible qu'à temporalités successives d'observation, c'est à dire pour $t' = t + 1$, alors que l'appariement stéréo à l'instant t de deux ensembles de points d'intérêt $\mathtt{p}_{j,t}$ et $\mathtt{p}_{j',t}$ n'est possible que lorsque les caméras j et j' présentent un champ recouvrant. La loi d'appariement temporel est importante en ce qui concerne le suivi de points d'intérêt, car elle implique qu'il existe au moins un appariement temporel associé à chaque point pour toutes les temporalités successives d'observation pendant lesquelles il a été suivi, ce qui signifie que la méthode proposée n'est pas robuste aux occlusions ni aux erreurs de détection des points auparavant triangulés.

2.4.1.5 Implémentation

L'algorithme 2 présente l'implémentation de l'étape d'extraction et d'appariement des points d'intérêt du module de détection et reconstruction de points mobiles.

2.4.2 Triangulation et transitivité des appariements

Lorsque tous les appariements $\mathtt{m}_{j,j'}^{t,t'}$ correspondants aux lois d'appariements multi-caméras détaillées dans la sous-section 2.4.1.4 ont été effectués, on procède ensuite à leur triangulation afin de tenter d'associer à chaque couple $m_\Gamma(p, p')$ un point 3D P. À noter qu'en ce qui concerne les appariement temporels, seulement ceux satisfaisant la contrainte épipolaire entre les deux images sont triangulés (le contraire n'aurait pas de sens), alors que les autres appariements associent simplement le point $p' \in \mathtt{p}_{j,t}$ le plus récent au point 3D P, associé au point $p \in \mathtt{p}_{j,t-1}$ et triangulé à la précédente temporalité, s'il existe. La vérification de la contrainte épipolaire s'effectue en deux étapes. La première étape consiste en l'application des contraintes d'évaluation des inliers $\mathsf{C}_{In}(p, p')$ et $\mathsf{C}_{In_\theta}(p, p')$, qui sont identiques aux contraintes d'appariement $\mathsf{C}_E(p, p')$ et $\mathsf{C}_{E_\theta}(p, p')$ mais comportent des seuils spécifiques T_{In} et T_{In_θ}. La deuxième étape consiste pour sa part à s'assurer que le point 3D nouvellement triangulé se situe bien en avant de l'axe optique de la caméra, tel qu'explicité dans l'équation (1.48). À l'exception de la première temporalité de la

Algorithme 2 : Extraction et appariement des points d'intérêt du module de détection et reconstruction de points mobiles.

Données : Images acquises par les caméras j à la temporalité t.
Résultat : Ensembles $\mathtt{m}_{j,j'}^{t,t'}$ respectant les lois d'appariement multi-caméras.

```
début
    pour chaque j faire
        S_1 = Extraction_S1();
        si Geometrie_Existante() alors
            S_2 = Extraction_S2(P_{j,t-1});
        fin
        p_{j,t} = Fusion_S1_S2();
    fin
    pour chaque j faire
        m_{j,j}^{t-1,t} = Appariement_C_L(p_{j,t-1}, p_{j,t});
        j' = j + 1;
        tant que j' <= J faire
            si Champ_Recouvrant(j, j') alors
                si Cameras_Fisheye(j, j') alors
                    m_{j,j'}^{t,t} = Appariement_C_{E_θ}(p_{j,t}, p_{j',t});
                sinon
                    m_{j,j'}^{t,t} = Appariement_C_E(p_{j,t}, p_{j',t});
                fin
            fin
        fin
    fin
fin
```

séquence à laquelle ne sont associés que des appariements stéréo, l'étape de triangulation respecte ensuite un ordre, c'est à dire qu'elle s'effectue d'abord sur les appariements temporels et ensuite sur les appariements stéréo. La raison en est simple, elle permet le suivi des points auparavant triangulés. Cette étape effectuée, il est ensuite nécessaire de s'assurer de la transitivité des appariements afin qu'ils correspondent bien au même point 3D. Plusieurs cas d'appariements non transitifs étant possibles dans un système multi-caméras, nous nous attacherons à expliquer la procédure mise en place afin de traiter le cas le plus simple à quatre vues, qui se généralise également à n vues. Pour des raisons de lisibilité, nous indicerons différemment les points 2D dans cette partie. Soit les points homologues $p_1 \in \mathtt{p}_{j,t-1}$, $p_2 \in \mathtt{p}_{j,t}$, $p_3 \in \mathtt{p}_{j',t-1}$ et $p_4 \in \mathtt{p}_{j',t}$, observés dans quatre vues distinctes et appareillés deux à deux par trois couples d'appariements $m_\Gamma(p_1, p_2) \in \mathtt{m}_{j,j}^{t-1,t}$, $m_\Gamma(p_3, p_4) \in \mathtt{m}_{j',j'}^{t-1,t}$ et $m_\Gamma(p_2, p_4) \in \mathtt{m}_{j,j'}^{t,t}$. Soit P_1 le point 3D issu de la triangulation de (p_1, p_2), P_2 le point 3D issu de la triangulation de (p_3, p_4) et P_3 le point issu de la triangulation de (p_2, p_4). Il convient de noter que selon l'ordre de triangulation

des appariements, c'est dans un premier temps P_1, puis P_2 et enfin P_3 qui sont triangulés. Si ces points sont homologues, il est nécessaire de s'assurer que $P_1 = P_2 = P_3$, or l'ordre de triangulation nous indique que lorsque l'appariement $m_\Gamma(p_2, p_4) \in \mathbb{m}_{j,j'}^{t,t}$ sera triangulé, p_2 sera déjà associé à P_1 et p_4 à P_2. La règle de transitivité des appariements consiste alors à réassocier chaque point d'intérêt au point 3D le plus ancien, en l'occurrence P_1, de manière récursive pour tous les points d'intérêts déjà associés à un point 3D plus récent. La figure 2.4 illustre une configuration à quatre vues de la transitivité des appariements.

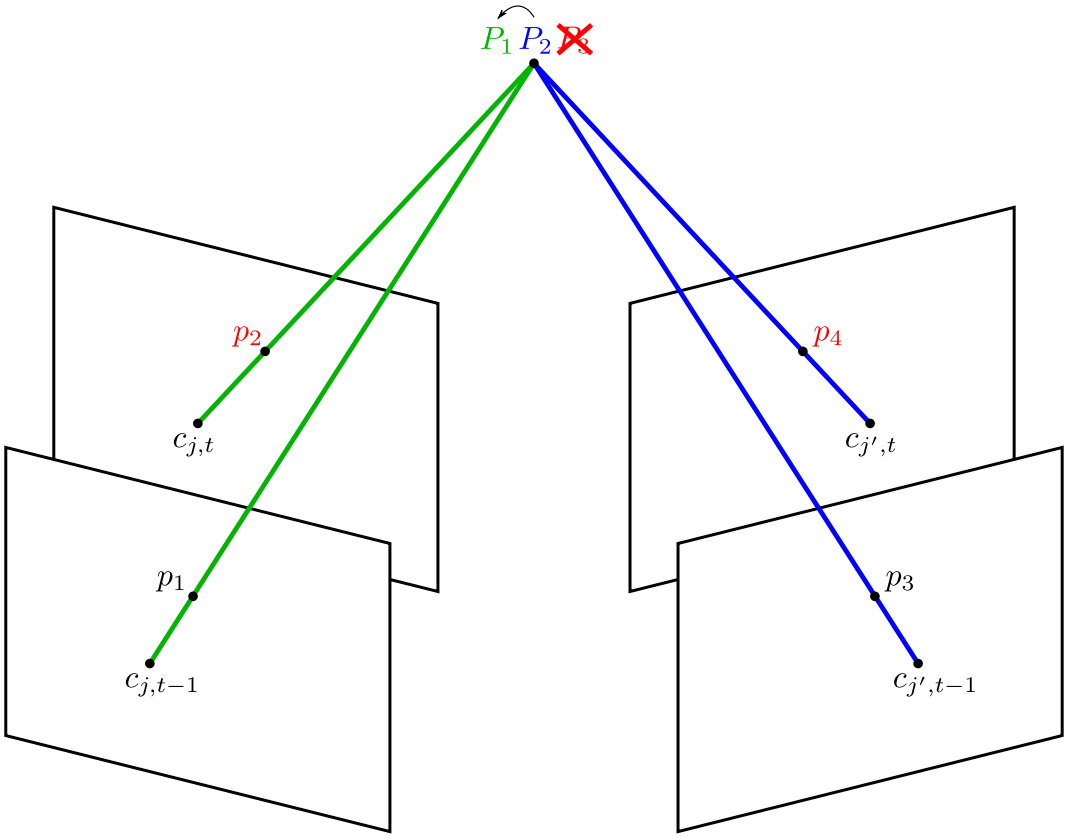

FIGURE 2.4 – Transitivité des appariements. Dans cette illustration, le point P_1 est d'abord triangulé suivant l'appariement temporel entre p_1 et p_2, P_2 est ensuite triangulé suivant l'appariement temporel entre p_3 et p_4, alors que l'appariement stéréo entre p_2 et p_4 devant produire le point P_3 provoque un conflit d'association. Suivant la règle de transitivité des appariements, le point P_2 est alors fusionné à P_1, tandis que les points d'intérêt p_3 et p_4 sont alors réassociés récursivement au point P_1 par transitivité.

L'application de la règle de transitivité des appariements permet enfin de s'assurer que tous les points triangulés sont bien uniques et associés à tous leurs points d'intérêt respectifs dans toutes les vues du système multi-caméras.

2.4.3 Classification des points 3D

Chaque point 3D P_n doit être associé à un minimum de deux observations $(o^n_{j,t}, o^n_{j',t'})$ par les caméras j et j' aux temps t et t' pour être reconstruit. Ces observations peuvent être issues du principe de *SfM*, c'est à dire temporelles ($j = j' \wedge t \neq t'$), ou issues du principe de *Stéréo Passive*, c'est à dire stéréo ($j \neq j' \wedge t = t'$), et correspondent à un appariement définitif $m_\Gamma(p^n_{j,t}, p^n_{j',t'})$ déterminé lors de l'étape d'appariement des ensembles de points d'intérêt $\mathtt{p}_{j,t}$ et $\mathtt{p}_{j',t'}$. Chaque point 3D P_n peut être associé à plus de deux observations, et l'on désigne par $\mathtt{o}^n$ l'ensemble des observations associées à P_n. Il est important de préciser que dans cette partie du manuscrit, tous les appariements de points d'intérêt entre chaque paire d'images $\mathtt{m}^{tt'}_{jj'}$ sont retenus afin de permettre la détection de points mobiles, au contraire de la méthode utilisée dans le module de SLAM visuel qui élimine les outliers ne satisfaisant pas la contrainte épipolaire générale de la scène. L'objectif de cette étape du module de détection et reconstruction de point mobiles consiste alors à déterminer la classe de chaque points 3D P_n à partir de l'ensemble de ses observations $\mathtt{o}^n$, c'est à dire les points *neutres* $P_n \in \mathcal{C}_N$, les points fixes ou *statiques* $P_n \in \mathcal{C}_S$, les points *candidats* $P_n \in \mathcal{C}_C$, les points *mobiles* $P_n \in \mathcal{C}_M$ et les points *outliers* $P_n \in \mathcal{C}_O$.

2.4.3.1 Consistance d'un point 3D

Un point 3D P_n est considéré *consistant* lorsqu'il satisfait la contrainte de consistance C_C. Cette contrainte impose que l'erreur de reprojection ou l'erreur angulaire de ce point soit inférieure aux seuils T_C et T_{C_θ} respectivement, pour toutes ses observations, tel que :

$$\mathsf{C}_C(\mathtt{o}^n) \iff \begin{cases} \forall o^n_{j,t} \in \mathtt{o}^n, (o^n_{j,t} - f(C_{j,t}, P_n)) < \mathsf{T}_C \\ \forall o^n_{j,t} \in \mathtt{o}^n, g(o^n_{j,t}, C_{j,t}, P_n) < \mathsf{T}_{C_\theta} \end{cases} \tag{2.8}$$

Avec $f(C_{j,t}, P_n)$ la fonction de projection du point P_n et $g(o^n_{j,t}, C_{j,t}, P_n)$ son erreur angulaire, définies dans les équations (1.56) et (1.61).

2.4.3.2 Immobilisme d'un point 3D

Une conséquence de la contrainte de consistance implique qu'un point 3D fixe ($P_n \in \mathcal{C}_S$) doit être consistant pour l'ensemble de ses observations $\mathtt{o}^n$. Deux autres contraintes spécifiques permettant de s'assurer de l'immobilisme d'un point sont néanmoins nécessaires.

Première contrainte d'immobilisme d'un point C_{S1}. Une première contrainte supplémentaire à la contrainte de consistance C_C consiste à s'assurer que le point a été

observé à au moins deux temporalités différentes afin de garantir son immobilisme dans le temps. Cette contrainte C_{S1} est définie telle que :

$$\mathsf{C}_{S1}(\mathsf{o}^n) \iff \forall o^n_{j,t} \in \mathsf{o}^n, \exists (o^n_{j,t}, o^n_{j,t'}) \in (\mathsf{o}^n)^2, t \neq t' \tag{2.9}$$

Deuxième contrainte d'immobilisme d'un point C_{S2}. La deuxième contrainte d'immobilisme d'un point 3D consiste à s'assurer que le point, s'il n'est observé qu'à deux temporalités différentes, est bien triangulé à la bonne place, et impose donc au moins une observation supplémentaire, permettant sa triangulation deux fois. Cette contrainte C_{S2} est alors définie telle que :

$$\mathsf{C}_{S2}(\mathsf{o}^n) \iff |\mathsf{o}^n| \geq 3 \tag{2.10}$$

Les points 3D consistants ne remplissant pas ces deux contraintes sont considérés comme neutres $P_n \in \mathcal{C}_N$, c'est à dire qu'ils n'ont été soit observés qu'en stéréo à la même temporalité, ou ne sont associés qu'à deux observations temporelles et n'ont donc pas été triangulés à la bonne place. Ce type d'observations ne permet alors pas de se prononcer sur l'immobilisme ou la mobilité de ces points. La figure 2.5 illustre les contraintes d'immobilisme des points 3D.

2.4.3.3 Mobilité d'un point 3D

Au contraire des points fixes, les points mobiles peuvent ne pas être consistants pour toutes leurs temporalités d'observation. En effet, leur déplacement, s'il n'est pas ambigu, implique généralement une erreur de reprojection ou une erreur angulaire importante dans les images acquises à des temporalités différentes, ce qui ne permet pas de vérifier la contrainte de consistance C_C. Les points 3D n'étant pas consistants sont alors évalués afin de déterminer leur classe parmi celles restantes, c'est à dire les points candidats $P_n \in \mathcal{C}_C$, les points mobiles $P_n \in \mathcal{C}_M$ et les points outliers $P_n \in \mathcal{C}_O$.

Première contrainte de mobilité C_{M1}. La première contrainte exprimant la mobilité d'un point 3D C_{M1} consiste à vérifier indépendamment sa consistance à chaque temporalité d'observation t, ce qui autorise ainsi une position dans l'espace différente pour chaque temporalité. La contrainte C_{M1} est donc définie telle que :

$$\mathsf{C}_{M1}(\mathsf{o}^n) \iff \begin{cases} \forall t, \forall o^n_{j,t} \in \mathsf{o}^n_t, (o^n_{j,t} - f(C_{j,t}, P_n)) < \mathsf{T}_{M1} \\ \forall t, \forall o^n_{j,t} \in \mathsf{o}^n_t, g(o^n_{j,t}, C_{j,t}, P_n) < \mathsf{T}_{M1_\theta} \end{cases} \tag{2.11}$$

Avec o^n_t l'ensemble des observations associées au point 3D P_n au temps t et les seuils T_{M1} et T_{M1_θ} définis tel que l'erreur de reprojection ou l'erreur angulaire maximales

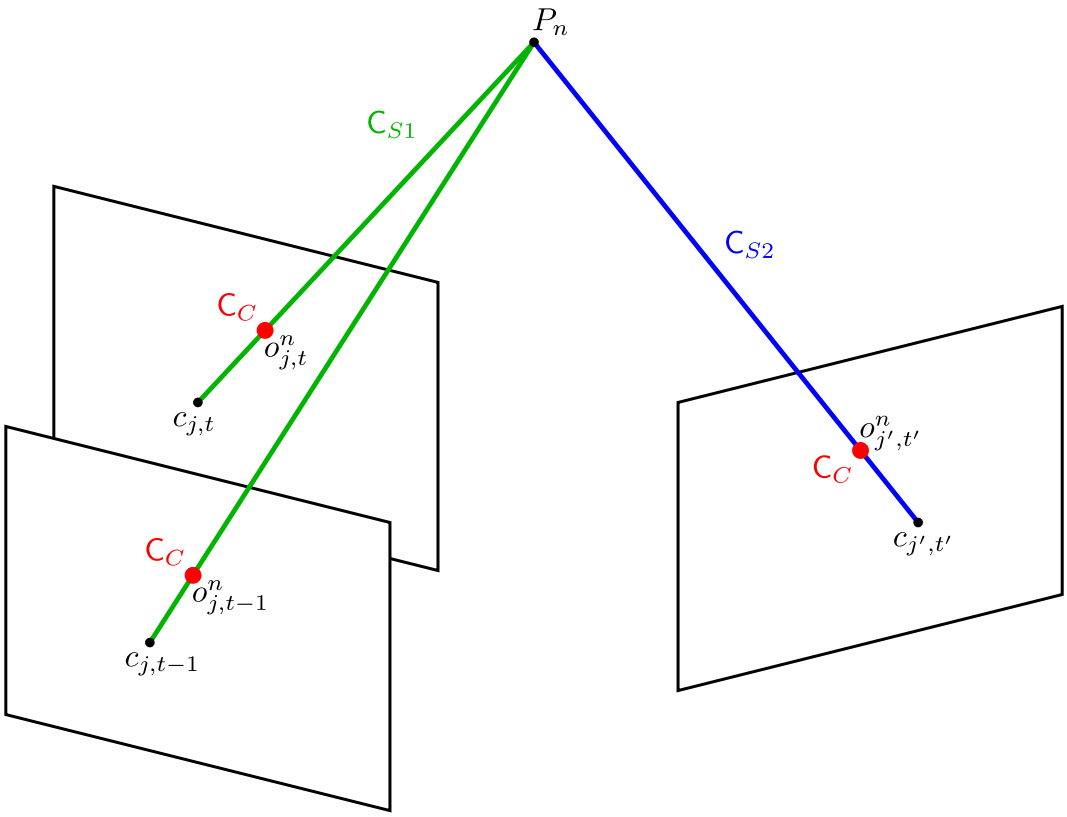

FIGURE 2.5 – Contraintes d'immobilisme des points 3D. Dans cette illustration, l'appariement temporel entre $o^n_{j,t-1}$ et $o^n_{j,t}$ satisfait la première contrainte d'immobilisme C_{S1} alors que la troisième observation $o^n_{j',t'}$ de P_n peut être une observation temporelle $(j' = j \wedge (t' \neq t-1 \neq t))$ ou stéréo $(j' \neq j \wedge (t' = t-1 \vee t' = t))$ et satisfait la deuxième contrainte d'immobilisme C_{S2}. À noter que la contrainte de consistance C_C est satisfaite pour toutes les observations de P_n.

permises pour satisfaire la contrainte. L'application de cette contrainte permet par la suite de considérer chaque point 3D P_n tel que la collection de plusieurs points 3D P^t_n, représentant la position dans l'espace de P_n à chacune de ses temporalités t.

Deuxième contrainte de mobilité C_{M2}. Considérant que le point P_n se déplace, seules les observations stéréo de ce point permettent sa triangulation au temps t. Il doit donc nécessairement exister au moins deux observations stéréo $(o^n_{j,t}, o^n_{j',t})$ pour chaque temporalité t, ce qui est assuré par la deuxième contrainte exprimant la mobilité d'un point 3D C_{M2}, définie telle que :

$$\mathsf{C}_{M2}(\mathsf{o}^n) \iff \forall t, |\mathsf{o}^n_t| \geq 2 \tag{2.12}$$

Troisième contrainte de mobilité C_{M3}. Enfin, la détection d'un point 3D mobile n'étant possible que sur plusieurs temporalités différentes d'observation, il doit au moins exister un couple d'observations temporelles $(o^n_{j,t}, o^n_{j,t'})$ dans l'ensemble o^n. La troisième contrainte exprimant la mobilité d'un point 3D C_{M3} exprime ainsi la relation :

$$\mathsf{C}_{M3}(\mathsf{o}^n) \iff \exists (o^n_{j,t}, o^n_{j,t'}) \in (\mathsf{o}^n)^2, t \neq t' \tag{2.13}$$

La figure 2.6 illustre les contraintes de mobilité des points 3D.

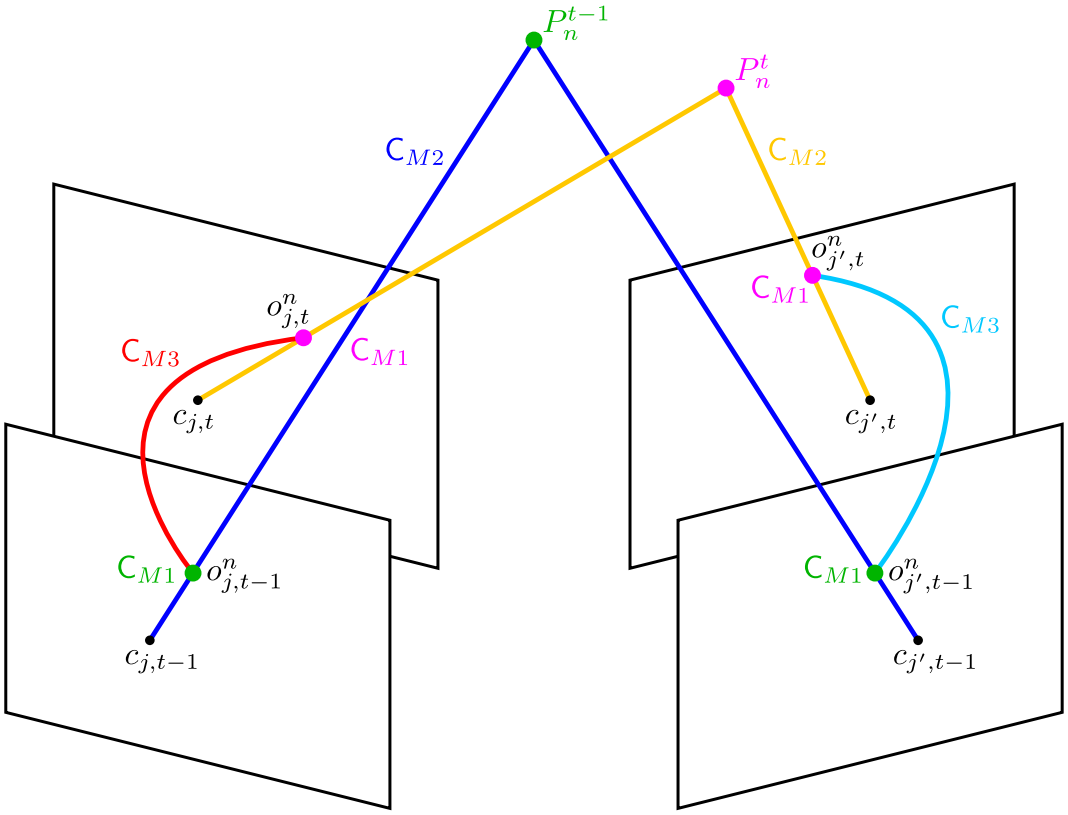

FIGURE 2.6 – Contraintes de mobilité des points 3D. Dans cette illustration, la première contrainte de mobilité C_{M1} de consistance stéréo est satisfaite pour toutes les observations à chaque temporalité $t-1$ et t, de même que la deuxième contrainte de mobilité C_{M2}, exigeant un minimum de deux observations stéréo par temporalité. Enfin, la troisième contrainte de mobilité C_{M3}, exigeant un minimum de deux temporalités d'observation différentes pour la détection d'un point mobile, est également satisfaite à deux reprises, bien qu'une seule ai été suffisante.

Consistance de la trajectoire des points mobiles. En pratique, alors qu'un minimum de deux temporalités d'observation est nécessaire afin de détecter un point 3D mobile, la méthode de segmentation proposée en exige trois afin d'évaluer la consistance

des trajectoires associées aux points reconstruits, impliquant leur suivi sur au moins trois temporalités successives. Cette étape permet le filtrage des points 3D dont la trajectoire ne satisfait pas certaines caractéristiques de régularité. Soit le point 3D P_n et sa position dans l'espace à chacune de ses temporalités P_n^t, c'est à dire sa trajectoire. Une première contrainte permettant d'évaluer la consistance de cette trajectoire est une contrainte de vitesse C_{T1}, qui n'autorise qu'une distance de déplacement limitée dans l'espace du point mobile entre chacune de ses temporalités successives. On définit donc cette contrainte telle que :

$$\mathsf{C}_{T1}(P_n) \iff \mathsf{T}_{T1_{min}} < d_{L_2}(P_n^t, P_n^{t+1}) < \mathsf{T}_{T1_{max}} \tag{2.14}$$

Avec $\mathsf{T}_{T1_{min}}$ et $\mathsf{T}_{T1_{max}}$ les seuils de distance acceptables au minimum et au maximum, respectivement. Similairement à la contrainte de vitesse, les objets mobiles supposés reposer sur le sol, on introduit une contrainte d'élévation de la trajectoire C_{T2}, définie telle que :

$$\mathsf{C}_{T2}(P_n) \iff |P_n^t(Y_w) - P_n^{t+1}(Y_w)| < \mathsf{T}_{T2} \tag{2.15}$$

Avec $P_n^t(Y_w)$ et $P_n^{t+1}(Y_w)$ les valeurs sur l'axe $\mathbf{Y}_w$ du repère monde de P_n^t et P_n^{t+1}, respectivement, et T_{T2} le seuil définissant le changement d'élévation maximum permis pour satisfaire la contrainte. À noter que les unités de mesure des seuils associés à ces deux contraintes sont dépendantes de l'étape de calibration des paramètres intrinsèques et extrinsèques du système multi-caméras. Lorsque ces paramètres de calibration s'inscrivent dans un référentiel métrique, les valeurs de ces seuils sont alors exprimées en mètres. Enfin, la dernière contrainte de consistance de la trajectoire d'un point mobile P_n concerne les changements de direction, c'est à dire l'angle 2D formé par la reprojection de chaque triplet de points consécutifs $(P_n^t, P_n^{t+1}, P_n^{t+2})$ sur le plan formé par les axes $\mathbf{X}_w, \mathbf{Z}_w$ du repère monde et désignés par $P_1 = \Pi_{\mathbf{X}_w,\mathbf{Z}_w}(P_n^t)$, $P_2 = \Pi_{\mathbf{X}_w,\mathbf{Z}_w}(P_n^{t+1})$ et $P_3 = \Pi_{\mathbf{X}_w,\mathbf{Z}_w}(P_n^{t+2})$, respectivement, avec $\Pi_{\mathbf{X}_w,\mathbf{Z}_w}(P_n) = \Pi_{\mathbf{X}_w,\mathbf{Z}_w}(X_w, Y_w, Z_w) = (X_w, Z_w)$. Cette contrainte C_{T3} est définie telle que :

$$\mathsf{C}_{T3}(P_n) \iff \widehat{P_1P_2P_3} > \mathsf{T}_{T3} \tag{2.16}$$

Avec T_{T3} l'angle minimum permis afin de satisfaire la contrainte. Les trois contraintes assurant la consistance de la trajectoire d'un point mobile permettent le filtrage des mouvements erratiques généralement engendrés par de mauvais appariements lors de l'étape d'appariement de points d'intérêt. Les points satisfaisant les trois contraintes de mobilité mais ne permettant pas un filtrage complet de leur trajectoire, c'est à dire les points observés sur deux temporalités successives seulement, sont considérés comme

points candidats $P_n \in \mathcal{C}_C$. Enfin, les points non consistants et ne satisfaisant pas les contraintes de mobilité ou de trajectoire sont considérés comme outliers $P_n \in \mathcal{C}_O$ et sont rejetés. La figure 2.7 illustre les contraintes sur la trajectoire des points 3D mobiles.

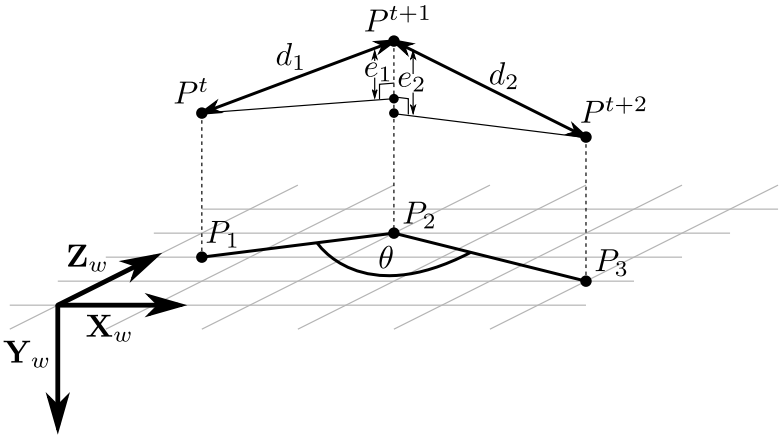

FIGURE 2.7 – Contraintes sur la trajectoire des points 3D mobiles. Dans cette illustration, $d_1 = d_{L_2}(P_n^t, P_n^{t+1})$ et $d_2 = d_{L_2}(P_n^{t+1}, P_n^{t+2})$ correspondent à l'évaluation de la contrainte C_{T1}, $e_1 = |P_n^t(Y_w) - P_n^{t+1}(Y_w)|$ et $e_2 = |P_n^{t+1}(Y_w) - P_n^{t+2}(Y_w)|$ correspondent à l'évaluation de la contrainte C_{T2} et $\theta = \widehat{P_1 P_2 P_3}$ correspond à l'évaluation de la contrainte C_{T3}.

2.4.3.4 Implémentation

L'algorithme 3 présente l'implémentation de la classification des points 3D reconstruits. Cette procédure affecte tous les points 3D observés à la dernière temporalité $t = T$. En conséquence, chacun de ces points est susceptible de changer de classe lors de l'étape de segmentation, ce qui permet une représentation correcte de l'évolution des mouvements se produisant au sein de la scène observée.

2.4.4 Optimisation globale des points 3D

Suivant l'étape de classification des points 3D, chaque point mobile $P_n \in \mathcal{C}_M$ est ensuite divisé en plusieurs points individuels P_n^t correspondant aux différentes positions du point P_n à chacune de ses temporalités d'observation t. Cette division de chaque point mobile en plusieurs points statiques permet alors l'optimisation de tous les points 3D indépendamment de leur classe, à l'exception des points candidats qui restent inconsistants lors de l'étape d'optimisation. Cette optimisation est alors effectuée par ajustement de faisceaux minimisant l'erreur de reprojection dans le cas général, et par ajustement de faisceaux minimisant l'erreur angulaire lorsque le système multi-caméras comporte une ou plusieurs caméras *fisheye*, comme détaillé dans la sous-section 1.1.7.

Algorithme 3 : Segmentation des points 3D observés à la dernière temporalité $t = T$.

Données : Ensemble des observations o^n du point 3D P_n
Résultat : Classe du point 3D P_n
début
  si $\mathsf{C}_C(\mathsf{o}^n)$ **alors**
    si $\mathsf{C}_{S1}(\mathsf{o}^n) \wedge \mathsf{C}_{S2}(\mathsf{o}^n)$ **alors**
      retourner $\mathcal{C}_S$
    sinon
      retourner $\mathcal{C}_N$
    fin
  sinon
    si $\mathsf{C}_{M1}(\mathsf{o}^n) \wedge \mathsf{C}_{M2}(\mathsf{o}^n) \wedge \mathsf{C}_{M3}(\mathsf{o}^n)$ **alors**
      si $\mathsf{C}_{T1}(P_n) \wedge \mathsf{C}_{T2}(P_n)$ **alors**
        si $\mathsf{C}_{T3}(P_n)$ **alors**
          retourner $\mathcal{C}_M$
        sinon
          retourner $\mathcal{C}_C$
        fin
      fin
    fin
  fin
  retourner $\mathcal{C}_O$
fin

Chapitre 3

Résultats

3.1 Introduction

La méthode proposée a été évaluée de manière quantitative et qualitative dans différentes conditions expérimentales. Dans ce chapitre seront notamment définis dans la section 3.2 tous les éléments permettant cette évaluation, c'est à dire les jeux de données utilisés, l'organisation, la sélection et l'initialisation de l'ensemble des paramètres de la méthode, les indicateurs de mesure permettant d'en évaluer les performances ainsi que le protocole de test pour chaque jeu de données. Les performances quantitatives et qualitatives obtenues pour chaque séquence ainsi que les limitations de la méthode seront enfin proposées dans la section 3.3.

3.2 Jeux de données et méthodologie d'évaluation

3.2.1 Motivations et objectifs

Deux jeux de données différents ont été utilisés afin de permettre l'évaluation de la méthode proposée dans des conditions expérimentales proches de la réalité. Dans chaque jeu de données, l'environnement observé se compose d'une partie fixe (route, véhicules arrêtés, aménagements urbains, espaces naturels) et d'une partie mobile (véhicules en mouvement, cyclistes, piétons). Le premier jeu de données utilisé fait partie de la suite KITTI, proposée par Geiger *et al.* [53] et très populaire dans la littérature. Ce jeu de données, *Visual Odometry / SLAM Evaluation 2012*, comporte plusieurs séquences de parcours urbains et autoroutiers en stéréo *short-baseline*, permettant l'évaluation de l'algorithme dans des conditions relativement favorables au module d'appariement de points d'intérêt. Le second jeu de données utilisé a été développé au sein du laboratoire d'accueil et permet l'évaluation de la méthode pour le cas de systèmes multi-caméras hétérogènes à champs recouvrants en *wide-baseline*. Plusieurs séquences correspondant à différents

scénarios de circulation routière ont été réalisées afin d'évaluer la robustesse du système et la qualité des reconstructions obtenues. Par ailleurs, la méthodologie développée afin d'évaluer l'algorithme proposé a pour objectif de mettre en exergue ses performances et limitations, en s'attachant notamment à mesurer l'impact engendré par la variation de ses paramètres grâce à des indicateurs quantitatifs et qualitatifs.

3.2.2 Jeu de données KITTI

Plusieurs jeux de données sont proposés par la suite KITTI afin de permettre l'évaluation de différents types de méthodes. Comme indiqué au précédent paragraphe, le jeu de données retenu est celui proposé par Geiger *et al.* [53], *Visual Odometry / SLAM Evaluation 2012*, notamment du fait de son utilisation dans plusieurs autres articles de la littérature, en particulier dans la méthode de SLAM visuel multi-objets éparse proposée par Sabzevari [143]. Sur la multitude de séquences proposées dans ce jeu de données, celles retenues l'ont été en raison de la présence d'un objet mobile dont la taille dans l'image et la durée d'observation ont été jugées significatives. En matière de spécifications techniques, ces séquences sont toutes composées d'images acquises à une fréquence de 10 Hz au moyen d'une paire stéréo *short-baseline* calibrée, en noir et blanc, d'une résolution de 1241×376 pixels et rectifiées.

3.2.3 Jeu de données IP

3.2.3.1 Véhicule d'acquisition et capteurs embarqués

Le véhicule expérimental Velac de l'Institut Pascal a été utilisé pour l'acquisition des différentes séquences. C'est un monospace sur lequel ont été installées quatre caméras rigidement liées à la caisse du véhicule et dont les spécifications sont détaillées dans le tableau 3.1. Ces quatre caméras sont identiques et ont été synchronisées à l'aide d'un déclencheur hardware permettant de capturer les images de chaque capteur simultanément. Trois des caméras sont associées à un objectif *fisheye* dont le champ de vue est de 185 degrés, tandis que la quatrième est associée à un objectif grand angle à focale plus longue, proposant un champ de vue d'environ 80 degrés. Les caméras *fisheye* ont été placées respectivement sur la calandre avant, pointant à l'horizontale dans l'axe longitudinal du véhicule, ainsi que sur les cotés gauche et droit du toit, au niveau des portes conducteur et passager et pointant légèrement vers le bas, perpendiculairement à l'axe longitudinal du véhicule. La caméra grand angle a, pour sa part, été placée sur le toit, au dessus du pare-brise et pointant légèrement vers le bas, dans l'axe longitudinal du véhicule. Comme expliqué en introduction de ce manuscrit, les choix du nombre, du positionnement et des caractéristiques optiques des caméras utilisées ont été motivés par la présence sur certains véhicules de grande série de systèmes aux caractéristiques similaires, tels que les

systèmes multi-caméra à 360° de type AVM (*Around View Monitor*, en anglais) ou les caméras pare-brise de type *Mobileye*. Un schéma vue de dessus permettant de visualiser les recouvrements des champs observés par chaque caméra ainsi qu'une photo de leur montage sur le véhicule sont visibles sur la figure 3.1.

Caméra	Point Grey Grasshopper3	
Modèle	GS3-U3-23S6C-C	
Chroma	Couleur	
Capteur	CMOS 1/1.2"	
Résolution	1920 × 1200	
Obturateur	Global shutter	
Fréq. max.	163 IPS	
Fréq. acquisition	20 IPS	
Interface	USB3 Vision	
Objectif	Fisheye	Grand angle
Focale	2.7 mm	6.5 mm
Champ de vue (H)	185°	82°
Champ de vue (V)	185°	57°

TABLE 3.1 – Caractéristiques des caméras.

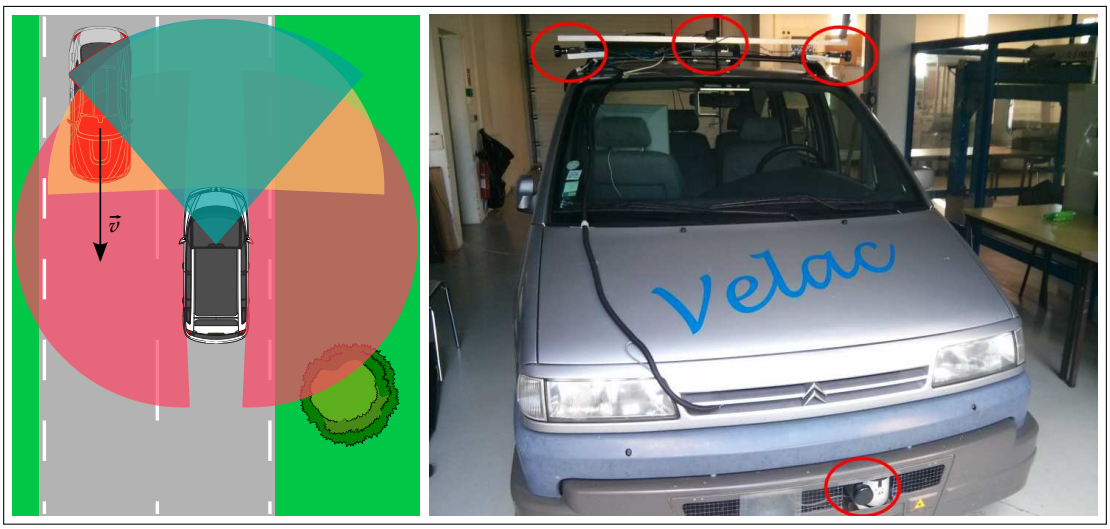

FIGURE 3.1 – Représentation vue de dessus du système multi-caméras et véhicule d'acquisition. À noter que les angles des secteurs circulaires représentant le champ de vision horizontal des caméras sont exacts.

3.2.3.2 Véhicules mobiles

Des véhicules électriques expérimentaux VIPALAB appartenant à l'Institut Pascal ont permis la réalisation des séquences présentant des véhicules en mouvement. Une

photo de plusieurs VIPALAB est visible figure 3.2.

3.2.3.3 Lieu d'acquisition

Les séquences du jeu de données ont été créées sur la Plate-forme Auvergne pour Véhicules Intelligents (PAVIN), rattachée à l'Institut Pascal et située sur le campus des Cézeaux d'Aubière. Le choix du site a été motivé par la crédibilité, le détail et le contrôle de l'environnement urbain qu'il propose. Une photo aérienne de PAVIN est visible figure 3.2.

FIGURE 3.2 – Véhicules VIPALAB et Plate-forme Auvergne pour Véhicules Intelligents.

3.2.3.4 Scénarios routiers

Chaque séquence a été développée selon plusieurs critères permettant l'évaluation de la méthode de détection et reconstruction d'objets mobiles dans des conditions de circulation différentes, à faible vitesse. Ces critères concernent notamment la direction et le sens de déplacement des objets mobiles, ainsi que leur visibilité dans les champs de vue de chaque caméra du système d'acquisition.

3.2.4 Méthodologie d'évaluation

3.2.4.1 Organisation des paramètres de la méthode

Différents paramètres influent sur les performances finales de la méthode proposée. Ces paramètres ont été organisés en quatre catégories différentes. La première catégorie de paramètres concerne l'extraction et l'appariement de points d'intérêt dans les images. Dans cette catégorie sont donc regroupés les types de détecteurs et descripteurs utilisés, mais aussi deux modifications relatives à la méthode d'extraction de ces points, détaillée dans la sous-section 2.4.1. Ces deux modifications concernent le filtrage ou non des points de l'ensemble S_1, ne prenant en compte que les meilleurs points de chaque case de la grille

n par n divisant l'image, tandis que la seconde modification concerne ou non le suivi des points précédemment triangulés, c'est à dire l'extraction de l'ensemble S_2. La seconde catégorie de paramètres se rapporte à la fonction utilisée afin de calculer l'erreur associée aux points évalués, c'est à dire l'erreur de reprojection ou l'erreur angulaire, comprenant également la mesure pixellique ou angulaire des contraintes $\mathsf{C}_E, \mathsf{C}_{E_\theta}, \mathsf{C}_{In}$ et C_{In_θ}. La troisième catégorie de paramètres concerne les seuils géométriques permettant d'évaluer cette erreur, qui comprend également les seuils T_L, T_E et T_{E_θ} associés à l'appariement temporel ou stéréo de points d'intérêt. Enfin, la dernière catégorie de paramètres porte sur les seuils utilisés afin d'évaluer la consistance de la trajectoire des points mobiles. Toutes ces catégories de paramètres sont reportées dans le tableau 3.2.

Extraction et appariement		Erreur	Seuils géométriques	Trajectoire
SIFT ORB AKAZE DAISY	Filtrage S_1 Extraction S_2	Reprojection Angulaire	T_L $\mathsf{T}_E, \mathsf{T}_{E_\theta}$ $\mathsf{T}_{In}, \mathsf{T}_{In_\theta}$ $\mathsf{T}_C, \mathsf{T}_{C_\theta}$ $\mathsf{T}_{M1}, \mathsf{T}_{M1_\theta}$	$\mathsf{T}_{T1_{min}}, \mathsf{T}_{T1_{max}}$ T_{T2} T_{T3}

TABLE 3.2 – Organisation des paramètres de la méthode.

3.2.4.2 Principe et indicateurs utilisés pour la mesure de performances

Bien qu'intrinsèquement importante, la mesure de la précision géométrique des reconstructions a été écartée lors de l'évaluation des performances. Cette décision est motivée par le fait, d'une part, que cette précision est dépendante des caractéristiques et de la calibration du système d'acquisition, et d'autre part, que la méthode proposée n'y apporte pas d'améliorations significatives en rapport à ce qui a déjà été publié dans la littérature, qu'il s'agisse de reconstruction par *SfM* ou *Stéréo Passive*. La fonctionnalité majeure de la méthode proposée se situe dans la classification et la reconstruction de points mobiles. Afin d'en évaluer les performances quantitatives, les deux indicateurs utilisés sont la précision et le rappel. La mesure de la précision consiste à évaluer le nombre de points correctement classés, c'est à dire les vrais positifs, par rapport aux nombre total de points de la classe évaluée, comprenant les vrais positifs ainsi que les faux positifs, tel que :

$$\text{Précision} = \frac{\text{Vrais positifs}}{\text{Vrais positifs} + \text{Faux positifs}} \tag{3.1}$$

En revanche, la mesure du rappel consiste à évaluer le nombre de points correctement classés en rapport au nombre de points appartenant réellement à la classe évaluée, c'est à dire l'ensemble des vrais positifs et des faux négatifs, tel que :

$$\text{Rappel} = \frac{\text{Vrais positifs}}{\text{Vrais positifs} + \text{Faux négatifs}} \tag{3.2}$$

Afin de mesurer les performances de la méthode pour la classification de points mobiles en termes de précision et de rappel, chaque objet mobile observé a été labellisé au sein de toutes les séquences retenues au moyen de masques binaires $\mathcal{M}_{j,t}$. La bonne ou mauvaise classification des points reconstruits est ensuite déterminée selon que leur point d'intérêt associé se situe ou non sur un masque dans l'image $p^n_{j,t} \in \mathcal{M}_{j,t} \vee p^n_{j,t} \notin \mathcal{M}_{j,t}$. Un vrai point mobile, ou vrai positif, est donc un point classé mobile et appartenant à un masque binaire. Un faux point mobile, ou faux positif, est un point classé mobile et n'appartenant pas à un masque binaire. Un vrai négatif, ou vrai point statique, est un point classé statique et n'appartenant pas à un masque binaire. Enfin, un faux négatif, ou faux point statique, est un point classé statique et appartenant à un masque binaire. Un récapitulatif de ces interprétations est proposé dans le tableau 3.3. La labellisation des différentes séquences a elle même été effectuée soit manuellement, soit à l'aide de la méthode par apprentissage profond MASK R-CNN, proposée par He *et al.* [66] et basée sur l'utilisation d'un réseau de neurones convolutif (*Convolutional Neural Network* ou CNN, en anglais). Cette méthode de labellisation automatique a été privilégiée car elle permet la segmentation par instance de chaque objet de la scène appartenant à la même classe (voitures, arbres, etc.), et donc une sélection plus aisée des objets mobiles seulement. À noter que malgré ses excellentes performances, cette labellisation automatique peut éventuellement introduire un biais lors de l'évaluation des performances, car les ombres des objets mobiles, elles aussi mouvantes, ne sont pas prises en compte. Enfin, s'agissant d'une tâche très fastidieuse, seules les images provenant d'une unique caméra par séquence ont été labellisées, c'est à dire les images acquises par la caméra gauche du jeu de données KITTI et les images acquises par la caméra pare-brise (grand-angle) du jeu de données IP. En pratique, si l'on rejette effectivement certains points mobiles de la comptabilisation, il s'agit d'une minorité de points, l'essentiel se trouvant au sein des images acquises pas les caméras retenues. Deux exemples de labellisation manuelle et automatique d'un même objet sont illustrés figure 3.3.

Résultat	Vrai	Faux
Positif	$P_n \in \mathcal{C}_M \wedge p^n_{j,t} \in \mathcal{M}_{j,t}$	$P_n \in \mathcal{C}_M \wedge p^n_{j,t} \notin \mathcal{M}_{j,t}$
Négatif	$P_n \in \mathcal{C}_S \wedge p^n_{j,t} \notin \mathcal{M}_{j,t}$	$P_n \in \mathcal{C}_S \wedge p^n_{j,t} \in \mathcal{M}_{j,t}$

TABLE 3.3 – Définitions des classes d'évaluation de la précision et du rappel.

FIGURE 3.3 – Labellisation d'un objet mobile pour la mesure de précision et de rappel. À gauche, une labellisation manuelle et à droite, une labellisation automatique avec MASK R-CNN ne prennant pas en compte l'ombre portée du véhicule.

3.2.4.3 Protocole de test

L'objectif du protocole de test proposé consiste à déterminer les valeurs de l'ensemble de paramètres permettant de s'approcher des meilleurs résultats quantitatifs de la méthode sur une séquence particulière, puis de réutiliser ensuite cet ensemble de paramètres pour toutes les séquences appartenant au même jeu de données. En conséquence, bien qu'il puisse y avoir de larges similarités de paramétrage entre plusieurs jeux de données différents, le protocole proposé peut éventuellement être assimilé à une étape de calibration de la méthode pour un jeu de données précis. Le choix de la séquence ou des séquences de calibration est effectué sur la base du mouvement des objets mobiles observés, avec une priorité donnée à la calibration de mouvements francs et prolongés, longitudinaux ou latéraux, d'objets de taille significative. Considérant le nombre de paramètres influant sur les résultats, ce protocole suit un ordre itératif affectant individuellement chaque catégorie de paramètres, afin d'éviter une explosion combinatoire du nombre de tests à effectuer. Cet ordre respecte en outre un certaine logique, visant à fixer un maximum de paramètres avant de tester par force brute l'influence d'autres paramètres. Ce protocole de test procède en cinq étapes, décrites ci-après, avec un récapitulatif du protocole de test proposé dans le tableau 3.4.

Sélection du type d'erreur. En premier lieu, le type d'erreur utilisé est sélectionné sur la base des caractéristiques du système d'acquisition considéré. L'erreur de reprojection sera préférée lorsque le système ne comporte pas de caméras *fisheye* et ne présente donc pas de distorsions importantes, alors que l'erreur angulaire sera préférée dans le cas contraire.

Évaluation sur un ensemble de points manuellement ou automatiquement extraits et appareillés. Dans un second temps, la séquence est évaluée sur un ensemble de points mobiles manuellement ou automatiquement extraits et appareillés, afin de déterminer sur l'ensemble de ces observations les intervalles de valeurs associés à chacun des seuils que comporte la méthode. Dans cette étape, chaque seuil est paramétré de manière empirique, de façon à maximiser la bonne détection des points mobiles. Les intervalles de valeurs permettent en outre de fixer le seuil de la contrainte de localité d'appariement T_L ainsi que les seuils de trajectoire $\mathsf{T}_{T1_{min}}, \mathsf{T}_{T1_{max}}, \mathsf{T}_{T2}$ et T_{T3}, mais aussi d'établir des valeurs minimales et maximales pour chacun des autres seuils géométriques. À l'issue de cette étape, la valeur de chaque paramètre correspond à une estimation brute mais fonctionnelle.

Première sélection des paramètres d'extraction et d'appariement. Une première sélection est opérée de manière empirique afin de retenir un couple de détecteurs et descripteurs de points d'intérêt produisant un nombre relativement élevé d'appariements corrects, de même que le choix du filtrage ou non de l'ensemble de points d'intérêt S_1 ainsi que de l'extraction de l'ensemble de points d'intérêt S_2. De manière générale, les points d'intérêt SIFT ont été sélectionnés dans cette étape du fait de leur bonnes performances et très grande popularité dans la littérature. Un choix concernant les modifications relatives à l'extraction des ensembles S_1 et S_2 est également opéré à cette étape, avec une priorité donnée à l'extraction d'un nombre relativement élevé de points en un temps de calcul raisonnable.

Tests itératifs par force brute afin de fixer les seuils géométriques. La troisième étape consiste ensuite à tester itérativement la méthode selon une détection et un appariement automatique de points d'intérêt afin de déterminer l'influence des différents seuils géométriques $\mathsf{T}_E, \mathsf{T}_{E_\theta}, \mathsf{T}_{In}, \mathsf{T}_{In_\theta}, \mathsf{T}_C, \mathsf{T}_{C_\theta}, \mathsf{T}_{M1}$ et T_{M1_θ} sur les performances. À noter que la plupart des valeurs de ces seuils ont volontairement été groupées du fait de leur lien géométrique évident. Ces groupes de paramètres sont définis tels que :

$$\mathsf{T}_E = \mathsf{T}_{In} = \mathsf{T}_C \qquad \mathsf{T}_{E_\theta} = \mathsf{T}_{In_\theta} = \mathsf{T}_{C_\theta} \tag{3.3}$$

À l'issue de cette étape, les valeurs des paramètres permettant d'obtenir les meilleures performances sont retenues.

Évaluation de l'influence des paramètres d'extraction et d'appariement. Enfin, la dernière étape permet d'évaluer les performances obtenues selon différents détecteurs et descripteurs de points d'intérêt, de même que selon les modifications portant sur l'extraction des ensembles de points d'intérêt S_1 et S_2.

Étape	Extraction et appariement	Paramètres
1	-	Erreur de reprojection ou erreur angulaire
2	Manuel ou automatique	$\mathsf{T}_L, \mathsf{T}_{T1_{min}}, \mathsf{T}_{T1_{max}}, \mathsf{T}_{T2}, \mathsf{T}_{T3}$
3	-	Détecteur, descripteur et modifications S_1, S_2
4	Automatique	$\mathsf{T}_E, \mathsf{T}_{E_\theta}, \mathsf{T}_{In}, \mathsf{T}_{In_\theta}, \mathsf{T}_C, \mathsf{T}_{C_\theta}, \mathsf{T}_{M1}, \mathsf{T}_{M1_\theta}$
5	Automatique	Détecteur, descripteur et modifications S_1, S_2

TABLE 3.4 – Récapitulatif du protocole de test.

3.3 Performances quantitatives, qualitatives et limitations

3.3.1 Format de présentation des résultats

Les vues des résultats proposées dans les sections suivantes sont extraites des tests pour lesquels les valeurs utilisées pour l'ensemble des paramètres de la méthode ont permis l'obtention des meilleures performances, sauf mention explicite dans le cas contraire. Dans chaque vue ou ensemble de vues, les points reconstruits sont présentés tels que des cercles de différentes couleurs. Ces couleurs sont respectivement le gris, représentant les points neutres $P_n \in \mathcal{C}_N$, le vert, représentant les points fixes $P_n \in \mathcal{C}_S$, le orange, représentant les points candidats $P_n \in \mathcal{C}_C$ ainsi que le rouge, représentant les points mobiles $P_n \in \mathcal{C}_M$. D'autre part, les masques permettant la mesure de la précision et du rappel sont représentés par les régions bleues. Les trajectoires en vue de dessus associées à chaque vue ou ensemble de vues, lorsque présentes, sont affichées à leur droite, avec en bleu la trajectoire du véhicule d'acquisition et en rouge les trajectoires des points mobiles reconstruits, visibles sur la ou les vues associées.

3.3.2 Jeu de données KITTI

3.3.2.1 Séquence 3

Présentation. La séquence 3 du jeu de données KITTI présente une voiture mobile en début de séquence, effectuant un mouvement latéral de gauche à droite. Les images 40 à 80 de la séquence sont évaluées, avec une apparition du véhicule mobile à l'image 50 pour la caméra gauche et à l'image 51 pour la caméra droite. La labellisation pour cette séquence a été effectuée de manière manuelle, et comprend pour chaque masque l'ombre visible du véhicule, qui est aussi en mouvement.

Sélection et initialisation des premiers paramètres. Le jeu de données KITTI propose pour chaque séquence des paires d'images rectifiées, c'est à dire qu'elles ne

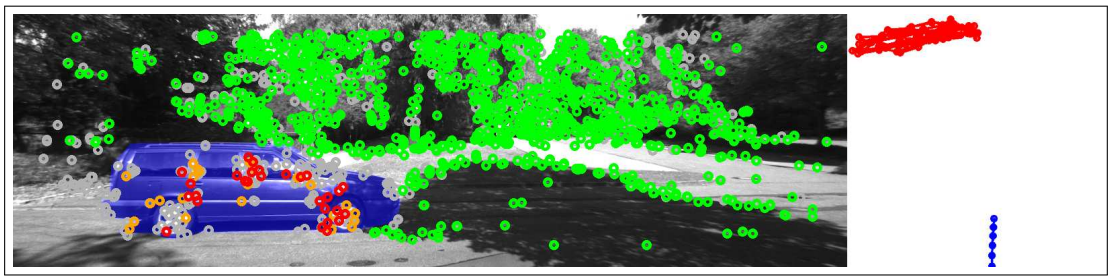

FIGURE 3.4 – Vue et trajectoires extraites de la séquence KITTI 3 avec les paramètres I du tableau 3.6.

présentent pas de distorsions particulières. L'erreur de reprojection a donc été sélectionnée pour ce jeu de données lors de la première étape du protocole de test. Lors de la seconde étape, l'évaluation des paramètres sur un ensemble de points mobiles manuellement extraits et appareillés a permis de fixer les premiers seuils tels que $\mathsf{T}_L = 100$ pixels, $\mathsf{T}_{T1_{min}} = 0.8$ mètre, $\mathsf{T}_{T1_{max}} = 1.5$ mètres, $\mathsf{T}_{T2} = 0.2$ mètre et $\mathsf{T}_{T3} = 110.0°$. Les détecteurs et descripteurs SIFT ont ensuite été sélectionnés lors de la troisième étape, alors que l'ensemble de points d'intérêt S_1 n'a pas été filtré et que l'ensemble de points d'intérêt S_2 n'a pas été extrait.

Influence des seuils géométriques. Les tests effectués lors de la quatrième étape portent sur l'évaluation de l'influence des différents seuils géométriques sur les performances de la méthode. Plusieurs couples de valeurs allant de 1 à 4 pixels d'erreur ont été testés pour les seuils $\mathsf{T}_E = \mathsf{T}_{In} = \mathsf{T}_C$ et T_{M1}, alors que les autres paramètres, dont les valeurs sont précisées au paragraphe précédent, sont fixés. Les résultats obtenus sont détaillés dans le tableau 3.5 alors qu'une représentation graphique est présentée figure 3.8. Une première observation de ces résultats montre que la valeur du seuil T_{M1} associé à la contrainte de mobilité C_{M1} n'influence que marginalement les performances, avec des écarts très faibles pour des valeurs comprises entre 1 et 4 pixels, lorsque les seuils T_E, T_{In} et T_C sont fixés au préalable. Ces derniers en revanche ont une incidence importante, non pas au niveau des performances, avec des écarts assez faibles entre valeur minimum et maximum, de 3.50% pour la précision et de 5.52% pour le rappel, mais au niveau du nombre de points mobiles reconstruits avec des valeurs minimum et maximum allant du simple au double, de 102 à 220. Les meilleures performances retenues, compte tenu de ce dernier point, sont établies pour l'ensemble de paramètres F du tableau 3.5, dont les seuils géométriques sont fixés à $\mathsf{T}_E = \mathsf{T}_{In} = \mathsf{T}_C = \mathsf{T}_{M1} = 2.0$ pixels et pour lesquels la précision, le rappel ainsi que le nombre de point mobiles atteignent respectivement 99.22%, 82.58% et 216.

ID	$\mathsf{T}_E, \mathsf{T}_{In}, \mathsf{T}_C$	T_{M1}	Précision	Rappel	Pts. mob.
A	1.0	1.0	96.49	79.71	102
B	1.0	2.0	96.49	78.57	102
C	1.5	1.0	**100.0**	80.67	170
D	1.5	2.0	**100.0**	82.53	181
E	2.0	1.0	99.15	**83.09**	202
F	2.0	2.0	99.22	82.58	216
G	2.0	3.0	99.22	82.58	216
H	2.0	4.0	99.22	82.58	216
I	3.0	2.0	99.25	81.59	219
J	3.0	3.0	99.25	81.59	219
K	3.0	4.0	99.25	81.59	219
L	4.0	2.0	97.72	77.71	217
M	4.0	3.0	97.76	77.05	**220**
N	4.0	4.0	97.76	77.05	**220**

TABLE 3.5 – Influence des seuils géométriques sur la séquence KITTI 3.

Influence de l'extraction et de l'appariement de points d'intérêt. La dernière étape du protocole de test vise à évaluer l'influence de l'étape d'extraction et d'appariement de points d'intérêt. Cinq couples de détecteurs et descripteurs ont été testés, SIFT, ORB, AKAZE et le descripteur DAISY couplé au détecteur FAST dans un premier temps puis au détecteur SIFT dans un second temps. Le couple obtenant les meilleurs résultats en termes de précision et de rappel est ORB, avec 100% de précision lorsque le filtrage S_1 est opéré, et 94.23% de rappel lorsque le filtrage de S_1 et l'extraction de S_2 sont opérés (paramètres F et H du tableau 3.6). À noter cependant que le nombre de points obtenus dans les deux cas est loin d'être important. Le descripteur AKAZE, pour sa part, sans filtrage de S_1 ni extraction de S_2 (paramètres I du tableau 3.6), offre des performances proches à ORB, avec -3.35% en précision et -3.65% en rappel, et détecte pratiquement le double de points mobiles avec 417 points contre 249 pour ORB dans le meilleur des cas. En ce qui concerne les opérations de filtrage de l'ensemble S_1 ainsi que l'extraction de l'ensemble S_2, on peut noter une tendance à la baisse des performances, à l'exception des tests effectués avec les descripteurs DAISY et dans une moindre mesure ORB, où l'on note une très légère hausse. Ces variations marginales sont probablement imputables aux caractéristiques du système d'acquisition, en particulier la résolution faible des images qui ne tire pas bénéfice du filtrage de l'ensemble S_1, ou encore la fréquence d'acquisition d'images assez élevée qui ne permet pas d'exploiter correctement le suivi des points précédemment triangulés, opéré par extraction de l'ensemble S_2. Tous ces résultats sont reportés dans le tableau 3.6 et la représentation graphique présentée figure 3.9.

ID	Dét. - Desc.	S_1	S_2	Précision	Rappel	Pts. mob.
A	SIFT	-	-	99.22	82.58	216
B	SIFT	×	-	99.18	84.13	208
C	SIFT	-	×	94.56	83.65	172
D	SIFT	×	×	95.60	85.29	159
E	ORB (3000)	-	-	97.50	90.0	249
F	ORB (3000)	×	-	**100.0**	90.21	148
G	ORB (3000)	-	×	95.78	89.21	200
H	ORB (3000)	×	×	98.0	**94.23**	93
I	AKAZE	-	-	96.65	90.58	417
J	AKAZE	×	-	94.79	87.50	332
K	FAST - DAISY	-	-	92.30	76.50	**801**
L	FAST - DAISY	×	-	96.92	86.06	478
M	SIFT - DAISY	-	-	97.43	80.85	284
N	SIFT - DAISY	×	-	98.59	82.35	260

TABLE 3.6 – Influence de l'extraction et de l'appariement sur la séquence KITTI 3.

3.3.2.2 Séquence 7

Présentation. La séquence 7 du jeu de données KITTI présente une voiture mobile suivie par le véhicule d'acquisition durant la majeure partie de la séquence selon un mouvement longitudinal et latéral avec un virage à gauche, puis un camion en toute fin de séquence arrivant en sens inverse. Les images 850 à 1100 de la séquence sont évaluées, avec une apparition du véhicule mobile dès le début de la séquence. La labellisation pour cette séquence a été effectuée de manière automatique à l'aide de la méthode de labellisation par apprentissage profond MASK R-CNN, ne comprenant pas pour chaque masque l'ombre visible du véhicule, qui est aussi en mouvement.

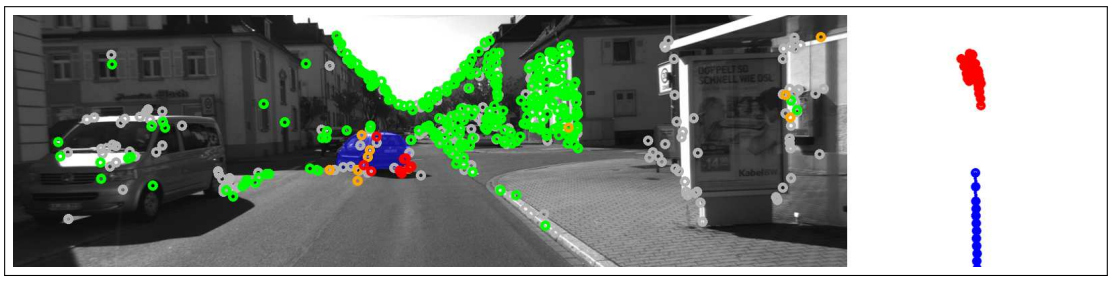

FIGURE 3.5 – Vue et trajectoires extraites de la séquence KITTI 7 avec les paramètres B du tableau 3.7.

Performances quantitatives. Les paramètres initiaux sélectionnés afin d'évaluer les performances de la méthode sur la séquence 7 du jeu de données KITTI sont identiques à ceux ayant permis d'obtenir les meilleurs résultats sur la séquence 3, c'est à dire prenant

pour valeur $\mathsf{T}_L = 100$ pixels, $\mathsf{T}_{T1_{min}} = 0.8$ mètre, $\mathsf{T}_{T1_{max}} = 1.5$ mètres, $\mathsf{T}_{T2} = 0.2$ mètre, $\mathsf{T}_{T3} = 110.0°$ et $\mathsf{T}_E = \mathsf{T}_{In} = \mathsf{T}_C = \mathsf{T}_{M1} = 2.0$ pixels, en utilisant le détecteur et descripteur AKAZE, sans filtrage de l'ensemble S_1 ni extraction de S_2 (paramètres A du tableau 3.7). Ces paramètres produisent des résultats de 80.81% et 31.61% en précision et rappel sur cette séquence, respectivement. Une rectification de ces paramètres a consisté à réduire la contrainte minimum sur le seuil de distance minimal pour l'évaluation de la trajectoire des points mobiles à $\mathsf{T}_{T1_{min}} = 0.5$ mètre, ce qui a permis d'augmenter les performances en rappel à 46.31% et de doubler le nombre de points mobiles, avec une précision légèrement inférieure à 79.21% (paramètres B du tableau 3.7). Enfin, deux autres tests visant à évaluer l'influence des paramètres géométriques $\mathsf{T}_E = \mathsf{T}_{In} = \mathsf{T}_C$ avec des valeurs de 3 et 4 pixels n'ont pas significativement amélioré la précision mais fortement diminué le rappel, plus de 10% inférieur à sa meilleure valeur. Ces résultats sont présentés dans le tableau 3.7 et la figure 3.10.

ID	$\mathsf{T}_{T1_{min}}$	$\mathsf{T}_E, \mathsf{T}_{In}, \mathsf{T}_C$	Précision	Rappel	Pts. mob.
A	0.8	2.0	80.81	31.61	484
B	0.5	2.0	79.21	**46.31**	**968**
C	0.5	3.0	82.68	36.66	867
D	0.5	4.0	**84.25**	30.86	757

TABLE 3.7 – Tests effectués sur la séquence KITTI 7.

3.3.2.3 Séquence 10

Présentation. La séquence 10 du jeu de données KITTI présente un camion mobile assez éloigné du véhicule d'acquisition en début de séquence, effectuant une marche arrière longitudinalement puis latéralement de gauche à droite. Les images 930 à 1020 de la séquence sont évaluées, avec une apparition du véhicule mobile présentant un mouvement très faible lors des premières images, puis plus marqué ensuite. La labellisation pour cette séquence a été effectuée de manière manuelle, et comprend pour chaque masque l'ombre visible du véhicule, qui est aussi en mouvement.

Performances quantitatives. Comme pour la séquence 7, les paramètres initiaux retenus pour cette séquence sont ceux pour lesquels les meilleurs résultats ont été observés sur la séquence 3 (paramètres A du tableau 3.8). Ces paramètres initiaux permettent d'obtenir respectivement 66.66% et 2.18% en précision et rappel. En réajustant les valeur des seuils $\mathsf{T}_{T1_{min}} = 0.3$ mètre et $\mathsf{T}_{T1_{max}} = 2.0$ mètres, les résultats en précision et rappel augmentent sensiblement, à 89.09% et 21.49%, alors que le nombre de points mobiles passe de 20 à 148 (paramètres B du tableau 3.8). De la même manière que

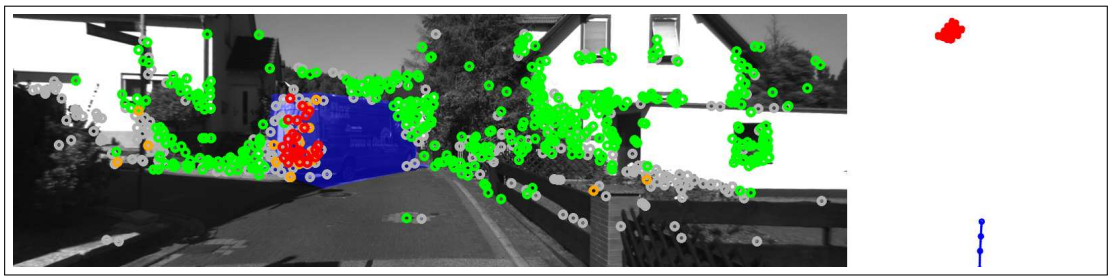

FIGURE 3.6 – Vue et trajectoires extraites de la séquence KITTI 10 avec les paramètres B du tableau 3.8.

pour la séquence 7, la variation des seuils géométriques $\mathsf{T}_E, \mathsf{T}_{In}$ et T_C ne permettent pas d'augmenter les performances. Ces résultats sont présentés dans le tableau 3.8 et la figure 3.11.

ID	$\mathsf{T}_{T1_{min}}$	$\mathsf{T}_{T1_{max}}$	$\mathsf{T}_E, \mathsf{T}_{In}, \mathsf{T}_C$	Précision	Rappel	Pts. mob.
A	0.8	1.5	2.0	66.66	2.18	20
B	0.3	2.0	2.0	**89.09**	**21.49**	**148**
C	0.3	2.0	3.0	84.78	12.58	118
D	0.3	2.0	4.0	73.33	8.68	112

TABLE 3.8 – Tests effectués sur la séquence KITTI 10.

3.3.2.4 Séquence 19

Présentation. La séquence 19 du jeu de données KITTI présente plusieurs voitures mobiles à une intersection en début de séquence, puis une voiture mobile suivie par le véhicule d'acquisition durant la majeure partie du reste de la séquence, selon un mouvement longitudinal et latéral avec un virage à gauche, et en fin de séquence, deux cyclistes effectuant un mouvement latéral de droite à gauche sur toute la largeur du champ de vue horizontal. Les images 2010 à 2270 de la séquence sont évaluées, avec une apparition des éléments mobiles dès le début de la séquence et ce jusqu'à la fin. La labellisation pour cette séquence a été effectuée de manière automatique à l'aide de la méthode de labellisation par apprentissage profond MASK R-CNN, ne comprenant pas pour chaque masque les ombres visibles des éléments mobiles, qui sont aussi en mouvement.

Performances quantitatives. Les paramètres initiaux retenus pour la séquence 19 sont les mêmes que ceux ayant permis d'obtenir les meilleures performances sur la séquence 3 du jeu de données KITTI (paramètres A du tableau 3.9). Les performances

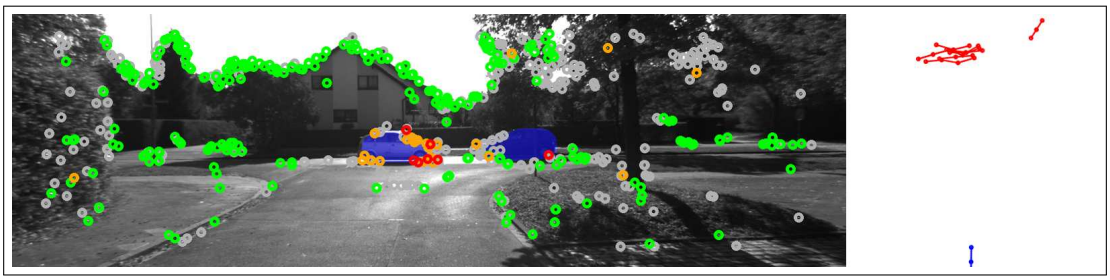

FIGURE 3.7 – Vue et trajectoires extraites de la séquence KITTI 19 avec les paramètres B du tableau 3.9.

obtenues avec cet ensemble de paramètres sont de 72.5% et 20.51% en précision et rappel, respectivement. Un réajustement des seuils de trajectoire à $\mathsf{T}_{T1_{min}} = 0.5$ mètre et $\mathsf{T}_{T1_{max}} = 2.0$ mètres (paramètres B du tableau 3.9) augmente sensiblement le rappel et le nombre de points reconstruits à 53.80% et 1009 contre 211 précédemment, alors qu'une baisse supplémentaire du seuil de distance minimum à $\mathsf{T}_{T1_{min}} = 0.3$ mètre (paramètres E du tableau 3.9) permet d'obtenir les meilleures performances en rappel à 57.40%. Comme sur les précédentes séquences du jeu de données KITTI, augmenter les valeurs des seuils géométriques $\mathsf{T}_E, \mathsf{T}_{In}, \mathsf{T}_C$ ne permet pas d'améliorer significativement les résultats. Les résultats relatifs à cette séquence sont présentés dans le tableau 3.9 et la figure 3.12.

ID	$\mathsf{T}_{T1_{min}}$	$\mathsf{T}_{T1_{max}}$	$\mathsf{T}_E, \mathsf{T}_{In}, \mathsf{T}_C$	Précision	Rappel	Pts. mob.
A	0.8	1.5	2.0	**72.5**	20.51	211
B	0.5	2.0	2.0	59.40	53.80	1009
C	0.5	2.0	3.0	60.28	39.38	880
D	0.5	2.0	4.0	58.78	31.12	824
E	0.3	2.0	2.0	53.62	**57.40**	**1204**
F	0.3	2.0	3.0	53.82	42.90	1107
G	0.3	2.0	4.0	54.65	33.84	956

TABLE 3.9 – Tests effectués sur la séquence KITTI 19.

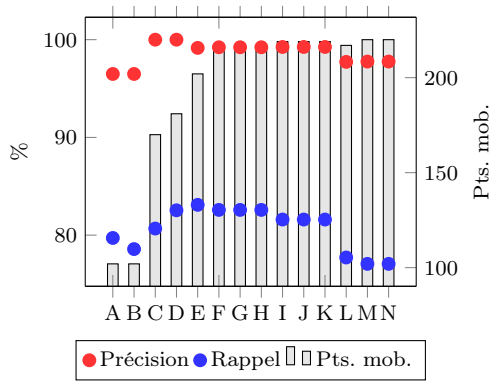

FIGURE 3.8 – Influence des seuils géométriques - Séquence KITTI 3.

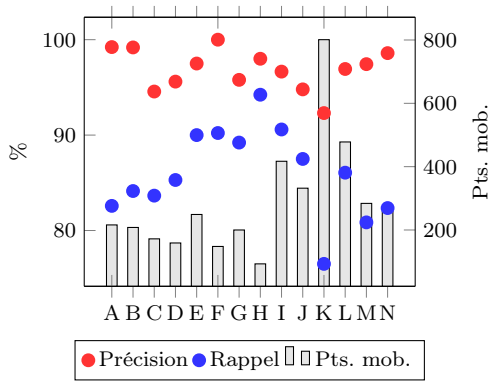

FIGURE 3.9 – Influence de l'extraction et de l'appariement sur la séquence KITTI 3.

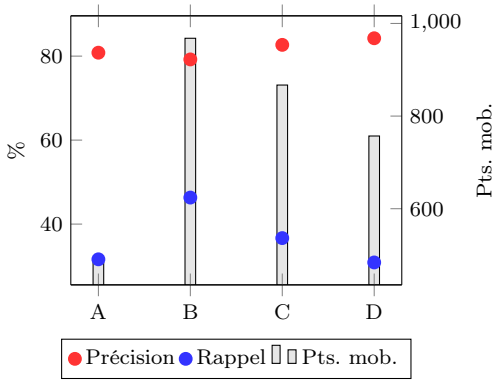

FIGURE 3.10 – Tests effectués sur la séquence KITTI 7.

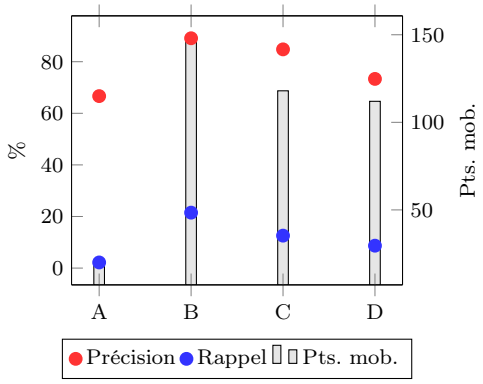

FIGURE 3.11 – Tests effectués sur la séquence KITTI 10.

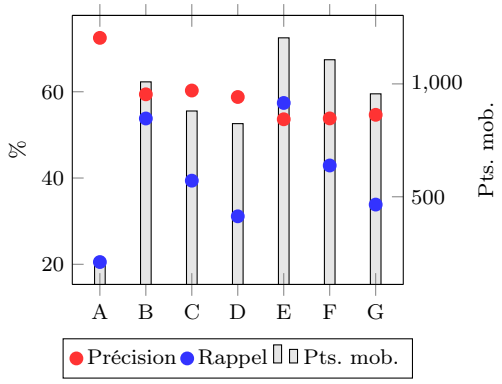

FIGURE 3.12 – Tests effectués sur la séquence KITTI 19.

3.3.2.5 Analyse qualitative des performances sur le jeu de données KITTI

Quatre illustrations, présentant chacune une vue de la séquence évaluée ainsi que ses trajectoires associées sont proposées dans les figures 3.4, 3.5, 3.6 et 3.7. Ces figures illus-

trent de manière convaincante les performances qualitatives atteignables par la méthode proposée. La plupart des points classés candidats ainsi que la totalité des points mobiles se trouvent sur les masques, alors que quelques points candidats ainsi que l'immense majorité des points fixes ne s'y trouvent pas. Alors que les performances quantitatives sur ce jeu de données sont relativement bonnes, en particulier sur la séquence 3, plusieurs facteurs permettent d'expliquer les performances plus en retrait en termes de précision et rappel sur les séquences 7, 10 et 19. En effet, alors que dans la séquence 3, le véhicule mobile se déplace latéralement à vitesse relativement constante, les véhicules mobiles labellisés dans les autres séquences ne se déplacent que très faiblement à de multiples occasions, ce qui ne permet pas la détection de leur mouvement car les points reconstruits sur ces véhicules sont alors consistants, et donc considérés comme statiques. Une illustration de ce type de situation est proposée figure 3.13.

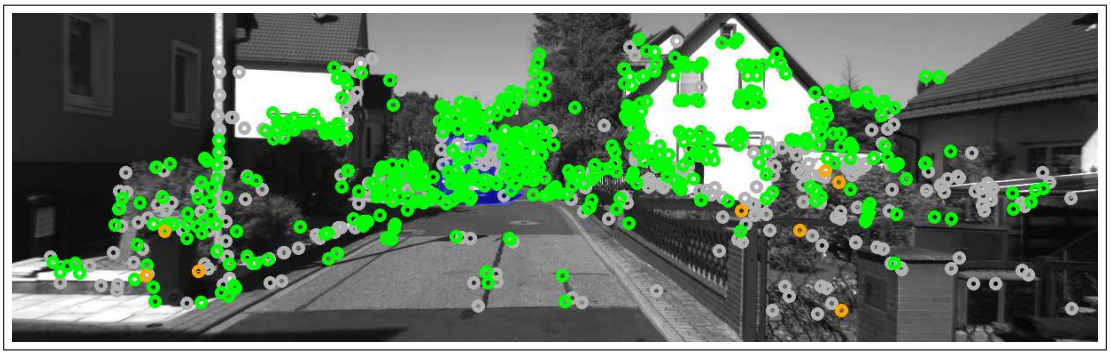

FIGURE 3.13 – Exemple de faux négatifs : points classés statiques sur un véhicule peu mobile, extrait de la séquence KITTI 10.

D'autre part, une performance faible en rappel peut s'expliquer par la reconstruction de points classés statiques lorsqu'ils se trouvent proches des contours de l'objet mobile et sur des surfaces relativement uniformes. Un exemple est proposé figure 3.14.

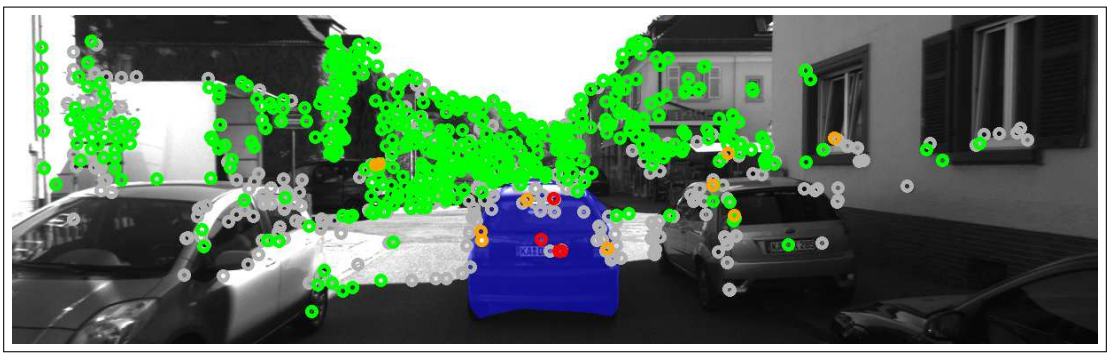

FIGURE 3.14 – Exemple de faux négatifs : points classés statiques sur les contours relativement uniformes d'un véhicule mobile, extrait de la séquence KITTI 7.

Enfin, la labellisation automatique effectuée grâce à la méthode MASK R-CNN et utilisée sur les séquences 7 et 19 ne permet pas la labellisation des ombres des véhicules mobiles, ce qui peut expliquer la précision relativement inférieure obtenue sur ces séquences. En effet, plusieurs des points reconstruits se trouvant sur les ombres des véhicules en mouvement sont classés mobiles, ce qui reflète effectivement leur déplacement, mais ne se trouvent alors pas sur un masque, ce qui engendre une baisse de précision. Une illustration est proposée figure 3.15.

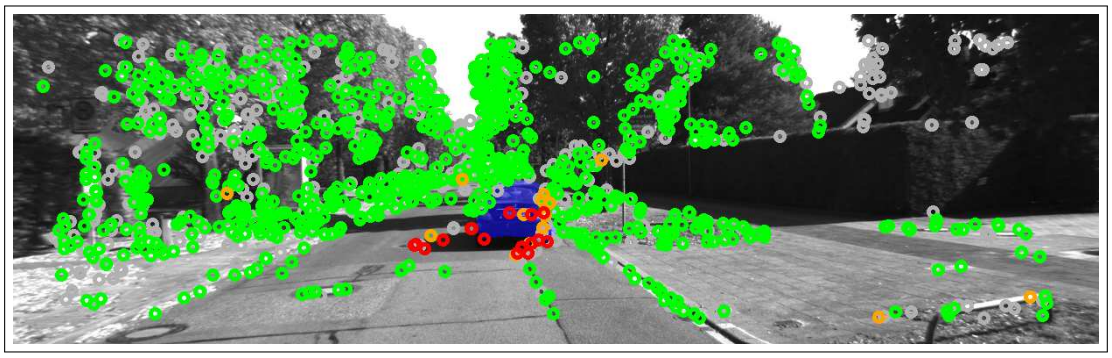

FIGURE 3.15 – Exemple de faux positifs : points classés mobiles sur l'ombre portée non labellisée d'un véhicule mobile, extrait de la séquence KITTI 19.

3.3.2.6 Conclusion des tests effectués sur le jeu de données KITTI

L'analyse quantitative et qualitative des performances obtenues sur ce jeu de données montre que la méthode proposée produit des résultats relativement bons en termes de précision, avec des valeurs maximales sur toutes les séquences comprises entre 72.5% et 100%, alors que les valeurs maximales pour le rappel, plus en retrait et dont les principales explications sont abordées au précédent paragraphe, sont comprises entre 21.49% et 94.23%. Suivant cette analyse, un ensemble de paramètres de référence, produisant des résultats satisfaisants sur l'ensemble du jeu de données, peut être proposé. Le tableau 3.10 récapitule ces paramètres de référence.

Extraction et appariement		Erreur	Seuils géométriques		Trajectoire	
Dét. - Desc.	AKAZE	Reprojection	T_L	100 px	$\mathsf{T}_{T1_{min}}$	0.3 m
Filtrage S_1	Non		T_E	2.0 px	$\mathsf{T}_{T1_{max}}$	2.0 m
Extraction S_2	Non		T_{In}	2.0 px	T_{T2}	0.2 m
			T_C	2.0 px	T_{T3}	110°
			T_{M1}	2.0 px		

TABLE 3.10 – Paramètres de référence pour le jeu de données KITTI.

3.3.3 Jeu de données IP

3.3.3.1 Séquence 10

Présentation. La séquence 10 du jeu de données IP présente un véhicule mobile VIPALAB effectuant un mouvement latéral puis longitudinal dans un rond-point. Les images 334 à 466 ont été évaluées, avec une apparition du véhicule mobile en début de séquence, alors que sa labellisation, comprenant également son ombre, a été effectuée manuellement.

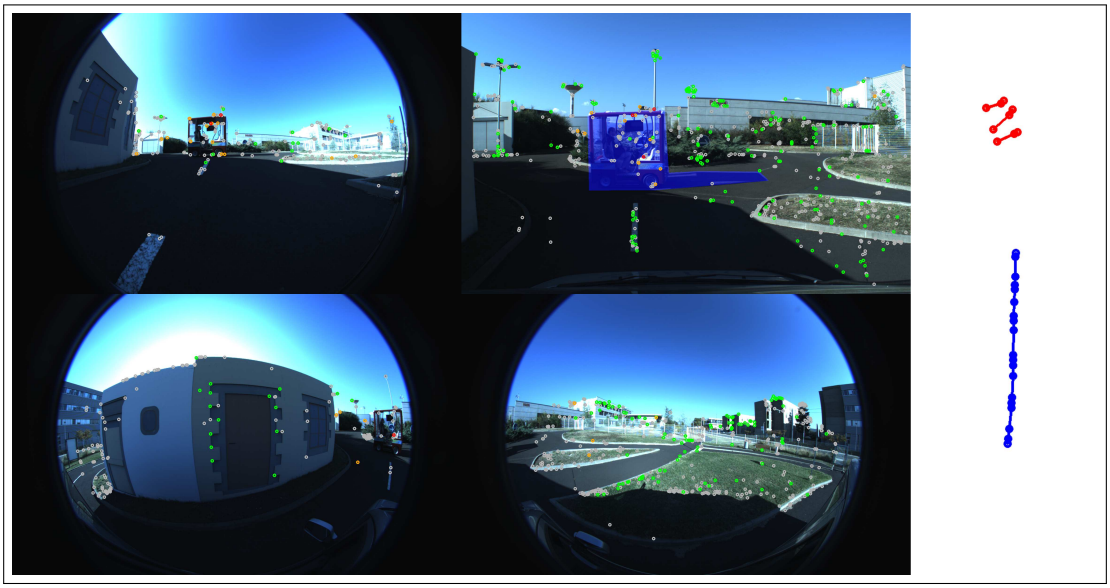

FIGURE 3.16 – Vues et trajectoires extraites de la séquence IP 10 avec les paramètres D du tableau 3.12. Deux faux négatifs sur les surfaces vitrées du véhicule mobile sont visibles sur la vue de la caméra pare-brise en haut à droite.

Sélection et initialisation des premiers paramètres. Le jeu de données IP propose pour chaque séquence quatre images par temporalité, dont certaines ont été acquises par des caméras *fisheye* et qui présentent donc de fortes distorsions. Bien que peu indiqué dans ce type de configuration, plusieurs tests ont été effectués par calcul de l'erreur de reprojection en complément de l'erreur angulaire afin de comparer les performances des deux méthodes. Lors de la seconde étape, l'évaluation des paramètres sur un ensemble de points mobiles manuellement extraits et appareillés a permis de fixer les premiers seuils tels que $\mathsf{T}_L = 400$ pixels, $\mathsf{T}_{T1_{min}} = 0.1$ mètre, $\mathsf{T}_{T1_{max}} = 5.0$ mètres, $\mathsf{T}_{T2} = 1.0$ mètre et $\mathsf{T}_{T3} = 110.0°$. Les détecteurs et descripteurs SIFT ont ensuite été sélectionnés lors de la troisième étape, alors que l'ensemble de points d'intérêt S_1 a été filtré et que l'ensemble de points d'intérêt S_2 n'a pas été extrait.

Influence des seuils géométriques. L'influence des seuils géométriques a été évaluée en faisant varier de manière uniforme, par pas de 0.05° d'angle pour l'erreur angulaire et de 1 pixel pour l'erreur de reprojection, l'ensemble des seuils géométriques $\mathsf{T}_{E_\theta}, \mathsf{T}_{In_\theta}, \mathsf{T}_{C_\theta}, \mathsf{T}_{M1_\theta}$ et $\mathsf{T}_E, \mathsf{T}_{In}, \mathsf{T}_C, \mathsf{T}_{M1}$, compte tenu de la faible incidence de valeurs spécifiques pour T_{M1} observée sur le jeu de données KITTI. Bien qu'attendue, la baisse de performances sur ce jeu de données est tout de même importante en comparaison des résultats obtenus sur les données KITTI. Une première observation des résultats présentés dans le tableau 3.11 et la figure 3.19 permet de remarquer la plus grande linéarité des courbes obtenues par calcul de l'erreur de reprojection, de même que de meilleures performances combinées en précision et rappel, quand les performances obtenues par calcul de l'erreur angulaire se croisent, avec des paramètres plus stricts favorables à un meilleur rappel et des paramètres plus laxes favorables à une meilleure précision. La précision a dans ce cas été privilégiée lors du choix des meilleurs paramètres, afin de donner plus d'importance à la réduction de faux positifs. Les meilleures performances combinées sont obtenues par calcul de l'erreur de reprojection avec les seuils $\mathsf{T}_E, \mathsf{T}_{In}, \mathsf{T}_C$ et T_{M1} fixés à 4 pixels, avec 47.88% et 52.30% de précision et rappel, respectivement (paramètres C du tableau 3.11), alors que les meilleures performances combinées par calcul de l'erreur angulaire sont obtenues pour des valeurs de $\mathsf{T}_{E_\theta}, \mathsf{T}_{In_\theta}, \mathsf{T}_{C_\theta}$ et T_{M1_θ} fixées à 0.25° d'angle, à 46.05% et 35.0% (paramètres I du tableau 3.11). À noter que malgré des performances marginalement inférieures, de 0.69% et 0.45% pour la précision et le rappel, respectivement, les valeurs des seuils géométriques retenues lors des autres étapes du protocole de test utilisant l'erreur de reprojection ont été établies à 5 pixels, compte tenu du nombre supérieur de points mobiles reconstruits, à 180 contre 138, dans le cas de seuils fixés à 4 pixels.

ID	Err.	$\mathsf{T}_E, \mathsf{T}_{In}, \mathsf{T}_C, \mathsf{T}_{M1}$ $\mathsf{T}_{E_\theta}, \mathsf{T}_{In_\theta}, \mathsf{T}_{C_\theta}, \mathsf{T}_{M1_\theta}$	Précision	Rappel	Pts. mob.
A	*rep.*	2.0	34.14	35.89	73
B	*rep.*	3.0	45.76	50.94	116
C	*rep.*	4.0	**47.88**	**52.30**	138
D	*rep.*	5.0	47.19	51.85	**180**
E	*ang.*	0.05	14.28	32.0	111
F	*ang.*	0.10	28.04	48.93	173
G	*ang.*	0.15	32.05	36.76	171
H	*ang.*	0.20	38.66	31.86	158
I	*ang.*	0.25	46.05	35.0	165
J	*ang.*	0.30	36.98	22.68	167

TABLE 3.11 – Influence des seuils géométriques sur la séquence IP 10.

Influence de l'extraction et de l'appariement de points d'intérêt. L'influence de l'étape d'extraction et d'appariement de points d'intérêt sur la séquence 10 du jeu de données IP a été évaluée en trois temps. Dans un premier temps, différents couples de détecteurs et descripteurs ont été évalués, utilisant l'erreur de reprojection ou l'erreur angulaire, avec filtrage de l'ensemble S_1. Dans un second temps, les tests ont été réalisés sans filtrage afin d'évaluer l'influence de cette étape sur les performances, alors que l'influence de l'extraction de l'ensemble S_2, c'est à dire le suivi des points précédemment triangulés, a été évaluée dans un dernier temps. Les meilleures performances obtenues en termes de précision lors de la première étape, dont les résultats sont présentés dans le tableau 3.12 et la figure 3.20, sont établies à 71.95% avec le couple de détecteurs et descripteurs FAST et DAISY (paramètres D du tableau 3.12), alors que le meilleur rappel est obtenu avec SIFT à 51.85%, par calcul de l'erreur de reprojection (paramètres A du tableau 3.11). Sans filtrage de l'ensemble S_1, les performances obtenues sont largement inférieures, avec la meilleure précision atteinte par le détecteur et descripteur ORB à 52.70% et le meilleur rappel obtenu avec le détecteur et descripteur AKAZE à 32.76%, ce qui justifie du bien fondé de cette étape en ce qui concerne le jeu de données IP. Ces résultats sont présentés dans le tableau 3.13 et la figure 3.21. Enfin, le suivi des points précédemment triangulés, c'est à dire l'extraction de l'ensemble S_2, n'améliore globalement pas les performances, comme montré dans le tableau 3.14 et la figure 3.22, à l'exception du rappel lorsque couplé à SIFT sans filtrage de S_1, où l'étape apporte 1.78%, et lorsque couplé à ORB avec et sans filtrage de S_1, où les gains s'établissent à respectivement 20.11% et 6.08%.

ID	Err.	Dét. - Desc.	S_1	S_2	Précision	Rappel	Pts. mob.
A	*rep.*	SIFT	×	-	47.19	**51.85**	180
B	*rep.*	ORB	×	-	59.25	23.88	58
C	*rep.*	AKAZE	×	-	48.03	38.88	181
D	*rep.*	FAST - DAISY	×	-	**71.95**	40.68	167
E	*rep.*	SIFT - DAISY	×	-	50.72	37.63	128
F	*ang.*	SIFT	×	-	46.05	35.0	165
G	*ang.*	ORB	×	-	61.76	11.29	73
H	*ang.*	AKAZE	×	-	36.60	18.55	229
I	*ang.*	FAST - DAISY	×	-	52.25	23.48	**242**
J	*ang.*	SIFT - DAISY	×	-	32.55	18.91	181

TABLE 3.12 – Influence de l'extraction et de l'appariement sur la séquence IP 10 - 1.

3.3.3.2 Séquence 2

Présentation. La séquence 2 du jeu de données IP présente un véhicule mobile VIPALAB suivi d'abord longitudinalement puis latéralement lors d'un virage à droite. Les

ID	Err.	Dét. - Desc.	S_1	S_2	Précision	Rappel	Pts. mob.
A	*rep.*	SIFT	-	-	32.66	19.14	309
B	*rep.*	ORB	-	-	**52.70**	24.07	279
C	*rep.*	AKAZE	-	-	37.19	**32.76**	431
D	*rep.*	FAST - DAISY	-	-	40.07	10.24	**817**
E	*rep.*	SIFT - DAISY	-	-	43.93	18.12	275
F	*ang.*	SIFT	-	-	19.09	10.85	435

TABLE 3.13 – Influence de l'extraction et de l'appariement sur la séquence IP 10 - 2.

ID	Err.	Dét. - Desc.	S_1	S_2	Précision	Rappel	Pts. mob.
A	*rep.*	SIFT	-	×	5.52	20.93	364
B	*rep.*	SIFT	×	×	4.51	33.33	327
C	*rep.*	ORB	-	×	**20.0**	30.15	**499**
D	*rep.*	ORB	×	×	14.86	**44.0**	184
E	*ang.*	SIFT	-	×	3.0	6.25	379
F	*ang.*	SIFT	×	×	2.63	12.50	322

TABLE 3.14 – Influence de l'extraction et de l'appariement sur la séquence IP 10 - 3.

images 0 à 600 ont été évaluées, avec une apparition du véhicule mobile en début de séquence, alors que sa labellisation, comprenant également son ombre pour une majeure partie de la séquence, a été effectuée manuellement.

Performances quantitatives. Les paramètres retenus pour l'évaluation des performances sur la séquence 2 du jeu de données IP sont respectivement $\mathsf{T}_L = 400$ pixels, $\mathsf{T}_{T1_{min}} = 0.1$ mètre, $\mathsf{T}_{T1_{max}} = 5.0$ mètres, $\mathsf{T}_{T2} = 1.0$ mètre, $\mathsf{T}_{T3} = 110.0°$, $\mathsf{T}_E = \mathsf{T}_{In} = \mathsf{T}_C = \mathsf{T}_{M1} = 5.0$ pixels, $\mathsf{T}_{E_\theta} = \mathsf{T}_{In_\theta} = \mathsf{T}_{C_\theta} = \mathsf{T}_{M1_\theta} = 0.25°$, le filtrage de l'ensemble S_1 ainsi que la non extraction de l'ensemble S_2. Plusieurs couples de détecteurs et descripteurs ont été évalués, avec les meilleures performances en précision obtenues par calcul de l'erreur de reprojection et l'utilisation du détecteur et descripteur AKAZE (paramètres B du tableau 3.15), à 80.15%, quand les meilleures performances en rappel sont obtenues avec le détecteur et descripteur SIFT (paramètres A du tableau 3.15), à 11.78%, valeur supérieure de seulement 0.46% au résultat obtenu avec le détecteur et descripteur AKAZE. Ces résultats sont présentés dans le tableau 3.15 et la figure 3.23.

3.3.3.3 Séquence 16

Présentation. La séquence 16 du jeu de données IP présente deux véhicules mobiles VIPALAB effectuant deux mouvements latéraux en sens inverse à une intersection. Les images 542 à 730 ont été évaluées, avec une apparition des véhicules mobiles en début de

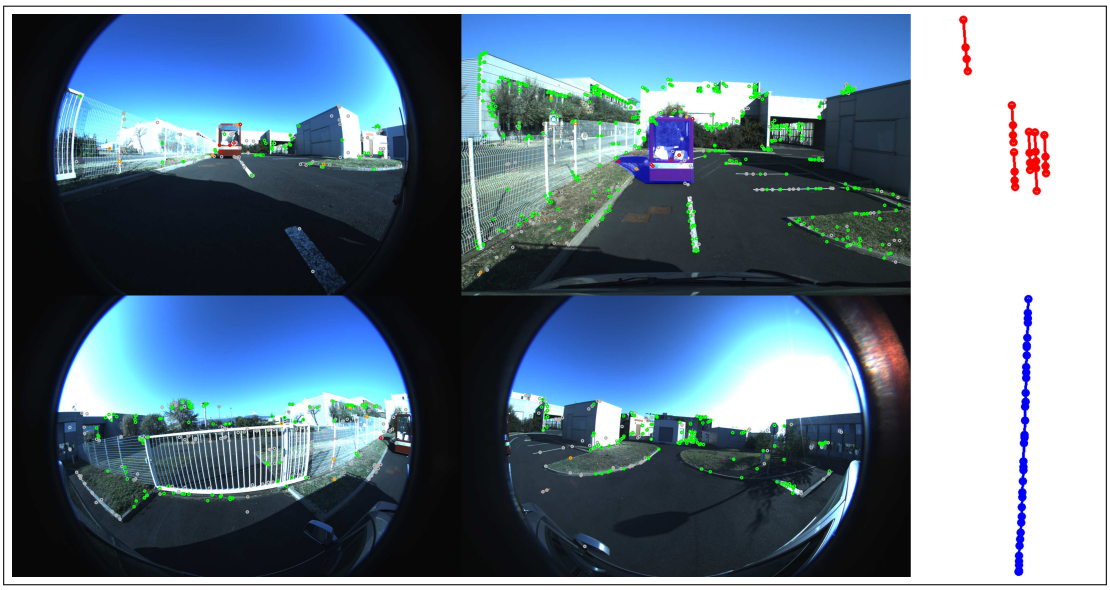

FIGURE 3.17 – Vues et trajectoires extraites de la séquence IP 2 avec les paramètres B du tableau 3.15. On peut remarquer sur la partie supérieure gauche du véhicule mobile plusieurs points classés statiques, dans la vue en haut à droite. Ce sont des appariements temporels, puisque ces points n'apparaissent seulement que dans cette vue.

ID	Err.	Dét. - Desc.	Précision	Rappel	Pts. mob.
A	*rep.*	SIFT	77.31	**11.78**	334
B	*rep.*	AKAZE	**80.15**	11.32	**467**
C	*rep.*	FAST - DAISY	60.55	8.26	372
D	*ang.*	SIFT	59.17	6.17	360
E	*ang.*	AKAZE	57.56	4.96	420

TABLE 3.15 – Tests effectués sur la séquence IP 2.

séquence, alors que leur labellisation, comprenant également les ombres des véhicules, a été effectuée manuellement.

Performances quantitatives. Les paramètres initiaux retenus pour l'évaluation de la séquence 16 du jeu de données IP sont identiques à ceux retenus pour la séquence 2, détaillés dans le paragraphe correspondant. Plusieurs couples de détecteurs et descripteurs ont été évalués sur cette séquence, de même que plusieurs valeurs pour les différents seuils géométriques. Les meilleures performances en termes de précision et de rappel sont obtenues grâce au détecteur et descripteur AKAZE, par calcul de l'erreur de reprojection et les seuils $\mathsf{T}_E, \mathsf{T}_{In}, \mathsf{T}_C$ et T_{M1} fixés à 2 pixels, à 90.90% et 47.6%, respectivement, malgré un nombre de points mobiles reconstruits relativement faible (paramètres C du tableau 3.16). On peut noter que les seuils géométriques ont sur cette séquence une incidence

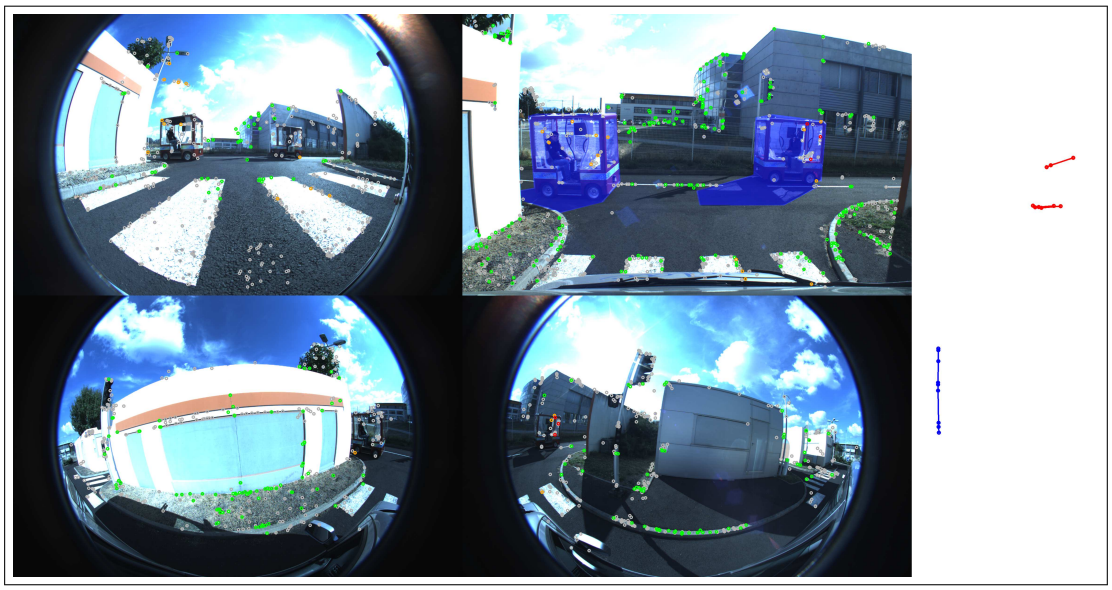

FIGURE 3.18 – Vues et trajectoires extraites de la séquence IP 16 avec les paramètres C du tableau 3.16.

importante, avec des écarts élevés entre plusieurs niveaux de seuillage pour la plupart des couples de détecteurs et descripteurs, par calcul de l'erreur de reprojection ou de l'erreur angulaire. En outre, alors que sur les autres séquences de ce jeu de données, des seuils géométriques plutôt laxes permettent l'obtention des meilleures performances, sur cette séquence les meilleures performances sont obtenues grâce à des seuils plus stricts. Le tableau 3.16 et la figure 3.24 présentent les résultats obtenus sur cette séquence.

ID	Err.	Dét. - Desc.	$\mathsf{T}_E, \mathsf{T}_{In}, \mathsf{T}_C, \mathsf{T}_{M1}$ $\mathsf{T}_{E_\theta}, \mathsf{T}_{In_\theta}, \mathsf{T}_{C_\theta}, \mathsf{T}_{M1_\theta}$	Précision	Rappel	Pts. mob.
A	*rep.*	SIFT	2.0	60.0	10.0	13
B	*rep.*	SIFT	5.0	52.38	23.40	43
C	*rep.*	AKAZE	2.0	**90.90**	**47.61**	25
D	*rep.*	AKAZE	5.0	45.45	25.0	30
E	*rep.*	FAST - DAISY	2.0	53.33	23.52	42
F	*rep.*	FAST - DAISY	5.0	65.21	23.80	53
G	*ang.*	SIFT	0.1	52.94	26.47	42
H	*ang.*	SIFT	0.25	50.0	27.27	**71**
I	*ang.*	AKAZE	0.1	12.50	18.75	58
J	*ang.*	AKAZE	0.25	52.38	25.58	50

TABLE 3.16 – Tests effectués sur la séquence IP 16.

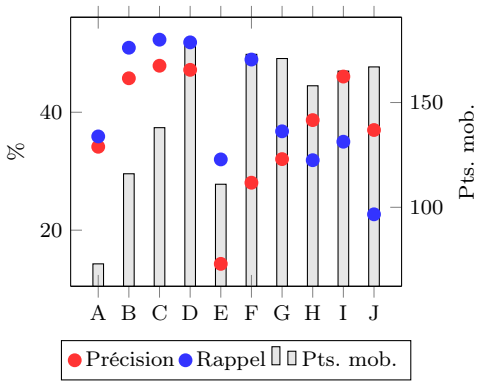

FIGURE 3.19 – Influence des seuils géométriques sur la séquence IP 10.

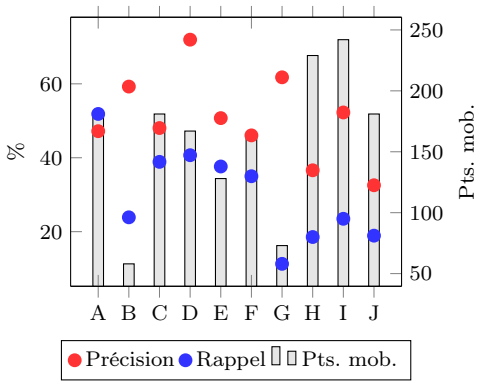

FIGURE 3.20 – Influence de l'extraction et de l'appariement sur la séquence IP 10 - 1.

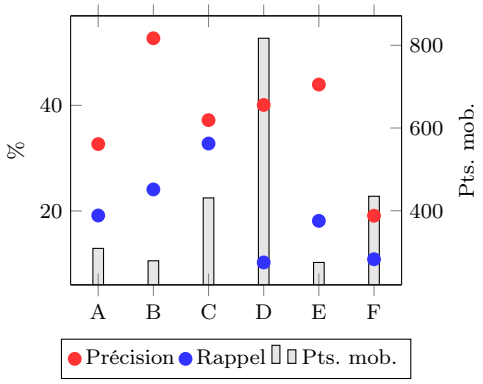

FIGURE 3.21 – Influence de l'extraction et de l'appariement sur la séquence IP 10 - 2.

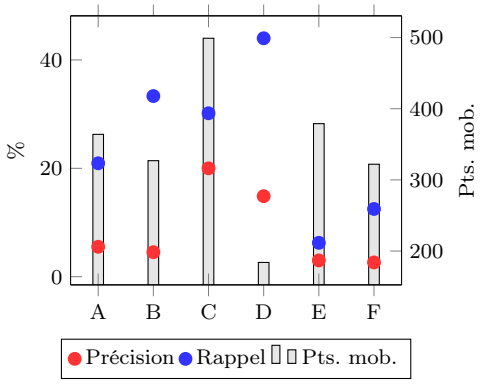

FIGURE 3.22 – Influence de l'extraction et de l'appariement sur la séquence IP 10 - 3.

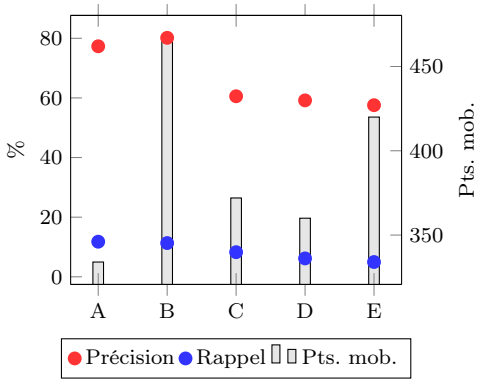

FIGURE 3.23 – Tests effectués sur la séquence IP 2.

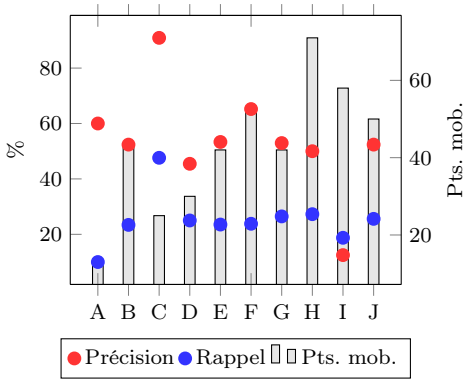

FIGURE 3.24 – Tests effectués sur la séquence IP 16.

3.3.3.4 Analyse qualitative des performances sur le jeu de données IP

Malgré des performances quantitatives relativement inférieures en comparaison à celles obtenues sur le jeu de données KITTI, les figures 3.16, 3.17 et 3.18 permettent de

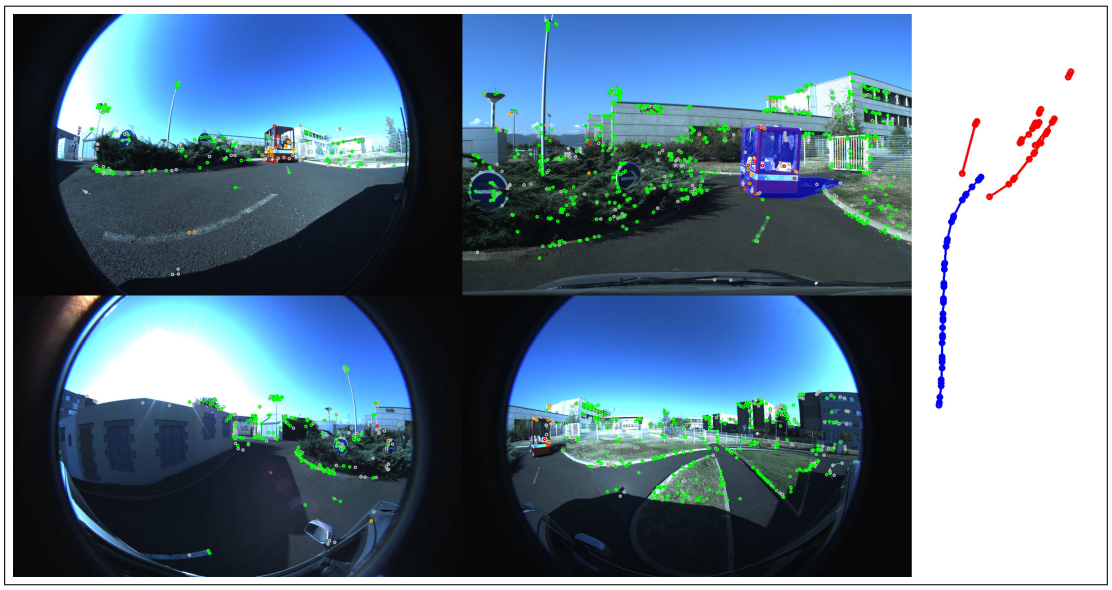

FIGURE 3.25 – Exemple de faux positifs : point classé mobile à gauche du véhicule, visible dans les vues en haut à droite et en bas à gauche, ainsi que sa trajectoire à gauche de celle du véhicule d'acquisition, extrait de la séquence IP 10 avec les paramètres C du tableau 3.12.

visualiser les performances qualitatives obtenues sur le jeu de données IP, plus difficile, et montrent bien une classification relativement juste de la plupart des points reconstruits. Une première observation au regard des performances en précision, notamment sur la séquence 10, avec une valeur maximale obtenue de 71.95%, indique la reconstruction de faux points mobiles. Ce problème est illustré dans la figure 3.25, où l'on peut apercevoir un faux positif reconstruit à gauche du véhicule, sur l'un des bâtiments, par la caméra pare-brise et la caméra gauche, qui correspondent respectivement aux vues en haut à droite et en bas à gauche de la figure. Dans cet exemple, le point reconstruit correspond à un mauvais appariement, c'est à dire que les points 2D sur les images ne correspondent pas au même point 3D de la scène, ce qui engendre dans ce cas précis sa classification erronée. La trajectoire de ce faux positif est en outre visible à gauche de la trajectoire du véhicule d'acquisition au bas de la figure, alors que d'autres mauvais appariements sur le véhicule mobile, même si placés sur le masque dans la vue en haut à droite, sont également visibles. Cette trajectoire de trois points, à gauche du véhicule, est de forme particulière, car composée de deux points très rapprochés ainsi que d'un troisième assez éloigné des deux autres. Cette forme particulière permet de déduire que le mauvais appariement a provoqué le changement de classe d'un point statique (les deux points rapprochés) vers la classe des points mobiles en invalidant sa contrainte de consistance C_C. D'autre part, ces mauvais appariements peuvent également mettre en défaut la contrainte de consistance stéréo utilisée par la contrainte de mobilité C_{M1}, ce qui produit alors le cas inverse, c'est à

dire la classification outlier d'un point auparavant mobile, et donc la perte de son suivi. À noter qu'un seul mauvais appariement suffit pour invalider ces deux contraintes, rendant d'autant plus ardue la tâche du suivi et de la détection de points mobiles sur de longs intervalles de temps. Une autre observation concerne les résultats obtenus par calcul de l'erreur angulaire, quasi-systématiquement inférieurs à ceux obtenus par calcul de l'erreur de reprojection. Là encore, ceci peut s'expliquer par la plus grande probabilité offerte par la contrainte épipolaire d'appariement appliquée aux caméras fisheye C_{E_θ} d'appareiller des points situés sur le bord du champ visible, ce qui augmente d'autant la probabilité de mauvais appariements. De manière plus générale, ces remarques, auxquelles s'ajoute le fait que la méthode n'est ni robuste aux occultations ni à la perte de détection des points auparavant triangulés lors du processus d'extraction, permettent d'expliquer le faible nombre et la faible durée de suivi des points mobiles reconstruits, qui n'excède bien souvent pas plus de trois à quatre temporalités successives sur ce jeu de données. Enfin, trois raisons peuvent principalement expliquer les mauvaises performances obtenues en termes de rappel sur ce jeu de données, notamment sur la séquence 2. La première, similaire à l'une de celles déjà énoncées sur le jeu de données KITTI et illustrée figure 3.26, provient du trop faible mouvement du véhicule mobile à de multiples occasions dans cette séquence, ce qui engendre la classification statique des points reconstruits sur le véhicule qui satisfont alors la contrainte de consistance C_C. La seconde, visible sur la figure 3.17 notamment, concerne les points seulement observés temporellement et placés sur un plan épipolaire. Dans cet exemple, les points statiques détectés sur le coin supérieur gauche du véhicule mobile sont effectivement consistants. Enfin, la troisième est plus évidente et concerne la classification statique de points reconstruits sur les zones vitrées des véhicules mobiles, qui correspondent effectivement à des points fixes de la scène observés à travers le véhicule, comme illustré dans la vue en haut à droite de la figure 3.16. Ce type d'erreur, au même titre que le problème observé sur le jeu de données KITTI concernant la détection de points mobiles sur les ombres non labellisées de véhicules mobiles, relève d'un problème sémantique et non géométrique, malgré un impact certain sur les performances quantitatives.

3.3.3.5 Conclusion des tests effectués sur jeu de données IP

L'analyse quantitative et qualitative des performances obtenues sur ce jeu de données montre que la méthode proposée produit des résultats relativement bons en termes de précision, avec des valeurs maximales sur toutes les séquences comprises entre 71.95% et 90.90%, alors que les valeurs maximales pour le rappel sont largement plus en retrait et comprises entre 11.78% et 52.30%. La sélection d'un ensemble de paramètres de référence pour ce jeu de données est plus difficile que sur le jeu de données KITTI, car plusieurs des

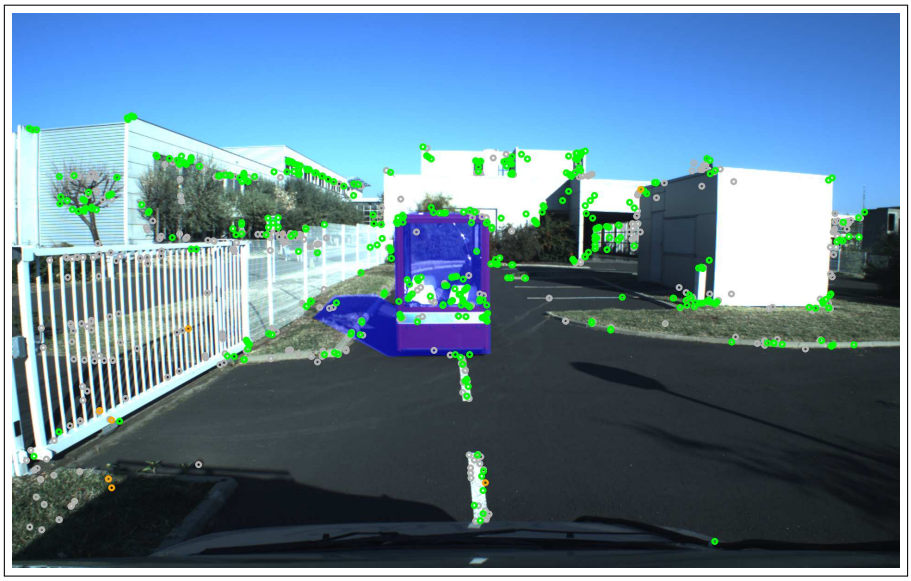

FIGURE 3.26 – Exemple de faux négatifs : points classés statiques sur un véhicule peu mobile, extrait de la séquence IP 2.

meilleurs résultats obtenus l'ont été grâce à des combinaisons différentes de paramètres. Malgré cela, si l'on excepte les résultats de la séquence 10 au regard de la précision, un ensemble de paramètres semble tout de même produire des résultats relativement convaincants. Le tableau 3.17 récapitule ces paramètres de référence.

Extraction et appariement		Erreur	Seuils géométriques		Trajectoire	
Dét. - Desc.	AKAZE	Reprojection	T_L	400 px	$\mathsf{T}_{T1_{min}}$	0.1 m
Filtrage S_1	Oui		T_E	5.0 px	$\mathsf{T}_{T1_{max}}$	5.0 m
Extraction S_2	Non		T_{In}	5.0 px	T_{T2}	1.0 m
			T_C	5.0 px	T_{T3}	110°
			T_{M1}	5.0 px		

TABLE 3.17 – Paramètres de référence pour le jeu de données IP.

3.3.4 Limitations de la méthode et de la méthodologie d'évaluation

Comme abordé dans la présentation des performances quantitatives et qualitatives, plusieurs limitations propres à la méthode proposée sont à relever, de même que certains biais engendrés par la méthodologie d'évaluation. En ce qui concerne les limitations de la méthode, on peut tout d'abord noter son paramétrage très sensible, pouvant faire varier de manière conséquente les résultats obtenus. Qu'il s'agisse des types de détecteurs et descripteurs utilisés, de la méthode de calcul de l'erreur associée aux points 3D, des seuils géométriques ou des seuils de trajectoire, tous ces paramètres ont une influence impor-

tante sur les performances, alors que leur nombre complique la recherche du meilleur compromis entre performances et nombre de points reconstruits. Une autre limitation concerne la dépendance très forte de la méthode aux résultats du processus d'extraction et d'appariement de points d'intérêt. Quand les mauvais appariements dans la plupart des méthodes de SLAM visuel classiques sont simplement rejetés, cela engendre deux conséquences majeures sur les résultats de la méthode proposée, qui sont, d'une part, l'apparition de faux positifs et, d'autre part, la perte de suivi des points mobiles. Enfin, la méthodologie d'évaluation comporte deux biais provenant du processus de labellisation. Un premier biais provient du fait que la labellisation automatique avec la méthode par apprentissage profond MASK R-CNN ne prend pas en considération les ombres mobiles des véhicules, ce qui peut engendrer une baisse de précision, alors qu'un second biais résulte du fait que la labellisation seulement partielle des vues, notamment sur le jeu de données IP, ne traduit pas complètement les performances de la méthode proposée.

Conclusion et perspectives

Les résultats présentés dans le chapitre précédent montrent que la méthode proposée répond bien à l'objectif initial défini en introduction de ce manuscrit, c'est à dire la reconstruction éparse d'une scène dynamique à l'aide d'un système multi-caméras hétérogène en stéréo *wide-baseline*. Deux jeux de données ont permis cette évaluation, avec dans un premier cas des performances quantitatives et qualitatives relativement bonnes sur le jeu de données KITTI, dont la configuration expérimentale en stéréo classique profite avantageusement au processus d'extraction et d'appariement de points d'intérêt, et dans un autre cas des performances plus modestes mais néanmoins prometteuses sur le jeu de données IP, développé durant ces travaux, plus difficile car composé de plusieurs caméras aux focales et points de vue très différents.

Principales contributions

Le module de détection et reconstruction incrémentale de points mobiles présenté dans la sous-section 2.4 de ce manuscrit constitue la contribution principale de ces travaux de thèse, en raison d'une conception permettant son fonctionnement sur plusieurs types de systèmes multi-caméras à large champs recouvrants, notamment dans le cas particulier et relativement difficile de systèmes hétérogènes en stéréo *wide-baseline*. Au sein de ce module, deux mécanismes permettent cette flexibilité d'utilisation. Le premier concerne le processus d'extraction et d'appariement de points d'intérêt, permettant le regroupement de toutes les observations temporelles et stéréo associées à chaque point 3D observé par le système multi-caméras de manière récursive. Le deuxième mécanisme se rapporte à la méthode de classification des points reconstruits, reposant d'une part sur l'évaluation de l'ensemble de leurs observations grâce à plusieurs contraintes géométriques visant à détecter leur mouvement, et d'autre part sur l'évaluation de leur trajectoire afin de filtrer ceux présentant un mouvement erratique ou dégénéré. En outre, l'autre contribution de ces travaux concerne la création d'un jeu de données représentatif de tels systèmes multi-caméras hétérogènes, répondant notamment aux contraintes industrielles initiales présentées en introduction de ce manuscrit et permettant d'envisager

l'utilisation éventuelle de la méthode proposée en conditions réelles. Enfin, les travaux présentés dans ce manuscrit ont donné lieu à une première publication [109], alors qu'une version plus complète est à soummettre [110].

Perspectives

L'analyse des performances quantitatives et qualitatives obtenues dans le chapitre 3 a permis la mise en évidence de plusieurs limitations relatives à la méthode proposée, en particulier son paramétrage complexe ou encore sa sensibilité aux mauvais appariements de points d'intérêt. Ces limitations peuvent faire l'objet de travaux d'amélioration, détaillés ci-après, alors que dans un second temps seront abordées les extensions possibles de la méthode en matière de fonctionnalités.

Perspectives d'amélioration. L'une des possibles améliorations de la méthode vise à faciliter son paramétrage. Afin d'y parvenir, une piste de réflexion consisterait en l'optimisation de chaque paramètre, par apprentissage ou utilisation de méthodes non-linéaires, sur plusieurs séquences d'images labellisées d'un jeu de données particulier, les paramètres obtenus constituant alors un point de référence relativement stable et adapté aux caractéristiques et conditions d'utilisation du système d'acquisition considéré. En outre, une optimisation de ce type, écartant les erreurs de paramétrage, permettrait certainement d'avoir une appréciation plus sensible des qualités et faiblesses de la méthode, et pourrait ainsi mettre en évidence d'autres perspectives éventuelles d'approfondissement. L'un des autres points d'amélioration concerne la gestion des mauvais appariements. L'implémentation d'une dimension probabiliste lors du processus de classification, en intégrant les nouvelles observations d'un point 3D à un modèle bayésien par exemple, permettrait éventuellement d'établir un score exprimant le potentiel de mouvement associé à chaque point reconstruit, évitant ainsi qu'un unique mauvais appariement ne puisse invalider les contraintes de consistance temporelle C_C et de consistance stéréo C_{M1} qui engendrent actuellement la perte de suivi ou la non détection de certains points mobiles. Dans le même ordre d'idée, l'utilisation de M-estimateurs, prenant en paramètre tous les points non consistants de la scène afin de déterminer la géométrie épipolaire propre à chaque objet rigide, pourrait également permettre le filtrage de ces fausses observations, avec cependant la possibilité d'un nombre d'appariements trop faible lors du calcul de la matrice essentielle pour une mise en place efficace dans le cas de systèmes hétérogènes en *wide-baseline*. À noter que ce dernier point entre par ailleurs dans le cadre des extensions possibles de la méthode à la reconstruction d'objets rigides. Une autre perspective, au regard du faible nombre de points mobiles actuellement reconstruits, consisterait à effectuer une étape de détection et d'appariement supplémentaire de points d'intérêt lo-

calisée au sein des zones de l'image présentant une concentration de points mobiles déjà significative, afin d'augmenter le nombre de points détectés se trouvant effectivement sur les objets mobiles sous-jacents. Enfin, le concept de détection localisée peut également s'envisager par utilisation conjointe de la méthode proposée à d'autres méthodes de classification, reposant sur des techniques d'apprentissage profond par exemple, afin de cibler les zones potentielles de mouvement dans les images avant même de lancer tout processus de détection et d'appariement de points.

Perspectives d'extension. Comme évoqué dans le paragraphe précédent, une extension de la méthode proposée à la reconstruction d'objets mobiles est envisageable grâce à l'utilisation de M-estimateurs. Cette fonctionnalité supplémentaire permettrait de surcroît d'envisager le suivi et la reconstruction monoculaire à l'échelle de chaque objet mobile par recalage de la reprojection des points 3D de son modèle sur les points d'intérêt détectés dans les images. Outre une certaine robustesse aux occultations, cela permettrait alors l'utilisation de tout le champ de vue offert par les systèmes multi-caméras de type AVM actuellement proposés sur certains véhicules de grande série pour le suivi d'objets mobiles.

Ouverture

Le dernier point abordé dans le précédent paragraphe, relevant de l'utilisation de systèmes multi-caméras dans le contexte des aides à la conduite et des véhicules autonomes, nous permet de nous interroger aujourd'hui sur la place des méthodes de SLAM visuel et de leur utilisation au sein de l'industrie automobile. Bien que de nombreuses expérimentations aient bénéficié ou bénéficient actuellement de ces méthodes dans une approche uni-modale, il est certainement peu probable que les techniques classiques de SLAM visuel, parmi lesquelles s'inscrivent ces travaux, exhibent à l'avenir la robustesse nécessaire à leur utilisation exclusive au sein d'applications critiques. En revanche, il est tout à fait envisageable que ce type de systèmes puisse s'intégrer dans une optique de redondance d'informations, toujours dans un souci de robustesse, en complément d'autres types de capteurs tels que des télémètres laser ou encore des RADARs. En outre, l'émergence de nouveaux outils informatiques, tels que l'apprentissage profond et notamment les réseaux de neurones convolutifs, qui bénéficient maintenant de matériel hautement spécialisé, pourrait de nouveau changer le paysage des perspectives d'utilisation qu'offrent les caméras numériques dans ce contexte, qu'il s'agisse de SLAM visuel ou de classification sémantique. Enfin, l'industrie automobile est actuellement en pleine mutation autour de la problématique du transport autonome, impliquant sans cesse de nouveaux acteurs issus de secteurs extrêmement divers et contribuant chacun à l'évolu-

tion des technologies et de leurs applications. La place de la recherche, dans ce contexte, est plus que jamais au centre des besoins de telles entreprises.

Bibliographie

[1] Pablo F Alcantarilla and T Solutions. Fast explicit diffusion for accelerated features in nonlinear scale spaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(7) :1281–1298, 2011.

[2] Pablo Fernández Alcantarilla, Adrien Bartoli, and Andrew J Davison. Kaze features. In *European Conference on Computer Vision*, pages 214–227. Springer, 2012.

[3] John Aloimonos. Shape from texture. *Biological Cybernetics*, 58(5) :345–360, 1988.

[4] Luis Alvarez, Rachid Deriche, Javier Sanchez, and Joachim Weickert. Dense disparity map estimation respecting image discontinuities : A pde and scale-space based approach. *Journal of Visual Communication and Image Representation*, 13(1-2) :3–21, 2002.

[5] Markus-Christian Amann, Thierry Bosch, Marc Lescure, Risto Myllyla, and Marc Rioux. Laser ranging : a critical review of usual techniques for distance measurement. *Optical Engineering*, 40(1) :10–19, 2001.

[6] Tim Bailey and Hugh Durrant-Whyte. Simultaneous localization and mapping : Part ii. *IEEE Robotics & Automation Magazine*, 13(3) :108–117, 2006.

[7] Simon Baker, Daniel Scharstein, JP Lewis, Stefan Roth, Michael J Black, and Richard Szeliski. A database and evaluation methodology for optical flow. *International Journal of Computer Vision*, 92(1) :1–31, 2011.

[8] Adrien Bartoli, Yan Gérard, François Chadebecq, Toby Collins, and Daniel Pizarro. Shape-from-template. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(10) :2099–2118, 2015.

[9] Tali Basha, Yael Moses, and Nahum Kiryati. Multi-view scene flow estimation : A view centered variational approach. *International Journal of Computer Vision*, 101(1) :6–21, 2013.

[10] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. SURF : Speeded up robust features. In *European Conference on Computer Vision*, pages 404–417. Springer, 2006.

[11] Rami Ben-Ari and Nir Sochen. Variational stereo vision with sharp discontinuities and occlusion handling. In *International Conference on Computer Vision*, pages 1–7. IEEE, 2007.

[12] Paul J Besl, Neil D McKay, et al. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2) :239–256, 1992.

[13] Michael Bleyer, Christoph Rhemann, and Carsten Rother. Extracting 3d scene-consistent object proposals and depth from stereo images. In *European Conference on Computer Vision*, pages 467–481. Springer, 2012.

[14] Michael Bleyer, Carsten Rother, Pushmeet Kohli, Daniel Scharstein, and Sudipta Sinha. Object stereo—joint stereo matching and object segmentation. In *Computer Vision and Pattern Recognition*, pages 3081–3088. IEEE, 2011.

[15] Jean-Yves Bouguet. Camera calibration tool-box for matlab. *California Institute of Technology Report*, 2002.

[16] Terrance E Boult and L Gottesfeld Brown. Factorization-based segmentation of motions. In *IEEE Workshop on Visual Motion*, pages 179–186. IEEE, 1991.

[17] Paul S Bradley and Olvi L Mangasarian. K-plane clustering. *Journal of Global Optimization*, 16(1) :23–32, 2000.

[18] Gary Bradski and Adrian Kaehler. *Learning OpenCV : Computer vision with the OpenCV library*. "O'Reilly Media, Inc.", 2008.

[19] Michael Brady and Alan Yuille. An extremum principle for shape from contour. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(3) :288–301, 1984.

[20] Thomas Brox, Andrés Bruhn, Nils Papenberg, and Joachim Weickert. High accuracy optical flow estimation based on a theory for warping. In *European Conference on Computer Vision*, pages 25–36. Springer, 2004.

[21] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. Brief : Binary robust independent elementary features. *European Conference on Computer Vision*, pages 778–792, 2010.

[22] Gerardo Carrera, Adrien Angeli, and Andrew J Davison. Slam-based automatic extrinsic calibration of a multi-camera rig. In *International Conference on Robotics and Automation*, pages 2652–2659. IEEE, 2011.

[23] Guangliang Chen and Gilad Lerman. Spectral curvature clustering (SCC). *International Journal of Computer Vision*, 81(3) :317–330, 2009.

[24] Yang Cheng, Mark Maimone, and Larry Matthies. Visual odometry on the mars exploration rovers. In *Systems, Man and Cybernetics*, volume 1, pages 903–910. IEEE, 2005.

[25] Peter Corke, Dennis Strelow, and Sanjiv Singh. Omnidirectional visual odometry for a planetary rover. In *Intelligent Robots and Systems*, volume 4, pages 4007–4012. IEEE, 2004.

[26] João Paulo Costeira and Takeo Kanade. A multibody factorization method for independently moving objects. *International Journal of Computer Vision*, 29(3) :159–179, 1998.

[27] Jonathan Courbon. *Navigation de robots mobiles par mémoire sensorielle.* PhD thesis, Université Blaise Pascal - Clermont-Ferrand II, 2009.

[28] Andrew J Davison, Ian D Reid, Nicholas D Molton, and Olivier Stasse. MonoSLAM : Real-time single camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6) :1052–1067, 2007.

[29] Richard O Duda, Peter E Hart, and David G Stork. *Pattern classification.* John Wiley & Sons, 2012.

[30] Jean-Denis Durou, Maurizio Falcone, and Manuela Sagona. Numerical methods for shape-from-shading : A new survey with benchmarks. *Computer Vision and Image Understanding*, 109(1) :22–43, 2008.

[31] Hugh Durrant-Whyte and Tim Bailey. Simultaneous localization and mapping : Part i. *IEEE Robotics & Automation Magazine*, 13(2) :99–110, 2006.

[32] Ehsan Elhamifar and René Vidal. Sparse subspace clustering. In *Computer Vision and Pattern Recognition*, pages 2790–2797. IEEE, 2009.

[33] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(3) :611–625, 2018.

[34] Jakob Engel, Thomas Schöps, and Daniel Cremers. Lsd-slam : Large-scale direct monocular slam. In *European Conference on Computer Vision*, pages 834–849. Springer, 2014.

[35] Jakob Engel, Jörg Stückler, and Daniel Cremers. Large-scale direct slam with stereo cameras. In *Intelligent Robots and Systems*, pages 1935–1942. IEEE, 2015.

[36] Jakob Engel, Jürgen Sturm, and Daniel Cremers. Semi-dense visual odometry for a monocular camera. In *International Conference on Computer Vision*, pages 1449–1456. IEEE, 2013.

[37] Sandro Esquivel, Felix Woelk, and Reinhard Koch. Calibration of a multi-camera rig from non-overlapping views. In *Joint Pattern Recognition Symposium*, pages 82–91. Springer, 2007.

[38] Carlos Hernandez Esteban, George Vogiatzis, and Roberto Cipolla. Multiview photometric stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(3) :548–554, 2008.

[39] Olivier Faugeras. *Three-dimensional computer vision : a geometric viewpoint*. MIT press, 1993.

[40] Olivier Faugeras, Quang-Tuan Luong, and Theo Papadopoulo. *The geometry of multiple images : the laws that govern the formation of multiple images of a scene and some of their applications*. MIT press, 2004.

[41] Olivier D Faugeras. What can be seen in three dimensions with an uncalibrated stereo rig ? In *European Conference on Computer Vision*, pages 563–578. Springer, 1992.

[42] Olivier D Faugeras, Quang-Tuan Luong, and Stephen J Maybank. Camera self-calibration : Theory and experiments. In *European Conference on Computer Vision*, pages 321–334. Springer, 1992.

[43] Paolo Favaro and Stefano Soatto. A geometric approach to shape from defocus. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(3) :406–417, 2005.

[44] Martin A Fischler and Robert C Bolles. Random sample consensus : A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6) :381–395, 1981.

[45] Andrew W Fitzgibbon and Andrew Zisserman. Automatic camera recovery for closed or open image sequences. In *European Conference on Computer Vision*, pages 311–326. Springer, 1998.

[46] Fernando Flores-Mangas and Allan D Jepson. Fast rigid motion segmentation via incrementally-complex local models. In *Computer Vision and Pattern Recognition*, pages 2259–2266. IEEE, 2013.

[47] Sergi Foix, Guillem Alenya, and Carme Torras. Lock-in time-of-flight (tof) cameras : A survey. *IEEE Sensors Journal*, 11(9) :1917–1926, 2011.

[48] Christian Forster, Matia Pizzoli, and Davide Scaramuzza. Svo : Fast semi-direct monocular visual odometry. In *International Conference on Robotics and Automation*, pages 15–22. IEEE, 2014.

[49] Wolfgang Förstner. A framework for low level feature extraction. In *European Conference on Computer Vision*, pages 383–394. Springer, 1994.

[50] Friedrich Fraundorfer and Davide Scaramuzza. Visual odometry - part ii : Matching, robustness, optimization, and applications. *IEEE Robotics & Automation Magazine*, 19(2) :78–90, 2012.

[51] Xiao-Shan Gao, Xiao-Rong Hou, Jianliang Tang, and Hang-Fei Cheng. Complete solution classification for the perspective-three-point problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(8) :930–943, 2003.

[52] Steffen Gauglitz, Tobias Höllerer, and Matthew Turk. Evaluation of interest point detectors and feature descriptors for visual tracking. *International Journal of Computer Vision*, 94(3) :335–360, 2011.

[53] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving ? the kitti vision benchmark suite. In *Computer Vision and Pattern Recognition*. IEEE, 2012.

[54] Andreas Geiger, Julius Ziegler, and Christoph Stiller. Stereoscan : Dense 3d reconstruction in real-time. In *Intelligent Vehicles Symposium*, pages 963–968. IEEE, 2011.

[55] Christopher Geyer and Kostas Daniilidis. A unifying theory for central panoramic systems and practical implications. In *European Conference on Computer Vision*, pages 445–461. Springer, 2000.

[56] Roland Goecke, Akshay Asthana, Niklas Pettersson, and Lars Petersson. Visual vehicle egomotion estimation using the fourier-mellin transform. In *Intelligent Vehicles Symposium*, pages 450–455. IEEE, 2007.

[57] Alvina Goh and René Vidal. Segmenting motions of different types by unsupervised manifold clustering. In *Computer Vision and Pattern Recognition*, pages 1–6. IEEE, 2007.

[58] Johann August Grunert. Das pothenotische problem in erweiterter gestalt nebst über seine anwendungen in der geodäsie. *Grunerts archiv für mathematik und physik*, 1(238-248) :1, 1841.

[59] Bert M Haralick, Chung-Nan Lee, Karsten Ottenberg, and Michael Nölle. Review and analysis of solutions of the three point perspective pose estimation problem. *International Journal of Computer Vision*, 13(3) :331–356, 1994.

[60] Chris Harris and Mike Stephens. A combined corner and edge detector. In *Alvey Vision Conference*, volume 15, page 50. BMVA, 1988.

[61] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision, Second Edition*. Cambridge Univ. Press, 2003.

[62] Richard I Hartley. Estimation of relative camera positions for uncalibrated cameras. In *European Conference on Computer Vision*, pages 579–587. Springer, 1992.

[63] Richard I Hartley. In defense of the eight-point algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(6) :580–593, 1997.

[64] Richard I Hartley, Rajiv Gupta, and Tom Chang. Stereo from uncalibrated cameras. In *Computer Vision and Pattern Recognition*, pages 761–764. IEEE, 1992.

[65] Richard I Hartley and Peter Sturm. Triangulation. *Computer Vision and Image Understanding*, 68(2) :146–157, 1997.

[66] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *International Conference on Computer Vision*, pages 2980–2988. IEEE, 2017.

[67] Martial Hebert and Eric Krotkov. 3d measurements from imaging laser radars : how good are they ? *Image and Vision Computing*, 10(3) :170–178, 1992.

[68] Berthold KP Horn and Brian G Schunck. Determining optical flow. *Artificial intelligence*, 17(1-3) :185–203, 1981.

[69] Andrew Howard. Real-time stereo visual odometry for autonomous ground vehicles. In *Intelligent Robots and Systems*, pages 3946–3952. IEEE, 2008.

[70] Frédéric Huguet and Frédéric Devernay. A variational method for scene flow estimation from stereo sequences. In *International Conference on Computer Vision*, pages 1–7. IEEE, 2007.

[71] Hossam Isack and Yuri Boykov. Energy-based geometric multi-model fitting. *International Journal of Computer Vision*, 97(2) :123–147, 2012.

[72] Pan Ji, Hongdong Li, Mathieu Salzmann, and Yuchao Dai. Robust motion segmentation with unknown correspondences. In *European Conference on Computer Vision*, pages 204–219. Springer, 2014.

[73] Zhiming Ji and Ming-Chuan Leu. Design of optical triangulation devices. *Optics & Laser Technology*, 21(5) :339–341, 1989.

[74] Rudolph Emil Kalman et al. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1) :35–45, 1960.

[75] Qifa Ke and Takeo Kanade. Transforming camera geometry to a virtual downward-looking camera : Robust ego-motion estimation and ground-layer detection. In *Computer Vision and Pattern Recognition*, volume 1, pages I–I. IEEE, 2003.

[76] Yeon-Ho Kim, Aleix M Martínez, and Avi C Kak. Robust motion estimation under varying illumination. *Image and Vision Computing*, 23(4) :365–375, 2005.

[77] Andreas Klaus, Mario Sormann, and Konrad Karner. Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure. In *International Conference on Pattern Recognition*, volume 3, pages 15–18. IEEE, 2006.

[78] Georg Klein and David Murray. Parallel tracking and mapping for small ar workspaces. In *International Symposium on Mixed and Augmented Reality*, pages 225–234. IEEE, 2007.

[79] Vladimir Kolmogorov and Ramin Zabih. Multi-camera scene reconstruction via graph cuts. In *European Conference on Computer Vision*, pages 82–96. Springer, 2002.

[80] Abhijit Kundu, K Madhava Krishna, and CV Jawahar. Realtime multibody visual slam with a smoothly moving monocular camera. In *International Conference on Computer Vision*, pages 2080–2087. IEEE, 2011.

[81] Abhijit Kundu, K Madhava Krishna, and Jayanthi Sivaswamy. Moving object detection by multi-view geometric techniques from a single camera mounted robot. In *Intelligent Robots and Systems*, pages 4306–4312. IEEE, 2009.

[82] Simon Lacroix, Anthony Mallet, Raja Chatila, and Laurent Gallo. Rover self localization in planetary-like environments. In *Artificial Intelligence, Robotics and Automation in Space*, volume 440, page 433, 1999.

[83] Bernhard Lamprecht, Stefan Rass, Simone Fuchs, and Kyandoghere Kyamakya. Extrinsic camera calibration for an on-board two-camera system without overlapping field of view. In *Intelligent Transportation Systems Conference*, pages 265–270. IEEE, 2007.

[84] Jean-Marc Lavest, Marc Viala, and Michel Dhome. Do we really need an accurate calibration pattern to achieve a reliable camera calibration ? In *European Conference on Computer Vision*, pages 158–174. Springer, 1998.

[85] Pierre Lébraly. *Etalonnage de caméras à champs disjoints et reconstruction 3D : Application à un robot mobile.* PhD thesis, Université Blaise Pascal - Clermont-Ferrand II, 2012.

[86] Pierre Lébraly, Eric Royer, Omar Ait-Aider, Clément Deymier, and Michel Dhome. Fast calibration of embedded non-overlapping cameras. In *International Conference on Robotics and Automation*, pages 221–227. IEEE, 2011.

[87] Gim Hee Lee, Friedrich Faundorfer, and Marc Pollefeys. Motion estimation for self-driving cars with a generalized camera. In *Computer Vision and Pattern Recognition*, pages 2746–2753. IEEE, 2013.

[88] Cheng Lei, Jason Selzer, and Yee-Hong Yang. Region-tree based stereo using dynamic programming optimization. In *Computer Vision and Pattern Recognition*, volume 2, pages 2378–2385. IEEE, 2006.

[89] Philip Lenz, Julius Ziegler, Andreas Geiger, and Martin Roser. Sparse scene flow segmentation for moving object detection in urban environments. In *Intelligent Vehicles Symposium*, pages 926–932. IEEE, 2011.

[90] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. Epnp : An accurate o (n) solution to the pnp problem. *International Journal of Computer Vision*, 81(2) :155–166, 2009.

[91] Maxime Lhuillier. Automatic structure and motion using a catadioptric camera. In *Proceedings of the 6th Workshop on Omnidirectional Vision, Camera Networks and Non-Classical Cameras*, 2005.

[92] Maxime Lhuillier. Effective and generic structure from motion using angular error. In *International Conference on Pattern Recognition*, pages 67–70. IEEE, 2006.

[93] Hongdong Li and Richard Hartley. Five-point motion estimation made easy. In *International Conference on Pattern Recognition*, pages 630–633. IEEE, 2006.

[94] Hongdong Li, Richard Hartley, and Jae-hak Kim. A linear approach to motion estimation using generalized camera models. In *Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.

[95] Ting Li, Vinutha Kallem, Dheeraj Singaraju, and René Vidal. Projective factorization of multiple rigid-body motions. In *Computer Vision and Pattern Recognition*, pages 1–6. IEEE, 2007.

[96] Bojian Liang and Nick Pears. Visual navigation using planar homographies. In *International Conference on Robotics and Automation*, volume 1, pages 205–210. IEEE, 2002.

[97] Tony Lindeberg. Scale-space theory : A basic tool for analyzing structures at different scales. *Journal of Applied Statistics*, 21(1-2) :225–270, 1994.

[98] Guangcan Liu, Zhouchen Lin, and Yong Yu. Robust subspace segmentation by low-rank representation. In *International Conference on Machine Learning*, pages 663–670, 2010.

[99] HC Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293(5828) :133–135, 1981.

[100] David G Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2) :91–110, 2004.

[101] Bruce D Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'81, pages 674–679, San Francisco, CA, USA, 1981. Morgan Kaufmann Publishers Inc.

[102] Quang-Tuan Luong. *Matrice Fondamentale et Autocalibration en Vision par Ordinateur*. PhD thesis, Université Paris Sud, 1992.

[103] Quang-Tuan Luong and Olivier Faugeras. Self-calibration of a stereo rig from unknown camera motions and point correspondences. Technical report, INRIA, 1993.

[104] Yi Ma, Harm Derksen, Wei Hong, and John Wright. Segmentation of multivariate mixed data via lossy data coding and compression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(9) :1546–1562, 2007.

[105] Mark Maimone, Yang Cheng, and Larry Matthies. Two years of visual odometry on the mars exploration rovers. *Journal of Field Robotics*, 24(3) :169–186, 2007.

[106] Donald W Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial & Applied Mathematics*, 11(2) :431–441, 1963.

[107] Worthy N Martin and Jagdishkumar Keshoram Aggarwal. Volumetric descriptions of objects from multiple views. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(2) :150–158, 1983.

[108] Larry Matthies and Stevena Shafer. Error modeling in stereo navigation. *Journal on Robotics and Automation*, 3(3) :239–248, 1987.

[109] Laurent Mennillo, Eric Royer, Frédéric Mondot, Johan Mousain, and Michel Dhome. Multibody reconstruction of the dynamic scene surrounding a vehicle using a wide baseline and multifocal stereo system. In *9th Workshop on Planning, Perception and Navigation for Intelligent Vehicles (satellite event of IROS'17)*. IEEE, 2017.

[110] Laurent Mennillo, Eric Royer, Frédéric Mondot, Johan Mousain, and Michel Dhome. Sparse multibody visual slam on wide-baseline and heterogeneous stereo systems. 2019.

[111] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *Computer Vision and Pattern Recognition*, pages 3061–3070. IEEE, 2015.

[112] Krystian Mikolajczyk and Cordelia Schmid. Indexing based on scale invariant interest points. In *International Conference on Computer Vision*, pages 525–531. IEEE, 2001.

[113] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10) :1615–1630, 2005.

[114] Annalisa Milella and Roland Siegwart. Stereo-based ego-motion estimation using pixel tracking and iterative closest point. In *Computer Vision Systems*, pages 21–21. IEEE, 2006.

[115] Yana Mileva, Andrés Bruhn, and Joachim Weickert. Illumination-robust variational optical flow with photometric invariants. In *Joint Pattern Recognition Symposium*, pages 152–162. Springer, 2007.

[116] Michael J Milford and Gordon F Wyeth. Mapping a suburb with a single camera using a biologically inspired slam system. *IEEE Transactions on Robotics*, 24(5) :1038–1053, 2008.

[117] Michael J Milford and Gordon F Wyeth. Single camera vision-only slam on a suburban road network. In *International Conference on Robotics and Automation*, pages 3684–3689. IEEE, 2008.

[118] Michael J Milford, Gordon F Wyeth, and David Prasser. Ratslam : a hippocampal model for simultaneous localization and mapping. In *International Conference on Robotics and Automation*, volume 1, pages 403–408. IEEE, 2004.

[119] Hans P Moravec. Obstacle avoidance and navigation in the real world by a seeing robot rover. Technical report, DTIC Document, 1980.

[120] Etienne Mouragnon, Maxime Lhuillier, Michel Dhome, Fabien Dekeyser, and Patrick Sayd. Real time localization and 3d reconstruction. In *Computer Vision and Pattern Recognition*, volume 1, pages 363–370. IEEE, 2006.

[121] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam : A versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31(5) :1147–1163, 2015.

[122] Raul Mur-Artal and Juan D Tardós. Orb-slam2 : An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, 33(5) :1255–1262, 2017.

[123] Rahul Kumar Namdev, K Madhava Krishna, and CV Jawahar. Multibody vslam with relative scale solution for curvilinear motion reconstruction. In *International Conference on Robotics and Automation*, pages 5732–5739. IEEE, 2013.

[124] Rahul Kumar Namdev, Abhijit Kundu, K Madhava Krishna, and CV Jawahar. Motion segmentation of multiple objects from a freely moving monocular camera. In *International Conference on Robotics and Automation*, pages 4092–4099. IEEE, 2012.

[125] Richard A Newcombe, Steven J Lovegrove, and Andrew J Davison. Dtam : Dense tracking and mapping in real-time. In *International Conference on Computer Vision*, pages 2320–2327. IEEE, 2011.

[126] Tal Nir, Alfred M Bruckstein, and Ron Kimmel. Over-parameterized variational optical flow. *International Journal of Computer Vision*, 76(2) :205–216, 2008.

[127] David Nistér. An efficient solution to the five-point relative pose problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6) :756–770, 2004.

[128] David Nistér, Oleg Naroditsky, and James Bergen. Visual odometry. In *Computer Vision and Pattern Recognition*, pages I–652. IEEE, 2004.

[129] Kemal E Ozden, Konrad Schindler, and Luc Van Gool. Multibody structure-from-motion in practice. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(6) :1134–1141, 2010.

[130] Frank Pagel. Calibration of non-overlapping cameras in vehicles. In *Intelligent Vehicles Symposium*, pages 1178–1183. IEEE, 2010.

[131] Matia Pizzoli, Christian Forster, and Davide Scaramuzza. Remode : Probabilistic, monocular dense reconstruction in real time. In *International Conference on Robotics and Automation*, pages 2609–2616. IEEE, 2014.

[132] Robert Pless. Using many cameras as one. In *Computer Vision and Pattern Recognition*, volume 2, pages II–587. IEEE, 2003.

[133] Jean-Philippe Pons, Renaud Keriven, and Olivier Faugeras. Multi-view stereo reconstruction and scene flow estimation with a global image-based matching score. *International Journal of Computer Vision*, 72(2) :179–193, 2007.

[134] Michael JD Powell. An efficient method for finding the minimum of a function of several variables without calculating derivatives. *The Computer Journal*, 7(2) :155–162, 1964.

[135] Clemens Rabe, Thomas Müller, Andreas Wedel, and Uwe Franke. Dense, robust, and accurate motion field estimation from stereo image sequences in real-time. In *European Conference on Computer Vision*, pages 582–595. Springer, 2010.

[136] René Ranftl, Vibhav Vineet, Qifeng Chen, and Vladlen Koltun. Dense monocular depth estimation in complex dynamic scenes. In *Computer Vision and Pattern Recognition*, pages 4058–4066. IEEE, 2016.

[137] N Dinesh Reddy, Iman Abbasnejad, Sheetal Reddy, Amit Kumar Mondal, and Vindhya Devalla. Incremental real-time multibody vslam with trajectory optimization using stereo camera. In *Intelligent Robots and Systems*, pages 4505–4510. IEEE, 2016.

[138] Jorma Rissanen. Universal coding, information, prediction, and estimation. *Information Theory*, 30(4) :629–636, 1984.

[139] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *European Conference on Computer Vision*, pages 430–443. Springer, 2006.

[140] Eric Royer, Maxime Lhuillier, Michel Dhome, and Jean-Marc Lavest. Monocular vision for mobile robot localization and autonomous navigation. *International Journal of Computer Vision*, 74(3) :237–260, 2007.

[141] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. ORB : an efficient alternative to SIFT or SURF. In *International Conference on Computer Vision*, pages 2564–2571. IEEE, 2011.

[142] Reza Sabzevari and Davide Scaramuzza. Monocular simultaneous multi-body motion segmentation and reconstruction from perspective views. In *International Conference on Robotics and Automation*, pages 23–30. IEEE, 2014.

[143] Reza Sabzevari and Davide Scaramuzza. Multi-body motion estimation from monocular vehicle-mounted cameras. *IEEE Transactions on Robotics*, 32(3) :638–651, 2016.

[144] Joaquim Salvi, Jordi Pages, and Joan Batlle. Pattern codification strategies in structured light systems. *Pattern Recognition*, 37(4) :827–849, 2004.

[145] Davide Scaramuzza. 1-point-ransac structure from motion for vehicle-mounted cameras by exploiting non-holonomic constraints. *International Journal of Computer Vision*, 95(1) :74–85, 2011.

[146] Davide Scaramuzza and Friedrich Fraundorfer. Visual odometry - part i : The first 30 years and fundamentals. *IEEE Robotics & Automation Magazine*, 18(4) :80–92, 2011.

[147] Davide Scaramuzza, Friedrich Fraundorfer, Marc Pollefeys, and Roland Siegwart. Absolute scale in structure from motion from a single vehicle mounted camera by exploiting nonholonomic constraints. In *International Conference on Computer Vision*, pages 1413–1419. IEEE, 2009.

[148] Davide Scaramuzza, Friedrich Fraundorfer, and Roland Siegwart. Real-time monocular visual odometry for on-road vehicles with 1-point ransac. In *International Conference on Robotics and Automation*, pages 4293–4299. IEEE, 2009.

[149] Davide Scaramuzza and Roland Siegwart. Appearance-guided monocular omnidirectional visual odometry for outdoor ground vehicles. *IEEE Transactions on Robotics*, 24(5) :1015–1026, 2008.

[150] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1-3) :7–42, 2002.

[151] Daniel Scharstein and Richard Szeliski. High-accuracy stereo depth maps using structured light. In *Computer Vision and Pattern Recognition*, volume 1, pages I–I. IEEE, 2003.

[152] Konrad Schindler, David Suter, and Hanzi Wang. A model-selection framework for multibody structure-and-motion of image sequences. *International Journal of Computer Vision*, 79(2) :159–177, 2008.

[153] Cordelia Schmid, Roger Mohr, and Christian Bauckhage. Evaluation of interest point detectors. *International Journal of Computer Vision*, 37(2) :151–172, 2000.

[154] Steven M Seitz, Brian Curless, James Diebel, Daniel Scharstein, and Richard Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *Computer Vision and Pattern Recognition*, volume 1, pages 519–528. IEEE, 2006.

[155] Jianbo Shi and Carlo Tomasi. Good features to track. In *Computer Vision and Pattern Recognition*, pages 593–600. IEEE, 1994.

[156] Natalia Slesareva, Andrés Bruhn, and Joachim Weickert. Optic flow goes stereo : A variational method for estimating discontinuity-preserving dense disparity maps. In *Joint Pattern Recognition Symposium*, pages 33–40. Springer, 2005.

[157] Frank Steinbrücker, Thomas Pock, and Daniel Cremers. Advanced data terms for variational optic flow estimation. In *Vision, Modeling and Visualization Workshop*, pages 155–164, 2009.

[158] Hauke Strasdat, José MM Montiel, and Andrew J Davison. Visual slam : why filter ? *Image and Vision Computing*, 30(2) :65–77, 2012.

[159] Jan Stühmer, Stefan Gumhold, and Daniel Cremers. Real-time dense geometry from a handheld camera. In *Joint Pattern Recognition Symposium*, pages 11–20. Springer, 2010.

[160] Peter Sturm. A historical survey of geometric computer vision. In *Computer Analysis of Images and Patterns*, pages 1–8. Springer, 2011.

[161] Peter Sturm, Srikumar Ramalingam, Jean-Philippe Tardif, Simone Gasparini, and João Barreto. Camera models and fundamental concepts used in geometric computer vision. *Foundations and Trends in Computer Graphics and Vision*, 6(1–2) :1–183, 2011.

[162] Deqing Sun, Jonas Wulff, Erik B Sudderth, Hanspeter Pfister, and Michael J Black. A fully-connected layered model of foreground and background flow. In *Computer Vision and Pattern Recognition*, pages 2451–2458. IEEE, 2013.

[163] Michael E Tipping and Christopher M Bishop. Mixtures of probabilistic principal component analyzers. *Neural Computation*, 11(2) :443–482, 1999.

[164] Engin Tola, Vincent Lepetit, and Pascal Fua. Daisy : An efficient dense descriptor applied to wide-baseline stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(5) :815–830, 2010.

[165] Miroslav Trajković and Mark Hedley. Fast corner detection. *Image and Vision Computing*, 16(2) :75–87, 1998.

[166] Bill Triggs, Philip F McLauchlan, Richard I Hartley, and Andrew W Fitzgibbon. Bundle adjustment—a modern synthesis. In *International Workshop on Vision Algorithms*, pages 298–372. Springer, 1999.

[167] Roberto Tron and René Vidal. A benchmark for the comparison of 3-d motion segmentation algorithms. In *Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.

[168] Paul Tseng. Nearest q-flat to m points. *Journal of Optimization Theory and Applications*, 105(1) :249–252, 2000.

[169] Levi Valgaerts, Andrés Bruhn, Markus Mainberger, and Joachim Weickert. Dense versus sparse approaches for estimating the fundamental matrix. *International Journal of Computer Vision*, 96(2) :212–234, 2012.

[170] Levi Valgaerts, Andrés Bruhn, Henning Zimmer, Joachim Weickert, Carsten Stoll, and Christian Theobalt. Joint estimation of motion, structure and geometry from stereo sequences. In *European Conference on Computer Vision*, pages 568–581. Springer, 2010.

[171] Luc Van Gool et al. Dense matching of multiple wide-baseline views. In *International Conference on Computer Vision*, pages 1194–1201. IEEE, 2003.

[172] Sundar Vedula, Simon Baker, Peter Rander, Robert Collins, and Takeo Kanade. Three-dimensional scene flow. In *International Conference on Computer Vision*, volume 2, pages 722–729. IEEE, 1999.

[173] Sundar Vedula, Peter Rander, Robert Collins, and Takeo Kanade. Three-dimensional scene flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(3) :475–480, 2005.

[174] René Vidal. Subspace clustering. *Signal Processing Magazine*, 28(2) :52–68, 2011.

[175] René Vidal and Richard Hartley. Three-view multibody structure from motion. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 30(2) :214–227, 2008.

[176] René Vidal, Yi Ma, and Shankar Sastry. Generalized principal component analysis (GPCA). *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(12) :1945–1959, 2005.

[177] René Vidal, Yi Ma, Stefano Soatto, and Shankar Sastry. Two-view multibody structure from motion. *International Journal of Computer Vision*, 68(1) :7–25, 2006.

[178] Christoph Vogel, Stefan Roth, and Konrad Schindler. View-consistent 3d scene flow estimation over multiple frames. In *European Conference on Computer Vision*, pages 263–278. Springer, 2014.

[179] Christoph Vogel, Konrad Schindler, and Stefan Roth. 3d scene flow estimation with a rigid motion prior. In *International Conference on Computer Vision*, pages 1291–1298. IEEE, 2011.

[180] Christoph Vogel, Konrad Schindler, and Stefan Roth. 3d scene flow estimation with a piecewise rigid scene model. *International Journal of Computer Vision*, 115(1) :1–28, 2015.

[181] George Vogiatzis and Carlos Hernández. Video-based, real-time multi-view stereo. *Image and Vision Computing*, 29(7) :434–441, 2011.

[182] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4) :395–416, 2007.

[183] Andreas Wedel, Thomas Brox, Tobi Vaudrey, Clemens Rabe, Uwe Franke, and Daniel Cremers. Stereoscopic scene flow computation for 3d motion understanding. *International Journal of Computer Vision*, 95(1) :29–51, 2011.

[184] Andreas Wedel, Clemens Rabe, Tobi Vaudrey, Thomas Brox, Uwe Franke, and Daniel Cremers. Efficient dense scene flow from sparse or dense stereo data. In *European Conference on Computer Vision*, pages 739–751. Springer, 2008.

[185] Walter Thompson Welford. *Aberrations of optical systems*. CRC Press, 1986.

[186] Manuel Werlberger, Thomas Pock, and Horst Bischof. Motion estimation with non-local total variation regularization. In *Computer Vision and Pattern Recognition*, pages 2464–2471. IEEE, 2010.

[187] Robert J Woodham. Photometric method for determining surface orientation from multiple images. *Optical Engineering*, 19(1) :191139, 1980.

[188] Jonas Wulff and Michael Julian Black. Modeling blurred video with layers. In *European Conference on Computer Vision*, pages 236–252. Springer, 2014.

[189] Koichiro Yamaguchi, David McAllester, and Raquel Urtasun. Robust monocular epipolar flow estimation. In *Computer Vision and Pattern Recognition*, pages 1862–1869. IEEE, 2013.

[190] Koichiro Yamaguchi, David A McAllester, and Raquel Urtasun. Efficient joint segmentation, occlusion labeling, stereo and flow estimation. In *European Conference on Computer Vision*, pages 756–771. Springer, 2014.

[191] Jingyu Yan and Marc Pollefeys. A general framework for motion segmentation : Independent, articulated, rigid, non-rigid, degenerate and non-degenerate. In *European Conference on Computer Vision*, pages 94–106. Springer, 2006.

[192] Christopher Zach, Thomas Pock, and Horst Bischof. A duality based approach for realtime tv-l 1 optical flow. In *Joint Pattern Recognition Symposium*, pages 214–223. Springer, 2007.

[193] Luca Zappella, Alessio Del Bue, Xavier Lladó, and Joaquim Salvi. Joint estimation of segmentation and structure from motion. *Computer Vision and Image Understanding*, 117(2) :113–129, 2013.

[194] Guofeng Zhang, Xueying Qin, Wei Hua, Tien-Tsin Wong, Pheng-Ann Heng, and Hujun Bao. Robust metric reconstruction from challenging video sequences. In *Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.

[195] Ruo Zhang, Ping-Sing Tsai, James Edwin Cryer, and Mubarak Shah. Shape-from-shading : a survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(8) :690–706, 1999.

[196] Teng Zhang, Arthur Szlam, Yi Wang, and Gilad Lerman. Hybrid linear modeling via local best-fit flats. *International Journal of Computer Vision*, 100(3) :217–240, 2012.

[197] Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11) :1330–1334, 2000.

[198] Henning Zimmer, Andrés Bruhn, and Joachim Weickert. Optic flow in harmony. *International Journal of Computer Vision*, 93(3) :368–388, 2011.

[199] M Zuliani, CS Kenney, and BS Manjunath. The multiransac algorithm and its application to detect planar homographies. In *International Conference on Image Processing*, pages III–153. IEEE, 2005.