Guillaume Caron Le 20 juin 2007

Odométrie visuelle pour un robot mobile en environnement industriel

Rapport de stage de Master Recherche STIC

Encadrants de stage

El Mustapha Mouaddib

Ouiddad Labbani-Igbida

Rapporteurs

Danielle Nuzillard

Alban Goupil

Remerciements

Je tiens tout particulièrement à remercier le Professeur El Mustafa Mouaddib, directeur du *Centre de Robotique, d'Électrotechnique et d'Automatique* à Amiens, pour m'avoir accueilli dans ses locaux pour la période du stage du Master Recherche en STIC de l'université de Reims. Je le remercie une fois encore, ainsi que Madame Ouiddad Labbani-Igbida, pour leur encadrement qui fut de grande qualité.

Je remercie par la même occasion la société BA Systèmes, instigatrice du projet *VOG*, dans le cadre duquel s'est inscrit mon stage.

J'adresse aussi une pensée particulière à mes collègues de bureau, thésards, stagiaires et post-doc de l'équipe robotique et des autres personnes du laboratoire car passer ces quelques mois en leur compagnie fut très agréable et enrichissant, grâce à la bonne ambiance qui y règne, tant scientifiquement qu'humainement.

Sommaire

I. Introduction

De nos jours, les robots sont de plus en plus utilisés, en particulier dans le domaine industriel et les usines. Ils peuvent avoir différents rôles, dont le déplacement de marchandises qui nous intéresse ici. En effet, le travail présenté dans ce document s'inscrit dans le projet *VOG* (*Vision Omnidirectionnelle pour le* Guidage) réunissant le *Centre de Robotique, d'Électrotechnique et d'Automatique* de l'*Université Picardie Jules Verne* à Amiens, l'*Institut de recherche en informatique et systèmes aléatoires* à Rennes ainsi que la société *BA Systèmes* située à Rennes aussi.

L'objectif de ce projet est de développer un nouveau système capable de guider un robot uniquement par la vision dans un environnement industriel. Le robot est alors composé d'un chariot pour le transport, de capteurs et d'une unité informatique de traitement qui connaît le but du déplacement du robot.

Actuellement, les capteurs permettent au robot de recevoir des signaux venant de l'environnement ou de balises émettrices dans le but de le guider ou de lui permettre de se localiser.

Une des technologies utilisées dans les usines pour ce type de robot repose sur la disposition d'émetteurs infrarouges ou laser sur le chemin que doit suivre le robot, ses capteurs lui permettant de recevoir les signaux de ces émetteurs. Une autre approche consiste à tracer une ligne au sol et, par une simple caméra placée sous le chariot, de suivre cette ligne. Enfin, on trouve aussi d'autres approches, plus globales cette fois-ci, où le robot n'embarque pas de capteur spécifique pour lui permettre de « percevoir » son environnement, mais où une ou plusieurs caméras sont positionnées dans l'usine, généralement en hauteur, et permettent de savoir où se trouve le robot pour lui transmettre des ordres de déplacement. Le gros inconvénient de ces méthodes, c'est qu'il faut équiper l'environnement spécifiquement pour permettre au robot de se localiser, ce qui entraîne des coûts supplémentaires et une période d'installation et de mise au point assez longue même si ces techniques sont très efficaces.

C'est pour répondre à ces problèmes qu'une autre approche est proposée, uniquement basée sur la vision cette fois-ci. L'idée est d'embarquer une seule caméra sur le robot et de n'utiliser que l'information visuelle perçue par ce biais pour lui permettre de se localiser, sans équiper particulièrement l'environnement.

Pour ce faire, plusieurs approches reposant sur différents capteurs sont envisageables. Il est possible de réaliser un système qui estime la trajectoire du robot à partir des images captées en continu. On appelle cela l'odométrie visuelle. On peut aussi développer une méthode de localisation qui consiste, à chaque fois que le robot capte une image, à la comparer à la mémoire qu'il a de son environnement, l'image la plus « proche » permettant d'estimer une position. Cette approche est, quant à elle, basée sur l'indexation d'images nécessitant une base de données (un ensemble d'images de l'environnement) construite durant une phase d'apprentissage. Le type de capteur entre aussi en jeu, les atouts n'étant pas les mêmes, que le système utilise une caméra perspective ou omnidirectionnelle.

Il faut donc faire la part des choses entre le type de caméra à utiliser, l'approche à adopter ou à inventer et la possibilité de réaliser ce système de navigation de sorte qu'il puisse être exécuté en temps-réel. Ces points sont à prendre en compte afin de réaliser un système optimal au possible.

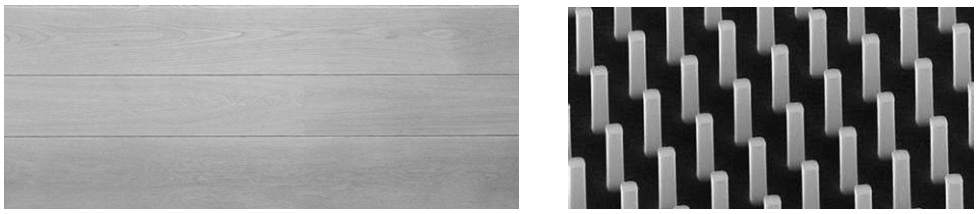
Dans un premier temps, c'est l'odométrie visuelle avec une caméra perspective qu'il a été choisi d'étudier et de développer afin d'en évaluer les avantages et les inconvénients.

Ce document va présenter le système réalisé après un rappel des travaux déjà réalisés. Ensuite, les résultats finaux seront présentés et discutés avant de prendre du recul sur ce qui a été fait et en tirer des conclusions. Mais commençons par présenter la problématique concrète.

II. Problématique et proposition

II.1. Problématique

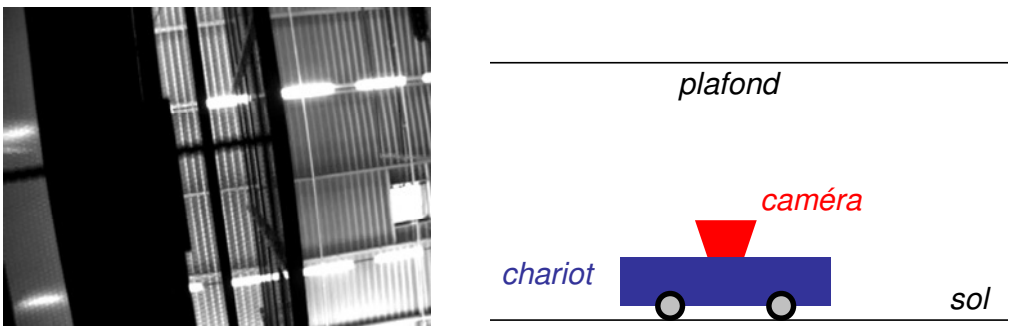
Comme nous allons le voir par la suite, l'odométrie visuelle repose bien souvent sur des techniques de suivi dans une séquence d'images. Pour que ce suivi soit possible, il faut qu'il soit possible de détecter des caractéristiques particulières dans les images, qui sont alors plus ou moins texturées (nous ne traiterons pas du cas de marqueurs positionnés dans l'environnement ce qui irait à l'encontre d'un des objectifs du projet). On entend par là que l'image contient des motifs ou que tout du moins, l'image n'est pas totalement uniforme, bref que sur certaines zones, au moins, il y ait des variations de contraste dans le cas des images en niveau de gris (Figure 1) dans lequel on se place. Tout ceci dans le but de pouvoir détecter des primitives visuelles pour les suivre d'une image à l'autre.



(a) **(b)**

Figure 1 : Exemple d'images sans (a) et avec motifs (b). Sur (b), on peut détecter des points d'intérêt

Revenons au cas précis qui doit être traité ici. Le robot mobile (le chariot) évolue dans une usine dans laquelle il doit déplacer des marchandises d'un point à un autre. Si on souhaite réaliser un système d'odométrie visuelle, il faut donc que la caméra embarquée par le robot, si c'est une caméra perspective, soit orientée dans une direction où il est toujours possible de détecter des caractéristiques visuelles. Ajoutons qu'on se place dans le cas où la caméra est fixée sur le robot et est donc immobile par rapport au robot de telle sorte qu'un mouvement de la caméra soit directement interprétable comme étant un mouvement du robot. Le choix a été fait d'orienter la caméra vers le plafond de telle sorte qu'on puisse toujours détecter des primitives visuelles (lampes, structures métalliques, etc) qui sont immobiles. On aurait pu choisir de suivre le sol mais le sol des usines est souvent dépourvu de motifs.



(a) **(b)**

Figure 2 : (a) Exemple d'image de plafond d'usine captée par la caméra du robot suivant le dispositif (b)

Enfin, la caméra étant pointée vers le haut, son axe optique est aligné avec la verticale du robot donc ses déplacements, dont ses rotations, seront réellement directement interprétables comme ceux du robot. La Figure 2 présente un exemple d'image captée par la caméra ainsi que le schéma du dispositif.

II.2. Proposition

Pour estimer le déplacement de la caméra entre deux positions, ou encore deux points de vue qui nous donnent deux images, comme l'état de l'art le présentera, une trame particulière est généralement suivie. Dans un premier, des primitives visuelles prépondérantes sont sélectionnées dans les deux images. Ensuite, les primitives des images sont mises en correspondance de telle sorte qu'à partir d'une primitive de la première image, on puisse connaître où elle se trouve dans la seconde et vis versa. Dès lors, les images, via leurs primitives visuelles, sont appariées et l'étape suivante consiste à estimer la relation, dans l'espace image, permettant de transformer l'ensemble des primitives de la première image en l'ensemble des primitives de la seconde. Enfin, c'est à partir de cette relation et éventuellement grâce à une contrainte connue de la scène observée que sont estimés les paramètres de déplacement de la caméra. C'est sur une déclinaison de ce schéma que se base la proposition présentée dans ce document.

L'idée et de détecter des points d'intérêt par la méthode de Harris dans deux images, prises à différents points de vue, qui sont en fait des pixels étant au centre d'une grande variabilité locale des intensités des pixels voisins. Une fois ces primitives visuelles extraites, il faut les apparier et là, c'est une méthode de mise en correspondance croisée par évaluation d'une fonction de corrélation entre le voisinage d'un point et de ceux candidats qui est utilisée. Une fois cet appariement effectué, la relation estimée pour transformer le nuage de points de la première image en celui de la seconde est une homographie définie dans l'espace projectif. Une fois cette homographie estimée, elle est séparée des paramètres intrinsèques de la caméra afin de passer du système de coordonnées pixelliques au système de coordonnées normalisées ayant une unité métrique réelle. On obtient ainsi une homographie euclidienne qu'on va décomposer en une matrice de rotation et un vecteur de translation correspondant au déplacement de la caméra entre les deux prises de vue.

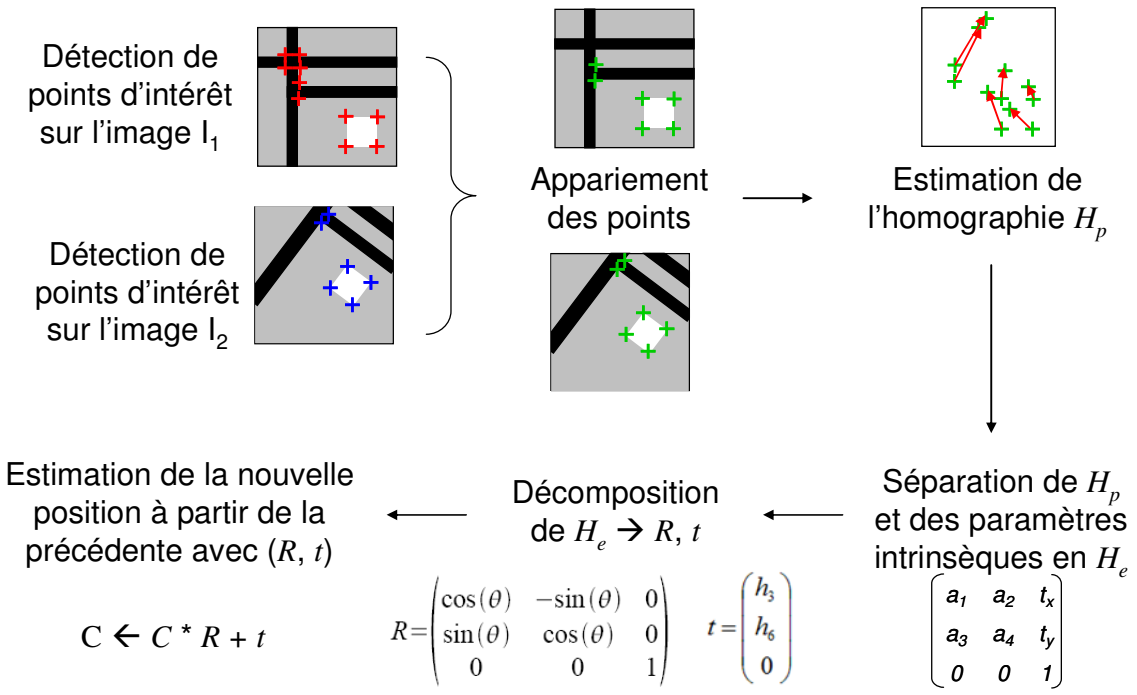


Figure 3 : Synoptique de l'algorithme d'estimation de déplacement de caméra entre deux prises de vues de la même scène

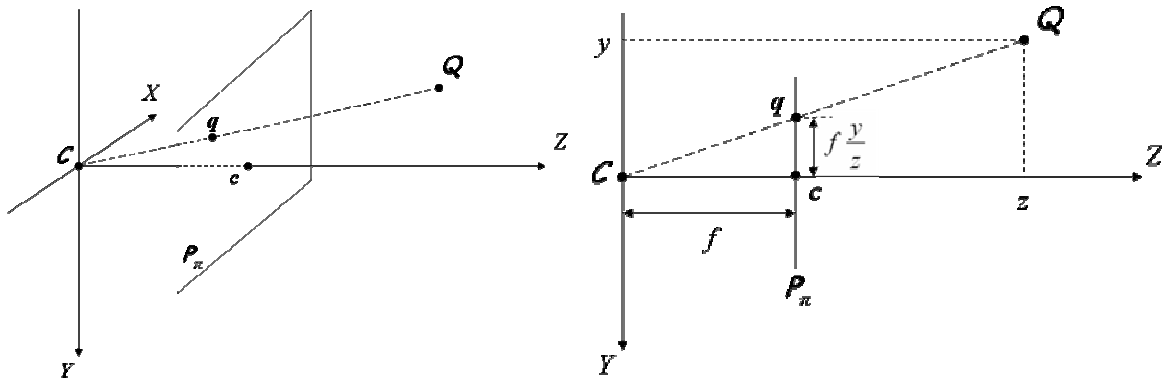
III. Géométrie des caméras et théorie

Cette section va présenter comment est modélisée une caméra perspective, par sa géométrie et ses paramètres ainsi que la modélisation des transformations entre points de vue.

La modélisation des caméras perspective et la théorie sur les transformations des images engendrées par le changement de point de vue sont basés sur les parties 1 et 2 de la thèse de Muriel Pressigout [3].

III.1. Modèle de projection d'une caméra perspective

Ce document présentant des travaux réalisés avec une caméra perspective, il convient d'en présenter les concepts et en particulier le modèle sténopé ou *pin-hole* en anglais. La Figure 4 présente ce modèle de projection perspective ainsi que les repères associés.



(a) **(b)**

Figure 4 : Le modèle sténopé : (a) la projection perspective de points de l'espace sur le plan image et (b) les repères associés à un tel modèle

Le modèle sténopé associe à la caméra un repère ($\mathcal{C}$, *X*, *Y*, *Z*), *X*, *Y* et *Z* étant les axes dont l'origine correspond au centre de projection. Le plan image est parallèle au plan (*X*, *Y*) et est situé à une distance *f*, appelée distance focale, de l'origine. La projection de $\mathcal{C}$ le long de l'axe *Z*, l'axe optique, coupe le plan image au point principal $\boldsymbol{c}$.

Un point de l'espace est défini par un vecteur de coordonnées homogènes $\boldsymbol{Q} = (x, y, z, 1)$ exprimées dans le repère caméra. Il se projette dans le plan image de la caméra en un point $\boldsymbol{q}$ dont les coordonnées pixels sont $\boldsymbol{q} = (u, v, 1)$. On passe de $\boldsymbol{Q}$ à $\boldsymbol{q}$ par une projection sur le plan image des coordonnées normalisées $\boldsymbol{q'} = (x', y', 1)$ puis par une transformation permettant de passer de l'espace métrique à l'espace pixellique.

La projection $\boldsymbol{q'}$ exprimée dans le repère caméra s'obtient de la façon suivante en appliquant le théorème de Thalès (Figure 4b) :

$$x' = f\frac{x}{z} \qquad y' = f\frac{y}{z} \tag{1}$$

ce qui peut se linéariser par une transformation projective A' :

$$A' = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad \text{telle que} \quad \boldsymbol{q'} = A'\,\boldsymbol{Q} \tag{2}$$

La transformation des coordonnées normalisées vers les coordonnées pixelliques exprimée dans le repère (O_π, U, V) prend en compte le changement d'unité ainsi qu'un changement d'origine (Figure 5) car O_π n'est pas le point principal. On a alors :

$$u = u_0 + \frac{x'}{l_x} \qquad v = v_0 + \frac{y'}{l_y} \tag{3}$$

Où $\boldsymbol{c}_p = (u_0, v_0, 1)$ définit les coordonnées pixelliques du point principal, l_x et l_y, les tailles d'un pixel le long des axes *X* et *Y*.

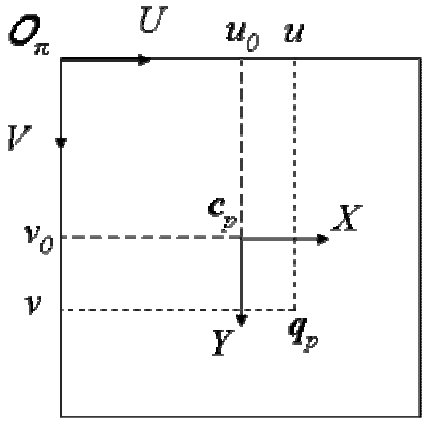


Figure 5 : Changement d'unité et de repère

L'équation **(3)** peut être linéarisée par une transformation projective *K'*:

$$K' = \begin{pmatrix} \frac{1}{l_x} & 0 & u_0 \\ 0 & \frac{1}{l_y} & v_0 \\ 0 & 0 & 1 \end{pmatrix} \quad \text{telle que} \quad \boldsymbol{q} = K' \boldsymbol{q}' \tag{4}$$

Il est courant de placer la focale dans la matrice *K'* (qui devient *K*) de façon à ce que la matrice de projection *A'* (qui devient *A*) ne contienne que des 1. Ceci est équivalent à considérer une caméra de focale 1 et dont les pixels seraient de taille $\left(p_x = \frac{f}{l_x}\right) \times \left(p_y = \frac{f}{l_y}\right)$. Une telle matrice *K* est appelée la matrice intrinsèque de la caméra.

Pour projeter un point $\boldsymbol{Q}$, dont les coordonnées sont exprimées dans le repère caméra, en un point $\boldsymbol{q}$ en coordonnées pixelliques, on procède comme suit :

$$\boldsymbol{q} \propto K' A' \boldsymbol{Q} = K A \boldsymbol{Q} \quad \text{avec} \quad A = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad \text{et} \quad K = \begin{pmatrix} p_x & 0 & u_0 \\ 0 & p_y & v_0 \\ 0 & 0 & 1 \end{pmatrix} \tag{5}$$

III.2. Transformations entre images

Dans cette partie, la description des transformations entre deux images d'une même scène vue par une seule caméra est abordée. Le but pour l'application présentée dans ce document est de pouvoir déterminer le mouvement de la caméra à partir des deux images, c'est pourquoi, il faut le définir. Deux possibilités sont alors envisageables : exprimer la relation entre ces deux images par la géométrie épipolaire ou en exploitant la matrice d'homographie pour effectuer le transfert d'une image à l'autre.

La géométrie épipolaire définit des transformations d'une image à l'autre, qu'on se trouve dans l'espace normalisé ou pixellique, respectivement par les matrices essentielle ou fondamentale. Cependant, il existe des cas où cette approche est dégénérée, notamment dans le cas d'une rotation pure entre les deux images où les épipoles ne sont pas définissables, ou dans le suivi de plan, ce qui est plus grave pour nous. En effet, en revenant à la Figure 2a, on remarque que ce qui est observé par la caméra (le plafond) est un environnement contenant un ensemble de plans. C'est pourquoi l'approche homographique est préférée car même s'il existe un cas où cette relation est dégénérée, il n'est pas susceptible de ce produire dans notre cas. Le cas dégénéré d'une homographie apparaît lorsqu'on suit un plan qui dans une des images est visible et dans l'autre, la caméra a subit une rotation telle qu'il passe maintenant par le centre optique de la caméra et est donc invisible. Ce cas est donc impossible dans l'application visée par notre projet puisque le plafond d'une usine ne viendra jamais passer par le centre optique de la caméra ou tout du moins, si cela se passe, c'est que l'application n'a même plus lieu d'être.

Développons alors maintenant l'expression d'une homographie planaire. Tout d'abord, rappelons qu'un plan $\boldsymbol{P}$ est défini par un point $\boldsymbol{Q}$ et un vecteur normal N. Notons d_1, la distance entre ce plan et le centre de projection $\boldsymbol{C}_1$ (Figure 6).

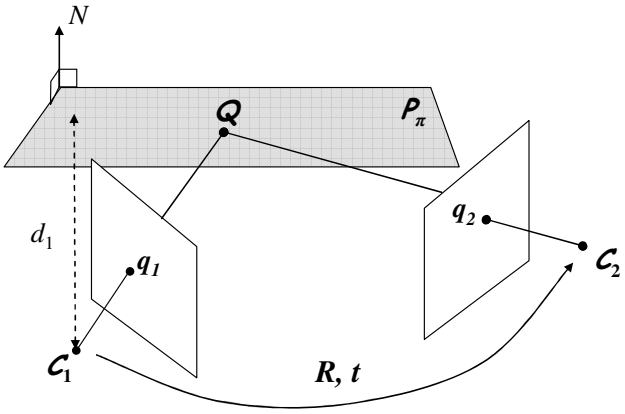


Figure 6 : Passage d'une vue à l'autre exprimé par homographie

Soit $\boldsymbol{Q}^{(1)}$ et $\boldsymbol{Q}^{(2)}$, les coordonnées de $\boldsymbol{Q}$, respectivement dans les deux repères caméra successifs. Ces deux vecteurs colonnes sont liés par une matrice de rotation R (3 x 3) et le vecteur de translation t de dimension 3 permettant de passer du premier repère caméra au second par la relation suivante :

$$\boldsymbol{Q}^{(2)} = R\,\boldsymbol{Q}^{(1)} + t \qquad \textbf{(6)}$$

qu'on homogénéise de la façon suivante :

$$\boldsymbol{Q}^{(2)} = T\,\boldsymbol{Q}^{(1)} \quad \text{avec} \quad T = \begin{pmatrix} R & t \\ 0_{1\times 3} & 1 \end{pmatrix} \qquad \textbf{(7)}$$

Si $\boldsymbol{Q}$ appartient au plan $\boldsymbol{P}_\pi$, alors ses coordonnées $\boldsymbol{Q}^{(1)} = (x^{(1)}, y^{(1)}, z^{(1)})$ dans le premier repère caméra vérifient :

$$N^{\mathrm{T}} \boldsymbol{Q}^{(1)} = d_1 \tag{8}$$

En utilisant cette relation, on peut factoriser l'équation **(6)** par $\boldsymbol{Q}^{(1)}$:

$$\boldsymbol{Q}^{(2)} = \left(R + \frac{tN^T}{d_1} \right) \boldsymbol{Q}^{(1)} \tag{9}$$

Il est possible de lier les points dans le système de coordonnées normalisées grâce à la relation de projection A donnant $Z^{(1)} \boldsymbol{q}'^{(1)} = \boldsymbol{Q}^{(1)}$ de la façon suivante :

$$Z^{(2)} \boldsymbol{q}'^{(2)} = \left(R + \frac{tN^T}{d_1} \right) Z^{(1)} \boldsymbol{q}'^{(1)}$$

$$\frac{Z^{(2)}}{Z^{(1)}} \boldsymbol{q}'^{(2)} = \left(R + \frac{tN^T}{d_1} \right) \boldsymbol{q}'^{(1)}$$

$$\text{Ainsi, } \boldsymbol{q}'^{(2)} \propto H' \boldsymbol{q}'^{(1)} \text{ avec } H' = R + \frac{tN^T}{d_1} \tag{10}$$

H est alors la matrice homographique de dimension (3 x 3) qui lie les points d'une image à l'autre par l'introduction des coordonnées du plan. La transformation est homogène à un facteur d'échelle près et H a huit degrés de liberté (tous les coefficients sauf le H(3,3) qui est fixé à 1).

La relation peut néanmoins s'exprimer à partir des coordonnées pixelliques avec $H = KH'K^{-1}$ comme suit :

$$\boldsymbol{q}^{(2)} \propto H \boldsymbol{q}^{(1)} \tag{11}$$

III.3. Conclusion partielle

Cette partie a visé a introduire certaines notions géométriques et théoriques nécessaires pour la suite des travaux dans la mesure où les homographies reposent sur la géométrie de la caméra et son modèle de projection et qu'elles seront utilisées dans plusieurs aspect du processus d'estimation de trajectoire.

IV. État de l'art en odométrie visuelle

IV.1. Généralités

On appelle « odométrie » l'estimation d'une trajectoire à partir d'un ensemble de relevés de positions relatives. Ainsi, on retrouve, dans le cas de robots mobiles se déplaçant sur des roues, des capteurs sur ces roues, composés d'un capteur infrarouge et d'une roue codeuse, dont les données relevées permettent d'interpoler le déplacement du robot dans un laps de temps très court. Malheureusement, ce type de dispositif engendre une erreur non bornée, surtout quand les trajectoires ne sont pas rectilignes. Dans la même famille, on compte aussi les capteurs inertiels qui entraînent le même type d'erreur. C'est pourquoi des méthodes de recalage par la vision sont apparues pour corriger l'erreur engendrée par ce type de capteur. Depuis, la puissance de la vision pour l'estimation de trajectoire s'est révélée et des méthodes uniquement basées sur la vision sont apparues.

L'odométrie visuelle est donc un sujet de recherche du moment mais qui prend ses racines dans les années 1980. Actuellement, dans cette branche de la recherche, on trouve tout type de caméra, monoculaires ou stéréo, perspective, omnidirectionnelle, fish eye, etc.

En environnement connu, deux grandes familles de méthodes existent, à savoir la recherche de références géométriques dans les images pour y reconnaître un objet ou une partie d'une scène 3D connue et préalablement modélisée (modèle CAO) ou le travail direct sur l'image. C'est cette dernière approche que nous allons développer car elle ne nécessite pas de connaissances précises de l'environnement dans lequel évolue le robot mobile a priori.

Ainsi, l'idée générale va être de détecter les similitudes entre deux images prises successivement par le robot et de déterminer le déplacement de la caméra qui a permis de passer de l'une à l'autre. Les méthodes proposées suivent presque toujours le schéma suivant : détection de primitives visuelles dans deux (ou plus) images successives, appariement de ces primitives entre images selon l'évaluation de corrélations, estimation d'une relation permettant de transformer les primitives d'une image en celles de l'autre et enfin, à partir de cette relation, décomposer le mouvement observé en rotation et translation de la caméra entre les deux prises de vue. Suivant ce qui est observé par la caméra, et la nature de son mouvement, on va estimer différemment cette transformation.

L'essentiel des recherches, du moins en ce qui concerne les parties théoriques servant de bases aux méthodes actuellement développées, a été rassemblé dans deux rapports de recherche de l'INRIA qui nous intéressent particulièrement. Le premier, intitulé *2D ½ Visual Servoing* a été écrit par Malis, Chaumette et Boudet [1]. Son objectif est de développer des méthodes d'asservissement visuel, c'est-à-dire de déplacement de caméra, placée par exemple au bout d'un bras robot (système « eye-in-hand »), afin de se placer devant un objet de telle sorte qu'on atteigne un état où l'objet ciblé soit dans l'orientation voulue selon un modèle préalablement défini. Cependant, ce rapport présente bien l'estimation de la structure de la scène à partir du mouvement de la caméra, appelé « Structure from motion » où l'estimation d'homographie est présentée. Le second rapport, *A projective framework for structure and motion recovery from two views of a piecewise planar scene* de Bartoli, Sturm et Horaud [2], présente le problème de déterminer la structure d'une scène à partir de deux vues dont le contenu est planaire par morceau. Enfin, les parties 1, 2 et 4 de la thèse de Muriel Pressigout, *Approches hybrides pour le suivi temps-réel d'objets complexes dans des séquences video* [3] sont un bon recueil pour les définitions géométriques des caméras, la présentation des primitives visuelles possibles et le développement des homographies. C'est d'ailleurs sur cette thèse que la section III est inspirée.

La question des primitives visuelles est importante. Rappelons que le but du travail est d'estimer une position de la caméra (donc du robot) à partir d'une autre grâce à deux images, l'une acquise à la première position et l'autre à la seconde. Bien entendu, il faut que ces deux images se recouvrent un minimum pour pouvoir en repérer les éléments communs afin de déterminer le déplacement de la caméra qui a changé le point de vue. Ce qui est assuré dans notre cas puisqu'une image est acquise toutes les 40 millisecondes et que la vitesse du robot permet un très important recouvrement entre deux images successives.

Même s'il est possible d'estimer le mouvement à partir de points, de droites ou les deux à la fois, ce sont souvent les points d'intérêt qui sont utilisés dans la littérature. Que ce soit le détecteur de Förstner, celui de Harris ou le détecteur SUSAN, l'idée est de détecter dans une image ce qui est identifiable comme des coins, c'est-à-dire une zone de forte variation d'intensité (gradient) dans plus d'une seule direction. Cet aspect est toujours en développement puisqu'en janvier 2006, un détecteur de coins basé sur un raisonnement flou a été publié [4].

Dans notre cas, nous souhaitons suivre un plafond, c'est pourquoi l'intérêt s'est porté sur des articles qui présentent l'estimation de position à partir du suivi d'un plan. Les articles de Pears et Liang [5] et [6] présentent des méthodes de détection et de suivi du plan du sol alors qu'une étude plus récente [7] présente un système d'estimation de position relative de caméra selon un plan parallèle à celui de la caméra, perpendiculaire au sol. Ces articles sont tous basés sur l'estimation d'homographie puisqu'ils cherchent à calculer un déplacement en se basant sur un plan de la scène. Cependant, Pears et Liang ou encore Nistér en stéréo [8], et d'autres se placent dans le cas où le plan de la caméra est perpendiculaire au sol sur lequel se déplace le robot.

IV.2. Travaux récents

Développons néanmoins ces travaux proches du sujet de ce document. Pears et Liang commencent donc par se placer dans un cadre d'étude précis en admettant que le robot se déplace sur une surface plane et en intérieur. Ils travaillent à partir de points d'intérêt détectés et appariés dans deux images proches (en terme de changement de point de vue) qui vont permettre d'estimer la transformation pour passer d'un ensemble de points à l'autre. Leurs développements sont intéressants puisqu'au lieu d'admettre que les points détectés sont coplanaires, ils cherchent à le démontrer dans [5] en sélectionnant un sous-ensemble de points réellement coplanaires. Pour ce faire, puisque l'axe optique de la caméra est supposé parallèle au plan du sol et donc pointe l'horizon, ils se limitent à travailler uniquement sur la moitié inférieure des images, qui contient le sol, dans laquelle ils calculent une matrice d'homographie dominante, c'est-à-dire vérifiant le plus grand nombre d'associations de points. Ces points sont alors considérés comme étant des inliers et le processus se répète sur cet ensemble de points jusqu'à ce qu'il se stabilise. Enfin, la vérification que les points coplanaires extraits reposent bien sur le sol passe par le calcul de la normale au plan formé par les points. Ce processus est en fait une classification qui sépare les points détectés en deux groupes : les points du sol et les autres. [5] présente ce processus principalement pour déterminer où le robot peut aller sur le plan du sol (évitement d'obstacles) alors que [6] utilise cette approche pour déterminer la distance entre le plan et la caméra ainsi que ses paramètres de mouvement entre les prises de vue. Le principal facteur d'erreur pour cette mesure est lié à la distorsion des images à proximité des bords avec seulement un écart moyen de 3% par rapport à la réalité. Les tests mentionnés pour l'estimation de la rotation de la caméra font état d'une erreur d'un angle de 4° en fin de mouvement (120 images). Le calcul de l'angle de rotation repose sur le calcul des valeurs propres de la matrice homographique estimée à partir des points. Malheureusement les expérimentations présentées dans leurs articles ne se font jamais sur un grand ensemble d'images et beaucoup de mouvement. Cependant, leur méthode

est intéressante car, contrairement aux méthodes présentées dans les rapport de recherche de l'INRIA présentés précédemment, il n'est pas nécessaire d'extraire l'homographie euclidienne de l'homographie projective (en séparant l'homographie projective estimée des paramètres intrinsèques de la caméra). En effet, c'est justement en utilisant les valeurs propres de cette matrice qu'il est possible de calculer directement l'angle de rotation.

Xu De se place dans un contexte où le plan observé est parallèle à celui de la caméra et perpendiculaire au sol, à l'initialisation. Le repère de la caméra est, comme dans notre système, défini avec l'axe Z confondu avec l'axe optique mais dans cas précis le plan (X, Y) de la caméra est parallèle au plan observé. Cependant, les rotations de la caméra se font autour de l'axe X dont l'angle est formé par les axes Z de deux repères caméra successifs. Aussi, même si l'homographie est mentionnée comme base théorique, la translation d'un centre de la caméra à l'autre et l'angle de rotation sont exprimés uniquement en fonction de coefficients obtenus à partir des coordonnées pixelliques d'un couple de points. Le calcul se fait donc pour chaque couple de points et les résultats sont moyennés pour n'obtenir qu'un seul angle et qu'un seul vecteur de déplacement. Cependant, contrairement au travail de Pears et Liang, les points d'intérêt sont sélectionnés et appariés manuellement. De plus, même si les résultats sont très précis, 1% d'erreur relative en moyenne pour les angles et les positions, les conditions d'expérimentation sont idéales et les déplacements très faibles.

Voilà pour l'essentiel des travaux récents les plus proches de notre cadre. Ces travaux sont apparus intéressants à de nombreux points de vue mais sont développés dans un contexte applicatif différent du notre. Aussi il apparaît intéressant, connaissant le processus général d'estimation de déplacement de caméra (détection de primitives, appariement, estimation de transformation) d'étudier plus précisément chaque point de ce processus.

IV.3. Détection de primitives

Abordons en premier lieu la détection de points d'intérêt. Avant de faire l'état de l'art des travaux développant de tels détecteurs, il est bon de préciser que leur champ d'application n'est pas limité à l'objet de ce document. En effet, on les retrouve dans bien des domaines de la vision par ordinateur, que ce soit en reconnaissance, en analyse des formes et du mouvement ou encore dans la reconstruction 3D d'une scène. C'est l'analyse du mouvement qui nous intéresse particulièrement et les points ont l'avantage sur les contours d'enlever toute ambiguïté sur le mouvement. Une approche pour sélectionner des points d'intérêt est de détecter des coins dans l'image. Les coins sont des extrémités ou des intersection de contours. Parmi les détecteurs les plus populaires, on retrouve ceux qui utilisent une matrice de structure locale dont les coefficients sont obtenus en combinant les dérivées partielles de l'image ; on parle encore de matrice liée à la fonction d'autocorrélation de l'image. Cette matrice contient les dérivées partielles de la fonction d'intensité. L'un de ces détecteurs est le détecteur de Harris [9]. Sa méthode est basée sur un seuillage : la mesure de la force du coin, définie à partir des valeurs propres de la matrice de structure, est comparée à un seuil savamment défini. Un autre détecteur apparaissant souvent dans la littérature est le détecteur SUSAN (Smallest Univalue Segment Assimiling Nucleus) basé sur une comparaison de luminosité sans dépendre des dérivées de l'image [10]. La fenêtre SUSAN est déplacée dans l'image et atteint un minimum quand elle repose sur un coin. Cette méthode est réputée pour être moins sensible au bruit mais utilise toujours un seuil. Ces méthodes sont néanmoins toute deux efficaces et la méthode de Harris reste très utilisée car assez rapide en exécution. Cependant, Varkonyi-Koczy [4], plus récemment, a publié un détecteur de coins basé sur un raisonnement flou qui donne des degrés d'intérêt au coin en vue de l'appariement suivant généralement l'étape de détection de coins. Cette méthode est basée sur un autre algorithme de détection de coins utilisant la matrice de structure déjà mentionnée, l'algorithme de Förstner [11] qui détecte les coins par un maximum local d'une fonction combinant les

différents éléments de la matrice de structure. Le raisonnement flou utilise une fonction d'appartenance continue entre les points détectés comme étant des coins et ceux détectés comme n'en étant pas. Par cette méthode, des caractéristiques pour l'appariement de points entre deux images sont directement définies.

Enfin, il existe une autre approche assez reprise dans la littérature malgré sa jeunesse et c'est l'algorithme des SIFT (Scale-invariant feature transform) qui est un algorithme d'extraction de primitives discriminantes des images. Les primitives sont invariantes aux changements d'échelle et aux rotations et sont robustes (*i.e.* partiellement invariant) au changement de point de vue et d'illumination. Cet algorithme a été conçu par David Lowe en 2003 [28].

IV.4. Appariement de primitives entre deux images

Une fois les points détectés dans deux images, il faut les apparier, c'est-à-dire mettre en correspondance chaque point situé au même endroit de la scène dans les deux images, en vue de calculer la transformation qui permet de passer d'un ensemble de points à l'autre (changement de point de vue).

Pour apparier deux points, on peut travailler sur l'intensité des pixels du voisinage ou sur des descripteurs locaux plus complexes utilisant des dérivées premières et parfois seconde voire plus. Bien entendu, la seconde classe de méthodes pour apparier des points est plus efficace, plus précise et moins source d'erreur (faux appariement) car plus robuste aux transformations géométriques et aux variations d'éclairage. Elle est d'ailleurs utilisée dans l'indexation d'image par attributs locaux comme par exemple dans les travaux de Cordelia Schmid ([12] et suivants). Mais elle nécessite de dériver la fonction d'intensité plusieurs fois ce qui joue en sa défaveur par rapport aux méthodes utilisant directement l'intensité du voisinage d'un point d'intérêt en terme de temps de calcul. En effet, il ne faut pas oublier que le système présenté dans ce document devra pouvoir travailler en temps réel. C'est pourquoi la première classe de méthodes a été préférée. Et un aperçu va en être donné ici.

Tout d'abord, présentons la méthode des SSD (Sum of Squares Differences). Un point dans une image va avoir un certain nombre de points candidats à l'appariement dans l'autre pour de multiples raisons. Une mesure d'écart (« distance » entre voisinages) est donc réalisée entre les voisinages locaux des points à apparier et c'est la mesure maximale qui défini la mise en correspondance. Le principe de la SSD est de calculer les différences pixels à pixels entre les voisinages des deux points pour lesquels ont effectue la mesure. Cette différence est mise au carré pour amplifier les fortes différences et minimiser les faibles. On en fait ensuite la somme et on obtient la mesure.

Une autre mesure de corrélation plus élaborée et moins sensible aux variations de luminosité est la ZNCC (Zero-mean Normalized Cross Corrélation). Cette méthode travaille aussi sur un voisinage d'un coin de taille définie. De la même manière, c'est cette mesure qui va départager les différents candidats à l'appariement. Les voisinages des points d'intérêt sont dans un premier temps centrés et réduits (moyenne nulle et amplitude maximale unitaire). Ensuite, le voisinage du point de la première image est multiplié par celui des points candidats et c'est la valeur de corrélation maximale qui détermine à quel point celui de la première image est apparié. La mesure est effectuée dans les deux sens (de l'image 1 vers l'image 2 et vis versa) et deux points ne sont réellement appariés au final que s'ils ont l'un pour l'autre une valeur de corrélation maximale. Un autre gros avantage de cette méthode est que la corrélation est exprimée entre -1 et 1. On peut donc définir un seuil (taux de corrélation) à partir duquel on peut avoir confiance en l'appariement de deux points. C'est généralement à partir de 0,8 (les points sont corrélés à plus de 80%) qu'on peut commencer à avoir une certaine certitude dans l'appariement. Ce seuil supplémentaire point nous préserve mieux des faux appariements.

Cette dernière méthode est déjà très efficace mais il en existe de plus complexes notamment celle de Scott et Longuet-Higgins [13] basée sur la décomposition en valeurs singulières. L'idée est de construire une matrice de corrélation *G* comme dans la méthode de la ZNCC, mais en lui ajoutant un poids gaussien prenant en paramètre la distance entre les points à apparier, et de décomposer cette matrice en valeurs singulières : $G=USV^T$ où *U* et *V* sont des matrices orthogonales et *S* diagonale contenant les valeurs singulières. En fixant la diagonale de *S* à 1 et en remultipliant les trois matrices, on obtient une matrice de score qui a la propriété intéressante de sélectionner les bonnes paires. Au final, un seul point est retenu. Cette approche développée dans [13] a été récemment reprise et enrichie par Kwolek [14].

IV.5. Estimation de la transformation entre images

Introduisons dans cette partie le livre de Hartley et Zisserman, *Multiple-view geometry* [15], la bible du domaine, s'il est possible de s'exprimer ainsi. Bon nombre de concepts nous concernant y sont développés, dont les estimations de transformation entre deux images caractérisée par les correspondances entre les points d'intérêt. A partir de ce moment, nous allons nous concentrer sur l'estimation du mouvement selon les hypothèses et les constatations qui ont été présentées dans la partie théorique, à savoir les transformations projectives 2D. Selon l'ouvrage susmentionné, il est possible d'estimer différemment le mouvement :

- Calcul de l'homographie projective 2D à partir d'un ensemble de correspondances entre deux images (i.e. des points)
- Calcul de la matrice fondamentale qui est une matrice singulière
- Calcul de tenseurs multi-focaux à partir de correspondances à travers *n* images

Ces problèmes ont des points communs et nous avons déjà rejeté le calcul de la matrice fondamentale dans notre étude théorique puisqu'elle peut être dégénérée. Quand aux tenseurs multi-focaux, ils n'ont pas été abordés dans ce travail qui s'est concentré sur les homographies qui sont plus aisément abordables et plus rapidement surtout. Cependant, si dans le futur, il est nécessaire d'augmenter la précision du système, ces tenseurs sont de bons outils pour diminuer les erreurs et sont plus performants dans le suivi de primitives tout au long de séquences d'images.

Le but de l'estimation de l'homographie projective 2D entre les deux ensembles de points appariés est de calculer une matrice 3x3 à partir de ces ensembles et les liant.

On mentionne « un ensemble de points », certes, mais combien précisément ? Ou du moins, combien de point faut-il au minimum pour pouvoir estimer la transformation ? Cette borne est connue en considérant le nombre de degrés de liberté du problème et de contraintes. La matrice homographique 3x3, *H*, contient neuf coefficients mais est définie à un facteur d'échelle près ce qui engendre huit degrés de liberté pour la transformation 2D. De l'autre côté, chaque correspondance compte pour deux contraintes, puisque pour chaque point de la première image les deux degrés de liberté (*x* et *y*) du point correspondant dans la seconde image doit correspondre à l'application de *H* sur le premier point. Les points sont définis par un vecteur homogène de dimension trois qui a aussi deux degrés de liberté puisque l'échelle est arbitraire. Par conséquent, il faut quatre points pour estimer *H* complètement. Dans le cas particulier de transformation rigide affine, seuls six degrés de libertés sont à estimer (le rectangle supérieur 2x3 de *H*), et donc à partir de trois points, car dans une telle transformation, la dernière ligne de *H* est fixée à (0 0 1).

Avec quatre points, dans le cas général, une solution exacte est possible pour *H*. C'est la solution minimale. Cependant, généralement, un nombre de points supérieur est à disposition et il arrive que certains de ces points soient du bruit. Le but sera alors d'estimer *H* de la meilleure façon en trouvant la transformation qui minimise une fonction de coût ou qu'il satisfasse un nombre maximum de correspondances selon l'approche RANSAC [16] qui

sélectionne quatre correspondances au hasard, estime *H*, l'applique à tous les points, et ne garde que l'homographie satisfaisant un maximum de correspondances en réitérant.

Plusieurs algorithmes existent pour estimer *H*. On rencontre d'abord l'algorithme DLT (Direct Linear Transform) [17] qui estime *H* en résolvant un système d'équations linéaires à partir d'exactement quatre correspondances. Si le problème est surdéterminé, *i.e.* il y a plus de quatre correspondances, une solution exacte peut-être trouvée s'il n'y a pas de bruit. Cependant c'est extrêmement rare et il faut alors chercher à minimiser la norme du vecteur contenant les coefficients de la matrice *H*. Pour éviter les cas dégénérés de l'estimation, les points sélectionnés ne doivent pas être colinéaires. Juste pour préciser, Hartley et Zisserman montrent que le problème de l'estimation de *H* à partir de lignes est équivalent à celui à partir de points.

Les approches minimisant des fonctions de coût utilisent une distance algébrique ou une distance géométrique dans le but de minimiser l'erreur de reprojection entre les deux images. Il est aussi montré dans [15] que dans le cas d'une transformation affine, les distances algébrique et géométrique sont identiques.

En plus de pouvoir estimer *H* à partir de points ou de lignes, il est possible, par la méthode de Sampson [18] de l'estimer à partir de coniques en minimisant l'erreur de Sampson. On retrouve aussi des fonctions de coût statistiques et l'estimation du maximum de vraisemblance. Mais ces méthodes nécessitent de modéliser l'erreur de mesure, le bruit.

Une déclinaison de l'estimation du maximum de vraisemblance (MLE : Maximum Likelihood Estimation) est aussi utilisée avec l'approche RANSAC mais se résume dans ce cas à l'erreur de reprojection pour déterminer grâce à un seuil sur cette quantité si l'homographie s'applique bien à un point donné (effectué sur tous les points : algorithme de vote).

Le « Gold Standard Algorithm » (GSA) pour l'estimation de *H*, défini comme tel dans [15], utilise quand à lui une décomposition en valeurs singulières sur l'ensemble des points, centré en zéro grâce à une translation *t* pour les points de la première image et *t'* pour ceux de la seconde image, afin de calculer le bloc linéaire de la matrice homographique affine.

Un des critères de ces dernières méthodes est leur robustesse car malgré celle des méthodes de mise en correspondance, il peut rester des outliers, liés ou non à l'appariement d'ailleurs, à l'étape d'estimation de la transformation, ce qui engendre de graves erreurs.

C'est pourquoi des méthodes robustes sont bien souvent utilisées. [19] de Malis et Marchand est un bon rassemblement de la théorie des méthodes robustes.

Récemment, un travail de rassemblement a été effectué par Jain en 2006, à l'occasion de la proposition d'une nouvelle méthode basée contours [20], sous la forme d'un tableau, repris ici (Tableau 1), qui lie les méthodes d'estimation de transformation, les primitives utilisées pour ce faire mais aussi le type de transformation estimable.

IV.6. Décomposition de l'homographie projective

L'homographie 2D nous permet de connaître la transformation qui lie un ensemble de points à un autre. Cependant, dans la plupart des cas, il n'est pas possible d'extraire directement le déplacement de la caméra lié à cette homographie projective. Le but est de déterminer la rotation et la translation qui caractérisent le déplacement de la caméra entre les deux prises de vue. L'obtention de ces paramètres à partir d'une matrice de transformation, telle que la matrice homographique, a fait l'objet de plusieurs développements.

En premier lieu, Tsai dans les années 1980, a publié un article [26] présentant l'estimation des paramètres de mouvement 3D en décomposant en valeurs singulières la matrice homographique. On y trouve comment décomposer la matrice de transformation en rotation, translation et récupérer la normale au plan observé suivant la multiplicité des valeurs singulières de *H* du moment que cette matrice est bien estimée à partir d'un plan rigide.

Plus récemment, Zhang et Hanson ont publié [27] une méthode basée à la fois sur la décomposition en valeurs singulières de H et sur la décomposition en valeurs propres. Ils séparent la décomposition en plusieurs cas selon la relation qui existe entre deux quantités calculées à partir des valeurs propres et l'ordre des valeurs singulières.

Source	Primitives	Technique	Transformation	Remarque
Nombreuses	Points, Patches	Correlation	Similitude	Populaire pour le recalage d'image. Bien développé dans la littérature du traitement d'images.
Livres, travaux antérieurs [15]	Points, Lignes	(DLT)	Projective	Solution directe mais très sensible au bruit.
Luong *et al.* [21]	Points avec des informations en plus	Utilisation de la calibration faible	Projective	Utilisation d'informations comme la matrice fondamentale qui nécessite des correspondances de points pour l'estimation.
Kanatani *et al.*, etc [22], [16]	Points, lignes, etc	RANSAC, ML, Estimation par Moindres carrés	Projective	Robuste à un grand nombre d'outliers et beaucoup moins sensible que le DLT ; très populaire
Kuthirummal *et al.* [23]	Contour non paramétrique	Transformée de Fourier de séquences	Affine	Calcul des invariants affins et des approximations polynomiales des contours dans le domaine de Fourier
Kruger et al., [24]	Texture	Transformée de Fourier de patches d'images	Affine	Correspondances minimes de lignes
Kumar et al., [25]	Coniques / polygones	Invariants projectifs	Projective	Deux correspondances de coniques
Jain [20]	Contour	Transformée de Fourier de séquences	Projective	Estimation de l'approximation affine et calcul de la profondeur projective itérativement pour le calcul robuste de l'homographie

Tableau 1 : Outils, méthodes et résultats pour l'estimation d'homographie (reproduit et traduit de [20])

V. Estimation de trajectoire

Fort des outils théoriques présentés en section III et de la connaissance des méthodes d'odométrie visuelle publiées récemment, un processus d'estimation de trajectoire a été élaboré et développé et c'est l'objet de cette partie.

Comme il a pu être remarqué, à quelques variantes près, le processus généralement suivi dans l'estimation de trajectoire consiste à calculer un ensemble de déplacements qui ont eu lieu entre les prises de vue réalisées par le robot en mouvement. Nous allons donc nous intéresser de plus près au calcul du déplacement du robot entre deux prises de vue.

En se basant sur le synoptique présenté en Figure 3, qui présente une idée générale du processus suivi pour estimer la trajectoire du robot, chaque point sera successivement repris et détaillé dans les sous-parties suivantes. Mais avant tout, la Figure 7 rappelle un exemple d'image acquise par la caméra du robot.

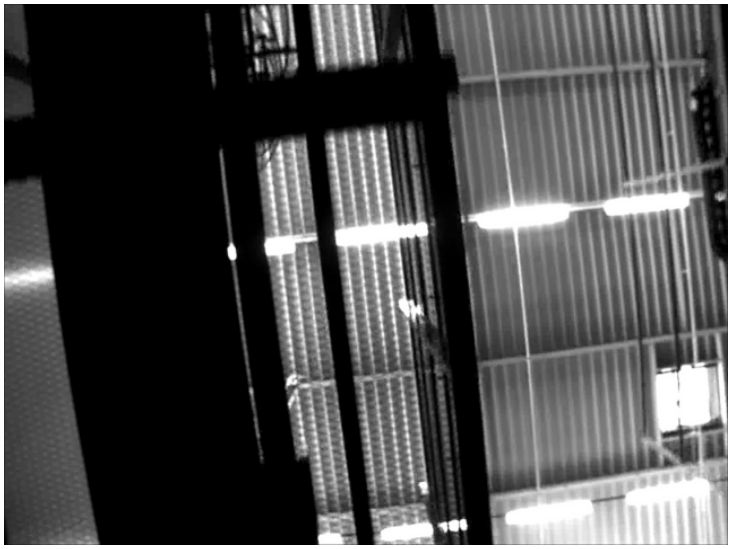


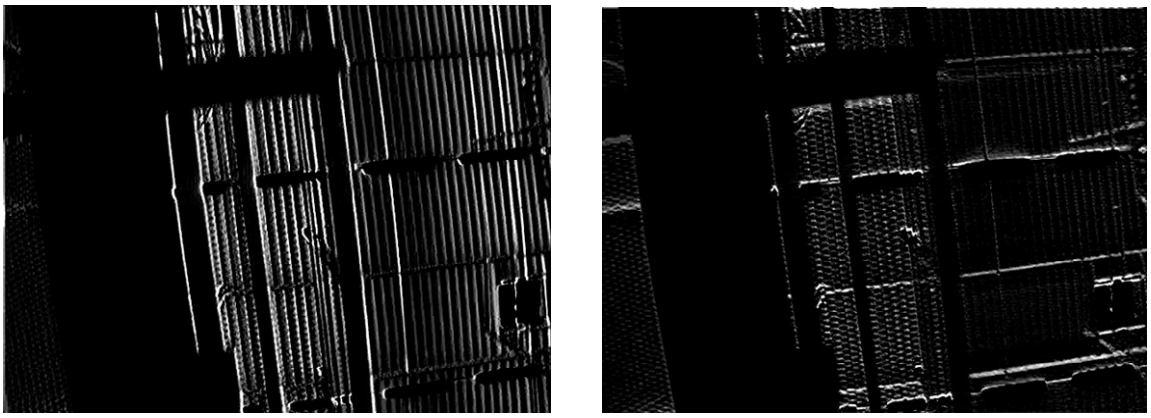
Figure 7 : Exemple d'une image acquise en situation par un robot mobile dans les conditions du projet

V.1. Détection des points d'intérêt

Afin de déterminer le mouvement de la caméra entre deux points de vue, il faut, dans un premier temps, calculer le mouvement entre les deux images. Pour ce faire, on ne travaille pas directement sur les pixels de l'image mais sur des points d'intérêt dans le but de n'utiliser que des zones discriminantes des images. Les points d'intérêt d'une image sont en fait ce qu'un humain interprète comme étant un coin dans la scène acquise sur l'image, c'est-à-dire un point (pixel ou zone de pixels suivant la résolution) où le contraste, ou encore la variation d'intensité, est fort dans plusieurs directions. Pour évaluer cette variation, on calcule classiquement les gradients de l'image à traiter en *X* et en *Y* (Figure 8).

La méthode de Harris [9] a été choisie pour la détection des points d'intérêt. Ce détecteur de coins se base sur une matrice de structure liée à la fonction d'autocorrélation calculée à partir des gradients de l'image à traiter **(12)**.

$$M = w \otimes \begin{pmatrix} \left(\frac{\partial I}{\partial x}\right)^2 & \left(\frac{\partial I}{\partial x}\right)\left(\frac{\partial I}{\partial y}\right) \\ \left(\frac{\partial I}{\partial x}\right)\left(\frac{\partial I}{\partial y}\right) & \left(\frac{\partial I}{\partial y}\right)^2 \end{pmatrix} \quad \textbf{(12)}$$



(a) **(b)**

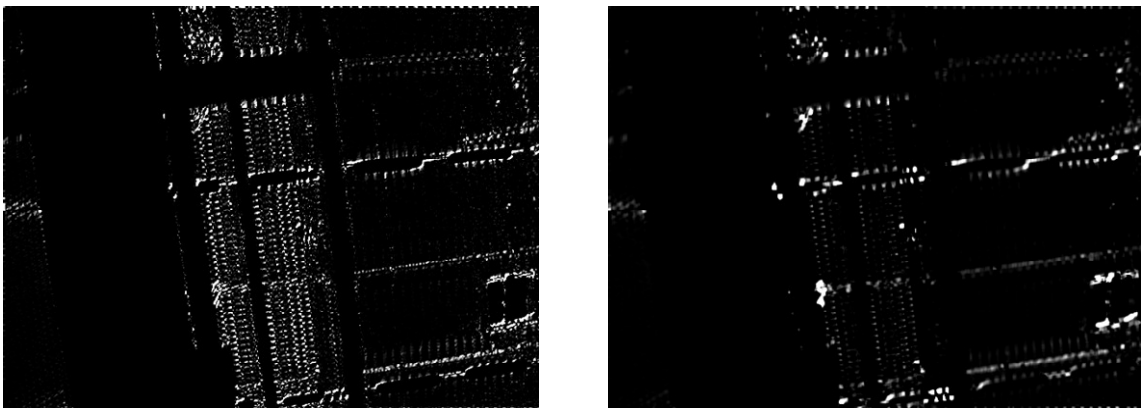
Figure 8 : Gradients en X (a) et en Y (b) de l'image de la Figure 7 pour les points d'intérêt.

Dans cette matrice, on retrouve donc les gradients autocorellés, mais aussi une convolution avec un noyau gaussien. Le but de ce lissage est d'éliminer les contours pour ne garder que les coins, réels points d'intérêt. La Figure 9 illustre ce principe en pratique sur l'image du produit des gradients selon les deux directions.

Une fois cette matrice de structure calculée, il faut calculer la « force de coin » en chaque pixel afin de ne garder que les plus forts au final. Dans son article original, Harris proposait de calculer les valeurs propres de cette matrice définie en chaque pixel et de seuiller sur la plus faible. En effet, si la plus faible des deux valeurs propres est supérieure à un seuil, la plus forte le sera aussi et cela signifie concrètement qu'il y a une variation dans deux directions pour le pixel courant. Par conséquent, ce point est un coin important.

Cependant, une approche équivalente a été développée par Harris et la force du coin peut être calculée à partir du déterminant et la trace la matrice M **(13)** avec k constant généralement égal à 0.04 (détermination empirique).

$$M = \begin{pmatrix} A & C \\ C & B \end{pmatrix}$$
$$\det(M) = \lambda_1 \lambda_2 = AB - C^2$$
$$trace(M) = \lambda_1 + \lambda_2 = A + B$$
$$C(x, y) = \det(M) - k(trace(M))^2 \qquad \textbf{(13)}$$

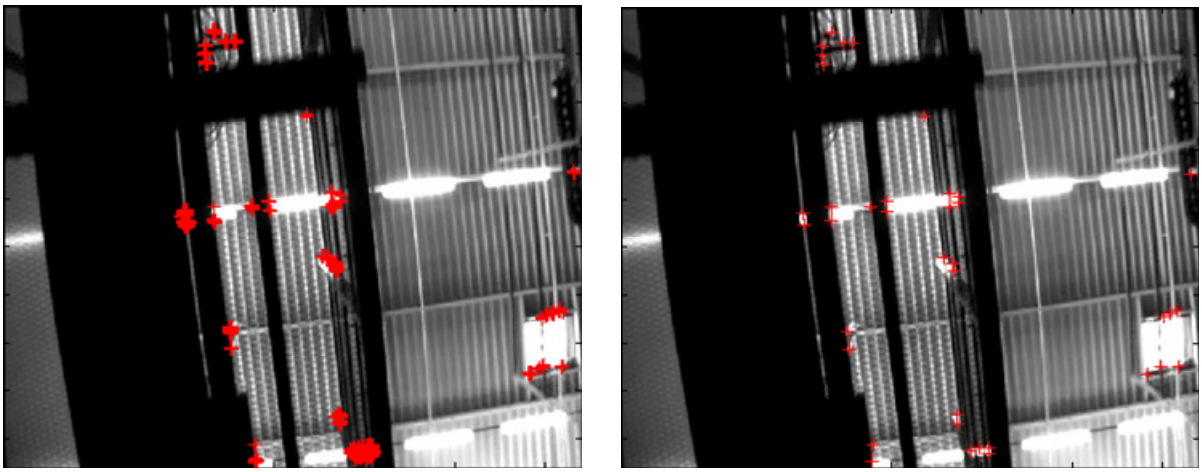


(a) **(b)**

Figure 9 : Résultat (b) et intérêt visuel du lissage gaussien sur l'image du produit des gradients (a) ; les points d'intérêt trop faible son affaiblis et seuls les forts persisteront après seuillage.

Une fois la force des coins $C(\cdot)$ calculée, un seuil y est appliqué comme discuté précédemment. Dans notre cas, il a fallut calculer un seuil dynamique, c'est-à-dire non constant pour toutes les images car ce seuil dépend fortement de l'image. Si la scène observée fourmille de détails, pour détecter les points d'intérêt les plus fort, le seuil doit être haut alors qu'à l'inverse, si l'image est assez lisse, pour détecter le peu de points d'intérêt imaginable, le seuil doit être faible. Dans les séquences video utilisées dans nos travaux, certaines parties des trajectoires du robot mobile sont plus ou moins éclairées ce qui engendre certaines images où le contraste est très fort et d'autres où il est plus diffus. Pour palier à ce problème, un coefficient qui sera utilisé pour le déterminer le seuil est calculé pour chaque image. Ce coefficient est en fait la valeur maximale de la carte des coins, *i.e.* l'ensemble des $C(\cdot)$ de toute l'image. En plaçant le seuil autour de 10% de cette valeur maximale, le problème qui apparaissait avec un seuil constant disparaît. C'est une des contributions de ce travail.

Une fois le seuil appliqué, il ne reste que les coins dignes d'intérêt (**Figure 10**a). Cependant, on note clairement que plusieurs points ont été détectés dans un voisinage proche comme étant des points d'intérêt. Pour alléger les calculs futurs et être plus robuste, on effectue une élimination des non maxima locaux à chaque pixel, si bien qu'un point est éliminé s'il n'a pas la force de coin la plus élevée de son voisinage. On épure ainsi un nombre important de point pour ne garder que les positions des coins réels (**Figure 10**b).



(a) **(b)**

Figure 10 : Points d'intérêt détectés (en rouge) sur l'image de la Figure 7. La méthode de détection de points de Harris donne le résultat (a). Cependant, on ne désire qu'un seul point par coin, d'où l'élimination non maximale (b).

V.2. Appariement des points entre deux images

Dès lors que les points d'intérêt sont détectés dans les deux images prises à différents points de vue, pour calculer le déplacement de la caméra qui a engendré la transformation de l'image, il faut apparier les points pour mettre en correspondance les différents éléments de la scène représentés dans les images. Pour ce faire, l'état de l'art a montré que plusieurs méthodes existaient. C'est la méthode par corrélation croisée normalisée et centrée qui a été retenue pour les avantages qu'elle présente. En effet, non seulement elle est moins sensible aux changements d'illumination que des méthodes plus simples du type SSD. De plus, la ZNCC (Équation 14) permet d'obtenir un taux de corrélation entre deux points puisque cette quantité est exprimée entre -1 et 1. Ainsi, on peut dire qu'au-delà d'un seuil de 0.8 (les points sont corrélés à 80%), on peut commencer à avoir confiance en l'appariement.

$$ZNCC(x,y,x',y') = \frac{\sum_{i=-w}^{w}\sum_{j=-w}^{w}(I(x+i,y+j)-\overline{I_w(x,y)})(I'(x'+i,y'+j)-\overline{I'_w(x',y')})}{\sqrt{\sum_{i=-w}^{w}\sum_{j=-w}^{w}(I(x+i,y+j)-\overline{I_w(x,y)})^2}\sqrt{\sum_{i=-w}^{w}\sum_{j=-w}^{w}(I'(x'+i,y'+j)-\overline{I'_w(x',y')})^2}}$$

Équation 14

Mais voyons de plus près comment cela se passe. Le principe est de déterminer un ensemble de points de la seconde image candidats à l'appariement d'un point de la première (et vis versa) pour chaque point et de ne sélectionner le point qui est non seulement le plus corrélé, *i.e.* celui qui a la valeur de ZNCC la plus élevée des candidats, mais dont cette corrélation est supérieure à 80%.

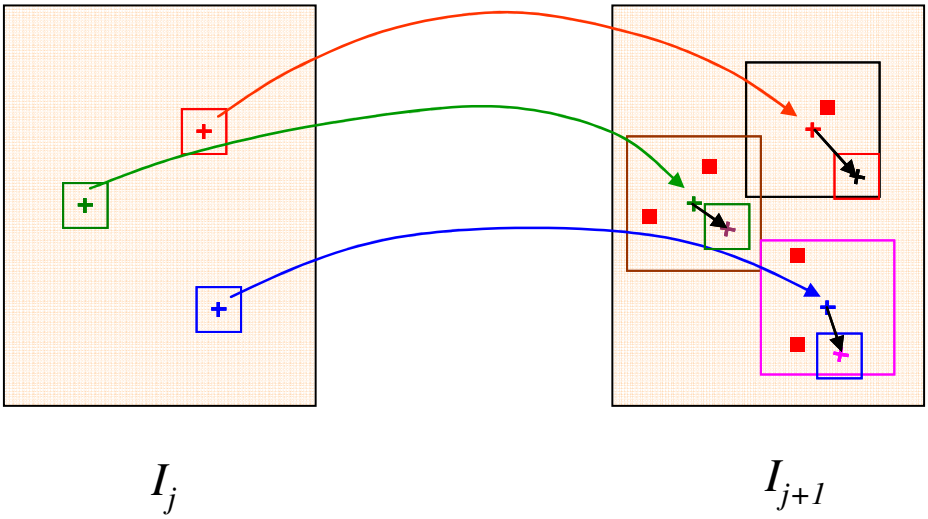
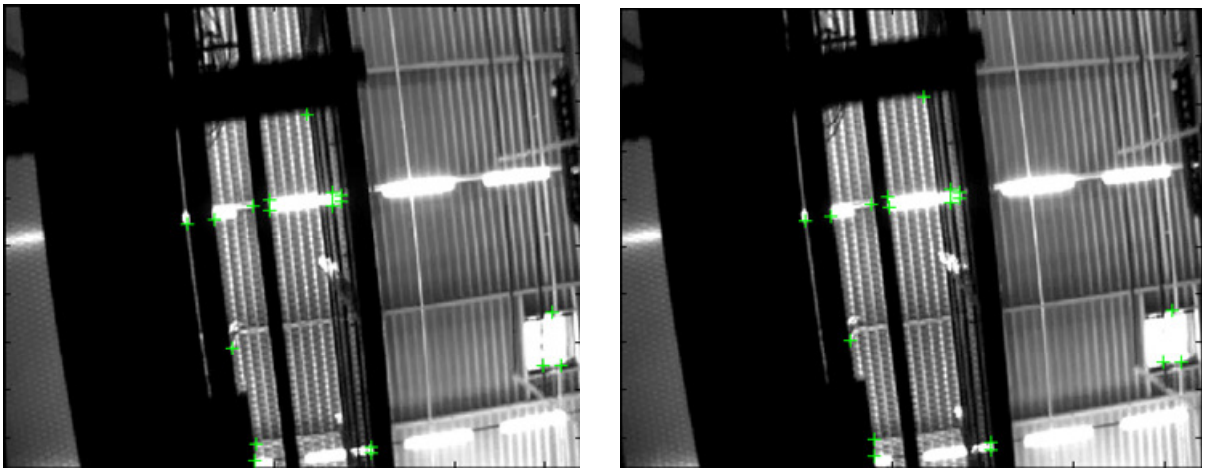


Figure 11 : Appariement croisé par corrélation centrée réduite (ZNCC). L'appariement se fait de l'image j vers l'image j+1 et inversement et au final, l'intersection des deux ensembles de correspondances obtenus donne les véritables appariements.

Pour ce faire, comme le montre la Figure 11, un voisinage est défini autour des points d'intérêt de manière à caractériser les images localement. Ensuite, un voisinage de recherche, centré sur les positions de chaque points de l'image *j*, est défini dans l'image *j+1* de telle sorte que les points de l'image *j+1* inclus dans ces fenêtre sont candidats à la mise en correspondance avec les points de la première image définissant les fenêtres de recherche. Ensuite, dans chaque fenêtre (et donc pour chaque point de la première image), la valeur de ZNCC est calculée entre les points inclus dans cette fenêtre et le point de la première image la définissant. Le point de l'image *j+1*, ayant le plus fort score de ZNCC avec le point de l'image *j*, est alors sélectionné si la corrélation est en plus supérieure à 0.8. Ensuite, le processus se déroule de l'image *j+1* vers l'image *j*. On obtient donc deux ensembles définissant les appariements d'une image à l'autre. Pour obtenir les correspondances les plus fiables possible, c'est l'intersection de ces deux ensembles qui donne l'ensemble des points réellement appariés (Figure 12).



(a) **(b)**

Figure 12 : Points (en vert) mis en correspondance entre (a) l'image de la Figure 7 et (b) la suivante dans la séquence video.

V.3. Estimation de la transformation entre images

Maintenant que les images sont mises en correspondance via leurs points d'intérêt (Figure 12) et afin de déterminer le déplacement de la caméra qui a engendré ce changement de point de vue, il faut déterminer la transformation géométrique qui lie les correspondances, *i.e.* les points de la première image appariés à ceux de la seconde.

V.3.1 Gold Standard Algorithm

Pour les raisons exposées dans la section III, c'est l'homographie qu'il a été choisi d'utiliser. Pour estimer la matrice homographique à partir des deux nuages de points mis en correspondance, l'algorithme « Gold Standard » de Hartley et Zisserman [15] utilisant la décomposition en valeurs singulières.

De ces nuages, deux translations sont extraites, t et t', qui permettent de centrer les deux nuages de points, c'est-à-dire de placer leur centre de gravité à l'origine du repère image. Ceci dans le but de n'avoir qu'à calculer la partie linéaire (rotation et changement d'échelle) de la transformation liant les nuages de points.

On construit ensuite la matrice A de dimension nx4, où n est le nombre de points, résultant de la concaténation des nuages de points, dont les lignes s'obtiennent de la façon suivante :

$$A_i^T = (P_i^T, P'^T_i) = (x_i, y_i, x'_i, y'_i)$$

On effectue alors la décomposition en valeurs singulières de A :

$$A = U\Sigma V^T$$

On sélectionne alors V_1 et V_2, les vecteurs singuliers à droite associés aux deux plus grandes valeurs singulières, définissant les deux axes principaux de A, c'est-à-dire caractérisant le sous espace 2D le plus significatif.

Dans [15], on peut voir que le point $A_i = (P_i^T, P'^T_i)^T$, repose sur $\mathcal{V}_H$, un sous-espace de codimension-2 de $\Re^4$, si et seulement si $[H_{2\times2} | -I_{2\times2}]A = 0$. Donc, connaissant A, la tâche d'estimation est équivalente à trouver l'espace de codimension-2 s'ajustant le mieux. [15] ajoute aussi que connaissant une matrice M, ayant pour lignes les A_i^T, le sous-espace s'ajustant le mieux aux A_i est couvert par les vecteurs singuliers V_1 et V_2 correspondant aux deux plus grandes valeurs singulières de M. Enfin, le carré supérieur gauche de la matrice

homographique définissant la partie linéaire de la transformation, $H_{2\times2}$, correspond au sous-espace couvert par V_1 et V_2 et est trouvé en résolvant le système suivant :

$$[H_{2\times2}|-I][V_1V_2]=0$$

en posant $[V_1V_2]_{4\times2}=\begin{bmatrix}B_{2\times2}\\C_{2\times2}\end{bmatrix}_{4\times2}$ et en remplaçant, on obtient :

$$[H_{2\times2}|-I]\begin{bmatrix}B_{2\times2}\\C_{2\times2}\end{bmatrix}=0$$

$$H_{2\times2}B_{2\times2}-I_{2\times2}C_{2\times2}=0$$

$$H_{2\times2}B_{2\times2}=C_{2\times2}$$

$$\boxed{H_{2\times2}=C_{2\times2}B_{2\times2}^{-1}} \quad \textbf{(15)}$$

Une fois que $H_{2\times2}$ est calculée, on peut déterminer complètement la matrice d'homographie affine en réutilisant les translations *t* et *t'* ayant servies à centrer les coordonnées des nuages de points :

$$H_A=\begin{bmatrix}H_{2\times2} & H_{2\times2}t-t'\\0^T & 1\end{bmatrix} \quad \textbf{(16)}$$

V.3.2 Filtre homographique

L'algorithme d'estimation d'homographie présenté précédemment a l'avantage indéniable de calculer la transformation satisfaisant au mieux les nuages de points mis en correspondance puisqu'il utilise une décomposition en axes principaux. Malgré cela, il faut néanmoins que les points servant d'entrée à cet algorithme soient coplanaires dans la scène sinon l'homographie estimée ne correspond à aucun plan existant. Cependant, déterminer un plan plus tôt dans la chaîne des processus, pour n'utiliser que les points reposant sur ce plan dans l'estimation de l'homographie, serait très coûteux en temps de calcul. L'idée pourrait alors être d'utiliser H_A et de calculer l'erreur de reprojection des ensembles de points *P* et *P'* (???) que cette matrice entraîne pour éliminer les points influant sur la mauvaise estimation de cette dernière, c'est-à-dire les points qui n'appartiennent pas au plan global du nuage de points dans le repère caméra. Une fois ces points éliminés, une nouvelle matrice homographique pourrait être calculée et on pourrait itérer le processus jusqu'à la stabilisation de l'erreur de reprojection.

L'intérêt de cette idée est confirmé par les travaux de Pears et Liang dans [5] qui segmentent le plan du sol grâce à des points coplanaires qui sont déterminés comme tel grâce à l'estimation de l'homographie sur les ensembles de correspondances successivement épurés des outliers.

Ce filtre particulier a été mis en place de telle sorte qu'une fois l'homographie H_A calculée, elle et sont inverse sont utilisés pour transformer les points des nuages de points pour donner l'image $\hat{P}'$ de *P* par la transformation H_A et l'image $\hat{P}$ de *P'* par H_A^{-1}, et calculer l'écart engendré *e* :

$$\hat{P}'=H_AP$$

$$\hat{P}=H_A^{-1}P'$$

$$e_i=\left|P'_i-\hat{P}'_i\right|^2+\left|P_i-\hat{P}_i\right|^2$$

Si e_i est supérieur à un seuil recalculé à chaque itération, alors le point i est considéré comme un outlier. Le seuil est recalculé à la fin d'une itération, en fonction de l'erreur moyenne des inliers.

Comme une majorité de points sont coplanaires dont le plan est parallèle au plan image, ce filtre permet de ne garder qu'un sous-ensemble des correspondances définissant un tel plan.

V.4. Décomposition de l'homographie

Il a été vu dans l'état de l'art que des méthodes de décomposition 3D du mouvement de la caméra à partir de l'homographie 2D existaient. Cependant, étant dans un cas particulier où le plan observé est parallèle au plan image et au sol, il est possible de déduire directement de l'homographie le déplacement de la caméra.

Il faut tout d'abord savoir que le bloc linéaire d'une transformation affine planaire contient les paramètres de rotation mais aussi de déformation. Ceci est rappelé dans [15] dans la partie sur la géométrie et les transformations projectives 2D.

Une relation homographique planaire se définit comme suit :

$$P'_i = H_A P_i = \begin{pmatrix} A & t \\ 0^T & 1 \end{pmatrix} P_i$$

où A est une matrice 2x2 non-singulière. Ce composant linéaire est composé de deux transformations fondamentales, à savoir les rotations et les changements d'échelle. La matrice A peut toujours être décomposée en :

$$A = R(\theta)R(-\phi)DR(\phi)$$

où $R(\theta)$ et $R(\phi)$ sont respectivement des rotations et D une matrice diagonale rassemblant les facteurs d'échelle :

$$D = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}$$

Cette décomposition provient directement de la décomposition en valeurs singulières en écrivant :

$$A = UDV^T = (UV^T)(VDV^T) = R(\theta)R(-\phi)DR(\phi)$$

puisque U et V sont des matrices orthogonales. A peut donc être décrite comme définissant une rotation, un changement d'échelle non isotrope, *i.e.* qui ne conserve pas les proportions, selon les deux axes transformés par la rotation précédente, une rotation inverse à la première pour retrouver les bons axes et enfin une autre rotation, la « réelle » rotation du plan, cette fois-ci ().

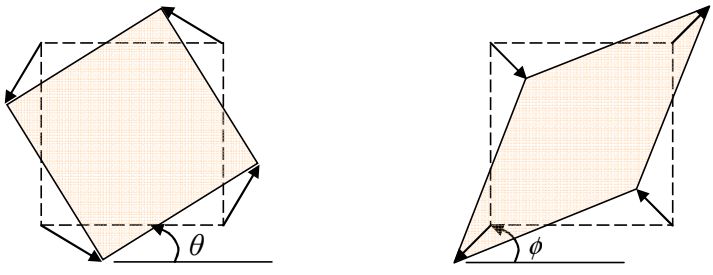


Figure 13 : Distorsions engendrées par une transformation affine planaire. (a) Rotation par $R(\theta)$. (b) Une déformation $R(-\phi)DR(\phi)$. Les directions du changement d'échelle dans la déformation sont orthogonales. Figure reproduite de [15].

En pratique, ayant estimé H_A, nous en effectuons la décomposition en valeurs singulières et ne gardons pour la rotation de la caméra que le produit :

$$R(\theta) = UV^T \text{ avec } H_{2\times 2} = U\Sigma V^T$$

Enfin, les paramètres de translation sont directement extraits de la matrice homographique, *i.e.* le vecteur colonne 1x2 haut droit.

V.5. Calcul de position et estimation de trajectoire

Une fois les paramètres de déplacement de la caméra connus, à savoir la translation et la rotation, tout est disponible pour calculer la nouvelle position du robot mobile, connaissant la précédente. Pour ce faire, le robot mobile évoluant sur un plan, son repère est défini par deux axes (Figure 14a). La translation étant définie dans le repère caméra après déplacement, on applique en premier la rotation d'angle θ au repère (Figure 14b). Selon ces axes transformés, on effectue la translation du centre du repère par (t_x, t_y) et la nouvelle position est alors connue (Figure 14c).

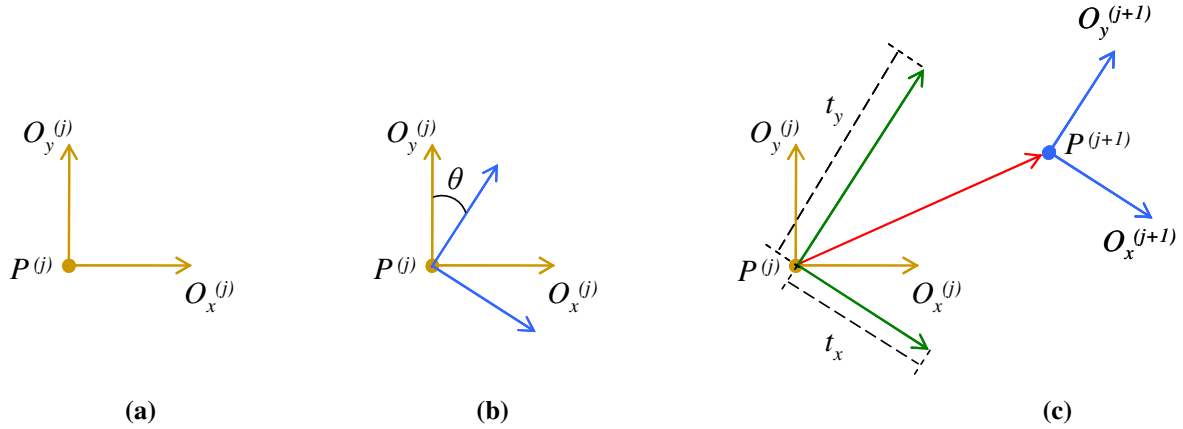
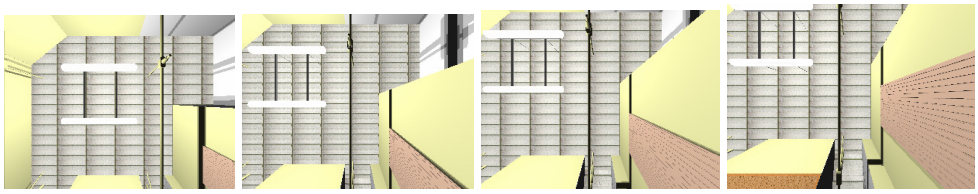


Figure 14 : Évolution schématique du calcul d'un nouvelle position par déplacement du repère « robot ». Partant du repère 2D à l'étape *j* (a), on y applique la rotation d'angle θ (b), suivie de la translation du centre du repère (c).

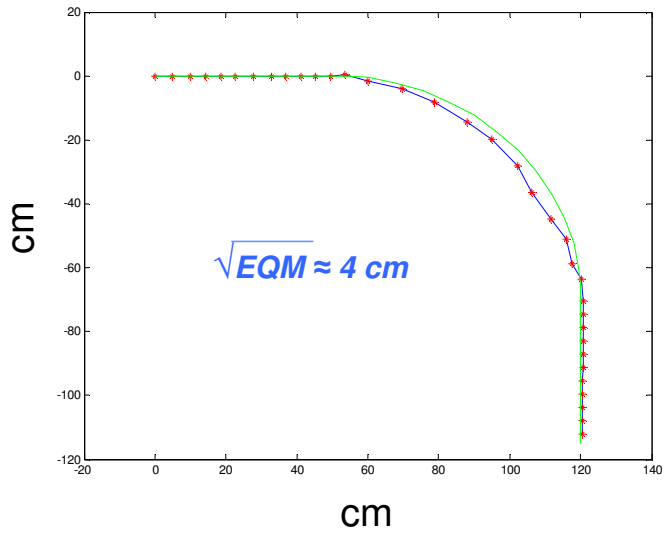
VI. Résultats

Ayant développé la méthode décrite dans la partie précédente avec le langage Matlab, des essais ont eu lieu sur des séquences d'images virtuelles et réelles. L'intérêt d'utiliser des images virtuelles réside dans le fait qu'elles sont dénuées de bruit, de vibrations du capteur, etc, bref des conditions qu'il est difficile d'atteindre avec des images réelles. Aussi, les premiers essais ont eu lieu sur des images de synthèse dans des conditions similaires au cas réel, du moins pour ce qui est l'architecture du système, à savoir que la caméra est pointée vers le plafond et qu'elle se déplace sur un plan parallèle. La Figure 15 présente le type d'image qui a été utilisé. Ces images ont été générées par Pov-Ray.

Un autre intérêt des séquences d'images virtuelles est qu'on sait parfaitement la position de la caméra à chaque prise de vue ce qui permet d'évaluer l'erreur d'estimation. Pour ce faire, on estime la trajectoire, *i.e.* les positions relatives de chaque prise de vue, selon le procédé présenté dans les parties précédentes de ce document. Dans un second temps, on calcule l'écart quadratique moyen entre les positions estimées et les positions « réelles ». Ceci nous permet d'évaluer l'erreur commise en unité métrique puisque l'homographie estimée dans l'espace pixellique est modifiée par la matrice des paramètres intrinsèques de la caméra pour être applicable dans l'espace normalisé du repère caméra.



(a)

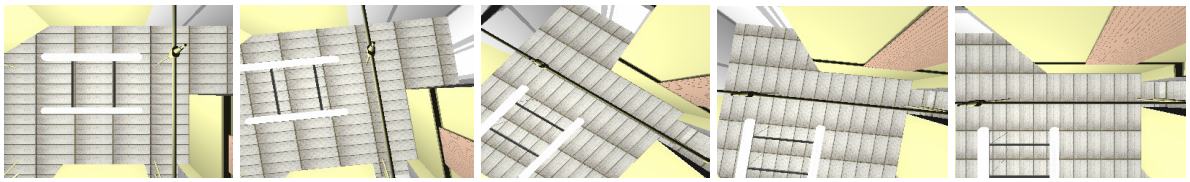


(b)

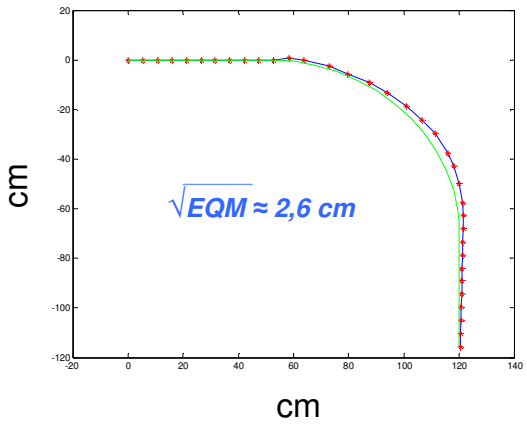
Figure 15 : Résultats (b) d'estimation de trajectoire sur une séquence de 35 images virtuelles (rectiligne, courbe, rectiligne) sans rotation de la caméra (a) d'un déplacement total de 120 cm en X et 120 cm en Y. On note la proximité de l'estimation (courbe bleue) et des positions réelles (courbe verte) puisqu'un écart quadratique moyen sous le centimètre est enregistré.

Ces résultats sont intéressants mais même si la trajectoire n'est pas rectiligne, la caméra ne subit aucune rotation. L'approche est certes limitée mais l'erreur engendrée est faible et montre la robustesse de la méthode dans un cas où la caméra ne subit que des translations.

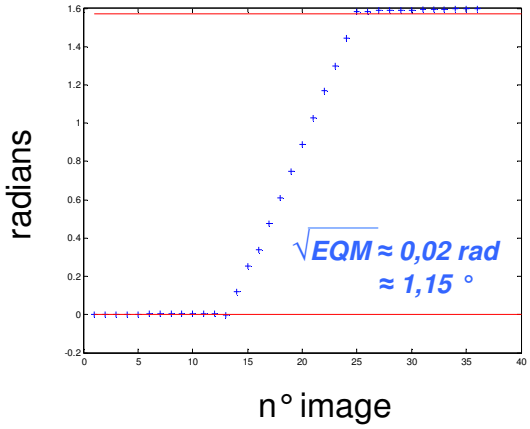
Mais comme il a été présenté précédemment, la méthode a été prévue pour fonctionner aussi bien dans le cas de translations, que de rotations, que d'un mélange des deux. Aussi, la Figure 16 montre une trajectoire dont les positions sont du même ordre que précédemment mais où le passage courbe est en fait le résultat d'un virage où cette fois-ci la caméra est en rotation.



(a)



(b)



(c)

Figure 16 : Estimation de trajectoire sur séquence de 35 images virtuelles (a) comprenant translations et rotations pour la trajectoire courbe engendrant un déplacement de la caméra de 120 cm en X et 120 cm en Y au total. (b) montre la superposition des véritables positions (en vert) et de la trajectoire estimée (en bleu). On note une erreur plus importante que dans le cas de translations pures. Néanmoins, elle reste faible puisque inférieure à 3 cm pour l'EQM et inférieure à 4 degrés pour l'orientation (c) avec un premier palier à zéro degré suivi d'un virage qui fait approximativement atteindre un palier à π/2.

Passons maintenant au cas de la séquence d'images réelles dont la trajectoire est plus longue avec plus d'images et dans laquelle on constate des vibrations de la caméra. La Figure 17b présente la trajectoire théorique telle qu'on pourrait se la représenter mais pour laquelle la vérité terrain, *i.e.* les positions exactes du robot à chaque prise de vue, est inconnue. Mais en regardant la video, l'œil humain peut bien se rendre compte que le chemin parcouru par le robot est tout d'abord rectiligne, vire à droite à angle droit, redevient rectiligne sur une distance importante durant laquelle des vibrations sont constatées, puis fait demi-tour dans le sens direct jusqu'à revenir au premier virage mais dans le sens inverse et continuer de manière rectiligne presque jusqu'au point de départ.

L'allure générale de la trajectoire est respectée. De plus, les orientations correspondent à ce à quoi on s'attendait à savoir un palier à zéro degrés (pas encore de rotation enregistrée), un virage à droite, un palier à $-\pi/2$, un demi tour suivi d'un palier à $\pi/2$ et enfin, le dernier virage qui mène l'orientation à π pour le retour au point de départ. Néanmoins, de petites erreurs persistent, ce qui va être développé dans la section suivante.

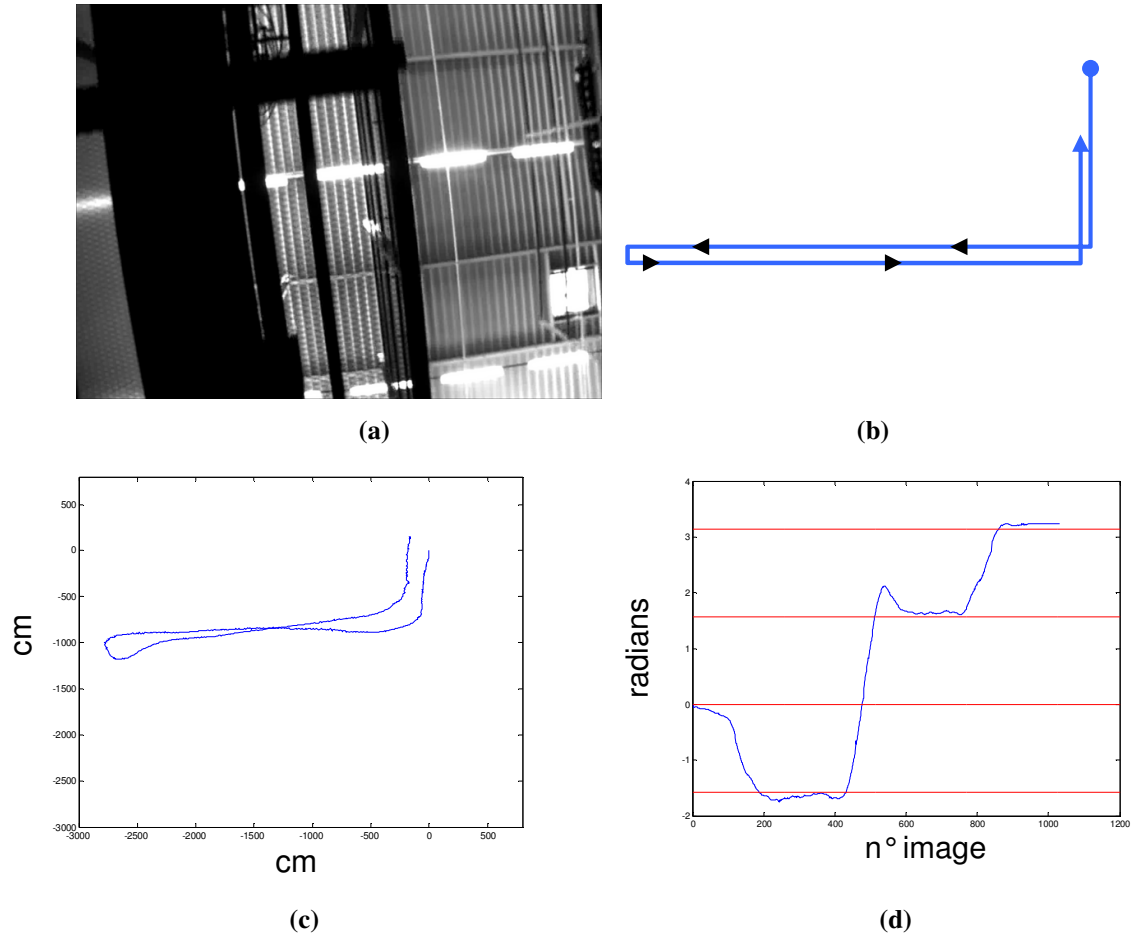


Figure 17 : Résultat d'estimation sur des images réelles (a). (b) présente schématiquement l'allure générale de la trajectoire suivie par le robot (donc la caméra). (c) et (d) sont les résultats de l'estimation. (c) présente la courbe des positions successives de la caméra et (d), l'orientation du robot en fonction des images avec les paliers (en rouge).

VII. Discussion

VII.1. Discussion sur les résultats

Comme il est possible de le voir dans la section précédente, les résultats sont assez proches de la réalité. Cependant, même si l'estimation de trajectoire est très précise dans le cas de translations pures, quand il y a des rotations, des imprécisions plus importantes apparaissent comme on le constate particulièrement sur la Figure 17c qui présente la trajectoire estimée sur la séquence d'images réelles. L'ennui, c'est que, s'il y a une imprécision trop importante à chaque estimation, au final, l'imprécision sera très importante mais le problème réside sûrement, en partie du moins, dans le fait que les points d'intérêt sont détectés dans l'espace pixellique et non sous-pixellique, tout comme l'appariement et donc l'estimation de la transformation se fait à partir de points en coordonnées pixelliques et non sous-pixelliques. Le passage du pixel au sous-pixel permettrait sûrement d'améliorer de façon significative les estimations de positions.

La sous partie suivante présente les travaux futurs et par là même un éventuel panel de solutions au problème constaté.

VII.2. Propositions pour la suite des travaux

Même si les résultats sont assez bons, il reste quelques problèmes dans l'estimation du déplacement, probablement à cause de la non parfaite coplanarité des points servant à estimer l'homographie. Le temps a manqué pour aller plus avant dans les développements mais une idée pourrait servir de base aux travaux futurs.

Actuellement, pour estimer l'homographie, les points sont filtrés par une homographie estimée itérativement en minimisant l'erreur de reprojection et en éliminant les outliers dans le but de ne garder qu'un ensemble de points coplanaires. Cependant, en utilisant la décomposition tridimensionnelle de l'homographie, et notamment en calculant la normale au plan observé, il serait intéressant, tout en minimisant l'erreur de reprojection pour éliminer les outliers, d'intégrer une contrainte selon laquelle les correspondances gardées (les inliers) permettent l'estimation d'une normale caractérisant un plan parallèle à la caméra. Mais il serait sans doute plus judicieux de développer cette idée selon une approche RANSAC qui sélectionnerait aléatoirement des correspondances dont l'homographie estimée ne serait validée que si le plan (par sa normale) formé par les correspondances est proche d'être parallèle à celui de la caméra. Ensuite seulement, la suite de RANSAC serait appliquée à savoir le vote selon lequel l'homographie planaire choisie convient à un maximum de points.

Si RANSAC a un inconvénient, c'est justement que les points sont choisis aléatoirement, ce qui rend difficile l'évaluation du temps d'exécution. Aussi, et un peu à la manière de , où les images sont découpées en zones pour estimer une homographie par zone et ne rassembler deux zones que si leurs homographies sont proches dans le but de détecter les plans de l'image, on pourrait, une fois les points d'intérêt détectés, les regrouper selon leur proximité, pourquoi pas en effectuant une classification hiérarchique. Ensuite il faudrait calculer une homographie planaire pour chaque groupe et rassembler deux groupes si leurs homographies sont proches. Cette idée repose sur la constatation que les points détectés dans une image ont plus de chance d'être coplanaires s'ils sont proches, un objet planaire pouvant, par exemple, avoir une surface de 5000 pixels seulement. Un autre cas envisageable est qu'on ait une structure plane dans l'image dont une partie est masquée de telle sorte qu'il n'y ait aucun lien visuel entre deux parties de cette structure plane. Cette approche permettrait de lier deux groupes de points aux antipodes d'une image tout en éliminant les outliers.

VIII. Conclusion et perspectives

Ce document a eu pour but de faire part d'une méthode d'estimation de trajectoire pour un robot mobile en environnement intérieur en utilisant d'une façon particulière la trame récurrente « détection de primitives visuelles dans les images, appariement de ces dernières, estimation de la transformation et extraction des paramètres de mouvement de la caméra ». Il a aussi eu la tâche de rassembler les bases théoriques nécessaires à la compréhension du problème ainsi que les travaux récents les plus proches du cas d'étude présent. Les algorithmes ont été implémentés et testés sur des séquences virtuelles et réelles dont les résultats ont été présentés et discutés et ont globalement montré que la réalisation de l'outil est en bonne voie.

Nous pouvons dors et déjà conclure que l'approche du problème de localisation de robot en environnement connu par odométrie visuelle a son intérêt. Les premiers résultats montrent qu'il est possible de développer cette approche. Cependant, le travail n'est pas terminé car il serait très intéressant de développer l'application dans un langage plus performant que Matlab, comme le C ou le C++ par exemple qui engendrerait une baisse monumentale des temps de calcul. Ceci a pu être observé par l'utilisation de la librairie OpenCV en langage C, pour la détection des points d'intérêt, qui a permis de diviser le temps de calcul par dix pour cette étape. Mais cela serait-il suffisant ? Si la réponse est non, il faudrait probablement s'orienter vers des architectures plus spécifiques qui permettent de paralléliser massivement les traitements comme le GPU (Graphics Processing Unit : carte graphique). Cette unité, habituellement dédiée au traitement de maillages pour le rendu 3D d'images de synthèse, voit sont utilisation détournée depuis quelques années, notamment pour le traitement d'images. Pour juger de l'intérêt d'utiliser cette architecture, l'algorithme du détecteur de Harris a été développé et a permis de gagner encore du temps sur une carte bas de gamme. Avec la nouvelle génération de GPU qui est en train d'apparaître et dont l'architecture est mieux pensée pour des utilisations non graphiques et en utilisant des cartes haut de gamme, la réduction du temps de calcul pour atteindre le temps réel ne semble pas être insurmontable, loin de là.

Mais les véritables prochaines étapes sont la validation des estimations des positions en comparaison de relevés réels et l'implantation des concepts de ce document sur un robot pour évaluer le résultat réel.

Il reste cependant des points en suspend. Qu'en est-il de la robustesse de la méthode ? De sa précision ? Comment l'intégrer dans un processus de contrôle de trajectoire ? Comment éviter d'éventuels obstacles ou du moins le détecter pour arrêter le robot ? L'ensemble de ces questions, et bien plus, constitue la suite du projet car le travail présenté dans ce document n'est qu'une partie de la toute première phase du projet *VOG* dont le but est de déterminer le type de caméra le plus intéressant, entre la caméra perspective utilisée dans ce travail et la caméra omnidirectionnelle, pour la localisation de robot mobile. On peut néanmoins directement affirmer qu'il ne sera pas possible de détecter un objet ou un obstacle étant sous le plan de la caméra, son champ de vision n'étant pas assez important, dans le cas d'une caméra perspective. Par contre, la solution a ce problème est envisageable par la vision omnidirectionnelle mais elle apporte son lot de difficultés aussi.

En résumé, le travail réalisé pendant ce stage a été très enrichissant. D'une part, il a été nécessaire d'acquérir un grand nombre de connaissances pour bien situer le problème et envisager une solution, ce qui est toujours d'un grand intérêt pour avoir une meilleure capacité à affronter les problèmes futurs. D'autre part, en contrepartie des études théoriques et bibliographiques, l'élaboration de certains algorithmes et leur développement fut une phase importante afin de montrer que la direction envisagée n'était pas dénuée de sens, ni d'intérêt.

IX. Références

[1] E. Malis, F. Chaumette et S. Boudet, *2D 1/2 visual servoing*, Rapport de recherche de l'INRIA, RR-3387, 1998.

[2] A. Bartoli, P. Sturm et R. Horaud, *A Projective Framework for Structure and Motion Recovery from Two Views of a Piecewise Planar Scene*, Rapport de recherche de l'INRIA, RR-4070, 2000.

[3] M. Pressigout, *Approches hybrides pour le suivi temps-réel d'objets complexes dans des séquences video*, Thèse, IRISA, 2006.

[4] A. R. Varkonyi-Koczy, A. Rövid et E. Selényi, "A new corner detector supporting feature extraction for automatic 3D reconstruction", *SAMI* 2006.

[5] N. Pears et B. Liang, "Ground plane segmentation for mobile robot visual navigation", *Proceedings of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp 1513-1518, 2001.

[6] B. Liang et N. Pears, "Visual navigation using planar homographies", *ICRA*, pp 205-210, 2002.

[7] XU De, TU Zhi-Guo et TAN Min, "Study on visual positioning based on homography for indoor mobile robot", *Acta Automatica Sinica*, 31 (3): 464-469, 2005.

[8] D. Nistér, O. Naroditsky et J. Bergen, "Visual Odometry", *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, (1) pp 652-659, 2004.

[9] C. Harris et M.J. Stephens, "A combined corner and edge detector", *Alvey Vision Conference*, pp 147–152, 1988.

[10] S.M. Smith, "A new class of corner finder", *British Machine Vision Conference*, pp 139-148, 1992.

[11] W. Förstner, "A feature based correspondance algorithm for image matching", *Int. Arch. Photogramm. Remote Sensing*, (26), pp. 150-166, 1986.

[12] C. Schmid, *Appariement d'images par invariants locaux de niveaux de gris*, Thèse, INPG, Juillet 1996.

[13] G. Scott et H. Longuet-Higgins, "An algorithm for associating the features of two patterns", *Proc. Royal Society London*, (B244), pp. 21-26, 1991.

[14] B. Kwolek, "Visual odometry based on gabor filters and sparse bundle adjustment", ICRA, pp. 3573–3578, 2007.

[15] R. Hartley et A. Zisserman, *Multiple view geometry in computer vision*, seconde edition, Cambridge university press, 2003.

[16] M. A. Fischler et R. C. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography", *Comm. of the ACM*, (24) pp 381-395, 1981.

[17] Y.I. Abdel-Aziz et H.M. Karara, "Direct linear transformation from comparator coordinates into object space coordinates in close-range photogrammetry", *Proceedings of the Symposium on Close-Range Photogrammetry*, pp. 1-18, 1971.

[18] P. D. Sampson, "Fitting conic sections to very scattered data: An iterative refinement of the Bookstein algorithm", *Computer Vision, Graphics and Image Processing*, (18) pp 97-108, 1982.

[19] E. Malis et E. Marchand, "Méthodes robustes d'estimation pour la vision robotique", *Journées nationales de la recherche en robotique, JNRR'05*, France, 2005.

[20] P. K. Jain et C. V. Jawahar, "Homography Estimation from Planar Contours", *Third International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT)*, North Carolina, Juin 2006.

[21] Q. Luong et T. Vieville, « Canonical Representation for the Geometries of Multiple Views", *Computer Vision and Image Understanding (CVIU)*, 64(2) pp 193-229, 1996.

[22] K. Kanatani, *Statistical Optimization for Geometric Computation: Theory and Practice*. Elsevier Science, 1996.

[23] S. Kuthirummal, C. V. Jawahar et P. J. Narayanan, "Planar Shape Recognition across Multiple Views", *Proceedings of the International Conference on Pattern Recognition (ICPR)*, pp 482-488, 2002.

[24] S. Kruger et A. Calway, "Image Registration Using Multiresolution Frequency Domain Correlation", *British Machine Vision Conference (BMVC)*, pp 316-325, 1998.

[25] M. P. Kumar, C. V. Jawahar et P. J. Narayanan, "Geometric Structure Computation from Conics", *ICVGIP*, pp 9-14, 2004.

[26] R. Y. Tsai, T. S. Huang et W.-L. Zhu, "Estimating three-dimensional motion parameters of a rigid planar patch, II: Singular value decomposition", *IEEE transactions on acoustics, speech and signal processing*, (ASSP-30:4), pp 525-534, 1982.

[27] Z. Zhang et A. R. Hanson, "Scaled euclidean 3D reconstruction based on externally uncalibrated cameras", ISCV'95, p. 37, 1995.

[28] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints", *International Journal of Computer Vision*, 60, 2, pp. 91-110, 2004.

[29] G. Simon, "Automatic online walls detection for immediate use in AR tasks", ISMAR'06, pp 39-42, 2006.