Analyse de séquences d'images à cadence vidéo pour l'asservissement d'une caméra embarquée sur un drone

Benoit Louvat

INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

N° attribué par la bibliothèque

THESE

pour obtenir le grade de

DOCTEUR DE L'INPG

Spécialité : «Signal, Image, Parole et Télécoms»

préparée au Laboratoire GIPSA-lab

dans le cadre de l'École Doctorale «Électronique, Électrotechnique, Automatique, Télécommunications et Signal»

présentée et soutenue publiquement

par

Benoît Louvat

le 5 Février 2008

Titre :

Analyse de séquences d'images à cadence vidéo pour l'asservissement d'une caméra embarquée sur un drone

Directeur de thèse : Jean-Marc Chassery

JURY

M. Philippe Poignet	Président
M. Jacques Gangloff	Rapporteur
M. Eric Marchand	Rapporteur
M. Olivier Strauss	Examinateur
M. Jean-Marc Chassery	Directeur de thèse
M. Gérard Bouvier	Encadrant
M. Laurent Bonnaud	Encadrant
M. Nicolas Marchand	Encadrant

Nous reconnaissons les choses, nous ne les connaissons pas.
G. Deleuze, Proust et les signes

Remerciements

J'adresse de sincères remerciements à Jacques Gangloff et Eric Marchand pour l'intérêt qu'ils ont montré pour mon travail en acceptant de le relire.

Je remercie aussi Phillippe Poignet pour avoir accepté de présider mon jury ainsi que Olivier Strauss pour son rôle d'examinateur.

Je souhaite remercier mon directeur de thèse, Jean-Marc Chassery pour m'avoir accueilli au sein du LIS puis de GIPSA-lab. De la même manière, j'adresse de chaleureux remerciements à mes trois encadrants : Gérard Bouvier pour sa disponibilité, son implication et ses talents de pilote, Laurent Bonnaud et Nicolas Marchand pour leur encadrement technique et leurs conseils qui ont permis de grandement améliorer ce manuscrit. Chacun dans leur spécialité ont grandement contribué à cette thèse.

Un grand merci à Grégoire et Moussa qui m'ont accompagné dans le dur travail de laboratoire. Merci à Guillaume, Vincent, Pascal, Renée et Lola pour tous ces moments passés ensemble qui m'ont fait oublier que j'avais une thèse à rédiger. Merci à l'ensemble des personnes avec qui j'ai partagé un bon moment que ce soit dans les coursives du GIPSA-lab ou bien sous d'autres horizons. Je ne les nomme pas ici, mais ils se reconnaîtront. Merci aussi à ma famille qui a toujours été présente pour moi sans qui je ne serais pas en train d'écrire ces remerciements. Merci à ma Maman pour son soutien aussi bien moral que technique infaillible depuis toutes ces années. Enfin, un merci spécial pour Marie qui me supporte tous les jours et qui, malgré le propos incompréhensible de ce manuscrit, a fait l'effort de le lire et relire afin d'éliminer toute trace de faute d'orthographe, merci pour cela, merci pour tes encouragements et pour ta présence à mes côtés.

Table des matières

Table des figures

Introduction

LA ROBOTIQUE est un domaine très prisé en ce début de XXIe siècle. Le nombre de travaux de recherche traitant de ce sujet est de plus en plus important. Les applications robotiques connaissent un grand engouement dans l'industrie. Le nombre de robots dans les milieux professionnels mais aussi personnels grandit chaque année. Les premières applications utilisant des robots étaient assez rudimentaires, le robot était programmé pour faire une tâche bien précise, il n'avait aucune interaction avec son environnement, ni aucune capacité d'adaptation. Au cours des années, les robots se sont perfectionnés et aujourd'hui, les robots sont des outils de plus en plus autonomes, ils sont capables d'effectuer des tâches de plus en plus complexes comme la navigation autonome dans des milieux hostiles, la marche bipède ou encore le choix, la préemption et l'utilisation d'outils. On peut assimiler un robot a un système automatique à contrôler. La robotique est donc une branche de l'automatique. La commande en robotique est un sujet bien couvert mais la complexité de certaines tâches nécessite des capteurs ayant un fort potentiel de mesures afin de pouvoir fournir des informations pertinentes pour développer des lois de commande permettant de réaliser la tâche. En effet, la principale difficulté pour réaliser des tâches complexes et donner une certaine autonomie aux robots est de disposer de capteurs fournissant assez d'informations. Un des capteurs fournissant le plus d'informations sont les caméras vidéo. Au cours des dernières décennies, les caméras vidéo sont devenues des capteurs importants dans le domaine de la robotique. L'ajout de la vision dans la boucle de commande des robots a permis de développer des lois de commande prenant en compte plus d'informations, apportant la plupart du temps une autonomie et une interaction avec l'environnement plus importantes. A l'instar des hommes, le robot a appris à voir. La caméra fournit des images à partir desquelles il est possible d'extraire un grand nombre de mesures comme la position du robot dans l'espace, la vitesse de déplacement du robot, la distance par rapport à des objets d'intérêt, mais aussi des mesures plus sémantiques sur l'environnement comme par exemple le type d'objet présent dans le champ de vision du robot. Ce domaine, réunissant à la fois la vision et la commande, est nommé l'asservissement

visuel.

Dans cette thèse, nous développerons un système d'asservissement visuel pour notre matériel expérimental : une caméra embarquée sur un drone. Notre objectif est de commander la caméra en utilisant uniquement les images filmées par celle-ci. Ceci afin de réaliser des tâches telles que le suivi d'objets fixes au sol quels que soient les mouvements du drone. Une grande partie des travaux menés s'inscrit dans un contexte expérimental, c'est pourquoi, nous nous attacherons tout au long de ce document à respecter les différentes contraintes imposées par notre système et à proposer des solutions en adéquation avec celles-ci. En effet, notre contexte expérimental nous oblige à prendre en compte un ensemble de contraintes tant au niveau de l'analyse d'image qu'au niveau de la commande. Contraintes qui viennent s'ajouter aux contraintes habituelles en asservissement visuel telle que la nécessité de limiter le temps d'analyse d'image pour pouvoir mettre à jour la commande le plus souvent possible. De plus, notre drone ne pouvant embarquer une électronique d'un poids important, nous nous plaçons dans un contexte de système embarqué. Notre approche a été de développer une solution permettant de réaliser les objectifs fixés, à partir de notre matériel expérimental.

La plupart des études en asservissement visuel s'intéressent :

- Soit au développement de nouvelles lois de commande en utilisant un environnement très simple permettant de ne pas se soucier de l'extraction des informations dans l'image,
- Soit à la partie vision en s'intéressant à la mesure d'information dans l'image et en proposant de nouveaux algorithmes d'analyse d'image généraux sans se préoccuper de l'intégration de ceux ci dans une boucle de commande.

Ici, nous nous intéresserons à la fois à l'extraction de l'information dans l'image et à la commande du système. En raison de l'étude simultanée des deux thématiques composant l'asservissement visuel, nous proposons une nouvelle commande permettant une plus grande imbrication entre la loi de commande et l'analyse d'image.

Le manuscrit se décompose en trois chapitres dans lesquels nous présenterons le contexte de ce travail et développerons l'ensemble des travaux réalisés. Le plan du document est le suivant :

- Le chapitre 1 où l'on parlera du contexte scientifique et expérimental dans lequel s'inscrit ce manuscrit. Le contexte scientifique présentera un état de l'art en asservissement visuel dans lequel nous donnerons le classement classique des différentes structures d'asservissement visuel. Les critères pour les discriminer seront l'espace de mesure ainsi que le lieu où est implantée la loi de commande du robot. A partir de ce classement, nous

décrirons rapidement les asservissements visuels 2D et 3D ainsi que les asservissements visuels hybrides basés sur les différents types de mesure. Nous décrirons aussi l'asservissement visuel indirect et l'asservissement visuel direct basé sur une structure de contrôle différente. Puis, nous présenterons quelques tâches d'asservissement visuel et quelques systèmes expérimentaux. Ensuite, nous présenterons notre contexte expérimental afin d'avoir une vue d'ensemble de notre problématique, nous détaillerons les différents matériaux utilisés dans notre système.

- Le chapitre 2 où l'on parlera de la partie correspondant à la thématique vision du travail exposé dans ce mémoire. Dans ce chapitre, nous ferons un bref état de l'art en estimation du mouvement apparent dans l'image. Nous discuterons des contraintes imposées par notre système pour ce qui est de l'analyse d'image et confronterons à notre problématique les différentes techniques présentées dans l'état de l'art. Nous proposerons une approche basée sur une estimation globale et un raffinement local permettant de réaliser une tâche de suivi d'objets quelconques. Nous montrerons la pertinence de notre algorithme par des résultats quantitatifs effectués en laboratoire et des résultats qualitatifs effectués sur le terrain. Nous introduirons une autre approche basée sur des points invariants permettant d'améliorer l'approche précédente et de développer une nouvelle tâche de reconnaissance d'objets perdus lors du suivi.

- Le chapitre 3 où l'on parlera de la partie correspondant à la thématique automatique du travail exposé dans ce manuscrit. Dans ce chapitre, nous ferons un bref état de l'art de la commande en asservissement visuel. Nous nous appuierons sur les lois de commande existantes pour développer une loi de commande basée sur une double boucle d'asservissement afin de réaliser le suivi d'objet. Cette loi de commande assure une bonne réactivité tout en respectant les contraintes liées aux systèmes embarqués. Dans ce chapitre, nous présenterons aussi une approche innovante pour commander notre système. Nous proposerons une nouvelle loi de commande que nous avons appelée commande par sur-échantillonnage. Cette approche permet de commander le système avant la convergence de l'algorithme d'analyse d'image. Nous montrerons des résultats démontrant la pertinence d'une telle approche. Ensuite, nous nous intéresserons à améliorer la commande de notre système et en particulier, aux différents problèmes de non-linéarités et de quantification en proposant différents types de contrôleurs pour la boucle de vision.

Chapitre 1

Contexte

Sommaire

1.1 Introduction

LE PRÉSENT TRAVAIL s'inscrit dans un contexte applicatif portant sur l'analyse de séquences d'images acquises par une caméra montée sur un drone et la commande d'une tourelle permettant d'orienter cette caméra. Plus précisément, nous nous sommes intéressés au couplage vision/commande afin de commander une tourelle à l'aide des informations fournies par l'analyse de séquence vidéo. De nombreux travaux existent sur le couplage/vision commande communément appelé asservissement visuel. Ce couplage entre une caméra et un robot (dans notre contexte une tourelle) permet de déplacer celui-ci d'une position courante à une position désirée vis-à-vis de la scène observée. Le couplage vision/commande n'est pas toujours utilisé pour positionner une caméra à l'aide d'une tourelle. Les techniques de stabilisation gyroscopique sont aussi utilisées. Leur principal inconvénient est le poids du système comprenant l'ensemble des capteurs nécessaire à la stabilisation. De plus, l'espoir de miniaturiser de tels capteurs est minime. Dans notre contexte où le matériel est embarqué dans un drone, le poids et la taille de l'électronique sont des aspects importants qui rendent inadapté ce type de méthode. L'avantage du couplage vision/commande par rapport aux autres techniques est que le capteur caméra possède une grande richesse d'information, l'utilisation de celui-ci possède un avantage indéniable pour la réalisation de tâches complexes demandant une grande précision. Pour résumer, les techniques utilisées relèvent de la recherche en asservissement visuel, ou plus précisément, comme le laisse présupposer cette dénomination, sur la commande d'un système en boucle fermée, ici une tourelle, grâce à des informations visuelles extraites des images acquises par la caméra à l'aide d'un algorithme d'analyse d'image adéquat.

Dans les paragraphes suivants, nous ferons une présentation des différentes classes de méthodes d'asservissement visuel, nous nous intéresserons tout d'abord à l'asservissement visuel indirect et à l'asservissement visuel direct. Puis, nous détaillerons les asservissements visuels 2D et 3D. Ensuite, nous présenterons quelques tâches d'asservissement visuel, en particulier pour des robots mobiles. Nous montrerons un rapide tour d'horizon des différentes applications utilisant comme support des drones. Ensuite, nous présenterons notre contexte, détaillerons notre système expérimental sur lequel l'ensemble des travaux menés dans ce manuscrit seront validés. Enfin, nous conclurons, en précisant les contributions que nous avons apportées.

1.2 Contexte scientifique : état de l'art

Un nombre très important d'approches de couplage vision commande a vu le jour. Ces approches peuvent être différenciées selon l'utilisation faite de la caméra, embarquée ou déportée, mais aussi suivant l'utilisation faite des informations visuelles, de leur nature, de l'objet

observé, plan ou non, fixe ou mobile, de la connaissance ou non d'un modèle de cet objet, conduisant à autant d'approches possibles. Les approches les plus classiques seront décrites ci-après. La figure 1.1 représente un robot avec une caméra embarquée (cas a) et un robot avec une caméra déportée (cas b). Nous restreindrons notre description au cas d'une caméra embarquée, la transcription au cas d'une caméra déportée pouvant généralement se déduire du cas précédent. Néanmoins, nous invitons le lecteur intéressé par ce type de configuration à se référer à [HCM95, Dor95, HDE98, RH99], voire à [Fla01] pour la coopération entre une caméra déportée et une caméra embarquée, ou encore [Kru03, GGdM+04] pour la robotique chirurgicale.

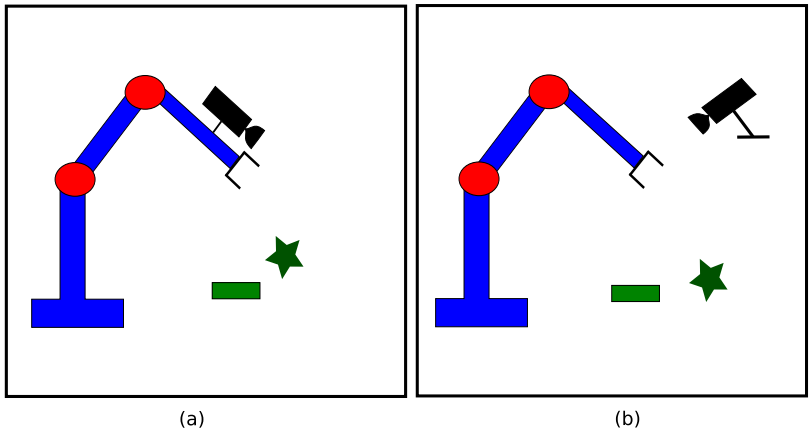

Fig. 1.1 – Couplage entre un robot et une caméra embarquée (a) et déportée (b)

Le principe de l'asservissement visuel consiste à prendre en compte des informations visuelles issues d'une ou plusieurs caméras dans la boucle de commande d'un robot afin d'en contrôler le mouvement. Le simple fait de l'inclure dans une boucle de commande permet d'augmenter significativement la précision du robot. Les premiers travaux [Ros76] et [TAO77] relatant l'interaction commande et vision fonctionnaient sur le principe de la boucle ouverte. On les retrouve sous la dénomination *look then move*. On peut les résumer ainsi : le robot se déplaçait, le capteur vision renvoyait l'information sur sa position dans la scène observée, puis, le cas échéant, le robot se déplaçait de nouveau suite à la modification de sa position dans cette même scène. La précision obtenue dépendait directement du capteur vision, mais aussi des asservissements articulaires, ou encore de la justesse des modèles utilisés et des phases de calibration. De plus, les types de commande développée dans ces études, de par leur simplicité, se révélaient efficaces uniquement dans le cas où la scène observée était statique. Nous

reviendrons plus en détails sur la commande de ce type d'asservissement dans le Chapitre 3.

Depuis, de nombreuses recherches ont amené de nouveaux développements en asservissement visuel. On trouve une littérature abondante exposant des études permettant la réalisation de tâches aussi complexes que le suivi ou la préhension d'objets mobiles, ou la modification de la trajectoire d'un robot mobile en cas d'obstacle, et cela, de plus en plus, à l'aide d'un ensemble robot/caméra partiellement ou non calibré.

La première utilisation de la vision en boucle fermée est due à Shirai et Inoue [SI73]. Dans leur approche, le capteur vision permettait un meilleur positionnement du robot. Mais l'apparition du terme asservissement visuel est due à Hill et Park [HP79]. Plusieurs méthodes ont ensuite vu le jour, elles ont été caractérisées par Sanderson et Weiss dans [SW80] en quatre grandes catégories, en fonction de l'utilisation des informations visuelles et du type de commande. Un premier critère basé sur la mesure dans l'image permet de distinguer les techniques dites d'asservissement en situation ou d'asservissement visuel 3D (en anglais *position-based control*), et les techniques dites d'asservissement basé image ou d'asservissement 2D (en anglais *image-base control*). Un autre critère basé sur la commande du robot est pris en compte dans la classification de Sanderson. Ce critère permet de distinguer les techniques dites d'asservissement visuel indirect (en anglais *dynamic look and move*) et les techniques dites d'asservissement visuel direct (en anglais *direct visual servo*). Depuis le classement de Sanderson, la recherche en asservissement visuel a beaucoup avancé et de nouvelles techniques découlant directement des précédentes sont apparues. Nous proposons de classer les techniques ne correspondant pas entièrement à la classification précédente dans une catégorie à part dite asservissement visuel hybride. Pour résumer, on pourrait classer l'ensemble des techniques d'asservissement visuel en trois groupes suivant trois critères :

1. Le calcul des articulations. Si le calcul des articulations se fait par l'intermédiaire du contrôleur du robot, c'est-à-dire que le contrôleur vison ne donne qu'un état de consigne au robot, alors, on parle d'asservissement indirect (*dynamic look and move*). Si par contre, le calcul des articulations se fait directement dans le contrôleur vision, c'est-à-dire que l'estimation de l'état du robot est réalisée dans le contrôleur vision, alors, on parle d'asservissement visuel direct (*direct visual servo*).

2. La mesure ou plus précisément l'erreur à minimiser. Si l'erreur est définie dans l'espace cartésien, alors, on parle d'asservissement en situation ou d'asservissement 3D (*position-based*). Si l'erreur à minimiser est définie directement dans le plan image, alors, on parle d'asservissement visuel 2D ou d'asservissement visuel basé image (*image-base*).

3. Lorsque les méthodes utilisent une combinaison des différentes approches exposées ci-dessus, en particulier en ce qui concerne la mesure, nous les classerons dans la catégorie d'asservissement hybride.

Le tableau 1.2 donne un aperçu de cette classification en fonction de la mesure et du contrôle du robot. Il est à noter que les asservissements visuels hybrides sont en général hybrides par leur mesure (par exemple à la fois 2D et 3D) et non par leur contrôle.

Critères	Contrôle direct	Contrôle indirect
Mesure 2D	Asservissement visuel 2D direct (figure 1.6)	Asservissement visuel 2D indirect (figure 1.4)
Mesure 3D	Asservissement visuel 3D direct (figure 1.5)	Asservissement visuel 3D indirect (figure 1.3)
Mesure hybride	Asservissement visuel hybride	

Fig. 1.2 – Classement des méthodes d'asservissement visuel

1.2.1 Asservissement visuel indirect / direct

Dans ce paragraphe, nous détaillerons la différence entre les asservissements visuels indirects et directs. La discrimination de ces deux types de technique se fait par rapport au premier critère (le calcul des articulations) de l'énumération du paragraphe 1.2.

1.2.1.1 Asservissement visuel indirect

Les techniques d'asservissement visuel indirect sont très abondantes dans la littérature [Cor93, CG96, HHC96, KV98]. Leur principal avantage est qu'elles permettent une commande articulaire découplée de l'algorithme d'analyse d'image et de sa limitation au niveau temps de calcul. En effet, dans l'asservissement visuel indirect, le calcul des commandes données aux articulations du robot pour satisfaire la tâche d'asservissement se fait dans un contrôleur interne au robot. Le système est divisé en deux boucles : une boucle de bas niveau pour le calcul de l'état des articulations du robot à l'aide du contrôleur interne au robot, cette boucle a une fréquence de l'ordre de 100Hz et une boucle de haut niveau comprenant le contrôleur vision fonctionnant à une fréquence inférieure car limitée par l'algorithme d'analyse d'image (typiquement 25 HZ).

Outre l'avantage de découplage entre la commande et l'analyse d'image, ces techniques présentent d'autres caractéristiques intéressantes. Beaucoup de robots possèdent une interface

acceptant des consignes en vitesse ou en position dans le plan cartésien, ce qui facilite leur mise en œuvre. De plus, dans ce type de méthode, le robot est considéré comme un outil de positionnement indépendant du système, simplifiant ainsi le contrôleur visuel qui n'a pas besoin de prendre en compte les singularités du robot. Tout ceci confère à l'asservissement visuel indirect des qualités telles que la robustesse, la simplicité ou encore l'adaptation du contrôleur visuel à différents types de robot. Les performances du contrôleur visuel dépendent uniquement de sa conception et du retard introduit par l'analyse d'image dans la boucle.

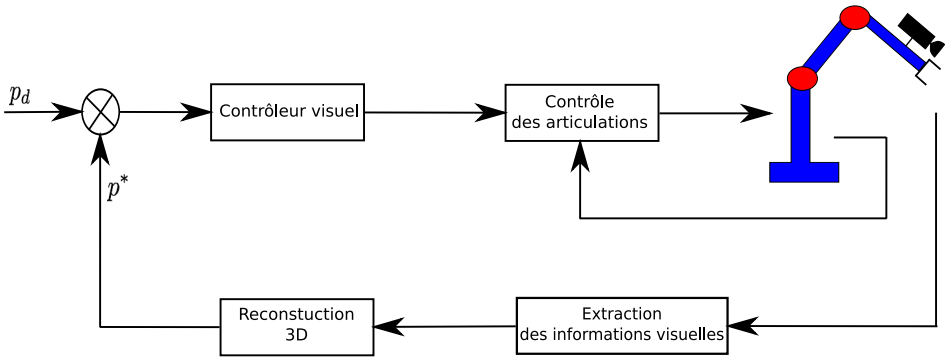

Fig. 1.3 – Schéma bloc d'un asservisement 3D indirect avec p une position 3D dans l'espace cartésien

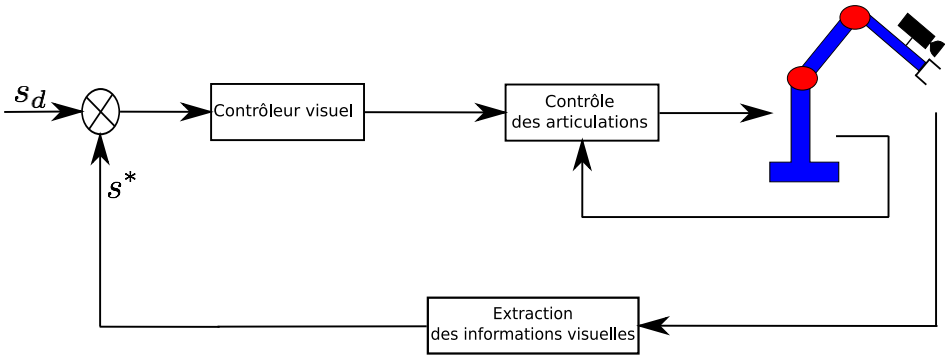

Fig. 1.4 – Schéma bloc d'un asservisement 2D indirect avec p une position 2D dans l'image

La mesure de l'erreur à stabiliser peut se faire dans l'espace cartésien ou dans le plan image. La figure 1.3 représente le schéma bloc d'un asservissement visuel indirect 3D. Dans ce cas, l'erreur est stabilisée dans l'espace cartésien (3D). La figure 1.4 représente quant à elle, le schéma bloc d'un asservissement visuel indirect 2D. Dans ce cas, l'erreur est stabilisée dans le plan image (2D). Nous détaillerons dans les paragraphes suivants la différence entre l'asservissement visuel 2D et 3D. Cette différence est la même dans le cadre d'un asservissement visuel indirect ou dans le cadre d'un asservissement visuel direct.

1.2.1.2 Asservissement visuel direct

Dans ce paragraphe, nous introduisons les méthodes d'asservissement visuel direct. Elles se différencient des techniques d'asservissement visuel indirect par le rôle du contrôleur visuel. Dans ces techniques [Cor96, GdM03, CGL+04], le contrôleur visuel prend en charge le calcul de toutes les articulations du robot afin de satisfaire la commande. L'état du robot est directement estimé dans le système de vision et le contrôleur vision se substitue au contrôleur bas niveau du robot, c'est-à-dire au contrôleur régulant les articulations du robot. Il n'y a plus de contrôleur dédié au robot, l'ensemble de la loi de commande se fait dans le même contrôleur. Il est évident qu'il est alors nécessaire de fournir une estimation de l'état du robot à une cadence élevée, ceci d'autant plus que la tâche à effectuer, nécessite une grande réactivité de la part du robot. Ce type de méthode se rapproche des méthodes d'asservissement classique de l'automatique.

La grande majorité des asservissements visuels se font selon ce schéma. En effet, l'augmentation de la puissance de calcul des ordinateurs ces dernières années permet d'avoir des algorithmes d'analyse d'image suffisamment rapides pour être embarqués dans une boucle d'asservissement visuel. De plus, l'augmentation de la fréquence d'acquisition des images par les caméras a permis d'obtenir des fréquences d'asservissement visuel très élevées, pouvant aller jusqu'à 1000Hz [NITM00].

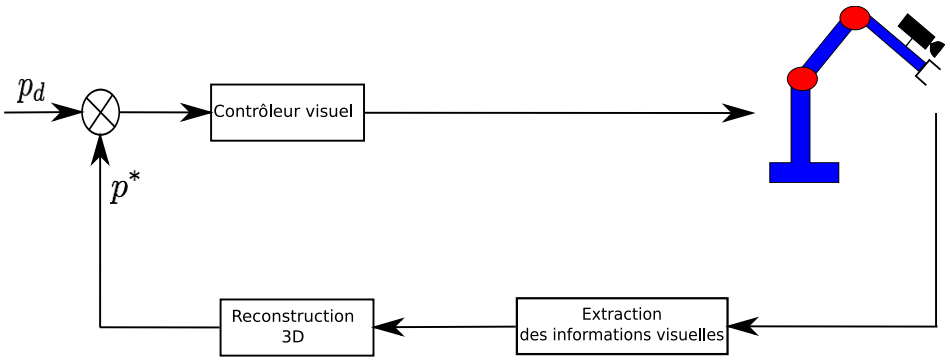

Fig. 1.5 – Schéma bloc d'un asservisement 3D direct avec p une position 3D dans l'espace cartésien

1.2.2 Asservissement visuel 2D/ 3D

Dans ce paragraphe, nous détaillerons la différence entre les techniques d'asservissement visuel 2D et 3D. La critère pour discriminer ces deux types de technique est la mesure utilisée dans la boucle d'asservissement (deuxième critère du paragraphe 1.2)

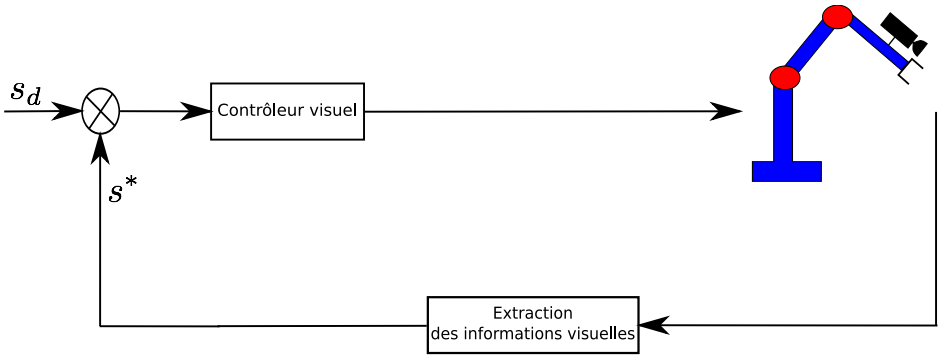

Fig. 1.6 – Schéma bloc d'un asservissement 2D direct avec s une position 2D dans l'image

1.2.2.1 Asservissement visuel 3D

Les techniques d'asservissement visuel 3D, par leur conception intuitive ont été les premières à voir le jour. La première implantation de ce type de méthode est l'œuvre de Shirai [SI73] et de Hill [HP79]. La fréquence de l'asservissement visuel était de l'ordre d'une dizaine de secondes, ce qui est assez loin de la fréquence d'un asservissement classique en automatique. Ces premiers travaux peuvent être considérés comme des pseudo-asservissements fonctionnant en boucle ouverte.

Les méthodes d'asservissement visuel 3D se basent sur une mesure 3D, impliquant une reconstruction 3D de l'information visuelle. C'est-à-dire une estimation à partir des informations visuelles extraites de la scène observée, de la position relative entre la caméra et un ou plusieurs objets de cette scène. En utilisant ces valeurs, on définit une erreur entre l'estimation de la position obtenue par reconstruction et la position désirée de l'objet dans l'espace de tâche. Cette erreur peut être définie *a priori* dans n'importe quel repère, l'espace de tâche n'étant pas restreint. Mais il existe des choix plus intéressants que d'autres. En effet, le choix d'un bon espace de tâche permet de découpler l'estimation de la position du robot et la loi de commande. Par exemple, le choix d'exprimer la position de la cible dans le repère de la caméra et non dans un repère lié à l'environnement permet d'éviter d'introduire dans l'estimation de la position le modèle géométrique du robot.

La plupart des travaux utilisant cette approche sont mono-caméra. Dans ce cas, la reconstruction de l'attitude de la cible (position et orientation) nécessite une connaissance *a priori* de la géométrie de la scène. Par exemple, le modèle CAO des objets observés ou les dimensions de la scène observée. Lorsque le système d'asservissement 3D dispose d'une tête stéréo, voire de plus de deux caméras, la reconstruction peut se faire sans aucune connaissance géométrique de la scène [RK96].

Les primitives géométriques utilisées dans ce type de méthode sont généralement relativement simples. La plupart des travaux se basent sur des primitives de type point [DD95] ou droite [AEH02]. Il existe quelques travaux utilisant des primitives plus complexes comme des coniques [Ma93] ou des sphères [DLPRR90]. Concernant le calcul de la situation de l'objet à partir de points, il peut être montré que trois points sont suffisants pour résoudre le problème. En effet, un point de l'objet fournit deux équations et la situation de l'objet par rapport à la caméra est entièrement définie par six inconnues (trois orientations, trois translations). Par contre, la résolution de ces équations ne fournit pas une solution unique. Toutefois, une solution unique est obtenue pour quatre points. De plus, dans [Yua89], il est démontré que l'utilisation de points supplémentaires non coplanaires améliore les résultats. Afin de résoudre ces équations non-linéaires, des méthodes itératives ont été proposées. Par exemple, dans [Low91] les auteurs proposent d'utiliser la méthode de Newton-Raphston avec comme principal inconvénient de nécessiter une estimation initiale.

Les droites ont été également très étudiées, par exemple, pour réaliser des tâches de suivi d'objets polyédriques [DC02], de sphères [CMC03] ou tout simplement pour faciliter le traitement d'image. Dans ce cas, le calcul de la situation de l'objet par rapport à la caméra se fait en utilisant trois droites, qui conduisent à la recherche des racines d'un polynôme de degré huit.

La figure 1.5 illustre le principe d'un asservissement visuel 3D direct. La figure 1.3 illustre un asservissement visuel 3D indirect. Dans les deux cas, p_d est un vecteur consigne représentant les coordonnées d'attitude désirées de l'objet par rapport à la caméra et p^* est le vecteur mesure estimé à l'aide d'un algorithme de reconstruction 3D.

La restriction à des objets simples, ainsi que les connaissances *a priori* nécessaires à la réalisation de ces techniques d'asservissement sont des limitations importantes. De moins en moins de travaux de recherche s'intéressent aux méthodes d'asservissement 3D, la plupart des travaux récents utilisent les techniques d'asservissement visuel 2D ou des techniques hybrides ne nécessitant pas une reconstruction 3D complète et étant donc beaucoup moins sensibles aux erreurs de calibrage. De plus, l'asservissement 3D ne satisfait pas certaines contraintes élémentaires comme la présence de l'objet dans le champ visuel. Par contre, on trouve des travaux basés sur l'asservissement visuel 3D dans la thématique de la réalité augmentée [GL04] où la pose d'un objet réel de l'image est estimé dans le repère caméra afin d'asservir un objet synthétique au mouvement de l'objet réel. Dans ce genre de travaux, on peut aussi citer l'asservissement visuel virtuel [MC02].

1.2.2.2 Asservissement visuel 2D

Les techniques d'asservissement visuel 2D utilisent une mesure 2D. L'idée est de ne plus utiliser une grandeur 3D reconstruite à partir des informations visuelles, mais d'utiliser directement les informations visuelles dans l'image 2D. Le premier à avoir introduit ce genre de technique est Weiss [Wei84]. Le but de ces méthodes est de faire converger les informations visuelles (primitives géométriques) s^* mesurées dans l'image vers les informations visuelles s_d désirées.

La clef de voûte de l'asservissement visuel 2D est la définition d'une matrice : le jacobien image. Dans le cas du formalisme de la fonction tâche, elle est appelée matrice d'interaction L_s^T. Il s'agit de l'interaction entre le mouvement relatif de la caméra par rapport à la scène observée représenté par un torseur cinématique T et la variation des mesures des primitives géométriques dans l'image représentée par un vecteur vitesse $\dot{s}$. On peut alors écrire la relation suivante :

$$\dot{s} = L_s^T T$$

avec comme fonction de tâche à minimiser, la différence entre les informations visuelles désirées et estimées :

$$e = s_d - s^*$$

Nous reviendrons en détails dans le chapitre 3 sur le jacobien de l'image et sur le formalisme de la fonction tâche. La plupart des travaux en asservissement visuel utilisent des primitives géométriques de type point pour calculer le jacobien de l'image. D'un point de vue pratique, ce genre de primitive est très avantageux. En effet, pour calculer le jacobien image, il est nécessaire que la scène observée contienne des informations visuelles permettant de le faire. C'est pourquoi les primitives ponctuelles sont utilisées dans bon nombre de travaux [Cha90, HHC96, MSCS06]. Même si la matrice d'interaction de telles primitives est parfaitement connue et très utilisée, il est possible de trouver une méthode générique pour l'obtention du jacobien image [ECR92]. Cette méthode permet d'obtenir une matrice pour des droites, des sphères, des cylindres. On trouve aussi des primitives beaucoup plus complexes comme par exemple les moments [Tah04].

Le principal avantage de ces techniques est le peu d'information nécessaire à leur réalisation. Dans le cas de primitives points, seuls la profondeur et les paramètres de calibration de la caméra sont nécessaires, mais dans la plupart des cas, une estimation grossière fonctionne très bien [Esp95]. Plus récemment, des méthodes dites d'asservissement visuel sans modèle ont été développées, dans ces cas là, aucun paramètre n'est connu à l'avance. Elles reposent sur l'identification en ligne de la matrice d'interaction [JFN97], ou sur différents types d'optimisation : dans [PML99] les auteurs proposent une optimisation de type quasi-newton pour un suivi de cible, le jacobien est estimé en ligne. Dans [MGdM02], les auteurs utilisent la

méthode du polyèdre flexible pour une tâche de positionnement sans estimation du jacobien. Les inconvénients des asservissements sans modèle sont la convergence de l'optimisation et le temps de calcul de cette optimisation.

Le principal inconvénient de l'asservissement visuel 2D est la présence de minima locaux lors de la minimisation de la fonction de tâche. Plus la position de départ est éloignée de la position à atteindre, plus le risque de converger vers un minimum local est grand. Dans ce cas, les commandes calculées sont nulles même si la fonction de tâche n'est pas nulle.

Un autre inconvénient inhérent aux asservissements visuel 2D est le problème connu sous le nom de problème *dφavance/recul* [Mal04]. Il est dû aux trajectoires rectilignes imposées par la commande de l'asservissement visuel 2D des primitives dans l'image. Il est représenté sur la figure 1.7. La position courante de la caméra correspond aux points A, B, C, D dans l'image et la position désirée correspond aux points A^*, B^*, C^*, D^*. Les points se déplacent suivant les flèches, or, un tel déplacement des points correspond à un retrait de la caméra suivant son axe optique, en théorie, jusqu'à l'infini. C'est principalement pour contourner ce genre de problème qu'ont été imaginées les méthodes hybrides (cf paragraphe 1.2.3).

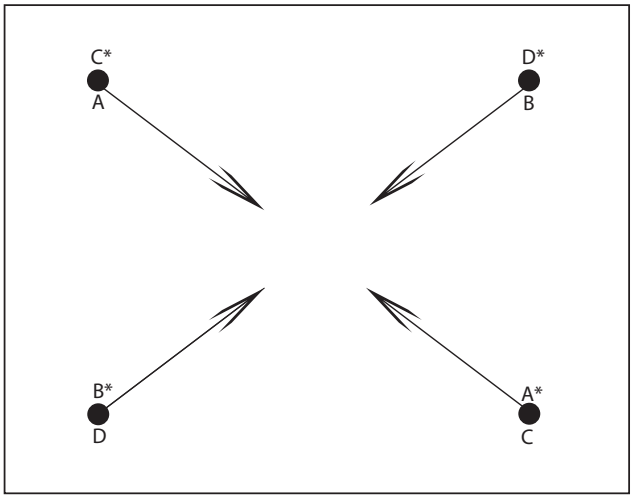

Fig. 1.7 – Illustration du problème d'avance/recul

La figure 1.6 illustre le principe d'un asservissement visuel 2D direct. La figure 1.4 illustre un asservissement visuel 2D indirect. Dans les deux cas, s_d est un vecteur consigne représentant les informations visuelles désirées dans l'image et s^* est le vecteur mesure des informations visuelles estimées à l'aide d'un algorithme d'analyse d'image.

1.2.3 Asservissement visuel hybride

La littérature en asservissement visuel hybride est très abondante. Le nombre d'études portant sur ce sujet a grandi énormément ces dernières années. Toutes les approches présentées ne sont pas toujours facilement classables. Dans cette partie, nous donnons quelques techniques hybrides d'asservissement visuel ne rentrant pas dans la classification précédente. Ce type de technique est apparu pour résoudre principalement les problèmes inhérents aux asservissements visuels 2D ou 3D. L'approche hybride peut en général se résumer à un mélange des grandeurs à asservir. Le plus souvent, les asservissements hybrides utilisent à la fois des mesures dans le plan image et des mesures dans l'espace cartésien ou encore dans certains cas des mesures de vitesse au lieu des mesures de position. Ce mélange des mesures se traduit généralement par un partitionnement de la commande. Le principe de l'approche hybride est donc d'utiliser à la fois les méthodes d'asservissement visuel en situation et les méthodes d'asservissement visuel basées sur l'image. Chaque technique étant utilisée au moment optimal pour optimiser la tâche d'asservissement.

1.2.3.1 Asservissement visuel 2D 1/2

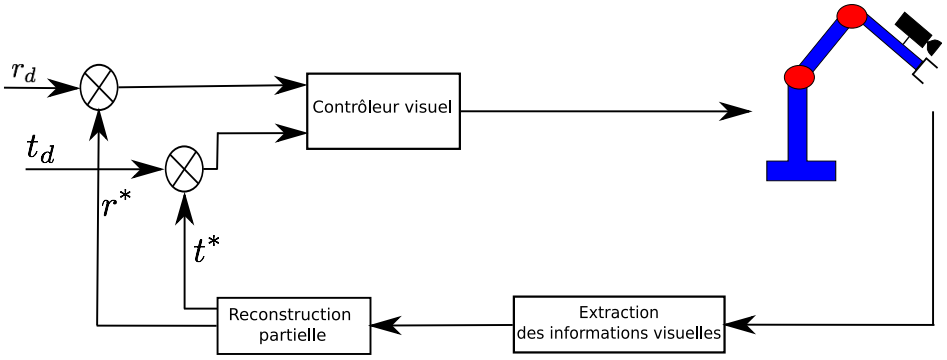

Fig. 1.8 – Schéma bloc d'un asservissement 2D 1/2

L'asservissement visuel 2D 1/2 développé par Malis [MCB99], est une technique d'asservissement visuel hybride basée sur des informations de mesures et de consignes définies à la fois directement dans l'image et dans le repère de la caméra. Ce type de technique a été utilisé pour réaliser des tâches de positionnement où la caméra est très éloignée de la position désirée [MCB98], cas où l'asservissement visuel 2D échoue. Plus récemment, l'asservissement visuel 2D 1/2 a été utilisé pour des tâches de suivi [MB05].

Plus précisément, l'idée de base est d'estimer l'homographie entre la situation du repère caméra désirée et la situation du repère caméra courante. Cette estimation est assez simple, par exemple dans le cas d'un objet plan, la mise en correspondance de quatre points sur

les images courantes et désirées permet l'estimation par l'intermédiaire de la résolution d'un simple système linéaire (pour un objet quelconque huit points sont nécessaires). A partir de cette homographie, le déplacement en rotation que la caméra doit réaliser pour rejoindre la position désirée est calculé. De plus, l'homographie permet d'estimer le rapport entre la distance courante de la caméra à l'objet et la distance désirée entre la caméra et l'objet ce qui permet de contrôler la translation le long de l'axe optique de manière à avoir ce rapport égal à un. Pour commander le robot en translation, l'asservissement 2D 1/2 utilise les informations 2D issues de l'image. En effet, l'homographie permet uniquement de connaître le déplacement en translation à un facteur d'échelle près. Pour résoudre ce problème, les coordonnées d'un point de l'objet sont utilisées pour contrôler le déplacement en translation du robot et ainsi garder l'objet au centre de l'image.

La figure 1.8 représente un asservissement visuel 2D 1/2 où r_d et t_d représentent respectivement le vecteur rotation désiré et le vecteur translation désiré et r^* et t^* représentent respectivement le vecteur en rotation mesuré et le vecteur en translation mesuré . Le principal avantage de cette technique est qu'elle ne nécessite que très peu d'informations *a priori* par rapport aux techniques classiques d'asservissement 3D . En effet, la convergence est assurée sans connaissance du modèle 3D de l'objet et avec uniquement une approximation de la profondeur désirée de l'objet.

Récemment, Benshimane et Malis [BM07] ont proposé une nouvelle technique d'asservissement visuel inspirée de l'asservissement 2D 1/2 mais ne nécessitant ni information 3D ni matrice d'interaction : l'asservissement visuel 2D basé homographie. Cette approche est basée sur la définition d'un isomorphisme entre la situation de la caméra et l'homographie estimée entre l'image courante et l'image désirée.

1.2.3.2 Asservissement visuel d2D/dt

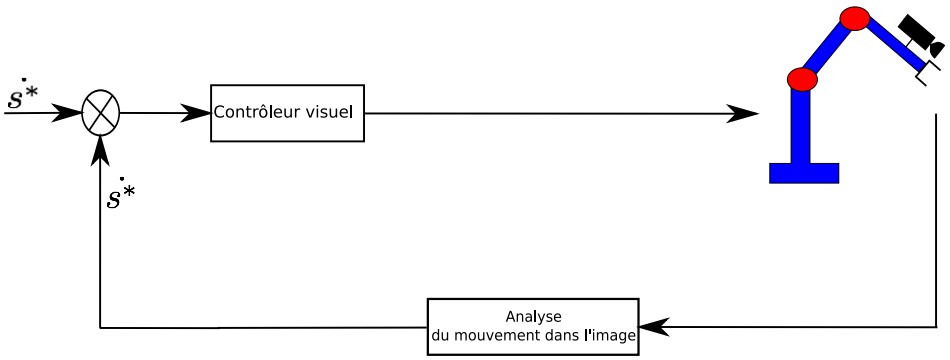

Fig. 1.9 – Schéma bloc d'un asservissement d2D/dt

Les techniques d'asservissement visuel 2D, 3D ou 2D 1/2 ont un point commun. Elles sont basées sur l'utilisation d'informations visuelles de type primitives géométriques. En effet, l'ensemble des méthodes présentées précédemment se basent sur une mesure de référence. Cette mesure de référence est la position de primitives géométriques dans l'image. L'inconvénient d'utiliser des primitives géométriques est, outre le fait qu'elles doivent être présentes dans l'image, la nécessité d'avoir à disposition des algorithmes d'extraction et de suivi de primitives géométriques robustes et précis. L'asservissement visuel $d2D/dt$ ou asservissement visuel dynamique n'utilise pas de primitive géométrique, il se base sur le champ de vitesse dans l'image. En effet, le calcul de la commande se résume à déplacer le robot afin que la vitesse du champ de mouvement mesurée corresponde à la vitesse du champ de mouvement désirée. La figure 1.9 représente un asservissement visuel d2D/dt avec $\dot{s_d}$ la vitesse du champ de mouvement désirée et $\dot{s^*}$ la vitesse du champ de mouvement mesurée.

Il est important de noter que dans ce type de système, la mesure des informations visuelles est dynamique et que ce sont ces mesures dynamiques qui sont directment utilisées dans la loi de commande. Cette approche a été proposée et validée par Cretual et Chaumette [CC01b]. Certaines études [QGS95, CAD95, SVS97] utilisent aussi des informations visuelles mesurées de façon dynamique dans l'image, c'est-à-dire par l'intermédiaire de l'analyse du mouvement dans l'image. Mais celles-ci reconstruisent l'information géométrique pour élaborer leur loi de commande. Dans ce cas, on peut rapporter ces techniques aux méthodes d'asservissement visuel 2D classique.

Dans les techniques d'asservissement dynamique, la loi de commande est élaborée directment à l'aide des mesures dynamiques dans l'image. Par exemple dans [CC01b], les auteurs se servent des paramètres d'un modèle de mouvement estimé dans l'image pour établir leur loi de commande. Le modèle utilisé dans le cas d'un objet plan peut s'écrire sous la forme :

$$\dot{x} = a_1 + a_3 x + a_5 y + a_7 x^2 + a_9 xy$$
$$\dot{y} = a_2 + a_4 x + a_6 y + a_8 y^2 + a_{10} xy$$

en partant de ce modèle, ils ont relié les paramètres a_i au mouvement du porteur. Plus précisément, ils ont relié les termes liés à la vitesse et à l'accélération du porteur. Ils ont ensuite développé des lois de commande en fonction de ces grandeurs. Cette méthode d'asservissement visuel a été validée pour de nombreuses tâches, comme le suivi de trajectoire parallèlement à un plan ou le suivi en rotation et une inclinaison d'un objet mobile afin de le garder au centre de l'image. Nous détaillerons la loi de commande de ce type d'asservissement dans le chapitre 3.

1.2.3.3 Autres types d'asservissement visuel hybride

Il existe un grand nombre de travaux sur les asservissements visuels hybrides. Dans les paragraphes précédents, nous avons détaillé les techniques d'asservissement visuel 2D 1/2 et d'asservissement visuel d2D/dt. D'autres méthodes d'asservissement visuel hybride existent, nous ne les détaillerons pas ici, mais on peut citer par exemple [CH01] qui propose un découpage de la commande selon les composantes des axes x et y et les composantes de l'axe optique z. Les grandeurs asservies sont dans le cas des axes x et y les mêmes que pour un asservissement visuel 2D, c'est-à-dire l'erreur de position entre les primitives (i.e. des points) désirées et courantes dans l'image. Par contre, pour les composantes de l'axe z, les grandeurs asservies correspondent à la surface des primitives pour le mouvement en translation et l'angle de rotation des primitives autour de cet axe pour le mouvement en rotation. Ce partitionnement de la commande et le découplage des mouvements qu'il entraîne permet de résoudre le problème d'*avance/recul* de la figure 1.7.

De nombreux travaux en asservissement visuel hybride se basent sur une approche 2D/3D. Par exemple dans [MBC99], l'estimation de la position de l'objet modélisé par une forme polyédrique se fait en deux étapes : une première étape consistant à faire une estimation du mouvement 2D du modèle de l'objet et une deuxième étape consistant, à partir de l'estimation du mouvement 2D fournie par la première étape, à estimer les paramètres 3D, c'est-à-dire ajuster le modèle CAO de l'objet.

Gans et Hutchinson proposent dans [GH02] une approche hybride 2D/3D permettant de commuter entre un asservissement visuel 2D et un asservissement visuel 3D afin de réaliser la tâche. La commande du système est divisée le long de l'axe de temps plutôt que le long des dimensions spécifiques de l'espace d'état. En utilisant les systèmes commutés hybrides, il peut être possible d'augmenter la région de stabilité, augmenter le taux de convergence et de commuter entre les systèmes instables dans un modèle afin de rendre stable le système entier [GH07].

1.2.4 Quelques tâches d'asservissement visuel

Dans ce paragraphe, nous exposerons quelques tâches d'asservissement visuel. Le nombre de travaux pléthoriques existant dans le domaine ne permet pas de faire un tour d'horizon complet de tout ce qui a été fait. Nous ferons simplement un bref descriptif de quelques tâches se rapprochant de nos travaux. Pour cela, nous nous intéresserons aux tâches de suivi de cible et aux robots mobiles.

1.2.4.1 Suivi de cible

Le principe du suivi de cible est de détecter un objet dans l'image et de suivre cet objet en fonction de son mouvement dans l'image. Le mouvement de l'objet dans l'image peut avoir deux origines :

- la caméra est fixe, alors le mouvement dans l'image est dû au mouvement propre de l'objet dans l'espace cartésien
- la caméra est montée sur un robot mobile. Dans ce cas, le mouvement de l'objet dans l'image peut être dû à son propre mouvement ou au mouvement du robot ou bien les deux.

Les tâches de suivi de cible peuvent se résumer ainsi : un objet est sélectionné dans l'image, il est suivi à l'aide d'un algorithme d'analyse d'image, sa position est envoyée au robot pour chaque image traitée et la commande déplace le robot en fonction de la position de l'objet. En général, la plupart des tâches de suivi ont comme objectif de garder l'objet au centre de l'image. Cette objectif est souvent réalisé à l'aide de techniques d'asservissement visuel 2D. La sélection de l'objet peut se faire de différentes manières : soit par une sélection en temps réel par l'utilisateur, soit par une reconnaissance d'un objet mémorisé lors d'une phase d'apprentissage ou encore par une détection des objets en mouvement dans l'image. Dans ce dernier cas, la détection est assez simple pour une caméra fixe : une simple différence d'images suffit dans la majorité des cas. Par contre, lorsque la caméra est mobile, la détection devient plus complexe. L'approche la plus utilisée est de détecter les mouvements incohérents par rapport au mouvement global.

La vitesse des objets dans l'image pouvant être très importante, l'algorithme d'image doit être rapide. Pour satisfaire cette contrainte, un grand nombre d'études utilise des cibles très simples, disposant parfois de marqueurs facilement identifiables dans l'image. Des travaux existent aussi sur le cas plus complexe où les objets sont réels. La principale difficulté dans ce contexte est la nécessité d'utiliser des algorithmes performants souvent gourmands en temps de calcul.

Pour la commande, la principale difficulté est d'annuler l'erreur de traînage entraînée par le mouvement de l'objet dans l'image. Une autre contrainte pour la commande apparaît lorsque l'objet se déplace rapidement. En effet, la plupart des travaux négligent la dynamique du manipulateur et de la boucle de vision. Si l'objet a un déplacement rapide, le robot se doit de réagir très rapidement et négliger sa dynamique entraîne la plupart du temps l'échec de la tâche de suivi.

La plupart des travaux se positionnent dans la première situation où la caméra est fixe. L'application première de ce type d'approche est la surveillance. Par exemple dans [CC01a], les auteurs proposent un suivi de piétons pour une caméra commandable en pan et tilt. Le piéton est détecté dans la première image à l'aide d'un algorithme d'analyse du mouvement global, puis, son déplacement est calculé entre chaque image et son centre de gravité est maintenu au centre de l'image. Afin de prendre en compte le déplacement propre du piéton et éviter les erreurs de traînage, la différence entre l'estimation du mouvement fournie par l'algorithme d'analyse d'image et le mouvement en rotation de la caméra mesuré par odométrie est filtrée par un filtre de Kalman à accélération constante. L'avantage est que le modèle utilisé pour le filtre de Kalman est simple et donc facilement implémentable. Par contre, un tel modèle ne décrit pas entièrement le mouvement du piéton et l'estimation perd en précision. Dans [Gan99], une tâche de suivi pour un robot à six degrés de liberté est proposée. Elle est basée sur une modélisation dynamique de la boucle de vision avec une consigne dans l'espace opérationnel du robot. Le correcteur utilisé est de type prédictif, il permet de considérer le mouvement de l'objet comme une perturbation à rejeter et donc de supprimer l'erreur de traînage.

Les systèmes à réalité augmentée se rapprochent des tâches de suivi de cible à la différence qu'ils ne commandent pas de robot. Le but est de contrôler la position dans l'image d'un objet de synthèse en fonction des mouvements d'objets réels. On peut citer [GL04] qui, à partir de points invariants dans l'image, estiment la position et la rotation d'un objet réel dans l'espace cartésien pour ensuite déplacer un objet de synthèse dans l'image en fonction de cette estimation de façon à ce que l'orientation et la position de l'objet de synthèse par rapport à l'objet réel reste les mêmes. Les travaux de Marchand et al [MC02, PM04] se rapprochent encore plus des tâches de suivi de cible robotique. En effet, ils proposent une technique d'asservissement visuel virtuel.

1.2.4.2 Les robots mobiles

Dans le paragraphe précédent, nous nous sommes intéressés à des tâches d'asservissement visuel dans le cas où le robot est immobile. Maintenant, nous allons présenter des tâches d'asservissement visuel dans le cas où le robot (le porteur de la caméra) est mobile. La principale difficulté dans l'utilisation de robots mobiles est leur mobilité, qui entraîne des changements importants dans la scène observée. De plus, dans le cas d'un suivi, le mouvement de la cible est dû à la fois au mouvement du robot et au mouvement de celle-ci et rendent plus difficile le suivi. Même dans le cas où la cible est fixe, le fait que le robot se déplace entraîne un déplacement de la cible dans l'image. Un autre problème est la difficulté d'avoir la position du robot dans un repère fixe en utilisant uniquement l'image. La conséquence de ces variations dans la scène pour les tâches de suivi est l'ajout de contraintes supplémentaires sur le déplacement du robot dans l'espace cartésien. Dans la suite, nous présenterons quelques tâches

d'asservissement visuel pour des robots mobiles et ferons un bref tour d'horizon sur les drones.

Navigation et suivi de cible pour les robots mobiles

Dans [BM05], l'application de suivi consiste à faire suivre une voiture électrique (meneuse) contrôlée par un pilote humain par une autre voiture électrique (suiveuse) contrôlée de façon automatique. Une affiche est adossée à la voiture meneuse et filmée par la voiture suiveuse. Une phase de calibration approximative est nécessaire pour avoir une reconstruction métrique. En plus, durant cette phase, une image de référence est sélectionnée dans l'affiche. Le suivi se fait en estimant l'homographie entre l'image de référence et l'image courante, la position et l'orientation dans l'espace cartésien relatives de la voiture suiveuse par rapport à la voiture meneuse sont obtenues en décomposant l'homographie estimée. Plus précisément, la décomposition correspond à l'extraction de la matrice homographique du déplacement en rotation et en translation de la caméra en fonction des paramètres intrinsèques de la caméra. En considérant que les deux voitures électriques roulent sur une surface plane, la commande à fournir à la voiture meneuse se résume à deux composantes en translation suivant les axes x et z et une composante en rotation autour de l'axe y. En fonction de cette décomposition, une commande est calculée afin de garder la voiture suiveuse à une distance constante de la voiture meneuse, cette distance étant préalablement initialisée.

Une grande partie des travaux en navigation de robot mobiles concerne les sous-marins semi-autonomes ou autonomes. Les principales applications sont la surveillance des canalisations sous-marines ou la cartographie du fond marin. Dans [CC00], une méthode de stabilisation du sous-marin en utilisant uniquement le mouvement dans l'image filmée est proposée. Dans [GSV00], une méthode de navigation de sous-marins autonomes est développée. Elle se décompose en deux opérations. La première opération est la création d'une mosaïque d'images représentant le fond marin. Cette mosaïque est construite à partir d'un ensemble d'images couvrant la zone d'intérêt, ces images sont filmées par un premier passage du sous-marin contrôlé manuellement. La deuxième étape est la navigation du sous-marin de façon automatique en utilisant la mosaïque précédemment créée. Une trajectoire est pré-calculée en utilisant comme représentation spatiale la mosaïque. Le contrôle du sous-marin se fait par l'intermédiaire de l'erreur de positionnement du sous-marin sur la mosaïque. Cette erreur est calculée en comparant l'image filmée courante à la mosaïque.

D'autres approches pour la navigation des robots mobiles s'inspirent directement de la nature. Par exemple dans [Ruf04], les auteurs utilisent les connaissances biologiques sur la technique de vol de la mouche pour construire un robot volant contrôlant son altitude en fonction de la vitesse du champ de mouvement apparent dans l'image.

Robots volants : drones

Un drone (ou en anglais un UAV pour *Unmanned Air Vehicule*) est un aéronef inhabité, piloté à distance, semi-autonome ou autonome. Récemment, les travaux de recherche concernant ce type d'engin ont crû très rapidement. La communauté des chercheurs en robotique s'est intéressée en particulier aux petits robots volants autonomes (hélicoptère, avion, hélicoptère quadri-rotor). L'application première de ces robots est la surveillance ou la reconnaissance semi-automatique de zones inaccessibles ou dangereuses pour une équipe humaine. On peut penser aux applications militaires mais aussi à des applications civiles comme la surveillance des glaciers ou des lignes électriques hautes tensions ou encore le contrôle du trafic routier. Cette notion de surveillance, s'accompagne d'un besoin d'avoir des informations visuelles de la zone survolée par le drone, c'est donc tout naturellement que les méthodes d'asservissement visuel pour des robots classiques se sont retrouvées dans les drones.

L'utilisation de la vision pour les drones est souvent une information supplémentaire pour contrôler le drone. En effet, la plupart des travaux portant sur le contrôle des drones n'utilisent pas l'information visuelle comme unique mesure, mais un ensemble de mesures issues de différents capteurs. A notre connaissance, il n'existe aucun drone commandé totalement automatiquement avec comme seul capteur une caméra. Par exemple, pour la navigation, l'information visuelle est souvent couplée avec une mesure GPS de la position du drone [Ami96]. Dans ce cas, la vision permet d'améliorer la précision du contrôle ou de réaliser des tâches spécifiques comme la stabilisation ou la poursuite de cible. A l'heure actuelle, la recherche sur les drones utilisant l'information visuelle se concentre sur les hélicoptères quadri-rotor et en particulier sur leur stabilisation en utilisant uniquement l'information visuelle [MSCS06, GHM07].

1.3 Contexte expérimental

Fig. 1.10 – Exemples d'images filmées

Le contexte de ce manuscrit s'inscrit dans le développement de tâches d'asservissement visuel pour un système drone. Nous nous sommes principalement intéressés à une tâche de suivi d'objets fixes et quelconques au sol, en proposant un système complet d'asservissement visuel.

Dans ce paragraphe, nous nous intéressons à notre plate-forme expérimentale qui se compose principalement d'un drone. Notre objectif est, à l'aide de cette plate-forme, de suivre des objets fixes quelconques au sol filmés par le drone. Deux exemples d'images sont exposés sur la figure 1.10. Dans l'image de droite, l'objet à suivre pourrait être la maison ou alors le parking. Dans l'image de gauche, l'objet d'intérêt pourrait être le croisement. Nous allons tout d'abord illustrer schématiquement cette plate-forme d'expérimentation.

1.3.1 Contexte de notre système expérimental

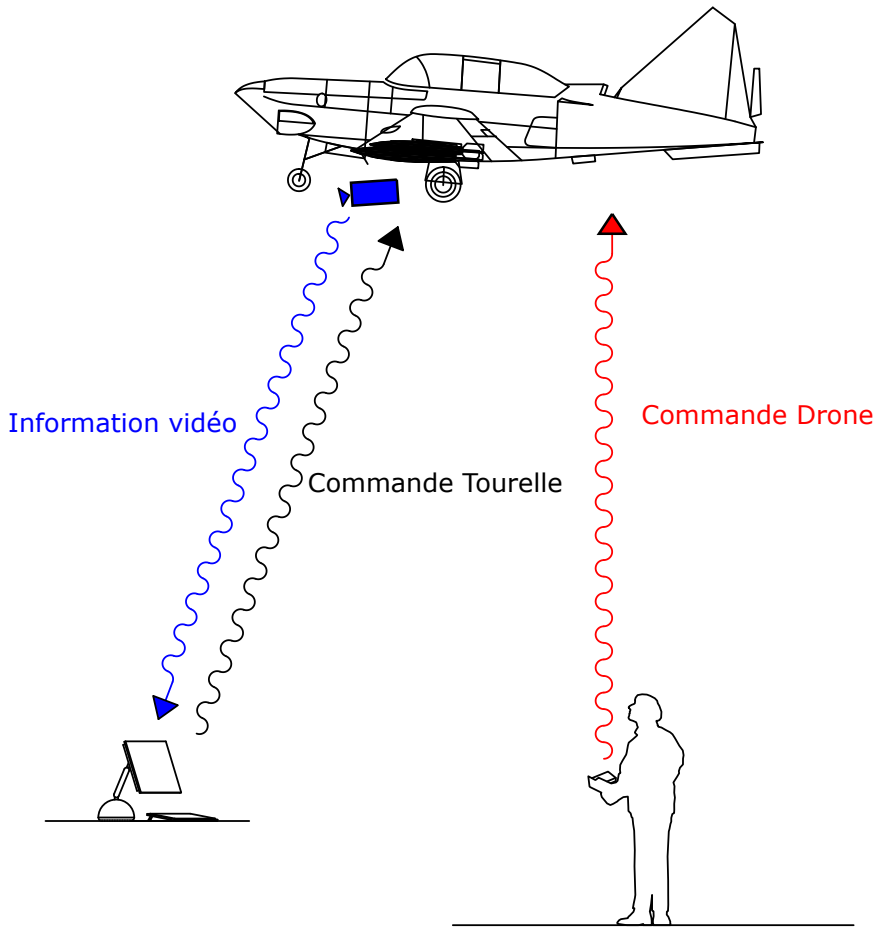

Fig. 1.11 – Schéma de contexte

Toute l'étude présentée dans ce manuscrit s'inscrit dans un contexte d'asservissement visuel pour une caméra embarquée dans un drone. Pour cela, nous avons à notre disposition un système expérimental pour tester les différents algorithmes d'analyse d'image et les différentes lois de commande et tâches d'asservissement visuel développées en laboratoire. Ce système est un drone, il s'agit en fait d'un avion de modélisme spécialement construit pour cette étude. Il permet d'embarquer une caméra montée sur une tourelle commandable en pan et tilt. Compte tenu des capacités de calcul des composants embarqués et de la nécessité d'un traitement d'image rapide pour pouvoir assurer la stabilité du système entier, l'algorithme d'analyse d'image est exécuté sur une station au sol. Les liaisons entre l'avion et la station au sol, que ce soit le retour des images filmées par l'avion ou l'envoi de commandes à la tourelle, se font sans fil.

Un schéma de notre contexte est présenté sur la figure 1.11. Le contrôle de l'avion (le drone) est réalisé par un pilote humain, tandis que la tourelle est commandée de façon automatique par un ordinateur au sol.

1.3.2 Détails sur l'équipement expérimental

La vision d'ensemble de notre système expérimental étant représentée figure 1.11, nous allons, à présent détailler chacun des éléments composant ce système.
Le drone est représentée sur la figure 1.12, la tourelle est représenté sur la figure 1.3.2 et la station au sol sur la figure 1.14. L'ensemble du dispositif expérimental se compose de :

- Un drone illustré sur la figure 1.12. Il a été conçu pour pouvoir embarquer sans problème des masses de plusieurs kilogrammes et d'un volume important. Envergure : 3,5m, masse à vide 15kg, moteur : essence, bicylindre, 100cc. Il embarque :

  - une caméra analogique pouvant acquérir 25 images/s,
  - un ensemble d'émission de télévision 2,4GHz,
  - un modem radio pour la liaison de données de contrôle de la tourelle de prises de vue,
  - un GPS à retransmission au sol pour le suivi de trajectoires,
  - un gyromètre pour la stabilisation du modèle sur l'axe de roulis, pour stabiliser les prises de vue,
  - une tourelle commandable en pan et tilt.

- La tourelle pan et tilt est illustrée sur la figure 1.3.2. Deux photos sont présentées surcette figure : une où la tourelle est embarquée dans l'avion et une autre où la tourelle

est montrée hors de l'avion. Elle dispose de plusieurs composants importants :

- deux moteurs à courant continu,
- deux potentiomètres permettant de mesurer la position angulaire des moteurs,
- deux micro-contrôleurs,
- une caméra.

– Une station au sol permettant la réalisation de la mission. Une photo de cette station est présentée figure 1.14. Elle est composée de :

- la réception de la vidéo analogique,
- un PC dédié au traitement des images en temps réel (le PC est un Pentium 4 cadencé à 3.2 GHz avec 1Go RAM et un disque dur rapide),
- la liaison modem vers la tourelle pour le contrôle du suivi des cibles (le débit est réglé à 9600 BAUDS),
- la réception des signaux GPS sur un PC dédié et le suivi du drone en temps réel sur des cartes IGN.

.

Fig. 1.12 – Drone

La boucle complète afin de réaliser un asservissement peut se décomposer de la manière suivante : les images sont acquises par la caméra embarquée dans l'avion. Elles sont transmises au sol par l'émetteur de vidéo analogique, puis traitées par les algorithmes d'extraction du

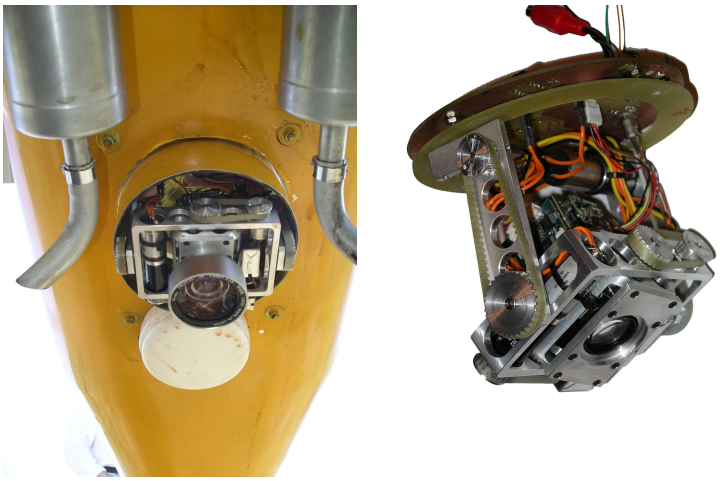

Fig. 1.13 – Tourelle supportant la caméra

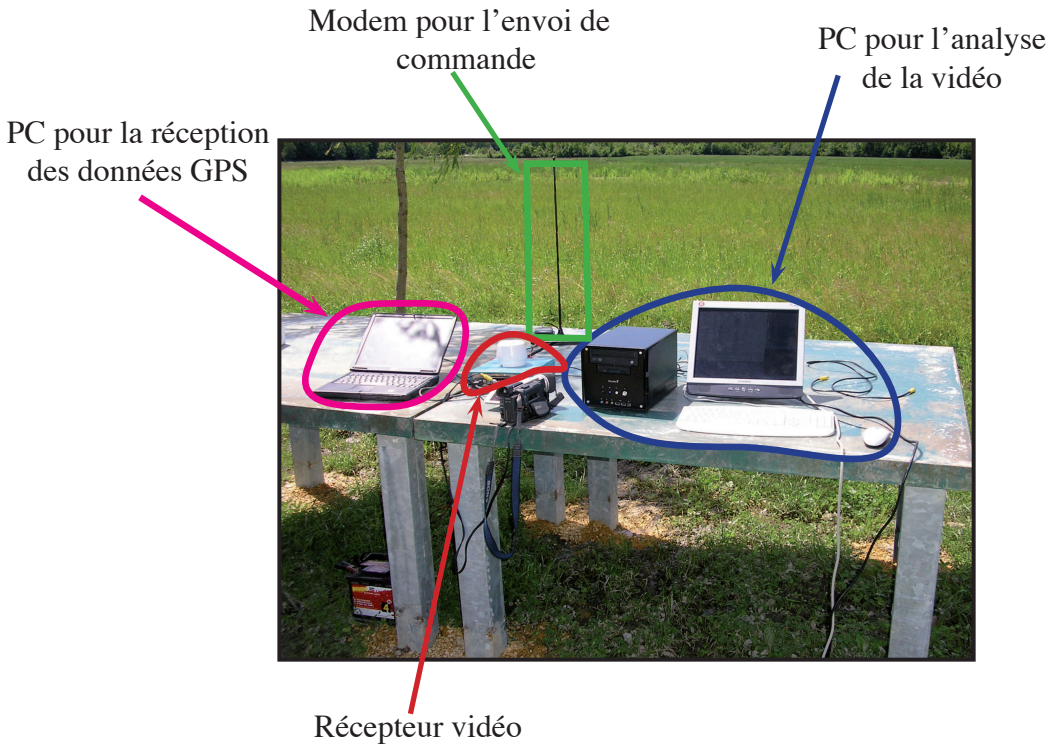

Fig. 1.14 – Station au sol

mouvement installés sur le PC. Les déplacements mesurés dans l'image permettent de calculer les consignes de commande nécessaires au bon déplacement de la tourelle pour le suivi de la cible. La liaison modem par modem radio assure la transmission des ordres à la tourelle. La durée d'une boucle de calcul et de commande est très courte : liaison TV analogique, temps de calcul optimisé, ce qui permet de travailler très proche de la cadence vidéo.

1.4 Conclusion

Nous venons de présenter différentes techniques d'asservissement visuel existantes ainsi que notre système expérimental. En confrontant l'ensemble de ces techniques à notre contexte et à notre système expérimental, plus particulièrement, aux différentes contraintes imposées (soit par le sytème soit par nous même), nous pouvons tirer un certain nombre de conclusions et donc éliminer un certain nombre de techniques présentées dans ce chapitre. Notre objectif est de développer un système d'asservissement visuel pour notre drone afin de réaliser une tâche de suivi d'objets quelconques au sol. Pour cela, nous devons disposer d'un algorithme d'analyse d'image permettant d'estimer la position d'un objet dans l'image et réaliser une commande permettant de centrer cet objet dans l'image et de le garder centré quels que soient les mouvements du drone. L'ensemble du système expérimental impose un certain nombre de contraintes au niveau matériel (puissance de calcul, précision des capteurs...) qui doivent être respectées pour la bonne réalisation de notre tâche. En effet, la plupart des travaux présentés précédemment se placent dans des conditions optimales pour réaliser leur tâche d'asservissement. Ces conditions sont souvent obtenues par l'utilisation de systèmes expérimentaux hautement dédiés à la tâche à réaliser et disposant de matériaux très performants. Ces systèmes ont l'avantage de permettre de s'astreindre de nombreuses contraintes, mais à un coût financier souvent prohibitif pour la commercialisation à grande échelle.

Notre approche a été plutôt de privilégier un système générique à moindre coût, c'est-à-dire proposer une méthode complète d'asservissement visuel facilement implantable et facilement configurable ne nécessitant pas le déploiement d'appareils très précis. Notre contribution se trouve plus particulièrement dans le système que nous proposons, système que nous avons en partie développé. En effet, après avoir réalisé des essais peu concluants avec une tourelle du *commerce*, nous avons (à partir de la tourelle existante) reconçu l'électronique et développé notre propre système tourelle.

Le nouveau système tourelle comme tous les systèmes embarqués, nous impose un certain nombre de contraintes en terme de consommation d'énergie et de poids. Ces contraintes sont valables pour toute l'électronique embarquée dans le drone, mais aussi pour la station au sol qui doit être composée de matériel facilement transportable. Comme souvent dans les sys-

tèmes embarqués, de telles contraintes ne permettent pas d'utiliser le matériel le plus puissant ou le plus précis ou encore d'avoir un système optimum disposant d'un ensemble de capteurs précis. Toujours dans une idée de simplicité, nous montrerons que notre système permet de réaliser une tâche de suivi assez simplement malgré ces imperfections. Il est à noter que notre approche perd beaucoup de son intérêt si on ne prend pas en compte cette motivation de simplicité et l'ensemble des contraintes qui viennent d'être énumérées.

Un des critères importants pour le choix de la technique d'asservissement visuel à utiliser dans notre contexte est la simplicité mécanique du système à commander. En effet, notre tourelle dispose uniquement de deux degrés de liberté : une rotation autour de l'axe x (pan) et une rotation autour de l'axe y (tilt). Ce système peut être vu comme un robot à deux degrés de liberté entièrement découplés. Avec un tel système, l'utilisation de techniques d'asservissement visuel complexe utilisant de fortes contraintes au niveau de la commande (génération de trajectoire, garantie que l'objet reste dans le champ visuel, problème d'avance/recul...) comme les techniques d'asservissement hybride ne se justifient pas. De plus, notre tâche se résume à un suivi d'objet pour un robot mobile sans connaissance *a priori* de la cible, ni de la scène filmée, ni de la position de notre robot dans l'espace, ce qui élimine de facto les techniques d'asservissement 3D. La consigne exprimée dans l'image et la simplicité des mouvements de notre tourelle, nous ont porté vers les techniques d'asservissement visuel 2D pour réaliser notre tâche.

D'autre part, la plupart de ces travaux en asservissement ne s'intéressent pas à la chaîne de traitement complète. Cest-à-dire les thèmes d'analyse d'image et de commande ne sont que rarement traités ensemble pour réaliser le couplage vision-commande. En général, les travaux se focalisent plus particulièrement sur un des deux aspects de cette chaîne, à savoir, l'analyse d'image ou la commande. Dans ce manuscrit, nous nous sommes intéressés à la fois à l'analyse d'image et à la commande. En effet, n'ayant aucun système antérieur et voulant étudier les différentes possibilités, nous proposons à la fois un algorithme d'analyse de vidéo et une loi de commande. Le couplage des deux permet de réaliser l'objectif fixé. L'algorithme d'analyse d'image basé sur une approche globale/locale que nous proposons est détaillé dans le chapitre 2. Notre approche de raisonner à la fois sur l'aspect commande et sur l'aspect image, nous a permis d'avoir une plus grande intégration de ces deux domaines dans la boucle d'asservissement visuel. Cette intégration a abouti à l'une des contributions majeures de ce manuscrit : la commande par sur-échantillonnage. Cette commande permet une imbrication très forte entre l'analyse d'image et la commande permettant ainsi de s'affranchir de certaines contraintes temporelles présentes dans les autres techniques d'asservissement visuel. Dans la suite du manuscrit, nous détaillerons les méthodes proposées pour estimer la position de l'objet dans l'image dans le chapitre 2 et la commande proposée dans le chapitre 3.

Chapitre 2

Estimation du mouvement dans l'image

Sommaire

2.1 Introduction

QUEL QUE SOIT LE SCHÉMA utilisé par chacune des méthodes d'asservissement visuel, un point qui leur est commun est la nécessité de travailler en temps-réel, c'est-à-dire d'être en mesure d'actualiser la commande des actionneurs en un temps très court par rapport à la vitesse d'évolution du robot. Dans le cas de l'asservissement visuel, la contrainte temporelle la plus forte concerne généralement le temps nécessaire à l'algorithme d'analyse d'image pour extraire une information visuelle pertinente. Cette contrainte sur le traitement d'image est d'autant plus facile à satisfaire que les formes recherchées pour la mise en correspondance sont rapidement identifiables car fortement adpatées à la tâche à effectuer. Par exemple, lorsque l'application dispose de marqueurs sur les objets d'intérêt, l'algorithme d'analyse d'image devient assez simple. *A contrario*, lorsque l'application visée permet à l'utilisateur le choix des informations visuelles extraites, l'algorithme d'analyse d'image est plus complexe donc plus gourmand en temps de calcul.

La partie la plus critique dans les techniques d'asservissement visuel est la manière d'extraire le mouvement dans l'image. Il est nécessaire qu'elle soit robuste et précise tout en respectant la contrainte temporelle. L'extraction du mouvement dans l'image conditionne entièrement la tâche d'asservissement visuel, car elle correspond à la mesure dont la commande a besoin pour réaliser la tâche désirée. Sans cette mesure, la commande à envoyer au robot ne peut pas être calculée et le système entier décroche.

A l'exception des techniques d'asservissement visuel basées sur des mesures dynamiques, c'est-à-dire des mesures de vitesse dans l'image, la grande majorité des asservissements visuels utilisent des informations visuelles de types primitives géométriques (position de points, droites...). Pour mesurer ces informations visuelles géométriques, il existe un grand nombre de solutions. En effet, dans le domaine de l'estimation du mouvement dans l'image, il existe un nombre important d'études et une quantité importante d'algorithmes différents. Deux grandes approches existent pour mesurer ces informations visuelles géométriques :

– La première approche consiste à estimer le flot optique dans l'image. On appelle ces méthodes les *techniques d'analyse du mouvement*. Le flot optique est défini comme le mouvement apparent 2D dans l'image, c'est-à-dire qu'il correspond au mouvement inter-image donné par les variations spatio-temporelles de l'intensité lumineuse. Le mouvement d'une image à la suivante n'étant qu'une projection 2D dans le plan image d'un mouvement 3D relatif à la scène et à la caméra, le flot optique est également appelé champ des vitesses apparentes. A partir de la vitesse apparente dans l'image obtenue par le flot optique, il est possible de retrouver les informations géométriques désirées. Par exemple, dans le cas d'un point, il est aisé de retrouver la position du point en

connaissant son emplacement initial. Cette approche est aussi utilisée dans les méthodes d'asservissement avec mesure dynamique. Dans ces cas là, le champ des vecteurs vitesses apparents est directement utilisé.

– La deuxième grande approche est l'extraction de formes géométriques et leur suivi image par image. On appelle ces méthodes les *méthodes à base de primitives géométriques*. Dans ce type de méthode, une première phase consiste à extraire les formes géométriques de l'image à l'aide d'un détecteur. Une deuxième phase consiste à les retrouver dans l'image suivante à l'aide d'une mise en correspondance, par exemple un critère de corrélation ou un suivi temporel de type prédiction ajustement. Le mouvement de la forme géométrique à suivre est donc résolu en estimant sa position dans chaque image. A la différence des méthodes précédentes, elle n'estime pas le mouvement dans l'image, mais uniquement le mouvement de la forme géométrique extraite en premier lieu.

On peut donc classer les différentes méthodes d'estimation du mouvement dans l'image en deux grandes catégories : les méthodes à base d'extraction de formes géométriques et les méthodes d'analyse du mouvement. Il est à noter qu'il existe des algorithmes qui ne sont pas classables dans ces deux catégories. Il s'agit des algorithmes utilisant l'espace fréquentiel pour l'estimation (transformée de fourrier, filtre de gabor,...). On les appelle les méthodes par filtrages fréquentiels, elles sont très rarement utilisées en asservissement visuel. En effet, la plupart de ces approches reposent sur une hypothèse de mouvement translationnel pur qui a pour effet de modifier la phase et sont assez coûteuses en temps de calcul. Nous n'évoquerons pas le principe des méthodes fréquentielles ici mais le lecteur peut se reporter aux travaux suivants : [JW87, CGN96, KC98, FR05].

Dans les paragraphes suivants, nous ferons un bref état de l'art en estimation du mouvement dans l'image. Nous nous intéresserons en particulier aux méthodes d'extraction de formes géométriques et d'analyse du mouvement dans l'image. On peut noter que ces dernières sont très utilisées dans les algorithmes de compression de vidéo. Ensuite, nous présenterons notre contexte pour l'estimation du mouvement et en particulier les contraintes imposées par notre système. Ensuite, nous présenterons l'approche basée sur une estimation globale et un raffinement local que nous proposons pour résoudre la position de l'objet dans l'image. Nous montrerons la pertinence de notre algorithme par des résultats quantitatifs effectués en laboratoire et des résultats qualitatifs effectués sur le terrain avec l'algorithme embarqué dans la boucle d'asservissement visuel. Nous introduirons une autre approche basée sur des points invariants pour résoudre des problèmes de robustesse. Enfin, nous conclurons, en précisant les avantages et inconvénients de notre approche ainsi que les perspectives concernant l'algorithme d'analyse d'image.

2.2 Etat de l'art en estimation du mouvement 2D apparent

Dans ce paragraphe, nous décrirons plusieurs méthodes d'estimation de mouvement apparent dans l'image. Toutes ces méthodes ont des hypothèses communes : il y a un vecteur de mouvement unique pour chaque pixel et il n'y a ni transparence, ni réflexion, ni ombre dans l'image. C'est-à-dire qu'il n'y a pas de mouvements apparents multiples pour un objet.

2.2.1 Estimation du mouvement par analyse du mouvement dans l'image

Dans ce type de méthode, le but est d'estimer le champ des vecteurs vitesses apparents dans l'image. Les algorithmes d'analyse du mouvement partent tous de l'hypothèse que l'intensité de l'image reste constante au cours du mouvement, ou alors, qu'elle varie d'une manière modélisable [HS81].
Cette hypothèse se traduit par l'équation des différences entre les images déplacées, c'est-à-dire entre les images aux instants t et $t + \delta t$, où $\delta t = +/ - 1$. L'équation est appelée DFD (*Displaced Frame Difference*) et s'écrit comme suit :

$$DFD = I(x + d_x, y + d_y, t + \delta t) - I(x, y, t) = 0$$

avec $I(x, y, t)$ l'intensité lumineuse du point $p = (x, y)$ à l'instant t et $d = (d_x, d_y)$ le vecteur de déplacement du point p entre les instants t et $t + \delta t$ dans le plan image. L'estimation du mouvement peut donc être vue de deux vue points différents : soit l'estimation du vecteur de déplacement $d(x, y, t) = (d_x(x, y, t), d_y(x, y, t))$, soit l'estimation du vecteur vitesse $v(x, y, t) = (v_x(x, y, t), v_y(x, y, t))$.
Dans le cas de l'estimation du vecteur de déplacement, il y a deux façons de poser le problème. Selon que l'estimation est posée dans le sens direct (t vers $t + 1$) ou dans le sens inverse (t vers $t - 1$). Prenons le cas du sens direct, alors, si on dispose des images $I(t)$ et $I(t + 1)$aux instants t et $t + 1$, l'estimation se résume à trouver le vecteur :

$$d(x, y, t) = (d_x(x, y, t), d_y(x, y, t))$$

pour le point $p = (x, y)$ à l'aide des deux images liées par l'hypothèse de conservation de l'intensité lumineuse :

$$I(x, y, t) = I(x + d_x, y + d_y, t + 1)$$

dans le cas de l'estimation inverse, le vecteur déplacement est estimé à l'aide de la relation :

$$I(x, y, t) = I(x - d_x, y - d_y, t - 1)$$

En général, l'estimation inverse est utilisée pour l'estimation de mouvement classique, par exemple dans les tâches d'asservissement visuel. L'estimation inverse est quant à elle souvent utilisée dans la compression vidéo avec prédiction causale d'image. La figure 2.1 résume les

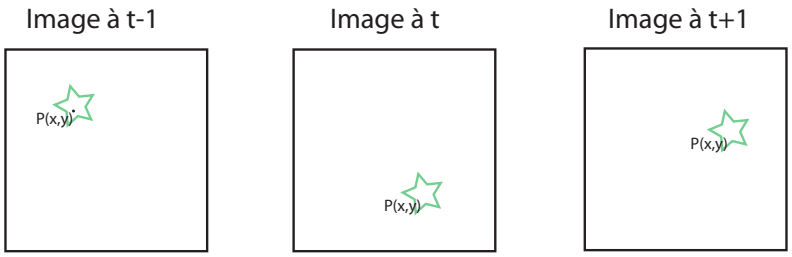

Fig. 2.1 – Exemple de de transformation de coordonnées spatiales

différents cas de figure exposés.

Dans le cas d'une estimation en vitesse, c'est-à-dire que l'on doit déterminer les vecteurs vitesses $v(x, y, t)$. On peut assimiler (si la vitesse reste constante sur l'intervalle δt et que cet intervalle est petit) la vitesse au déplacement :

$$v(x, y, t) = \frac{d(x, y)}{\delta t}$$

simplement, on peut dire que le champ de vitesse et le champ de déplacement sont identiques lorsque l'échantillonnage temporel est constant et connu. Par conséquent, dans la suite, nous nous intéresserons uniquement à l'estimation du champ de déplacement.

L'estimation de mouvement est un problème mal posé. Un problème est dit mal posé lorsqu'il ne dispose pas d'une solution unique. En effet, on n'observe pas directement le mouvement dans l'image, mais sa conséquence sur l'intensité lumineuse dans l'image. Dans le cadre de l'estimation de mouvement, deux constatations montrent que ce problème est mal posé :

- Le problème d'existence de la solution peut être illustré dans le cas pour lequel l'un des deux points en correspondance n'est pas visible dans l'image (occultation, sortie de l'image...).
- Le problème d'unicité de la solution est lié au problème d'ouverture [MU81].

Le problème d'occultation est dû au recouvrement d'une surface dans l'image. Ceci peut être dû à la translation ou à la rotation d'un objet dans le champ visuel ou à la disparition de l'objet du champ visuel lors de déplacements de la caméra. Ce recouvrement provoque la disparition de certains pixels de l'image I_t dans l'image I_{t+1}, entraînant l'impossibilité de mise en correspondance entre les deux images. Des techniques robustes d'estimation existent pour prendre en compte les discontinuités dues aux occultations. Par exemple, en se basant sur une segmentation des objets dont on veut estimer le mouvement.

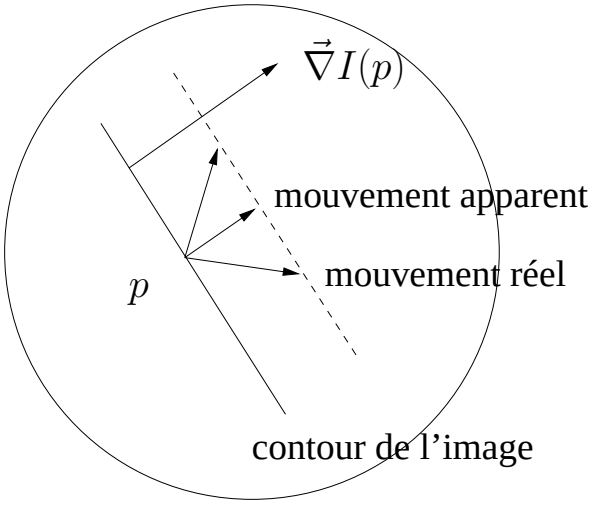

Fig. 2.2 – Illustration du problème d'ouverture

Le problème d'ouverture est une reformulation du fait que la solution n'est pas unique. Il est illustré sur la figure 2.2. Considérons les vecteurs de mouvement en chacun des pixels comme des variables indépendantes. Alors, dans le cas 2D, le nombre d'équations sera deux fois plus petit que le nombre d'inconnues, puisque chaque vecteur possède deux composantes pour un pixel. On peut montrer que, en l'absence de contraintes supplémentaires, on ne peut déterminer en chaque point, uniquement la composante normale du déplacement, qui est orientée dans la direction du gradient spatial de l'intensité, au point considéré. Un exemple bien connu de ce problème est l'enseigne du barbier représenté sur la figure 2.3.

Une solution pour éviter ce problème est de considérer le champ de vitesse lisse et de faire l'hypothèse que les voisins du point ont une vitesse proche, puis d'utiliser un bloc de voisins contenant suffisamment d'information.

Dans les méthodes d'analyse du mouvement, l'estimation du mouvement dans l'image est basée sur les gradients spatiaux et temporels d'intensité lumineuse. Dans le cas d'une image qui est une information discrète, les gradients sont approchés par des différences finies. Les paragraphes suivants présenteront brièvement l'approche utilisée par ces méthodes.

2.2.1.1 Equation de contrainte du mouvement apparent

La plupart des méthodes se basent sur l'équation de contrainte du mouvement apparent (ECMA) pour résoudre le mouvement dans l'image. A partir de l'équation de la DFD, et toujours sous l'hypothèse que l'intensité lumineuse est constante, on peut écrire pour un pixel

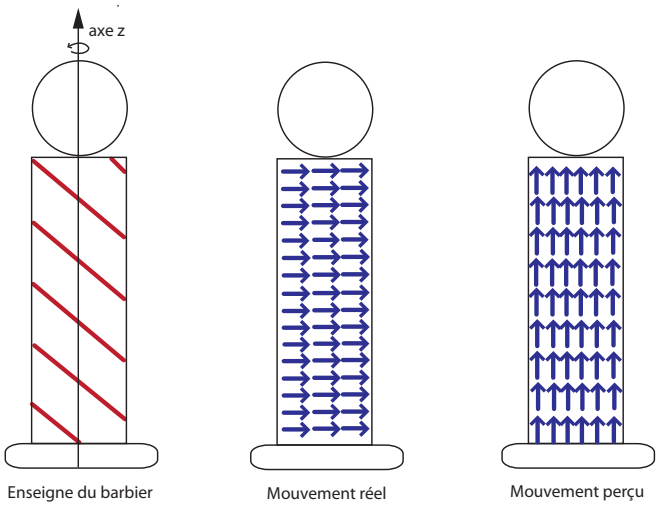

Fig. 2.3 – Problème de l'enseigne du barbier

de l'image $p = (x, y)$:

$$\frac{dI(x,y,t)}{dt} = 0 \tag{2.1}$$

où x et y varient en temps. En différenciant l'équation 2.1 [HS81], on obtient :

$$I(x + d_x, y + d_y, t + \delta t) = I(x,y,t) + \frac{\partial I(x,y,t)}{\partial x} d_x + \frac{\partial I(x,y,t)}{\partial y} d_y + \frac{\partial I(x,y,t)}{\partial t} \delta t \tag{2.2}$$

en remplacant ce développement dans l'équation de la DFD, on obtient :

$$DFD(x,y,t) = \frac{\partial I(x,y,t)}{\partial x} d_x + \frac{\partial I(x,y,t)}{\partial y} d_y + \frac{\partial I(x,y,t)}{\partial t} \delta t = 0 \tag{2.3}$$

en divisant par δt l'équation 2.3, on obtient l'équation de contrainte du mouvement apparent (ECMA) ou aussi équation du flux optique :

$$\frac{\partial I(x,y,t)}{\partial x} v_x(x,y,t) + \frac{\partial I(x,y,t)}{\partial y} v_y(x,y,t) + \frac{\partial I(x,y,t)}{\partial t} = 0 \tag{2.4}$$

où $v_x(x,y,t)$ et $v_x(x,y,t)$ sont, respectivement, les composantes de la vitesse en x et y. L'équation 2.4 peut être écrite pour tous les pixels de l'image sous la forme suivante :

$$(\nabla I)v + I_t = 0 \tag{2.5}$$

avec $\nabla I = (\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y})^T$ le gradient spatial de l'image, I_t est le gradient temporel et $v = (v_x, v_y)^T$ le vecteur vitesse.

L'hypothèse que l'intensité ne varie pas le long de la trajectoire du mouvement n'étant que partiellement vérifiée, ou complètement erronée dans le cas de changements d'illumination, une équation de contrainte du mouvement basée sur un gradient spatial constant dans la direction du mouvement a été proposée [BPT88, TP84] :

$$\frac{d\nabla I}{dt} = 0 \tag{2.6}$$

L'utilisation $\frac{d\nabla I}{dt}$ permet de s'abstraire de l'hypothèse d'intensité lumineuse constante mais requiert que les distorsions et les rotations dans l'image soient négligeables. On peut réécrire l'équation 2.6 sous la forme suivante :

$$\begin{pmatrix} \frac{\partial^2 I}{\partial x^2} & \frac{\partial^2 I}{\partial x \partial y} \\ \frac{\partial^2 I}{\partial x \partial y} & \frac{\partial^2 I}{\partial y^2} \end{pmatrix} v + \frac{\partial(\nabla I)}{\partial t} = 0 \tag{2.7}$$

avec $\frac{\partial^2 I}{\partial x^2}$, $\frac{\partial^2 I}{\partial x \partial y}$ et $\frac{\partial^2 I}{\partial y^2}$ les dérivées secondes dans l'image. En pratique, l'équation 2.7 est dure à implémenter à cause de la nature passe-haut de l'opérateur dérivée seconde.

Comme on peut le constater, l'équation ECMA ne suffit pas pour déterminer de manière unique le champ de vitesse. En effet, l'ECMA possède deux inconnues v_x et v_y pour chaque pixel. Ceci est la formulation mathématiques du problème de l'ouverture 2.2. Il est donc nécessaire d'introduire des contraintes supplémentaires sur le champ de mouvement pour estimer le mouvement. Il existe plusieurs méthodes que l'on peut classer comme suit :

- Les méthodes paramétriques qui introduisent un modèle de mouvement (constant, affine, projective) comme contrainte supplémentaire. Le champ de mouvement est estimé sur un support étendu dont on espère qu'il contienne des gradients d'orientations différentes.
- Les méthodes différentielles qui s'appuient sur l'estimation des gradients spatio-temporels en chacun des pixels. Ces méthodes partent du principe que le champ de mouvement est lisse et font l'hypothèse que le vecteur de déplacement varie lentement au voisinage du pixel [HS81, EK92, KMK05].
- Les méthodes de mise en correspondance qui supposent que l'image est divisée en plusieurs blocs. La mise en correspondance se fait sur chacun des blocs en utilisant l'intensité lumineuse. Pour chaque bloc de l'image courante, on recherche le bloc d'intensité le plus proche dans l'image suivante. On retrouve souvent ces méthodes sous le nom anglais de "block-matching" [GKC03, LCC98, LC97]. La mise en correspondance se fait à l'aide d'un critère, les plus utilisés sont la différence moyenne absolue (MAD) et l'erreur moyenne quadratique (MSE). Par leur simplicité d'implémentation, on les retrouve dans beaucoup de standards de compression vidéo : MPEG2, MPEG4, H.264
- Les méthodes statistiques qui utilisent une contrainte probabiliste de lissage. Les plus utilisées sont les méthodes Markoviennes et Bayesiennes [KD92, KD89, CK95, Ker04,

BDM04]. On peut aussi citer les méthodes d'estimation de mouvement par filtrage particulaire très bien adaptées au suivi d'objet mobile [PS06, Ros06]. Ces types de méthode pourraient être intéressantes dans notre contexte, mais malheureusement elles sont extrêmement coûteuses en temps de calcul et donc inadaptées pour une tâche d'asservissement visuel comme la nôtre.

Dans la suite, nous ferons un bref descriptif des méthodes différentielles et paramétriques.

2.2.1.2 Les méthodes paramétriques

Dans ces méthodes, la contrainte supplémentaire pour estimer le champ de vitesse est la définition explicite d'un modèle de mouvement. Les modèles sont généralement des modèles paramétriques qui prennent en compte les déformations dans l'image. Les transformations les plus courantes sont représentées sur la figure 2.4. Le modèle affine représente les transformations présentes dans une projection orthographique (rotation, transformation d'un carré en parallélogramme) alors que le modèle projectif linéaire représente les transformations présentes dans une projection perspective. Le modèle affine s'écrit :

$$v_x = a_1 + a_3 x + a_5 y$$
$$v_y = a_2 + a_4 x + a_6 y$$

avec x et y les coordonnées d'un point et v_x et v_y la vitesse en x et y du point. Le modèle projectif s'écrit :

$$v_x = \frac{a_1 + a_3 x + a_5 y}{1 + a_7 x + a_8 y}$$
$$v_y = \frac{a_2 + a_4 x + a_6 y}{1 + a_7 x + a_8 y}$$

Le modèle le plus réaliste pour décrire le mouvement d'un objet rigide dans l'image est le modèle quadratique [SW86]. Le modèle quadratique prend en compte les principales transformations dans l'image, mais n'a pas de correspondance physique particulière. Il s'écrit comme suit :

$$v_x = a_1 + a_3 x + a_5 y + a_7 x^2 + a_9 xy + a_{11} y^2$$
$$v_y = a_2 + a_4 x + a_6 y + a_8 x^2 + a_{10} xy + a_{12} x^2$$

L'estimation des différents modèles se fait à l'aide d'algorithmes de minimisation. De nombreuses méthodes peuvent être utilisées. Par exemple, on peut citer les algorithmes de type moindres carrés, descente de gradient, Gauss-Newton. Il existe aussi des estimateurs robustes [OB95, OB97]. Le problème des modèles paramétriques est que, souvent, ces modèles ne décrivent pas complètement le mouvement réel.

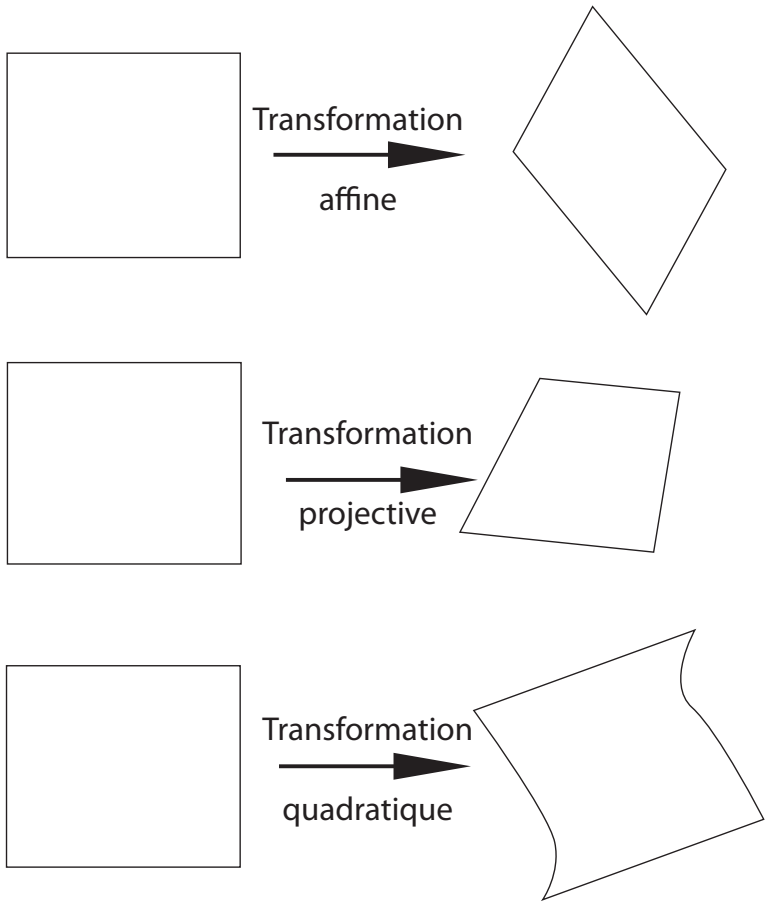

Fig. 2.4 – Exemple de transformations de coordonnées spatiales

2.2.1.3 Les méthodes différentielles

Dans ce paragraphe, nous présenterons uniquement la méthode différentielle de Horn et Shunck [HS81]. Ils ont été les premiers à proposer une solution. Elle est bien adaptée aux petits déplacements et est basée sur le développement en série de Taylor de l'EMCA. Par la suite, beaucoup d'autres travaux ont vu le jour, proposant des contraintes plus avancées [Hil84, Nag87, EK92].

Le but de la méthode de Horn et Schunk est de trouver le champ de vecteur qui satisfait l'EMCA en chacun des pixels de l'image. Soit J_{fux} l'erreur par rapport à l'EMCA en chaque pixel $p(x, y, t)$ de l'image :

$$J_{flux} = (\nabla I)v + I_t$$

Une contrainte supplémentaire qui suppose que tous les pixels voisins ont un mouvement semblable est utilisée. Pour cela, ils utilisent l'erreur $J_{uniformite}$ qui est faible si le champ de

vecteur vitesse est lisse :

$$J^2_{uniformite} = \|\nabla v_x\|^2 + \|\nabla v_y\|^2$$

La méthode consiste donc à minimiser de façon itérative l'énergie $J(v)$ suivante :

$$J(v)^2 = (1-\lambda)J^2_{flux} + \lambda J^2_{uniformite}$$

avec λ un coefficient de pondération entre les termes d'attache aux données et de lissage. Pour minimiser l'erreur, les équations de Euler-Lagrange [EH01] sont utilisées. La convergence est obtenue lorsque l'erreur est inférieure à un seuil choisi ou lorsque le nombre d'itérations maximum est atteint.

2.2.2 Estimation du mouvement par le suivi de primitives géométriques

Les techniques dites de primitives géométriques (*feature-based*) sont basées sur des primitives géométriques simples (points, lignes, segments, contours...). Le fonctionnement de ces méthodes peut se résumer ainsi : les primitives géométriques sont extraites dans la première image et sont ensuite suivies image par image dans le reste de la séquence. Le suivi se fait par la mise en correspondance des primitives géométriques extraites entre deux images à l'aide d'un critère. Certaines méthodes utilisent l'équation de contrainte du mouvement pour suivre les primitives géométriques. Dans ce cas, leur fonctionnement se rapproche en partie des méthodes exposées dans le paragraphe précédent.

Un grand nombre de travaux a été réalisé sur le sujet [CR76, Har87, KT91, ST94, SMB04, RAP06]. Le principal avantage de ces méthodes est leur simplicité et leur coût en temps de calcul. Par contre, elles dépendent beaucoup de la densité de primitives géométriques que l'on peut extraire dans l'image. De plus, elles sont complètement inefficaces en cas de recouvrement de la partie de l'image contenant la primitive à suivre. En d'autres termes, elle ne sont pas robustes aux phénomènes d'occultation.

D'autres méthodes découlant directement des méthodes à base de primitives géométriques sont utilisées en asservissement visuel, on les appelle les méthodes à base de modèles (*model-based*). On les retrouve principalement dans les asservissements visuels 3D, mais elles peuvent aussi être utilisées en asservissement visuel 2D. A la différence des méthodes précédentes, elles ne se contentent pas d'extraire des formes géométriques simples, mais définissent explicitement un modèle de l'objet à suivre. L'extraction de primitives géométriques consiste donc à reconstruire le modèle CAO de l'objet dans l'image. Elles sont plus coûteuses en temps de calcul, mais beaucoup plus robustes à d'éventuelles perturbations. Dans le cadre d'un asservissement visuel 3D, elles se résument le plus souvent à une estimation de l'attitude de l'objet (son orientation et sa position dans le repère cartésien). La tâche de suivi revient alors à faire

correspondre l'attitude estimée de l'objet à l'attitude désirée. Le principal inconvénient de ces approches est la nécessité de définir un modèle et donc d'avoir une connaissance *a priori* de l'objet à suivre. Il existe un large éventail de modèles pouvant être très différents allant d'un modèle de visage [BCL00] à la pose 3D d'une pièce métallique [MBC99] en passant par des modèles déformables [KH98].

Dans les paragraphes suivants, nous développerons certaines méthodes d'extraction de primitives de façon générale sans se rattacher à l'asservissement visuel. La présentation qui suit n'est pas exhaustive, elle comporte uniquement l'extraction et la mise en correspondance de points dans une image : formes géométriques qui rentrent dans le contexte de notre travail de recherche. Nous nous intéresserons uniquement aux primitives points en raison de leur généricité. En effet, le point est une primitive que l'on retrouve dans tous les objets réels. Nous détaillerons ce choix dans le paragraphe 2.3. Pour d'autres primitives géométriques, nous renvoyons le lecteur intéressé à la bibliographie. Par exemple, pour les droites, nous pouvons citer [MZ96, HK98, FS03] ou encore pour les contours [Can86].

2.2.2.1 Extraction de points dans une image

Dans la littérature, il existe trois grandes classes de détecteurs de points :

- La première classe est basée sur les contours. Par exemple, Mokhtarian et Suomela [MS98] propose le détecteur CSS qui repose sur une détection de contours à partir desquels des points d'intérêt sont extraits.
- La deuxième classe utilise la corrélation avec un gabarit. Par exemple, Smith et Brady [SB97] propose une approche très simple à partir de masques de forme circulaire. Ils proposent de balayer chaque point de l'image avec le masque circulaire et comptent le nombre N de pixels inclus dans le masque ayant une valeur proche de celle du pixel central. Après seuillage sur N, on détermine si oui ou non, un point d'intérêt est détecté.
- La troisième classe repose sur une mesure direct dans l'image. Cette approche, appelée approche signal est la plus utilisée. Elle est plus robuste que les deux autres classes et ne nécessite ni extraction de contour, ni connaissance approximative des points d'intérêt.

Nous allons ici exposer brièvement quelques détecteurs de points appartenant à la troisième classe. Beaudet [Bea78] propose un opérateur de détection invariant en rotation et translation. Cet opérateur utilise les dérivées secondes de l'image I. Soit H le Hessien de l'image I au point p de coordonnées (x, y) :

$$H = \begin{pmatrix} \frac{\partial^2 I(p)}{\partial x^2} & \frac{\partial^2 I(p)}{\partial x \partial y} \\ \frac{\partial^2 I(p)}{\partial y \partial x} & \frac{\partial^2 I(p)}{\partial y^2} \end{pmatrix}$$

avec $\frac{\partial^2 I}{\partial x^2}$, $\frac{\partial^2 I}{\partial x \partial y}$ et $\frac{\partial^2 I}{\partial y^2}$ les dérivées secondes. Alors,le détecteur $k = C(\frac{\partial^2 I}{\partial x^2}\frac{\partial^2 I}{\partial y^2} - \frac{\partial^2 I}{\partial x \partial y}^2$ est le déterminant de H multiplié par une constante C. Le point p est considéré comme point d'intérêt lorsque la valeur absolue de k est supérieure à un seuil fixé empiriquement.

Kitchen et Rosenfeld [KR82] propose un opérateur basé sur la courbure de la surface image multipliée par la norme du gradient au point $p = (x, y)$:

$$RK = \frac{\frac{\partial^2 I(p)}{\partial x^2}\frac{\partial I(p)}{\partial x}^2 + \frac{\partial^2 I(p)}{\partial y^2}\frac{\partial I(p)}{\partial x}^2 - 2\frac{\partial^2 I(p)}{\partial xy}\frac{\partial I(p)}{\partial x}\frac{\partial I(p)}{\partial y}}{\frac{\partial I(p)}{\partial x}^2 + \frac{\partial I(p)}{\partial y}^2}$$

Harris et Stephen [Har87] s'appuient sur l'auto-corrélation de l'image lissée par une Gaussienne. Cette auto-corrélation est calculée sur une fenêtre W de taille définie. Soit $p = (x, y)$ un point de l'image I, alors le détecteur de Harris s'écrit :

$$M(p) = G(\sigma) * \sum_{p \in W} \begin{pmatrix} \frac{\partial I(p)}{\partial x} & \frac{\partial I(p)}{\partial x}\frac{\partial I(p)}{\partial y} \\ \frac{\partial I(p)}{\partial x}\frac{\partial I(p)}{\partial y} & \frac{\partial I(p)}{\partial y} \end{pmatrix}$$

où $G(\sigma)$ est une Gaussienne de variance σ et $\frac{\partial I(p)}{\partial x}$ et $\frac{\partial I(p)}{\partial y}$ sont les gradients spatiaux de l'image I au point p. Les valeurs propres $[\lambda_1, \lambda_2]$ de la matrice $M(p)$ sont importantes, elles permettent de discriminer si le point $p = (x, y)$ est un point d'intérêt ou pas, c'est-à-dire est-ce qu'il correspond à un coin. La matrice $M(p)$ est symétrique, donc ses valeurs propres sont réelles. La discrimination se fait comme suit :

- Si λ_1 et λ_2 sont faibles par rapport à un seuil fixé selon l'image, la région considérée a une intensité constante.
- Si $\lambda_1 \gg \lambda_2$ ou $\lambda_2 \gg \lambda_1$, la région considérée présente un contour.
- Si λ_1 et λ_2 sont élevées par rapport au seuil, la région présente un coin et donc un point d'intérêt.

A partir des valeurs propres de la matrice $M(p)$, les auteurs proposent le détecteur $R(p)$ suivant :

$$R(p) = det(M(p)) - kTrace^2(M(p))$$

La valeur de k fixée par les auteurs est 0.04. Si $R(p)$ est supérieur à 0 et Trace(M(p)) est supérieur à un seuil fixé t ($Trace(M(p)) > t$), alors le pixel $p = (x, y)$ est détecté comme un point d'intérêt.
On peut noter que Noble [Nob88] propose à partir de la même matrice $M(p)$ un autre détecteur $RN(p)$:

$$RN(p) = \frac{Trace(M(p)}{Det(M(p))}$$

Si $RN(p)$ est supérieur à un seuil fixé, alors le point p est considéré comme un point d'intérêt.

Dans [Sch98], Schmid compare les différents détecteurs de points que nous venons d'exposer suivant deux critères : la répétabilité (même nombre de points détectés lors d'un changmenent d'illumination de la scène) et le contenu d'information (les points détectés doivent se distinguer les uns des autres). Il en conclut que le détecteur de Harris est celui qui répond le mieux aux deux critères.

2.2.2.2 Caractérisation des points extraits

Le problème de recherche des invariants dans une image consiste à rechercher des quantités caractéristiques de l'image indépendantes du point de vue, des conditions d'illumination, des transformations géométriques, des changements d'échelle, etc... Les invariants peuvent être utilisés dans la mise en correspondance d'image ou bien dans l'indexation d'images, c'est-à-dire la description de l'image par un résumé. Dans [GRC01], on trouve un état de l'art sur l'utilisation des invariants en robotique. L'ensemble des détecteurs présentés dans le paragraphe précédent sont invariants aux rotations et translations dans l'image (symétrie de la matrice de détection). Il est à noter qu'il existe une extension du détecteur de Harris invariant au changement d'échelle : le détecteur Harris multi-échelle [DSH00].

Plusieurs types d'invariants sont possibles. Par exemple, les invariants géométriques utilisant la géométrie d'un groupe de points (distance euclidienne entre deux points ou l'angle entre trois points). L'inconvénient des invariants géométriques est leur sensibilité aux mauvais points extraits. De plus, le voisinage des points détectés n'est pas pris en compte.

Autres que les invariants géométriques, on trouve les invariants différentiels. Ceux-ci prennent en compte la valeur de l'intensité du point et de ses dérivées. Par exemple, Sistiga dans [Sis00] définit un vecteur comprenant le niveau de gris du pixel extrait, la norme du gradient du point, le laplacien, l'opérateur de Beaudet et l'opérateur de Kitchen-Rosenfeld. Dans [Low04, Low99], Lowe définit un descripteur invariant pour des points d'intérêt. Il est appelé SIFT (*Scale Invariant Feature Transform*) et est basé sur les gradients dans huit directions des pixels au voisinage du point d'intérêt. Ce qui donne, pour chaque point d'intérêt, un descripteur de dimension 128. Il est invariant aux changements de luminosité et aux transformations géométriques. Nous le décrirons par la suite dans le paragraphe 2.7.1 pour augmenter la robustesse de notre estimation et pour une nouvelle tâche d'asservissement permettant la redécouverte d'objets perdus.

2.2.2.3 Mise en correspondance de points

La mise en correspondance de points consiste à retrouver les primitives extraites dans une image dans l'image suivante à l'aide d'un critère de ressemblance. Puis, à utiliser cette correspondance pour estimer la transformation (le déplacement dans le cas de l'asservissement visuel) entre les deux images.

Plusieurs critères de mesure de ressemblance existent dans la littérature. Le plus simple est un critère de corrélation sur un fenêtrage autour du point d'intérêt. Par exemple, la SSD (Sum of Squared Differences) représente la somme des différences quadratiques entre termes correspondants aux deux fenêtres. On trouve aussi la SAD (Sum of Absolute Differences) qui est la somme des valeurs absolues des différences entre les termes correspondants, la ZSSD (Zero Mean SSD) identique à la SSD mis à part que la moyenne des termes de la fenêtre est soustraite à chaque terme, la NCC (normalized Cross Correlation), un des critères les plus utilisés, est la somme des produits entre termes correspondants normalisée par le produit des moyennes quadratiques calculées sur chaque fenêtre. Il est possible de mettre en correspondance des points d'intérêt en utilisant non plus la fonction d'intensité lumineuse mais des descripteurs (en général invariants aux transformations géométriques) associés aux points. Par exemple dans l'algorithme SIFT, la mise en correspondance entre deux points d'intérêt se fait en comparant les descripteurs de chacun des points.

Le calcul de la transformation entre les deux images se fait à l'aide des critères de corrélation précédents. Il repose sur l'hypothèse que le déplacement inter-image est petit (hypothèse réaliste dans le cas d'une séquence vidéo). Nous pouvons alors considérer que si un point d'intérêt était dans l'image $I(t)$ à la position (x, y), alors, ce point dans l'image $I(t+1)$ se trouvera au voisinage de (x, y). Et donc, le pic de corrélation donnera la localisation du point correspondant dans la deuxième image.
D'autres méthodes citées dans [Zha93] existent pour calculer les transformations telles que les techniques de relaxation ou des techniques mettant en jeu la théorie des graphes.

2.3 Discussion par rapport à notre contexte

Comme nous l'avons exposé dans le paragraphe précédent, il existe de nombreuses études traitant de l'estimation du mouvement dans une séquence d'images. Notre système comporte de multiples contraintes. Cela nous oblige à choisir des méthodes en adéquation avec la tâche désirée. Les cinq principales contraintes de notre système sont :

1. Le suivi d'un objet *a priori* quelconque et non connu à l'avance. Par exemple dans l'image de droite de la figure 2.5 la cible est une maison.
2. La qualité fluctuante des images filmées (figure 2.6). Cette qualité fluctuante est due à des problèmes de transmission de la vidéo entre le drone et la station au sol.
3. La possibilité d'estimer de grands déplacements inter-image (à cause des mouvements très rapides du drone).
4. Le temps de calcul pour pouvoir l'inclure dans la boucle d'asservissement.
5. La précision afin de suivre des objets de petite taille. La maison dans la figure 2.5 représente une petite partie de l'image complète.

De plus, à partir du contexte particulier, un certain nombre de constations peuvent être faites sur les images filmées, entraînant des simplifications mais aussi des complications particulières au problème. On peut les résumer :

1. Simplifications liées au contexte particulier :

   – L'attitude de vol du drone nous permet de considérer le terrain filmé plat et donc de faire l'approximation que les images représentent un plan 2D.
   – Les occultations sont négligeables compte tenu de la hauteur de vol du drone.
   – Il n'y a pas d'ombre portée, ni de transparence, ni de réflexion (pas de survol de plan d'eau) dans les images.

2. Complications liées au contexte particulier :

   – Durant le suivi, l'attitude du drone change entraînant un changement de pose graduel de l'objet suivi.
   – L'hypothèse de conservation de l'intensité lumineuse n'est pas vérifiée. En effet, l'intensité lumineuse peut varier en fonction de l'éclairage extérieur (le soleil) et du gain automatique de la caméra.
   – Dans certaines images (figure 2.6), on aperçoit les roues de l'avion tournant sur elles mêmes provocant un mouvement parasite.

En reprenant les deux grandes catégories de méthode d'estimation du mouvement que nous avons présentées au paragraphe précédent et en les confrontant à nos contraintes, nous obtenons :

- La première grande classe de méthode : l'estimation du mouvement à partir de primitives géométriques s'appuie sur l'extraction de primitives géométriques (points, lignes, contours,...) dans l'image et leur suivi. Le choix de bonnes primitives permet une esti-

mation précise de la position de l'objet dans l'image. Leurs principaux avantages sont leur faible consommation en temps de calcul et leur précision. Par contre, elles sont très sensibles aux perturbations dans l'image. Dans le cas de notre tâche, des perturbations dues à la transmission vidéo endommagent fortement les images à traiter.

- La deuxième classe de méthode : l'estimation du mouvement par analyse du mouvement est fondée sur l'analyse du mouvement dans une région ou dans la globalité de l'image à l'aide de l'intensité lumineuse. Plus robustes que les méthodes précédentes, elles offrent une meilleure représentation du mouvement par l'utilisation de modèles prenant en compte des déformations 2D complexes (affine, projective,...). Leurs principaux inconvénients sont la nécessité d'une grande quantité d'informations pour converger vers une bonne estimation et leur temps de calcul important. De plus, dans le contexte de notre tâche d'asservissement visuel, compte tenu de la taille des objets à suivre, et compte tenu que le mouvement de l'objet peut différer légèrement du mouvement global dans l'image, l'estimation fournie par ce genre d'algorithme peut être imprécise.

Fig. 2.5 – Exemple de cible

2.4 Méthode proposée pour estimer le mouvement

Dans le cadre de notre système d'asservissement visuel, nous avons besoin d'un algorithme d'analyse d'image rapide, précis et robuste. Ce dernier point est primordial compte tenu du type d'image à analyser. Dans le paragraphe précédent, nous avons fait un bref tour d'horizon des différentes classes de méthode possibles pour réaliser notre tâche d'asservissement visuel.

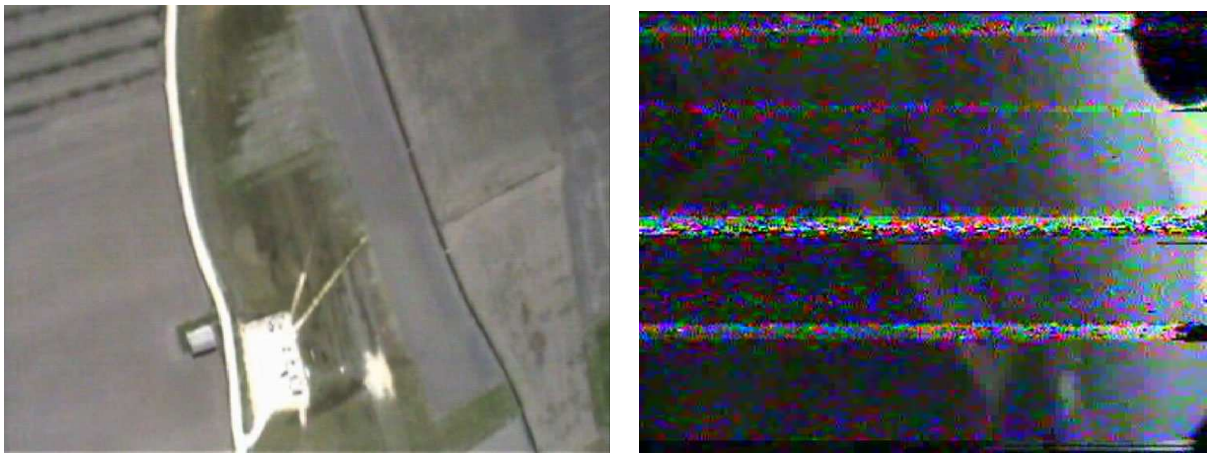

Fig. 2.6 – Qualité fluctuante des images filmées

Nous avons confronté ces différentes méthodes aux contraintes de notre tâche en détaillant pour chacune ses inconvénients et ses avantages. A partir de ces constatations et pour faire en sorte que notre algorithme d'analyse d'image soit robuste au problème de perturbation, tout en gardant une bonne précision sur la position de la cible, nous proposons un algorithme basé sur une combinaison des méthodes d'analyse du mouvement et de primitives géométriques.

Pour cela, nous sommes partis de deux algorithmes existants choisis pour leurs bonnes propriétés pour chacune des deux classes de méthodes. Notre algorithme repose sur l'algorithme KLT [KT91] pour l'extraction de primitives points et sur l'algorithme RMRm [OB94] pour l'analyse du mouvement dans l'image. Cette approche peut être comparée aux approches de recherche globale/locale. En effet, dans notre algorithme, l'analyse du mouvement se comporte comme une recherche globale du mouvement dans l'image et le suivi de primitives géométriques comme un raffinement local de la recherche de la cible. L'algorithme développé est représenté dans le paragraphe 2.4.4.

Par la suite, nous détaillerons l'algorithme RMRm et l'algorithme KLT puis l'algorithme que nous proposons. En premier lieu, nous présenterons rapidement l'analyse pyramidale dont se servent les différents algorithmes présentés.

2.4.1 Analyse multirésolution pyramidale

L'analyse multirésolution pyramidale est la structure utilisée pour l'estimation de mouvement dans la quasi totalité des algorithmes récents. Elle est utilisée dans les trois algorithmes (KLT, RMRm et SIFT) que nous présenterons. Cette analyse se décompose en deux étapes : un filtrage permettant de créer une série d'images à différentes résolutions et un sous-échantillonnage permettant de construire la pyramide.

2.4.1.1 Analyse multirésolution

Une analyse multirésolution est l'analyse d'une série d'images issues de la même scène, mais à des niveaux de résolution spatiale différents. Afin de générer cette série d'images à différentes résolutions, l'image à pleine résolution (l'image d'origine) est convoluée avec un filtre passe-bas. Cette convolution de l'image par un filtre passe-bas est similaire à une défocalisation de la caméra qui observe la scène. Toute la théorie de l'analyse multirésolution repose sur le fait qu'aucune primitive n'est générée lors du passage d'un niveau de résolution à un niveau inférieur de résolution. Plus simplement, le filtrage sélectionne les primitives les plus significatives sans ajouter d'artefact menant à l'apparition de primitives indésirables. Pour cela, il faut que le filtre passe-bas respecte le principe de causalité. Dans [BWBD86], il est démontré que le filtre gaussien respecte cette contrainte pour des signaux à une dimension, puis ,Yuille et Poggio [YP86] ont étendu la démonstration aux images.

Dans ce paragraphe, nous nous intéresserons uniquement aux techniques multirésolution utilisant un filtre gaussien. En effet, la plupart des techniques multirésolution utilisent un tel filtre. De plus, les algorithmes que nous utiliserons pour réaliser notre tâche d'asservissement sont basés sur des pyramides de type gaussienne et laplacienne. Ces pyramides sont construites à partir d'un filtre passe-bas de type gaussien. Il existe d'autres approches. Par exemple, Wu et Xie [WX90] utilisent un filtre exponentiel et un filtre moyenneur en plus du filtre gaussien. On peut citer aussi l'utilisation de filtre morphologique pour construire une représentation multirésolution [CY89, OCD95, Toe89].

Nous rappelons maintenant brièvement la façon d'obtenir un filtre gaussien bidimensionnel discret. Soit le filtre gaussien bidimensionnel $G(x, y)$, il est séparable en deux filtres unidimensionnels $G(x)$ selon l'axe x et $G(y)$ selon y. Les réponses impulsionnelles de ces filtres sont les suivantes :

$$G(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{x^2}{2\sigma^2}} \text{ et } G(y) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{y^2}{2\sigma^2}} \tag{2.8}$$

Avec σ la variance de la gaussienne permettant de régler la largeur de bande du filtre.
L'image étant un signal discret, il est nécessaire de faire une approximation discrète de $G(x)$

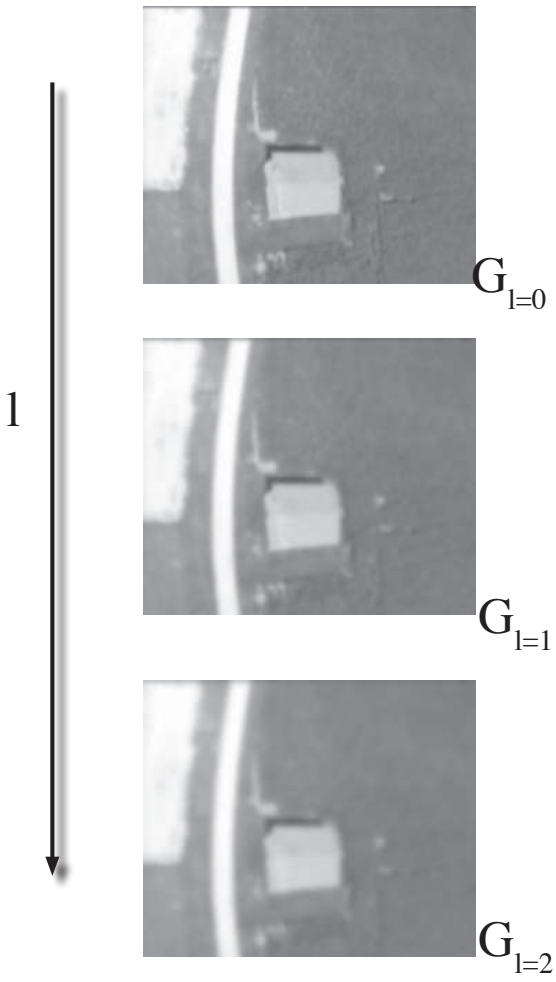

Fig. 2.7 – Images à plusieurs résolutions

et de $G(y)$ afin de disposer d'un vecteur horizontal et d'un vecteur vertical pour réaliser la convolution. En théorie, la distribution gaussienne n'est jamais nulle, ce qui entraîne une dimension dim (le nombre de coefficient) infinie pour le filtre gausssien numérique. En pratique, les valeurs du filtre peuvent être assimilées à zéro à partir d'une déviation égale à trois fois la variance [SHB07]. De plus, par commodité, une dimension impaire est choisie afin de pouvoir centrer le pixel à filtrer avec le noyau du filtre. Donc, la dimension d'un filtre gaussien numérique est calculée de la façon suivante :

$$dim = 1 + 2 * 3 * \sigma$$

ainsi, pour une valeur de $\sigma = 1$, on obtient comme dimension $dim = 7$. Les coefficients sont calculés en utilisant l'équation 2.8 discrétisée :

$$G(u) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{u^2 - ((dim-1)/2)}{2\sigma^2}}$$

ce qui donne comme filtre :

$$G(x) = \begin{bmatrix} 0.006 & 0.061 & 0.24 & 0.38 & 0.24 & 0.061 & 0.006 \end{bmatrix} \text{ et } G(y) = \begin{bmatrix} 0.006 \\ 0.061 \\ 0.24 \\ 0.38 \\ 0.24 \\ 0.061 \\ 0.006 \end{bmatrix}$$

d'où

$$G(x,y) = \begin{bmatrix} 0.0000 & 0.0004 & 0.0015 & 0.0023 & 0.0015 & 0.0004 & 0.0000 \\ 0.0004 & 0.0037 & 0.0148 & 0.0234 & 0.0148 & 0.0037 & 0.0004 \\ 0.0015 & 0.0148 & 0.0586 & 0.0927 & 0.0586 & 0.0148 & 0.0015 \\ 0.0023 & 0.0234 & 0.0927 & 0.1467 & 0.0927 & 0.0234 & 0.0023 \\ 0.0015 & 0.0148 & 0.0586 & 0.0927 & 0.0586 & 0.0148 & 0.0015 \\ 0.0004 & 0.0037 & 0.0148 & 0.0234 & 0.0148 & 0.0037 & 0.0004 \\ 0.0000 & 0.0004 & 0.0015 & 0.0023 & 0.0015 & 0.0004 & 0.0000 \end{bmatrix}$$

la matrice $G(x,y)$ n'est pas utilisée en pratique. En effet, il est plus rapide de convoluer successivement l'image avec les deux filtres $G(x)$ et $G(y)$ que de convoluer l'image avec le filtre $G(x,y)$.

Une autre solution pour obtenir le noyau du filtre gaussien est d'utiliser une approximation de la réponse impulsionnelle présentée dans l'équation 2.8 . Une approximation de cette réponse est fournie par le théorème de la centrale limite [Rio00] qui donne comme coefficients du noyau les coefficients du polynôme du triangle de pascal d'ordre p . L'ordre p est choisi de façon à avoir $\sigma = \frac{\sqrt{p}}{2}$. Si on reprend une variance $\sigma = 1$, on obtient $p = 4$, ce qui donne comme filtre :

$$G(x,y) = \frac{1}{246} \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$$

où $\frac{1}{246}$ est le coefficient de normalisation égal à la somme des coefficients du noyau. Comme précédemment, on peut séparer $G(x, y)$ en deux filtres $G(x)$ et $G(y)$:

$$G(x) = \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \end{bmatrix} \text{ et } G(y) = \begin{bmatrix} 1 \\ 4 \\ 6 \\ 4 \\ 1 \end{bmatrix}$$

La représentation multirésolution est donc une série d'images représentant la même scène filmée mais à des niveaux de résolution spatiale de plus en plus faibles par l'application d'un filtre passe-bas. La figure 2.7 représente une série d'images à trois niveaux ($l = 3$) de résolution différents. G_l représente l'image au niveau de résolution l. L'image G_0 est l'image d'origine ($l = 0$) issue d'une vidéo filmée par notre drone au cours d'un vol. L'image G_1 ($l = 1$) est obtenue en filtrant l'image d'origine. L'image G_2 est obtenue à partir de l'image G_1 convoluée par le filtre gaussien.

2.4.1.2 Structure pyramidale

L'opération de filtrage permet, grâce au théorème de Shannon, de sous-échantillonner l'image sans perdre d'information. Ce sous-échantillonnage de la scène observée a donné naissance à des structures de représentation : les structures pyramidales [BA83, Dye87, Hum87, REE93]. La figure 2.8 est une représentation schématique d'une pyramide de n niveaux.

Une structure pyramidale est facilement construite. La base de la pyramide contient l'image d'origine qui provient, par exemple, d'une caméra. Les niveaux successifs de la pyramide contiennent les images à des résolutions de plus en plus basses et sous-échantillonnées d'un facteur N sur chaque axe. L'image à chaque niveau est obtenue en filtrant l'image du niveau juste au dessous, comme sur la figure 2.7. Les coefficients correspondent aux paramètres du filtre passe-bas utilisé. Ensuite, l'image est sous-échantillonnée par un facteur N : seulement un pixel sur N^2 est conservé afin d'obtenir une image de taille inférieure, avec N^2 comme facteur de division. En pratique, les opérations de filtrage et de sous-échantillonnage sont effectuées en même temps. Une structure pyramidale comporte plusieurs avantages :

– Les images sous-échantillonnées de chaque niveau de pyramide sont divisées par N par rapport au niveau précédent, ce qui fait de la représentation sous forme de pyramide une structure très compacte. Par exemple, si $N = 2$, elle ne nécessite que 33% d'espace mémoire en plus, par rapport à l'image en pleine résolution seule.

– Le filtrage est simple et rapide. Chaque niveau est généré par filtrage linéaire du niveau précédent. Le même filtre peut être réutilisé puisque l'image précédente est déjà sous-échantillonnée, ce qui déplace la fréquence de coupure effective. Un filtre de petite taille peut être appliqué niveau par niveau, ce qui rend la construction de la pyramide très rapide.

La plupart des algorithmes d'estimation du mouvement utilisent une structure pyramidale avec un schéma d'estimation en v (cf figure 2.8).

Une pyramide Gaussienne de trois niveaux est représentée figure 2.9. Cette pyramide a été générée à l'aide d'une image issue d'une séquence filmée par notre caméra durant un vol. Nous utilisons, une pyramide de ce type dans les algorithmes KLT et RMRm. Dans notre cas, entre chaque niveau de pyramide, l'image résultante est divisée par un facteur deux dans chacune des directions.

Par contre, dans le cas de l'algorithme SIFT, il est nécessaire d'avoir des images filtrées par un filtre passe-bande plutôt qu'un filtre passe-bas. Pour cela, il existe une autre représentation pyramidale qui est la pyramide laplacienne. Cette structure pyramidale peut être obtenue à partir d'une analyse multirésolution par filtrage gaussien. Reprenons les figures 2.7 et 2.8, avec G_l l'image au niveau de résolution l. Considérons que G_l est échantillonnée au niveau n de la pyramide gausienne et prenons L_n le niveau n de la pyramide laplacienne. Alors L_n peut être approximé à la différence de deux images G_l de niveau de résolution contigu, L_l s'obtient donc de la façon suivante :

$$L_n = G_l - G_{l+1}$$

Pour obtenir L_{n-1}, il faut répéter la même opération, mais en prenant des images G_l correspondant à l'échantillonnage du niveau $n-1$ de la pyramide gausienne. La figure 2.10 représente une pyramide Laplacienne.

L'approche pyramidale permet d'éliminer le bruit en diminuant l'échelle et la résolution dans l'image. De plus, l'image à résolution grossière ne contient pas les nombreux petits détails (au sens spatial) que nous montre l'image à pleine résolution, ce qui améliore la convergence de l'algorithme. Par exemple, pour un grand déplacement inter-image, la structure pyramidale permet d'initialiser l'estimation à un niveau d'échelle plus petit et donc, avec un déplacement inter-image réduit.

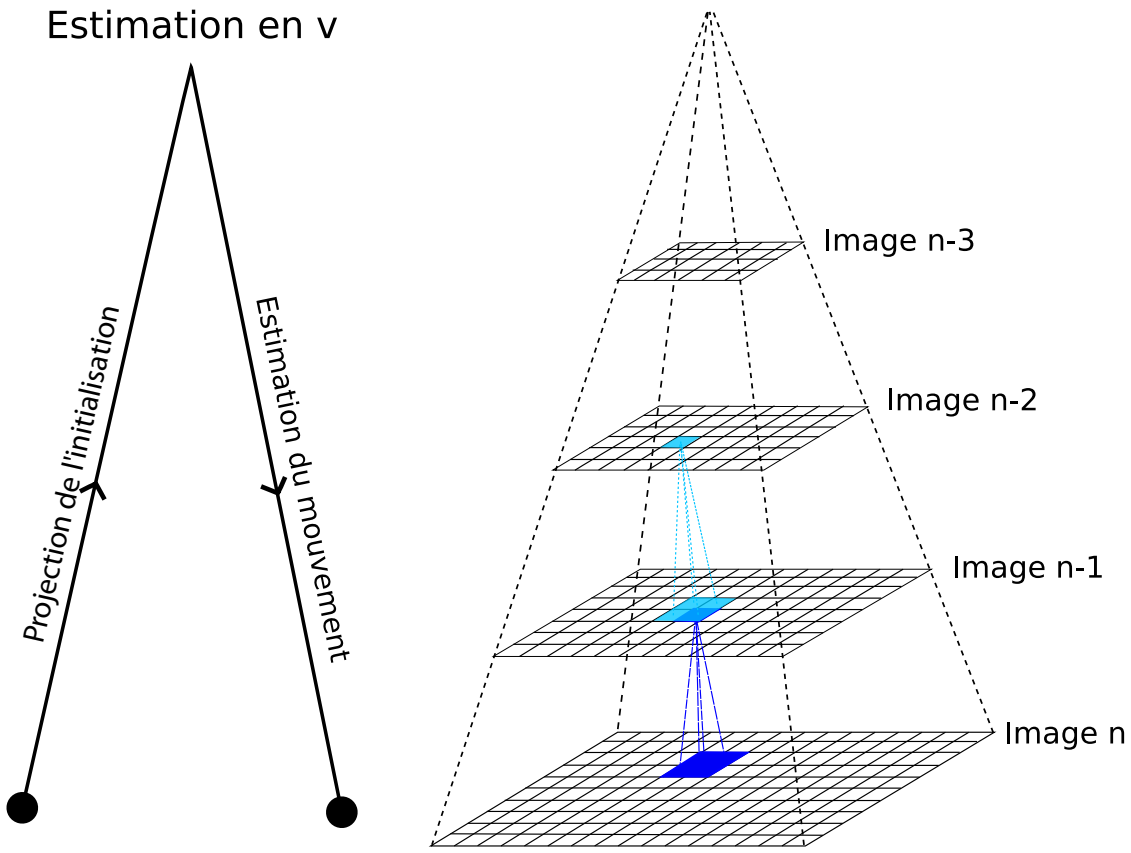

Fig. 2.8 – Représentation pyramidale

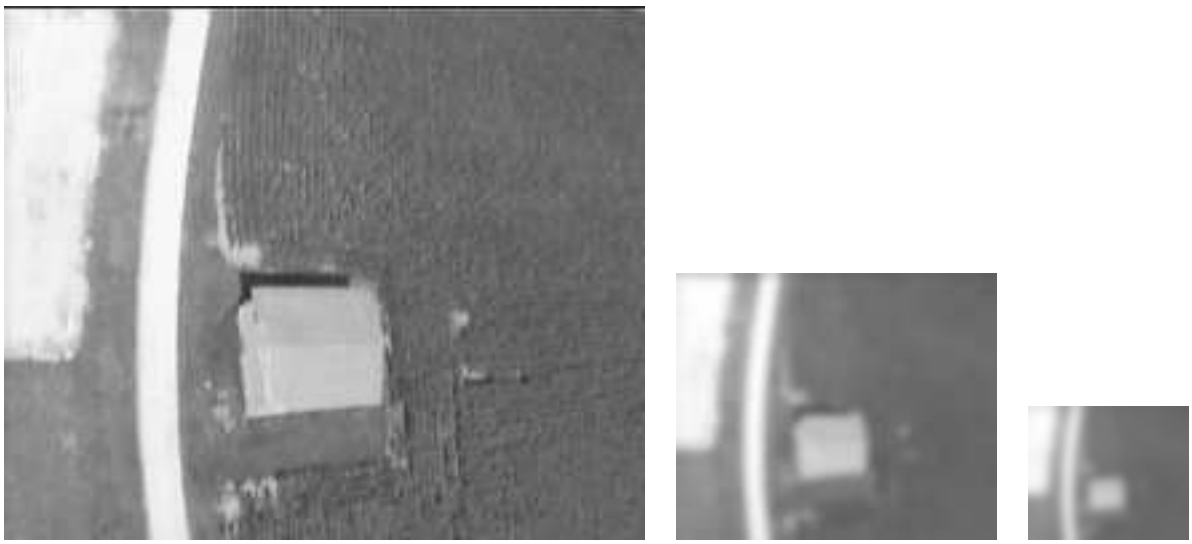

Fig. 2.9 – Pyramide Gaussienne

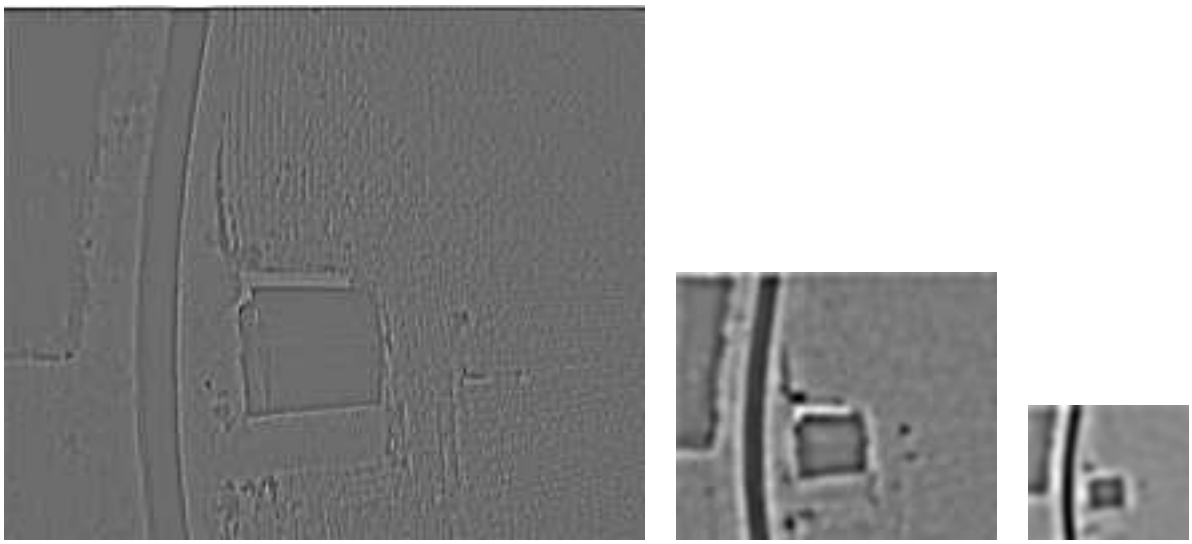

Fig. 2.10 – Pyramide Laplacienne

2.4.2 Algortihme KLT

Pour l'extraction de points dans l'image et le suivi de ces points, nous avons utilisé l'algorithme KLT. Il a été développé par Kanadé et Lucas [KT91], on le retrouve dans de nombreuses tâches de suivi d'objets. Il existe un nombre important d'implantations et de dérivées de l'algorithme originel [ST94, SMB04, ZGN04, RAP06]. Nous résumerons en premier lieu l'algorithme général proposé par Shi et Tomasi [ST94], puis nous détaillerons l'implémentation que nous utilisons.

2.4.2.1 Algorithme général

Compte tenu de la présence de bruit, les approximations du modèle de mouvement et le besoin de plus d'une équation par pixel pour résoudre le mouvement, on considère le déplacement sur un voisinage de pixels décrit par un modèle unique. Les points caractéristiques à suivre comme P ne représentent pas un pixel, mais le centre d'une fenêtre d'analyse W contenant un ensemble de pixels p. L'ensemble des calculs pour l'extraction et le suivi des points caractéristiques se font dans cette fenêtre. La première étape consiste à extraire les points à suivre. Pour cela, une adaptation du détecteur de Harris 2.2.2.1 est utilisée. Soit G la matrice suivante :

$$G = \sum_{p \in W} \begin{pmatrix} {g_x}^2 & g_x g_y \\ g_x g_y & {g_y}^2 \end{pmatrix} \tag{2.9}$$

avec $g_x \text{et} g_y$ représentant respectivement les gradients spatials en x et y au point p. Un point caractéristique est considéré comme bon candidat au suivi si :

$$\min(\lambda_1, \lambda_2) < \lambda.$$

où $\lambda_1 \text{et} \lambda_2$ sont les valeurs propres de G et λ un seuil défini empiriquement en fonction de la scène observée.

La deuxième étape est le suivi des points extraits dans l'image suivante. Soit une image I à l'instant t et la même image à l'instant $t + \tau$, alors on peut écrire sous l'hypothèse que l'intensité lumineuse est conservée :

$$I(x, y, t + \tau) = I(x + v_x, y + v_y, t)$$

ainsi une image postérieure prise au temps $t + \tau$ peut être obtenue en déplaçant chaque point dans l'image courante, prise à l'instant t, par la quantité appropriée. Le vecteur de mouvement $v = (v_x, v_y)$ correspond au déplacement du point $P = (x, y)$

Le vecteur de mouvement v est fonction de la position du point P, mais aussi de l'ensemble des déplacements des pixels de la fenêtre d'analyse W. Or, l'ensemble de ces déplacements ne

sont pas identiques, différentes déformations apparaissent, elles sont dues à la non régularité spatiale. Un modèle affine de la quantité de mouvement prenant en compte ces déformations permet une meilleure représentation, on le définit comme suit :

$$v = DP + d$$

où

$$D = \begin{pmatrix} d_{xx} & d_{xy} \\ d_{yx} & d_{yy} \end{pmatrix} \text{ et } d = (d_x, d_y)^T$$

D étant la matrice de déformation et d la translation du centre de la fenêtre d'analyse définie par le point P. Le déplacement de P entre l'image d'origine à l'instant t et l'image suivante à l'instant $t + \tau$ revient à :

$$I(AP + d, t + \tau) = I(P, t)$$

où

$$A = 1 + D$$

et 1 est la matrice identité 2x2
L'estimation du vecteur de mouvement v revient à un problème d'optimisation où il faut trouver les paramètres des matrices D et d minimisant le résidu suivant pour une fenêtre d'analyse :

$$E = \sum_{p \in W} [I(p + v, t + \tau) - I(p, t)]^2$$

où W est la fenêtre d'analyse, p un pixel de cette fenêtre. On peut approximer le modèle de l'intensité lumineuse en utilisant un développement en série de Taylor du premier ordre :

$$I((p + v), t + \tau) = I(p, t + \tau) + \nabla I(p, t + \tau)^T . v$$

Où $\nabla I(p, t + \tau)^T = (g_x, g_y)$ représente le gradient spatial de l'image calculée au point p. Ce qui donne comme résidu :

$$E = \sum_{p \in W} [I(p, t + \tau) + \nabla I(p, t + \tau)^T . v_k - I(p, t)]^2$$

La minimisation du résidu E se fait de façon itérative, avec v_k l'estimation du vecteur v à l'itération k. On peut mettre sous la forme matricielle [ST94] le résidu E. Ce qui donne pour chaque itération k le système linéaire 6x6 à résoudre :

$$T z_k = e$$

où z_k est le vecteur paramètre estimé l'itération k :

$$z_k = \begin{pmatrix} d_{xx} \\ d_{yx} \\ d_{xy} \\ d_{yy} \\ d_x \\ d_y \end{pmatrix} \text{ et } e = \sum_W [I(p,t) - I(p,t+\tau)] \begin{pmatrix} xg_x \\ xg_y \\ yg_x \\ yg_y \\ g_x \\ g_y \end{pmatrix}$$

et

$$T = \sum_W \begin{pmatrix} U & V \\ V^T & G \end{pmatrix}$$

avec

$$U = \begin{pmatrix} x^2g_x^2 & x^2g_xg_y & xyg_x^2 & xyg_xg_y \\ x^2g_xg_y & x^2{g_y}^2 & xyg_xg_y & xyg_y^2 \\ xyg_y^2 & xyg_xg_y & y^2g_x^2 & y^2g_xg_y \\ xyg_xg_y & xyg_y^2 & y^2g_xg_y & y^2g_y^2 \end{pmatrix}$$

$$V = \begin{pmatrix} xg_x^2 & xg_xg_y & yg_x^2 & yg_xg_y \\ xg_xg_y & xg_y^2 & yg_xg_y & yg_y^2 \end{pmatrix}$$

$$G = \begin{pmatrix} {g_x}^2 & g_xg_y \\ g_xg_y & {g_y}^2 \end{pmatrix}$$

Un critère de convergence est fixé sur le résidu E afin de discriminer les bons et les mauvais points. Si le critère est suivi, le point est considéré comme suivi, sinon le point est éliminé.

2.4.2.2 Implémentation utilisée

Notre implémentation est basée sur la version de Stan Birchfield [Bir97] de l'algorithme KLT. Le processus de suivi peut se décomposer en trois étapes :

- La création des points qui est similaire à la présentation du paragraphe précédent.

- Le suivi qui est basé sur l'algorithme général avec certaines simplifications. L'estimation de la matrice D représentant les déformations affine dans la fenêtre d'analyse dépend fortement de la taille de celle-ci [ST94]. La contrainte temps réel nous imposant l'utilisation de fenêtres de petites tailles, il nous a donc paru plus judicieux de ne prendre en compte que les translations pures. De plus, comme nous le verrons plus tard, les déformations affines dans l'image seront prises en compte dans l'estimation du mouvement

global. Le vecteur v se résume donc à :

$$v = d$$

Ce qui simplifie le résidu E à calculer :

$$E = \sum_{p \in W} [I(p, t+\tau) + \nabla I(p, t+\tau)^T . d_k - I(p, t)]^2$$

où d_k est l'estimation du vecteur d à l'itération k. Ce qui donne sous forme matricielle le système linéaire 2x2 à résoudre pour chaque itération k :

$$G d_k = e$$

où

$$G = \sum_{p \in W} \begin{pmatrix} {g_x}^2 & g_x g_y \\ g_x g_y & {g_y}^2 \end{pmatrix}$$

et

$$e = \sum_{p \in W} [I(p, t) - I(p, t+\tau)] \begin{pmatrix} g_x \\ g_y \end{pmatrix}$$

Le résidu E est minimisé en utilisant la méthode itérative de Newton-Raphson.

- Vérification des points, deux critères sont utilisés pour éliminer les mauvais points permettant ainsi leurs destructions au cours du suivi :

  – La convergence : si Newton-Raphson ne converge pas au bout d'un nombre d'itérations maximum, alors le point est éliminé.
  – La moyenne de la différence d'intensité entre les deux fenêtres de calcul : si la moyenne est supérieure à un seuil fixé, le point est éliminé.

Finalement, nous utilisons un suivi multirésolution (construction de pyramides gaussiennes pour chaque image). Le principe est le même que celui présenté dans la paragraphe 2.4.1. Le facteur de division est de 2. C'est-à-dire que la taille d'une image au niveau n correspond à une division par deux de la taille de l'image au niveau $n-1$. L'estimation commence au niveau le plus grossier de la pyramide et est ensuite projetée sur le niveau supérieur jusqu'à atteindre le niveau le plus haut. La structure pyramidale permet d'initialiser Newton-Raphson à un niveau de définition plus grossier, moins bruité afin d'obtenir de meilleures performances. Ceci permet aussi d'estimer de plus grands déplacements.

La figure 2.11 illustre l'algorithme KLT utilisé sur une de nos images, avec la zone englobant les points d'intérêt qui correspond simplement à la zone de l'image à suivre et l'agrandissement de cette zone montrant les différentes fenêtres de calcul W et leurs centres : les

points P d'intérêt.

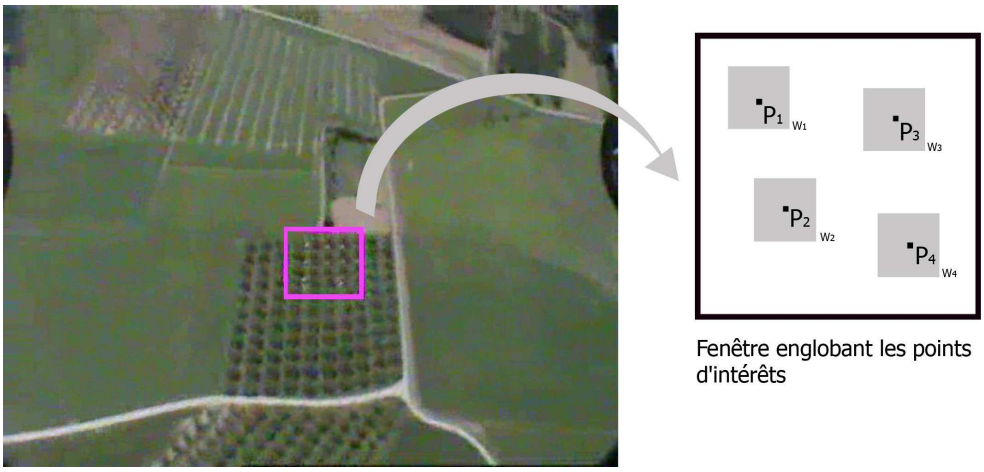

Fig. 2.11 – Illustration de l'algorithme KLT

2.4.3 Algorithme RMRm

Pour estimer un modèle de mouvement global dans l'image, nous avons utilisé l'algorithme RMRm développé par J.-M. Obodez [OB94]. Comme exposé au paragraphe 2.2.1.2 il existe plusieurs modèles de mouvement 2D. Ici, nous utiliserons un modèle affine défini comme suit :

$$v_x = a_1 + a_2 x_i + a_3 y_i$$
$$v_y = a_4 + a_5 x_i + a_6 y_i$$

où $p = (x, y)$ est la position spatiale d'un point dans l'image, $(a_1, ..., a_6)$ sont les paramètres du modèle affine et v_x, v_y le vecteur vitesse au point p. Pour estimer les paramètres du modèle, nous allons tenter de résoudre l'équation de contrainte du mouvement apparent pour chaque point p. Elle peut s'écrire sous la forme suivante :

$$\frac{dI}{dt}(p,t) = V(p,t)\nabla I(p,t) + I_t(p,t) = 0 \tag{2.10}$$

où $\nabla I(p_i, t)$ est le gradient spatial au point p_i, $I_t(p_i, t)$ est le gradient temporel au point p_i et $V(p,t) = (v_x, v_y)$.

A présent, nous résumons l'algorithme RMRm : soit deux images successives acquises par la caméra. La première étape consiste à construire deux pyramides gaussiennes comme présentée dans la paragraphe 2.4.1. Le facteur de division entre chaque niveau est deux. C'est-à-dire que la taille d'une image au niveau n correspond à une division par deux de la taille de l'image au niveau $n-1$ permettant ainsi l'estimation de grands déplacements. La deuxième

étape est l'estimation proprement dite. Soit θ_t le vecteur des six paramètres du modèle affine du mouvement à l'instant t à estimer. La première estimation de ce vecteur se fait au sommet de la pyramide. C'est-à-dire au niveau le plus grossier de résolution. Le problème d'estimation se résout en minimisant, par rapport à θ_t, le critère suivant :

$$C(\theta_t) = \sum_{p \in S} \rho(r(p, \theta_t))$$

où

$$r(p, \theta_t) = \nabla I(p, t).V_{\theta_t}(p) + I_t(p, t)$$

avec p l'ensemble des points de l'image, S le support d'estimation, $\nabla I(p, t)$ le gradient spatial de I au point p, $I_t(p, t)$ la dérivée temporelle de I au point p, V_{θ_t} le vecteur vitesse calculé au point p en appliquant le modèle défini par la valeur d'estimation courante de θ_t et ρ un estimateur robuste. Dans ce cas, l'estimateur robuste est la fonction bi-ponderée de Tukey [OB94]. Cet estimateur permet, au cours du processus itératif, de pondérer l'adéquation du déplacement du point p avec l'estimation courante du modèle de mouvement θ_t. Plus cette adéquation est faible, plus le poids est proche de zéro. Cet estimateur permet de rejeter les points dont le mouvement est aberrant par rapport au mouvement global et donc de rendre plus robuste l'estimation. Compte tenu de la non-linéarité de $C(\theta_t)$, un processus incrémental basé sur des approximations successives de $C(\theta_t)$ est utilisé. Soit $\widehat{\theta_t^k}$ l'estimé de θ_t à l'itération k. Nous pouvons écrire :

$$\widehat{\theta_t^k} = \widehat{\theta_t^{k-1}} + \widehat{\Delta\theta_t^k}$$

où $\Delta\theta_t^k$ représente le raffinement donné par l'estimation à l'itération k. En notant, $\widehat{p_k}$ l'estimation de la position du point p à l'itération k, nous obtenons alors en considérant le temps entre deux images égal à 1 :

$$\widehat{p_k} = p + V_{\widehat{\theta_t^k}}(p)$$

une approximation de taylor au premier ordre de r au point $p + V_{\widehat{\theta_t^k}}(p)$ à l'instant $t + 1$ est faite :

$$r'(\Delta\theta_t^k) = \nabla I(p + V_{\widehat{\theta_t^k}}(p), t + 1).V_{\Delta\theta_t^k}(p) + I(p + V_{\widehat{\theta_t^k}}(p), t + 1) - I(p, t)$$

et chaque incrément est obtenu en minimisant l'erreur suivante par rapport à $\Delta\theta_t^k$:

$$D(\Delta\theta_t^k) = \sum_{p \in S} \rho(r'(p, \Delta\theta_t^k))$$

$D(\Delta\theta_t^k)$ étant linéaire par rapport à $\Delta\theta_t^k$, l'algorithme IRLS peut être appliqué pour la minimisation avec :

$$\Delta\theta_t^k = \arg\min_{\Delta\theta_t^k} \Sigma\rho(r'(\Delta\theta_t^k))$$

Le processus est itéré et les incréments sont cumulés jusqu'à ce qu'un critère de convergence prédéfini soit atteint. L'estimation au niveau de résolution le plus fin est initialisée par la valeur obtenue au niveau de résolution plus grossier. L'algorithme RMRm utilise un processus d'estimation aux moindres carrés itératifs pondérés. Il ne fait appel qu'au calcul des dérivées spatio-temporelles de la fonction d'intensité.

L'algorithme RMRm permet de prendre en compte les variations d'intensité lumineuse. Dans ce cas, l'équation 2.10 est réécrite de la façon suivante :

$$\frac{dI}{dt}(p,t) = V(p,t)\nabla I(p,t) + I_t(p,t) = -\xi$$

où ξ est un scalaire représentant la variation d'intensité lumineuse à estimer. Le même processus d'estimation par incréments est utilisé, mais le vecteur à estimer est cette fois $\phi = (\theta, \xi)$.

2.4.4 Algorithme proposé

Pour réaliser des tâches d'asservissement visuel avec notre système, nous avions besoin d'estimation du mouvement robuste compte tenu de la qualité des images à traiter et d'une grande précision sur la zone à suivre en raison de la taille des objets caractéristiques de nos images. Pour cela, nous avons choisi de combiner deux méthodes d'estimation de mouvement. Une méthode d'estimation robuste du mouvement dominant en utilisant un modèle paramétrique et une méthode à base d'extraction et de suivi de points caractéristiques dans l'image. Nous avons utilisé les deux algorithmes précédemment résumés, KLT pour le suivi de point et RMRm pour l'estimation du modèle de mouvement. L'algorithme proposé est résumé sur la figure 2.12. Notre algorithme se décompose en plusieurs phases :

- Une première phase d'initialisation est nécessaire. Elle comporte l'extraction des points caractéristiques de la zone à suivre dans notre image et l'initialisation du support pour l'estimation du mouvement global. C'est-à-dire la fenêtre où est exécuté le calcul du modèle de mouvement. Ce support est inférieur à l'image complète pour des raisons de temps de calcul. Par contre, il doit être assez grand pour que l'on dispose d'un nombre d'échantillons suffisants pour assurer la convergence de l'algorithme. En pratique, sa taille est d'environ 200x200 pour une image de 320x240. Cette taille permet aussi de limiter les effets dus aux bords des images. La figure 2.13 représente le suivi d'une zone de l'image avec les différentes fenêtres de calcul.