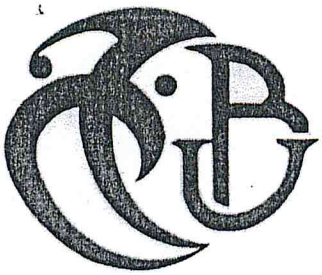
République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



Université Saad Dahlab, Blida
USDB.

Faculté des Sciences.
Département Informatique.

Mémoire pour l'obtention
Du diplôme de Master en Informatique.
Option : Ingénierie de logiciel

Sujet :

Contribution à l'analyse de mouvement dans une séquence d'images appliquée à la robotique mobile

Présenté par : Kahlouche souhila

Promoteur : Belhocine Mahmoud

Co-promotrice : Benblidia Nadjia

Devant le jury

Hamouda	**Président**
Reguegue	**Examinateur**
Hannachi	**Examinateur**

Organisme d'accueil : Centre de Développement des Technologies Avancées (CDTA)

- 2010/2011-

Remerciements

Avant tout, je remercie DIEU le Tout-puissant de m'avoir donné le courage, la volonté, la patience et la santé durant toutes ces années.

Honneur à ceux qui sont à l'origine de ce travail, je parle bien évidemment de mon promoteur Monsieur Belhocine Mahmoud et de ma co-promotrice Madame Benblidia Nadjia pour leurs conseils et pour la confiance qu'ils m'ont accordée

J'exprime toute ma gratitude à Monsieur Djekoune Oualid, pour ses critiques constructives et ses suggestions qui ont amélioré ce travail.

Je remercie le directeur du Centre de Développement des Technologies Avancées de nous avoir facilité l'inscription au Master2.

J'exprime ma sincère reconnaissance au président du jury et aux examinateurs pour l'honneur qu'ils m'ont fait en acceptant de juger ce travail

Mes remerciements vont également à tous ceux qui ont permis de loin ou de près, de contribuer à l'aboutissement de ce travail et à tous les membres de la division "Productique et Robotique" pour l'ambiance de travail.

Un remerciement particulier pour mes parents, mon époux et mes enfants, ainsi qu'à toute ma famille pour m'avoir donné la force d'aller au bout de ce travail, pour leur patience et leur amour.

Résumé

Le travail présenté dans ce mémoire a pour objectif d'exploiter le flux des images acquises par une caméra embarquée sur un robot mobile, pour l'estimation du déplacement 2D relatif des objets ou de la caméra. Une synthèse de plusieurs méthodes d'estimation de mouvement dans une séquence d'images ainsi qu'une stratégie d'évitement d'obstacles basée sur le flot optique sont présentées. Cette stratégie est implémentée sur le robot mobile *B21r* pour guider ce dernier dans ses déplacements sans collisions avec les obstacles présents dans son environnement.

Mots clés

Robotique mobile, évitement d'obstacle, analyse de mouvement, flot optique.

Abstract

The presented work aims to use the flow of images acquired by a camera mounted on a mobile robot to estimate the relative 2D motion of objects or of the camera. A summary of several methods for motion estimation in an images sequence and a strategy for obstacles avoidance based on optical flow are presented. This strategy is implemented on the mobile robot *B21r* to guide it during its moves, without collisions with obstacles in its environment.

Key words

Mobile robotics, obstacles avoidance, motion analysis, optical flow.

ملخص

العمل المقدم في هذه المذكرة يهدف إلى تسخير تدفق الصور التي حصلت عليها كاميرا محمولة على الروبوت المتحرك لتقدير الحركة ثنائية الأبعاد النسبية للأجسام أو الكاميرا. يتم تقديم ملخص لعدة أساليب لتقدير الحركة في سلسلة من الصور وكذلك إستراتيجية لتجنب العقبات على أساس التدفق الضوئي. ويتم تنفيذ هذه الإستراتيجية على الروبوت 21B ر المتحرك لترشده في التنقل من دون الاصطدام مع العقبات في بيئته.

مفـــاتيح

الروبوتات المتحركة، تجنب العقبات ، تحليل الحركة، تدفق الضوئي

Sommaire

Sommaire

Introduction générale 4

Chapitre 1 : Analyse de mouvement dans une séquence d'images

1. Introduction 6
2. Détection de mouvement 6
   - **2.1. Méthodes différentielles** 7
     - *2.1.1 La différence temporelle des images:* 7
     - *2.1.2. Double différence temporelle des images et Caractère de Contour :* 7
     - *2.1.3. Différence au fond de l'image* 8
   - **2.2. Méthodes de modélisation Markovienne** 9
3. Estimation globale du mouvement 9
   - **3.1. Modèle à mouvement translationel** 10
   - **3.2. Modèle à mouvement affine** 10
   - **3.3. Modèle à projection lineaire** 10
   - **3.4. Modèle à mouvement quadratique** 11
4. Estimation locale du mouvement 11
   - **4.1. Méthode basée sur la corrélation temporelle** 12
     - *4.1.1. Le critère SSD (Sum of Squared Difference)* 12
     - *4.1.2. Critère ZSSD (Zero Sum of Squared Differences)* 13
     - *4.1.3. Le critère ZNSSD (Zero Normed Sum of Squared Differences)* 13
     - *4.1.4. Le critère CC (cross-correlation)* 13
     - *4.1.5. Le critère ZNCC (Zero mean Normalized Cross Correlation)* 13
   - **4.2. Méthodes différentielles** 14
     - *4.2.1. Méthode de Horn et Shunk* 15
     - *4.2.2. Méthode Lucas et Kanade* 16
     - *4.2.3. La méthode Lucas et Kanade pyramidale* 17
5. Conclusion 19

Chapitre 2: Le flot optique et son utilisation en robotique

1. Introduction 20
2. Les applications sur robots manipulateurs 21
3. Les applications en robotique aérienne 22
   - **3.1. Atterrissage automatique** 22

3.2. Suivi de terrain 23
3.3. Evitement d'obstacles 24
4. Les applications en robotique mobile 25
4.1. Amarrage automatique 25
4.2. Navigation dans un couloir 26
5. Comprendre et interagir avec l'environnement par le mouvement 27
5.1. Calcul du Foyer d'expansion à partir du flot optique 28
5.2. Calcul du de la profondeur à partir du flot optique 29
5.3. La stratégie de la balance 32
5.4. Reconstruction 3D des points par le mouvement 32
6. Conclusion 35

Chapitre 3: Implémentation et mise en œuvre

1 Implémentation de la méthode Horn et Schunck 36
3. Implémentation de la méthode Lucas et Kanade 40
4. Implémentation de la méthode Lucas et Kanade pyramidale 43
5. Implémentation de la méthode de corrélation de bloc (Bloc matching) 45
6. Comparaison et discussions 47
7. Implémentation de la stratégie de la balance 49
7.1. Cas de scènes simples 50
7.2. Cas de scènes complexes 52
8. Evitement d'obstacles appliqué sur le robot B21r 53
8.1. Descriptif du robot mobile B21r 53
8.2. Scenario d'évitement d'obstacles 54
9. Conclusion 56

Conclusion générale 57

Bibliographie

Annexe A

Annexe B

Introduction Générale

Introduction Générale

L'analyse de scènes statiques a connu beaucoup d'intérêt dans les années 70, comparée à l'analyse de mouvement dans une séquence d'images, ceci était dû principalement aux problèmes techniques liés au stockage et à la manipulation des données qui sont très volumineuses par rapport aux données manipulées lors d'une analyse de scènes statiques.

Par la suite, avec l'apparition des machines de plus en plus puissantes et la mise en place de nouvelles technologies de traitement de l'information, le mouvement dans l'image a connu une nouvelle vision, dans laquelle il est considéré comme un critère de segmentation tout comme la couleur et la texture.

L'analyse de mouvement par traitement de séquence d'images n'a réellement été développée qu'à la fin des années 80 mais les applications sont surtout orientées vers la détermination des paramètres de mouvement d'objet dans la scène, cette analyse comporte :

- L'estimation du flot optique : il s'agit ici d'estimer le champ des vitesses instantanées en tout point de l'image ;
- La segmentation d'image au sens du mouvement : le but est d'obtenir des régions formées des points ayant le même mouvement ;
- La détection du mouvement : il s'agit de situer dans les images les zones correspondant aux objets en mouvement ;
- L'estimation des paramètres du mouvement tridimensionnel des objets de la scène.
- Le suivi : la trajectoire d'un point ou d'une primitive est estimée en suivant ses positions dans la séquence ;
- L'étude des mouvements non-rigides comme les mouvements d'objets articulés ou de fluides.

Actuellement le mouvement dans une image est de plus en plus utilisé dans le processus de vision. Parmi les domaines d'application de l'analyse de mouvement, nous pouvons citer de manière non exhaustive les domaines suivants:

- La météorologie par l'intermédiaire des images satellitaires ;
- La surveillance et le suivi de cibles pour le trafic urbain, la protection de zones sensibles et applications militaires ;
- La biomédecine et la biomécanique à des fins thérapeutiques ou sportives ;

- La reproduction synthétique du mouvement pour la réalité virtuelle ;
- La navigation de véhicules autonomes en robotique mobile.

En effet, grâce aux observations faites sur les yeux des insectes, les chercheurs ont pu remarquer qu'ils étaient très sensibles aux mouvements : le champ de mouvement perçu par les yeux des insectes n'est que le flot optique, ils se sont donc inspirés du comportement des insectes pour élaborer des stratégies de navigation appelées bio-inspirées. Ainsi, les techniques d'évitement d'obstacles, de suivi de terrain et d'atterrissage observé chez les insectes ont servi de modèle à de nombreuses applications en robotique aérienne et mobile.

C'est au tour de ce dernier axe que se situe le travail présenté dans ce mémoire. Il concerne d'exploitation du flux des images acquises par une caméra embarquée sur un robot mobile pour l'estimation du déplacement relatif 2D des objets ou de la caméra. Pour cela, le développement de quelques méthodes d'analyse et d'estimation de mouvement utilisées dans une navigation basée vision, s'avère nécessaire.

Notre mémoire s'articule autour de trois chapitres, organisés comme suit :

Le premier chapitre consiste en une étude comparative des différentes méthodes d'estimation de mouvement, et plus particulièrement les méthodes différentielles.

Le deuxième chapitre présente un état de l'art sur l'utilisation du flot optique en robotique.

Le troisième chapitre sera consacré aux résultats expérimentaux de l'utilisation du flot optique pour l'évitement d'obstacles. Un test sur le robot mobile B21r disponible au niveau de la division productique et robotique du CDTA, sera présenté, interprété et commenté.

A la fin de ce mémoire, une conclusion générale et des perspectives seront présentées.

Chapitre1

ANALYSE DE MOUVEMENT DANS UNE SEQUENCE D'IMAGES

Chapitre1 : Analyse de mouvement dans une séquence d'images

1. Introduction

Les capteurs optiques, tels que la rétine humaine ou une simple caméra, peuvent être considérés comme des matrices constituées d'un ensemble de cellules, sur lesquelles viennent se projeter les photons réfléchis par les objets soumis à l'éclairage d'une source lumineuse.

Lorsque la scène change, ou bien lorsque le capteur change de position, la projection de la scène sur la surface du capteur optique change également. Des bouleversements surviennent dans les intensités lumineuses enregistrées dans chacune des cellules. Toutefois, la projection de la scène apparaît toujours sur la surface du capteur.

Les changements dans la répartition des intensités lumineuses sur la surface du capteur rendent compte des changements opérés réellement : le flux optique est une mesure des changements réels par leur projection sur la surface du capteur.

Le flux optique n'en est pas pour autant une mesure exacte des changements réels comme le précise Horn dans [Horn 81]. Il s'agit de la perception par le système de vision des changements dans la scène. Cette perception peut être bruitée par des conditions d'éclairage variables, ou bien être totalement absente dans certains cas comme dans la situation suivante : comment déceler la rotation sur elle-même d'une sphère parfaitement lisse, de couleur uniforme et soumise à un éclairage régulier ?

Plus concrètement, le calcul du flux optique dans une image permet d'associer à chacun de ses pixels un vecteur vitesse 2D représentant l'estimation de la vitesse correspondant à ce pixel.

En analysant le mouvement dans une séquence d'images on s'intéresse à tout ce qui concerne les changements provoqués par ce mouvement entre des images consécutives de la même scène, en commençant par la détection et l'estimation du mouvement jusqu'au suivi d'objets et la segmentation au sens du mouvement.

2. Détection de mouvement

La détection de mouvement dans une séquence d'images consiste à distinguer les zones mobiles des zones fixes d'une scène. Comme le mouvement d'un objet induit des différences temporelles de la fonction de luminance et si le capteur est fixe et que l'éclairement de la scène varie peu, alors nous concluons que toute variation temporelle de l'intensité est liée au mouvement d'un objet ou à du bruit.

La détection de mouvement est utilisée dans de nombreuses applications comme : la compression et l'indexation vidéo, la robotique mobile (détection d'obstacles) ou encore la télésurveillance (intrusion, trafic routier ou poursuite de cibles), ainsi, un grand nombre d'algorithmes de détection de mouvement sont proposés. Nous pouvons les classer en trois catégories en fonction de la façon dont la différentiation est menée [Richefeu 06].

2.1. Méthodes différentielles

La plupart des techniques de détection du mouvement dans la séquence d'images sont basées sur une estimation du module du gradient temporel. Si les conditions d'éclairage de la scène varient lentement, alors une variation significative du niveau de gris d'un pixel entre deux images impliquera qu'il y a un mouvement en ce point. Nous pouvons subdiviser ces méthodes en trois groupes [Desachy 01].

2.1.1 La différence temporelle des images:

La façon la plus simple pour détecter un mouvement entre deux trames consécutives consiste à mesurer les changements instantanés entre elles, puis à effectuer la différence pixel à pixel, ainsi; ce qui reste correspond aux positions consécutives des objets en mouvement. Partant de ce principe, l'équation à l'état brut du mouvement *M(s)* s'écrira :

$$M(x,y) = \left|\frac{dI(x,y)}{dt}\right| = |\Delta_{It,I(t+1)}(x,y)| = |I_t(x,y) - I_{t-1}(x,y)| > seuil \quad \text{..... (I.1)}$$

Les figures1.1 illustre l'application de la différence temporelle des images.

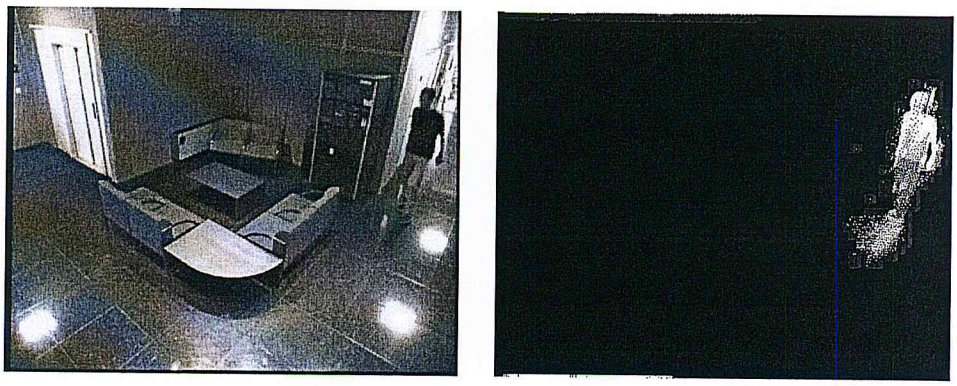


Fig. 1.1 : Application de la différence temporelle des images

2.1.2. Double différence temporelle des images et Caractère de Contour :

Cette méthode utilise aussi la différence entre les pixels des trames consécutives dans un flux vidéo comme la précédente. Mais trois trames consécutives seront utilisées dans ce cas. Cette façon de faire nous donne un résultat meilleur que celui de la méthode citée ci-dessus.

Soit I_t l'image à l'instant t, I_{t-1} l'image à l'instant (t-1) et I_{t-2} l'image à l'instant (t-2). L'objet en mouvement se compose des pixels qui satisferont les équations suivantes :

$$I_1(x,y) = I_t(x,y) - I_{t-1}(x,y) > seuil$$

$$I_2(x,y) = I_{t-1}(x,y) - I_{t-2}(x,y) > seuil$$

$$I_{résultat}(x,y) = I_1(x,y) \cap I_2(x,y) \text{ .....(I.2)}$$

2.1.3. Différence au fond de l'image

La soustraction de l'image du fond est l'approche la plus largement utilisée pour la détection d'un mouvement dans une séquence d'images, le problème est défini comme suit : étant donné une vidéo (ou bien séquence d'images) dans laquelle se présente un nombre quelconque d'objets mobiles, l'objectif est de fournir en sortie l'image du fond qui décrit le mieux la scène. Cette méthode est très populaire et elle est utilisée dans plusieurs applications. Elle détecte la région de mouvement en soustrayant pixel par pixel l'image courante de l'image du fond. Soit I_t l'image à l'instant t et $Fond_t$ est l'image du fond à l'instant t. L'objet en mouvement se compose des pixels qui satisferont l'équation suivante :

$$Mouvement = |I_t(x,y) - Fond_t(x,y)| > Seuil \quad \text{.....(I.3)}$$

Cette méthode nous permet d'obtenir la forme complète de l'objet et de compter exactement le nombre d'objets en mouvement parce que l'image du fond ne contient pas d'objets mouvants. La figure 1.2 représente une application de cette méthode (détection de pixels en mouvement par la soustraction de l'image du fond).

(a) (b) (c)

Fig1.2 : Détection de mouvement par soustraction du fond.

(a) : Image de référence, (b) : Image courante, (c) : Différence

2.2. Méthodes de modélisation Markovienne

C'est une modélisation probabiliste du problème de détection de mouvement dans laquelle on cherche la solution et la réalisation la plus probable d'un phénomène aléatoire, en passant par les étapes : modélisation, simulation et optimisation. Leurs avantages majeurs sont la robustesse et la qualité des résultats, leur principal inconvénient est leur lenteur d'exécution due au grand volume de calcul. La figure 1.3 montre un exemple de détection de mouvement par modélisation markovienne.

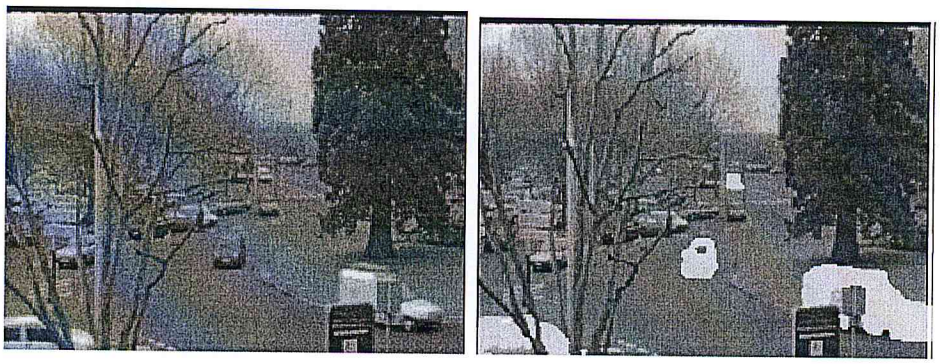


Fig1.3 : Détection de mouvement par modélisation markovienne

Généralement la détection de mouvement dans une séquence d'images, est succédée par d'autres traitements indispensables dans beaucoup d'applications, tel que l'estimation, le suivi ou même la segmentation au sens du mouvement [Kahlouche 07 a]. Dans la suite de ce chapitre, nous allons nous intéresser à un aspect très important dans le domaine de l'analyse de mouvement dans une séquence d'images qui est l'estimation de mouvement, par l'estimation du vecteur de déplacement associé à chaque pixel de l'image, pour ce là, nous présenterons les différentes méthodes existantes dans la littérature.

3. Estimation globale du mouvement

L'estimation globale du mouvement s'effectue par l'estimation d'un modèle paramétrique de mouvement appliqué à tout le support de l'image. L'estimation obtenue, consistant en un nombre réduit de paramètres de mouvement, est *globale* et permet de calculer en tout point de la scène un vecteur vitesse. Les modèles utilisés doivent être suffisamment riches pour approximer de façon satisfaisante les mouvements complexes présents dans les séquences d'images. Un grand nombre de modèles paramétriques de mouvement ont été développés. Ils ont été utilisés à des fins variées, telles que l'estimation du flot optique, la segmentation au sens du mouvement, la construction d'images mosaïques ou encore la reconnaissance

d'activités, nous citons ci-dessous quelques exemples de modèles de mouvement [Odobez 95].

3.1. Modèle à mouvement translationel

Il décrit la projection d'une translation rigide 3D, sur le plan image, c'est un mouvement 2D constant qui préserve toutes les formes (Figure1.4). C'est un modèle à deux paramètres (a, b) :

$$\begin{cases} X = X + a \\ Y = Y + b \end{cases} \quad \text{.....(1.4)}$$

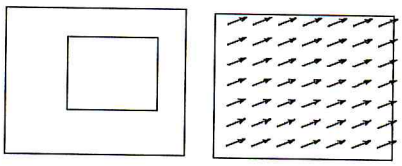


Fig. 1.4 : Mouvement translationnel.

3.2. Modèle à mouvement affine

C'est un modèle à six paramètres dans lequel les droites sont préservées (Figure 1.5)

$$\begin{cases} X = a_0 + a_1 X + a_2 Y \\ Y = a_3 + a_4 X + a_5 Y \end{cases} \quad \text{.....(I.5)}$$

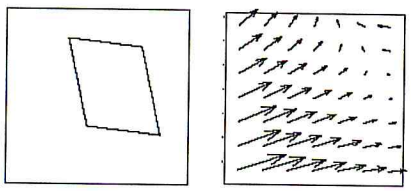


Fig.1.5 : Mouvement affine.

3.3. Modèle à projection lineaire

C'est un modèle à huit paramètres, il resulte de la projection d'un mouvement 3D affine. Avec ce modèle les droites sont preservées (figure 1.6).

$$\begin{cases} X = a_0 + a_1 X + a_2 Y + a_3 X^2 + a_4 XY \\ Y = b_0 + b_1 X + b_2 Y + b_3 X^2 + b_4 XY \end{cases} \quad \text{.....(I.6)}$$

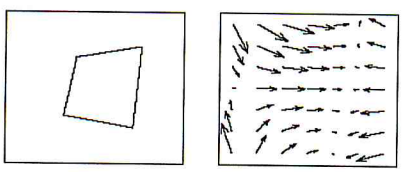


Fig.1.6: Mouvement à projection linéaire

3.4. Modèle à mouvement quadratique

C'est un modèle à douze paramètres. Il décrit un mouvement 3D affine des surfaces paraboliques, ce modèle ne préserve pas les droites (Figure 1.7).

$$\begin{cases} X = a_0 + a_1 X + a_2 Y + a_3 X^2 + a_4 XY + a_5 Y^2 \\ Y = b_0 + b_1 X + b_2 Y + b_3 X^2 + b_4 XY + b_5 Y^2 \end{cases} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (I.7)$$

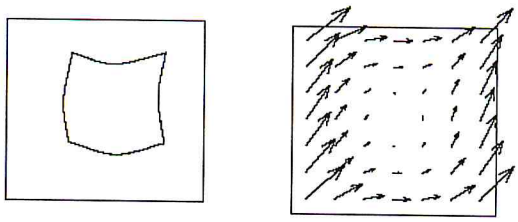


Fig. 1.7: Mouvement quadratique

La figure 1.8 montre quelques exemples de mouvement à modèles parametrés.

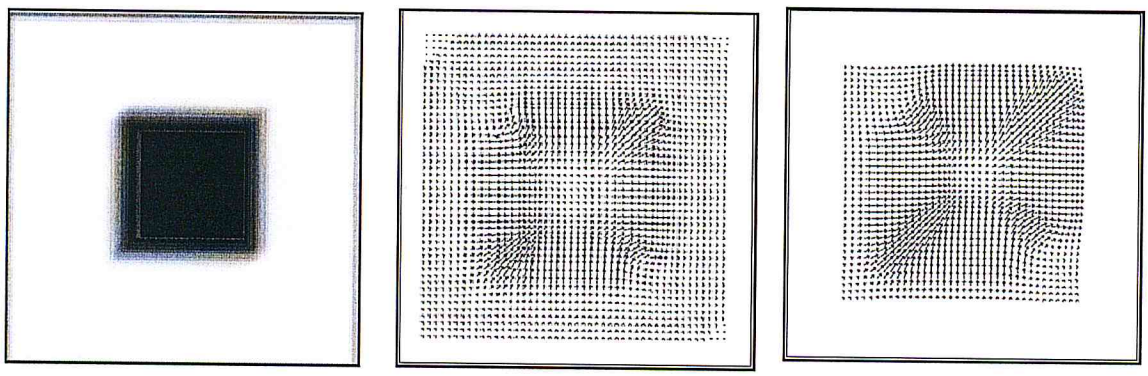


Image originale Mouvement affine Mouvement quadratique

Fig.1.8 : Exemples de mouvements à modèles parametrés

Les méthodes parametriques s'appliquent bien dans la description des mouvements rigides, cependant, leur inconvénient majeur est : que tous les modèles existants décrivent certaines classes de mouvement, et ne sont pas capables de décrire un mouvement arbitraire.

4. Estimation locale du mouvement

Afin d'obtenir une mesure locale du mouvement, les techniques d'estimation du flot optique sont contraintes d'intégrer l'information sur un voisinage spatial et parfois spatio-temporel. Ces techniques sont groupées en trois grandes familles selon que la vitesse est estimée à partir de la luminance $I(x, y, t)$, de la transformé de Fourier de $I(x, y, t)$ ou des dérivées spatio-temporelles de $I(x,y,t)$. Elles sont appelées respectivement méthodes de mise en correspondance ou corrélations, méthodes fréquentielles et méthodes différentielles. Les méthodes par corrélation consistent à maximiser l'intercorrélation de régions de deux images

acquises à deux instants successifs, à minimiser la somme des carrés des différences entre les régions, ou encore à minimiser la somme des valeurs absolues des différences. Les méthodes basées énergie utilisent une famille de filtres spatio-temporels accordés sur certaines fréquences (par exemple : filtre de *Gabor*, filtres sélectifs, transformations de *Hartley*) pour extraire l'information sur le mouvement. Les méthodes différentielles utilisent le gradient spatial et temporel de l'image, en supposant, la conservation de l'intensité de l'image dans un intervalle de temps..

4.1. Méthode basée sur la corrélation temporelle

La méthode de corrélation dans le cas de l'analyse de mouvement est basée sur la recherche du correspondant de chaque pixel d'une image de la séquence étudiée à l'instant t dans l'image suivante à l'instant $t+dt$. On définit une fenêtre de corrélation autour du pixel considéré qui va balayer l'ensemble de l'autre image *(figure 1.9)*. La convolution de cette fenêtre avec toutes les zones de l'image permet le calcul de coefficients de corrélation. L'ensemble des coefficients calculés définit les scores de corrélation associés aux pixels. Le point choisi est celui ayant le plus grand score.

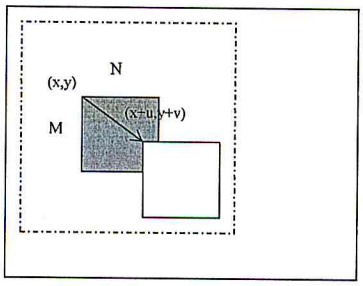


Fig.1.9 : Le suivi de pixels par corrélation

Il existe plusieurs critères de corrélation, les plus connus sont : [Aschwander 92]

4.1.1. Le critère SSD (Sum of Squared Difference)

Il représente la distance euclidienne notée entre deux vecteurs formés par les intensités du pixel (i_1, j_1) de l'image à l'instant t et celle du pixel (i_2, j_2) de l'image suivante à l'instant $t+dt$. Ce critère consiste à minimiser la somme des différences des intensités sur l'ensemble de la fenêtre de corrélation :

$$SSD(i_1, j_1, i_2, j_2) = \sum_{i=-n}^{n} \sum_{j=-m}^{m} \left[I_1(i_1 + i, j_1 + j) - I_2(i_2 + i, j_2 + j) \right]^2 \quad \ldots\ldots\ldots\ldots \quad (I.5)$$

La fenêtre de corrélation est de taille $(2n+1)*(2m+1)$.

Ce critère basé sur la différence d'intensité entre les pixels est très sensible aux différences d'illumination entre les deux images, et génère quelques faux appariements et donc des vecteur de flot optique erronés (figure1.10).

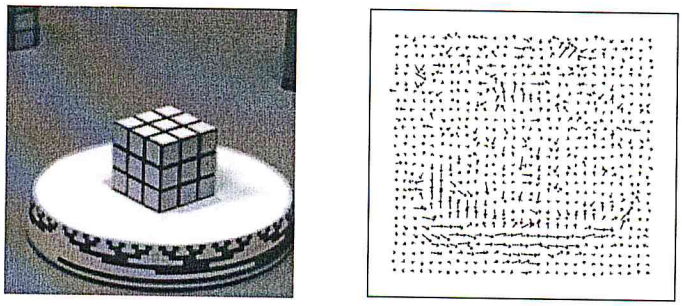


Fig.1.10 : Calcul du flot optique par la methode de correlation avec le critère SSD

4.1.2. Critère ZSSD (Zero Sum of Squared Differences)

Pour rendre ces critères invariants aux intensités moyennes des fenêtres de corrélation, on retranche la moyenne des intensités de la façon suivante :

$$ZSSD\ (i_1,j_1,i_2,j_2)=\sum_{i=-n}^{n}\sum_{j=-m}^{m}\left[\ \left(I_1(i_1+i,j_1+j)-\overline{I_1(i_1,j_1)}\right)-\left(I_2(i_2+i,j_2+j)-\overline{I_2(i_2,j_2)}\ \right)\ \right]^2 \quad \text{..... (I.6)}$$

4.1.3. Le critère ZNSSD (Zero Normed Sum of Squared Differences)

La normalisation du critère *ZSSD* par la variance locale des intensités donne le critère *ZNSSD*. Ce critère permet d'améliorer la précision des résultats, néanmoins les calculs deviennent plus importants.

$$ZNSSD=\frac{\sum_{i=-n}^{n}\sum_{j=-m}^{m}\left[\left(I_1(i_1+i,j_1+j)-\overline{I_1(i_1,j_1)}\right)-\left(I_2(i_2+i,j_2+j)-\overline{I_2(i_2,j_2)}\right)\right]^2}{\sqrt{\sum_{i=-n}^{n}\sum_{j=-m}^{m}\left[I_1(i_1+i,j_1+j)-\overline{I_1(i_1,j_1)}\right]}\ \sqrt{\sum_{i=-n}^{n}\sum_{j=-m}^{m}\left[I_2(i_2+i,j_2+j)-\overline{I_2(i_2,j_2)}\right]}} \quad (II.7)$$

4.1.4. Le critère CC (cross-correlation)

Le critère *CC* (*Corrélation Croisée*) est en fait le produit scalaire des deux vecteurs d'intensité :

$$CC\ (i_1, j_1, i_2, j_2) = \sum_{i=-n}^{n}\sum_{j=-m}^{m}[I_1(i_1+i,j_1+j)\times I_2(i_2+i,j_2+j)] \quad (II.8)$$

4.1.5. Le critère ZNCC (Zero mean Normalized Cross Correlation)

Le critère ZNCC est défini par :

$$ZNCC=\frac{\sum_{i=-n}^{n}\sum_{j=-m}^{m}[(I_1(i_1+i,j_1+j)-\overline{I_1(i_1,j_1)})\times(I_2(i_2+i,j_2+j)-\overline{I_2(i_2,j_2)})]^2}{\sqrt{\sum_{i=-n}^{n}\sum_{j=-m}^{m}[I_1(i_1+i,j_1+j)-\overline{I_1(i_1,j_1)}]^2}\sqrt{\sum_{i=-n}^{n}\sum_{j=-m}^{m}[I_2(i_2+i,j_2+j)-\overline{I_2(i_2,j_2)}]^2}} \quad (II.9)$$

Le retranchement de la valeur de chaque niveau de gris d'un terme égal à la moyenne des niveaux de gris, permet de rendre ce critère invariant à la différence de valeur des niveaux de gris à quelques pixels près.

Les techniques de corrélation sont robustes, simples à mettre en œuvre et se retrouvent dans la majorité des standards de compression vidéo. De plus, elles permettent de mesurer des déplacements d'amplitude quelconque (grande ou faible) et sont très utilisées en stéréovision. Cependant elles sont imprécises (du fait de la discrétisation du déplacement estimé), et très coûteuse en temps de calcul.

4.2. Méthodes différentielles

L'approche différentielle fait l'hypothèse que l'intensité lumineuse, d'un point de l'image, se conserve dans le temps.

Ceci se traduit par :

$$E(x, y, t)= E(x+dx, y+dy, t+dt) \quad \text{(I.10)}$$

Le développement de la partie $E(x+dx, y+dy, t+dt)$ en série de Taylor limitée au premier ordre, nous donne la relation suivante :

$$\frac{\partial E}{\partial x}dx+\frac{\partial E}{\partial y}dy+\frac{\partial E}{\partial t}dt=0 \quad \text{(I.11)}$$

On obtient une équation linéaire à deux inconnues : u et v, appelée équation de contrainte du flot optique (CFO)

$$E_x.u + E_y.v + E_t = 0 \quad \text{(I.12)}$$

On remarque ainsi que deux inconnues sont à estimer pour déterminer le déplacement. L'équation (I.12), ne suffit donc pas pour déterminer le flot optique, le problème est mal posé (une équation pour deux inconnues). Ce dernier, connu sous le nom de problème d'ouverture. Pour que l'équation (I.12) ait une solution unique, il est alors nécessaire d'ajouter une contrainte sur le flot optique.

4.2.1. Méthode de Horn et Shunk

Dans [Horn 81], une contrainte globale de régularité est ajoutée elle, stipule que le champ de vitesse recherché est régulier, l'algorithme consiste alors à minimiser la fonction suivante :

$$\varepsilon_c^2 = \left(\bar{u} - u\right)^2 + \left(\bar{v} - v\right)^2 \quad \text{.....(I.13)}$$

où $\overline{u_{ij}}$ et $\overline{v_{ij}}$ sont les moyennes des composantes du flot optique.

Cette quantité est combinée avec l'erreur fonctionnelle de CFO:

$\varepsilon_b = E_x u + E_y v + E_t$ .....(I.14)

L'erreur fonctionnelle totale: $\varepsilon^2 = \iint_{\Omega} \lambda^2 \varepsilon_c^2 + \varepsilon_b^2$ .....(1.15)

Où Ω est le support d'estimation et correspond en général à toute l'image.

λ Contrôle l'influence du terme de lissage.

Une solution au problème d'estimation de mouvement est obtenue par plusieurs étapes de relaxation de la méthode de Gauss-Seidel (pour plus de détails voir annexe A). Les équations récursives sont:

$$\begin{cases} u^{n+1} = \bar{u}^n - \frac{E_x \bar{u}^n + E_x \bar{v}^n + E_x}{\lambda^2 + E_x^2 + E_y^2} \lambda E_x \\ v^{n+1} = \bar{v}^n - \frac{E_x \bar{u}^n + E_x \bar{v}^n + E_x}{\lambda^2 + E_x^2 + E_y^2} \lambda E_y \end{cases} \quad \text{..... (I.16)}$$

Où n : est le nombre d'itération.

E_x , E_y , E_t : les dérivées partielles de E respectivement par rapport à x,y et à t.

La méthode de Horn et Schunck a apporté une solution à l'équation (CFO) grâce à la régularité qu'elle impose sur tout le flot optique, ce qui donne un champ de vitesse dense obtenu d'une manière artificielle. Cependant, le champ de mouvement calculé par cette méthode est lisse partout dans l'image, ce qui n'est pas toujours vrai. Un autre problème qui demeure sans solution avec cette méthode c'est qu'elle est inadaptée lorsqu'il s'agit de traiter de grands déplacements.

4.2.2. Méthode Lucas et Kanade

L'algorithme de Lucas-Kanade (LK), comme l'a proposé à l'origine [Lucas 81], a apporté une solution à l'équation du flot optique (CFO), en se basant sur l'hypothèse de cohérence de mouvement localement, en effet si une fenêtre locale de pixels se déplace de manière cohérente, alors nous pouvons calculer le mouvement du pixel central en utilisant ses pixels voisins, en résolvant un système d'équations. Par exemple pour une fenêtre n x n pixel nous avons un système de 2 x n x n équations.

$$\underbrace{\begin{bmatrix} I_x(p_1) & I_y(p_1) \\ I_x(p_2) & I_y(p_2) \\ . & . \\ . & . \\ . & . \\ I_x(p_n) & I_y(p_n) \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} u \\ v \end{bmatrix}}_{d} = - \underbrace{\begin{bmatrix} I_t(p_1) \\ I_t(p_2) \\ . \\ . \\ . \\ I_t(p_n) \end{bmatrix}}_{b} \qquad \text{..... (I.17)}$$

A x d = b

En multipliant les deux membres de l'équation I.17 par A^T, le nombre d'équations diminue et le système s'écrira sous la forme :

$$\begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} \sum I_x I_t \\ \sum I_y I_t \end{bmatrix} \text{ ..... (1.18)}$$

Le système d'équations (1.18) est un système surdéterminé que nous pouvons résoudre au sens des moindre carrées. Ce qui revient à minimiser l'équation$\|Ad - b\|^2$.

Le format standard de la solution est donné par : $(A^T A)d = A^T b$

A partir de cette relation nous pouvons obtenir les composantes du mouvement u et v :

$$\begin{bmatrix} u \\ v \end{bmatrix} = (A^T A)^{-1} A^T b \text{.....(I.19)}$$

Cette technique fournit un résultat dense du champ de mouvement, mais peut être utilisée dans un contexte épars, du fait qu'elle s'appuie uniquement sur des informations locales qui sont dérivées d'une fenêtre autour de chacun des points d'intérêt.

La figure I.11 montre l'influence de la taille de la fenêtre sur le résultat de calcul du flot optique. Plus la fenêtre est large plus le résultat est meilleur.

Il est à noter que cette méthode a l'inconvénient de ne mesurer que de faibles déplacements, car les grands mouvements peuvent déplacer les points à l'extérieur de la fenêtre locale ce qui rend leurs suivi impossible par l'algorithme.

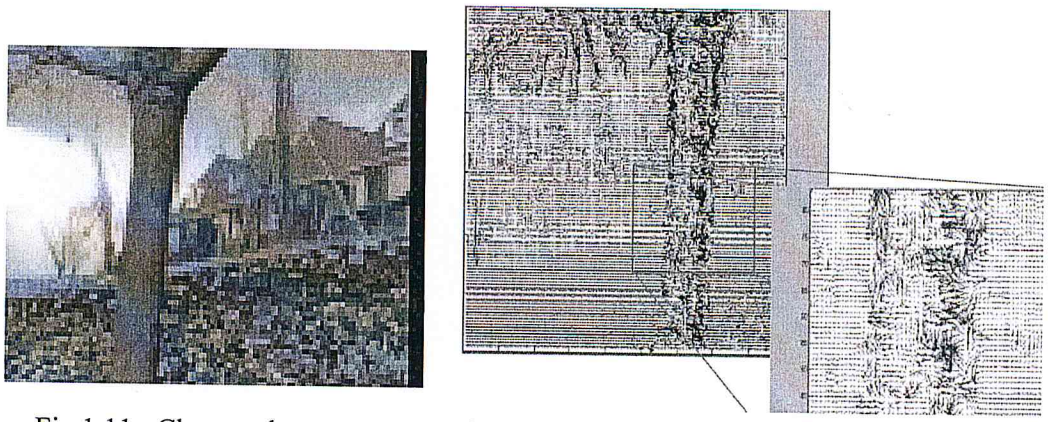


Fig.1.11 : Champs de mouvement calculés par la méthode Lucas et Kanade.

Pour pallier au problème des grands déplacements, il existe des méthodes multi-résolutions qui consistent à estimer le flot optique sur des résolutions plus faibles et à revenir progressivement vers la résolution initiale afin d'affiner l'estimation [Bouguet 99]. Il existe également des techniques mixtes comme utilisant l'appariement de blocs et les méthodes différentielles [Beghdadi 03], ce qui permet d'allier les avantages des deux méthodes.

4.2.3. La méthode Lucas et Kanade pyramidale

Le principe de cette méthode est de réduire le bruit en diminuant la résolution de l'image *[Tomasi 91]*, et ceci en construisant une pyramide qui commence par l'image initiale (niveau 0) et calcule le flot optique pour ce niveau, ensuite propage le résultat au niveau supérieur, comme une première valeur de déplacement du pixel, et calcule le flot optique du niveau en cours, et ainsi de suite jusqu'au parcours de toute la pyramide (figure 1.12).

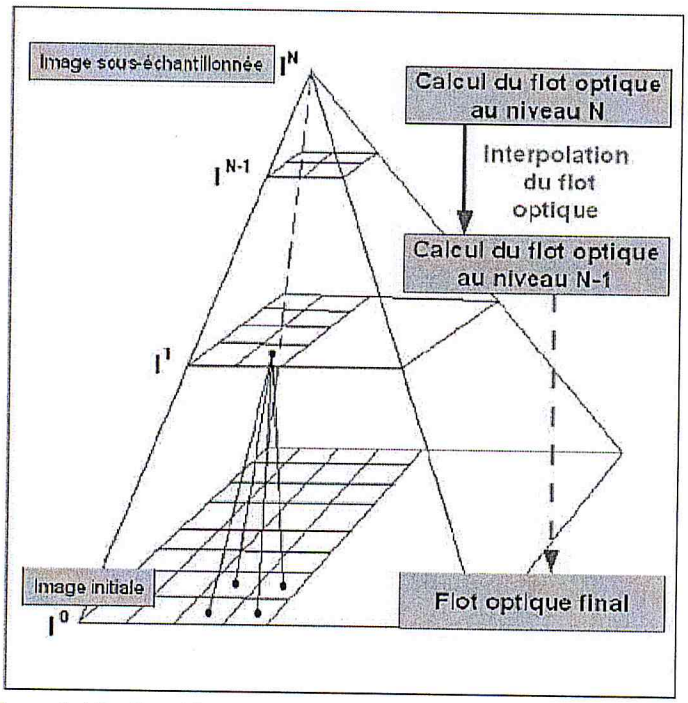


Figure1.12 : Implémentation pyramidale de Lucas et Kanade

4.2.3.1.Construction de la pyramide

Une pyramide d'images à différentes résolutions est construite à partir de chaque image de la séquence. Une méthode simple de construction est basée sur le calcul de la moyenne des niveaux de gris de quatre pixels voisins. La pyramide construite par cette méthode est appelée pyramide quaternaire. Cette méthode est rapide car elle ne nécessite que de simples calculs de moyennes. Si l'on note par I^0 le niveau de l'image initiale, l'image de niveau I^N est obtenue à partir de l'image de niveau I^{N-1} par un simple moyennage. Le niveau de gris $I_N(i, j)$ de chaque pixel est donné par :

$$I_N(i,j) = Int \quad [(I_{N-1}(2i-1,2j-1) + I_{N-1}(2i-1,2j) + I_{N-1}(2i,2j-1) + I_{N-1}(2i,2j))/4] \qquad (I.20)$$

Où Int[x] désigne le nombre entier le plus proche de x. Ainsi, à chaque nouveau niveau, le nombre de lignes et de colonnes est divisé par deux. En partant de l'image initiale, ce processus est itéré jusqu'au dernier niveau correspondant à la résolution la plus grossière. A titre d'exemple, si l'on dispose d'une image initiale de taille 512*512, la construction de la pyramide correspondante est illustrée dans la figure 1.13.

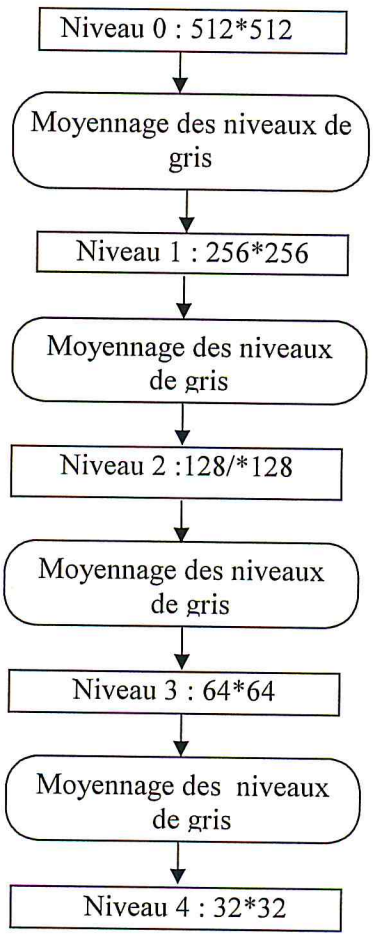


Fig.1.13 : Exemple de construction de la pyramide d'une image de taille 512x512

La figure1.14 montre le résultat du calcul du champ de déplacement de la séquence précédente par la méthode Lucas et Kanade pyramidale, ce résultat est visiblement meilleur comparé au résultat obtenu avec la méthode de Lucas et Kanade simple. L'approche pyramidale permet donc de gérer les grands déplacements dans l'image.

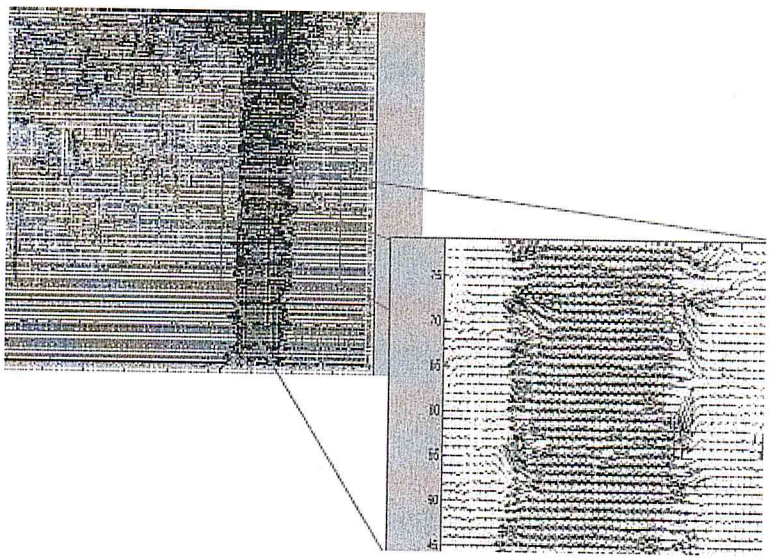


Fig.1.14 : Champs de mouvement calculés par la méthode Lucas et Kanade pyramidale.

5. Conclusion

Dans ce chapitre, nous avons présenté un état de l'art des méthodes d'analyse de mouvement dans une séquence d'images, en commençant par la détection de mouvement ainsi que les différentes approches existantes dans ce contexte, ensuite nous avons abordé l'estimation de mouvement en présentant une étude comparative de quelques méthodes existantes, à partir de laquelle nous avons conclu que : Les méthodes paramétriques sont généralement employées pour des formes de mouvements spécifiques tels que les expressions faciales ou les mouvements articulés, il reste difficile de les généraliser à des classes de mouvement plus larges. Quand aux méthodes de corrélations, dans lesquelles le mouvement estimé est éparse (par bloc), nous avons vu qu'elles sont lourdes en temps de calcul ce qui rend leur implémentation dans un système en temps réel cautionnée par les ressources de calcul disponibles. Tandis que les méthodes différentielles ont un coût de calcul faible, elles permettent une estimation dense et subpixelique du mouvement, elles sont les plus prometteuses au niveau des résultats et les plus intéressantes à développer. Dans ce qui suit nous allons monter l'utilisation de l'estimation de mouvement dans le domaine de la robotique en mettant l'accent surtout sur la robotique mobile.

Chapitre2

LE FLOT OPTIQUE ET SON UTILISATION EN ROBOTIQUE

Chapitre2 : Le flot optique et son utilisation en robotique

1. Introduction

Les premières approches de la navigation des robots étaient basées sur l'architecture composée des trois modules qui sont la perception- la Planification- et l'action, comme indiqué dans Figure 2.1. Le module perception reçoit les données sensorielles (comme la vision, les sonars, capteurs tactiles etc), et les utilise pour maintenir une représentation de l'environnement. Ensuite le module de planification, en se basant sur le modèle de l'environnement ainsi que le but à atteindre, définit une séquence de mouvements pour atteindre ce but, cette séquence est ensuite envoyée au module Action où le plan est exécuté par l'envoi de commandes aux actionneurs du robot. Le rôle de la vision dans cette architecture réside dans le module de perception pour produire un modèle du monde 3D. Le système de navigation quant à lui, repose sur la maintenance complète et exacte de la reconstruction de l'environnement. Toutefois, le calcul nécessaire pour maintenir le modèle de l'environnement, restreint le robot aux faibles vitesses et généralement à des environnements simples. Par conséquence l'architecture Perception-Planification-Action présente un manque de robustesse dans des environnements réels et c'est pour ces raisons, elle est de moins en moins utilisée [Arkin. R 98].

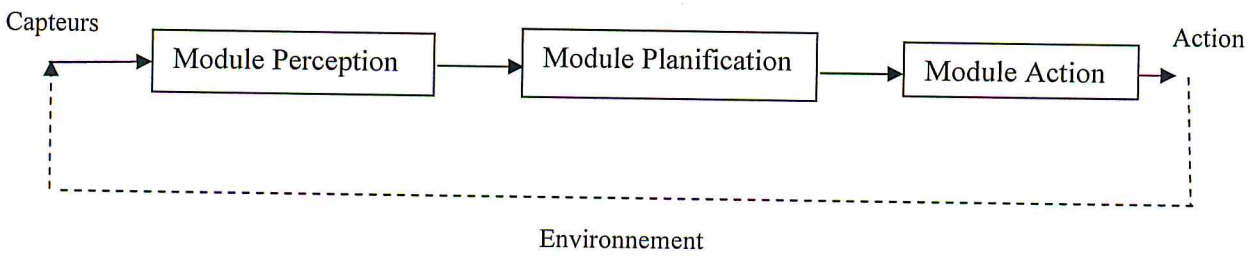


Fig. 2.1 : L'architecture Perception-Planifucation-Action

Par la suite d'autres approches de navigation de robots ont été développées et cela sans avoir besoin des modèles du monde 3D, notamment : la vision basée perception active, qui contrairement à l'approche classique de perception qui répond aux besoins de la navigation dans un environnement, l'approche active considère la perception comme une tâche de recherche et d'exploration et inverse cette dépendance de manière à ce que l'action soit motivée par le besoin de percevoir.

La richesse de l'information contenue dans le flot optique a intéressé les roboticiens dans le but de donner à leurs robots la possibilité d'interagir de façon autonome avec leur environnement. L'observation du comportement des insectes montre d'ailleurs que ce champ de déplacement est essentiel pour la navigation dans un environnement naturel texturé. La mesure du foyer d'expansion permet de connaître la direction de déplacement à chaque instant, ce qui permet de la contrôler. La mesure du temps à collision permet de détecter la proximité des obstacles et de contrôler cette proximité. Ceci permet de s'approcher des obstacles en toute sécurité ou encore de les éviter. L'information de vitesse étant incluse dans le flot optique, cela permet de stabiliser le mouvement du robot par rapport à son environnement, etc.

Dans ce qui suit, nous présentons de façon plus détaillée, mais non exhaustive, un état de l'art des applications utilisant le flot optique comme information pour la commande de robots. Nous verrons principalement trois domaines d'applications : les applications sur les robots manipulateurs pour des tâches de positionnement ou de suivi de cibles, les applications en robotique aérienne et les applications en robotique mobile, auxquelles nous nous intéresseront plus particulièrement dans ce mémoire. Par la suite, nous allons montrer comment extraire du vecteur flot optique, des informations utiles à la navigation de robots.

2. Les applications sur robots manipulateurs

Le flot optique est très utilisé dans des applications de suivi de points d'intérêt dans une scène, nécessitant un champ de vision large, un élément robotique de type tête articulée *pan-tilt* [Crétual 01] est alors utilisé. Ces applications sont nombreuses, on peut, par exemple, citer la surveillance et le suivi automatique de piétons. Dans ce cas, la commande du robot consiste à faire en sorte que le mouvement de l'objet dans l'image soit nul. Pour les applications sur les robots manipulateurs, ce sont généralement les tâches de positionnement de l'effecteur par rapport à un objet qui intéressent la communauté des roboticiens. Pour le positionnement du robot par rapport à un objet, une position relative ou une image désirée à atteindre est prédéfinie. Les techniques d'asservissement visuel permettent alors d'assurer la convergence vers le point désiré. La commande n'est cependant pas robuste lorsque l'objet observé est mobile, une erreur de traînage dépendant de la dynamique de l'objet perturbe la convergence. Le flot optique est souvent utilisé en complément pour compenser ces perturbations en estimant le mouvement de l'objet et limiter ainsi les l'erreur de positionnement [Chaumette 93, Bensalah 95, Wilson 94, Feddema 93]. D'autres travaux utilisent le flot optique pour

positionner le robot en poursuivant une trajectoire désirée [Colombo 95, Grosso 96]. Plutôt que d'utiliser le flot optique comme simple terme de compensation dans les boucles de commande, une autre approche consiste à réguler le flot optique ou, plus précisément, les paramètres du mouvement extraits du flot optique. Dans [Crétual 01a], les mouvements de quelques points de l'image sont intégrés et un asservissement visuel est appliqué, ceci permet de suivre un objet mobile dans une scène. Cependant, cette méthode ne diffère pas d'un asservissement visuel géométrique classique, seule la manière d'obtenir la position des points change. Une application plus intéressante consiste à réguler le temps avant contact obtenu à partir de l'estimation des paramètres d'un modèle affine du mouvement [Questa 95, Cipolla 97a, Cipolla 97b]. L'idée, théorisée la première fois par [Lee 76], est de freiner le robot dans la direction de l'axe optique de façon à ne pas entrer en collision avec l'objet planaire observé. Dans [Questa 95], le flot optique est également utilisé pour aligner le plan image avec le plan objet en utilisant le fait que les paramètres affines du mouvement sont fonction de l'orientation du plan objet. Dans [Crétual 01b], une approche plus robuste est développée en utilisant un modèle quadratique du mouvement. Dans ce dernier travail, les paramètres affines sont également utilisés pour suivre une trajectoire parallèlement à une surface plane.

3. Les applications en robotique aérienne

L'utilisation du flot optique connaît un grand succès dans le domaine de la robotique aérienne. En effet, le fait qu'un véhicule aérien se déplace dans l'espace 3-D représente un défi très important, d'autant plus que la mesure de vitesse n'est généralement pas disponible sur un tel engin. Dans le domaine de la robotique aérienne, de nombreux travaux utilisant le flot optique s'inspirent du comportement des insectes. Dans [Collett 80, Wagner 82, Tammero 02], les auteurs ont observé que les insectes utilisent le flot optique pour éviter les obstacles, pour stabiliser leur vol [Collett 75] et pour atterrir [Srinivasan 96, Srinivasan 00]. Des algorithmes de commande ont alors été élaborés et utilisés sur des robots pour valider ces observations.

3.1. Atterrissage automatique

Dans [Srinivasan 00], les auteurs montrent que les abeilles atterrissent avec un angle approximativement constant en conservant un flot optique constant. Ils observent également que la vitesse d'avance est proportionnelle à la vitesse de descente pendant l'atterrissage. Ils vérifient alors que ce comportement assure un atterrissage en douceur puisque la vitesse de l'insecte décroit exponentiellement en s'approchant du sol. Dans [Netter 99], une idée

similaire est proposée pour la commande d'un véhicule aérien en utilisant un capteur constitué de deux photo-récepteurs dirigés vers le sol. Le principe est alors de réguler le flot optique à une valeur constante tout en réduisant la vitesse d'avance de l'engin. Ainsi, le flot optique étant fonction du rapport entre la vitesse du véhicule et la distance au sol, la distance diminue proportionnellement à la vitesse. Cependant, dans cette approche, la vitesse d'avance n'est pas régulée directement, seul l'angle de tangage est contrôlé. Pour réduire la vitesse d'avance, l'angle de tangage est réduit [Ruffier 04, Ruffier 05]. Une approche similaire est proposée dans [Beyeler 09a] pour la commande d'un engin à voilure fixe. La mesure du flot optique est utilisée pour contrôler l'assiette du véhicule. Pour cela, des capteurs de souris sont utilisés. Les capteurs de souris sont des capteurs bas coût, légers et très rapides (environ 1500 mesures par seconde) qui fournissent directement une mesure du flot optique. Un capteur de souris est constitué de photo-diodes (18 × 18 pixels typiquement) qui détectent les mouvements relatifs par appariement de blocs (figure 2.2). Cependant les champs de vision sont très restreints et le capteur ne fournit qu'une seule mesure du flot optique. Dans [Beyeler 09b, Barber 05, Dahmen 09], les capteurs de souris sont disposés uniformément en plusieurs points dirigés vers le sol et vers l'avant. Des poids sont associés à chaque capteur et pour chaque entrée de commande (roulis et tangage). Pour l'atterrissage, les moteurs sont coupés, ce qui ralentit progressivement le véhicule. Le flot optique perçu diminue donc également, ce qui joue sur la commande de tangage et entraine l'atterrissage en douceur.

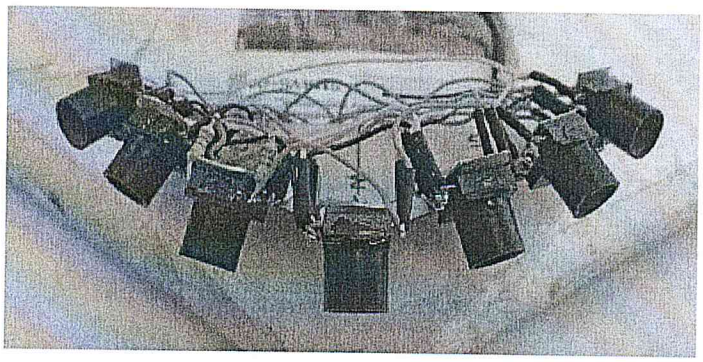


Fig.2.2 Les sept capteurs de souris embarqués sur un robot aérien [Beyeler 09b]

3.2. Suivi de terrain

On a vu que le principe de l'atterrissage automatique consistait à conserver le flot optique constant et à réduire la vitesse d'avance. Cependant, si la vitesse d'avance est également régulée à une valeur constante, on remarque que le véhicule est capable de suivre le sol à distance constante. Cette approche est utilisée dans les travaux de [Ruffier 05, Beyeler 09b].

Dans [Ruffier 05], un capteur dirigé vers le sol est utilisé pour suivre un terrain plat ou un terrain pentu. Cependant, les auteurs ont remarqué que le contrôleur ne fonctionne que pour un terrain de pente faible. Pour améliorer la technique, un deuxième capteur, dirigé vers l'avant du véhicule, a été ajouté [Ruffier 08]. En combinant les deux signaux acquis par les deux capteurs, le contrôleur d'altitude permet de suivre des terrains plus raides. Dans [Humbert 05b, Humbert 06], le flot optique est décomposé en série de Fourier et les coefficients de Fourier sont utilisés pour contrôler l'orientation et l'altitude du véhicule, la vitesse d'avance étant régulée séparément. Le flot optique étant calculé sur un champ de vision large, le contrôleur permet en théorie de suivre des terrains très pentus. Cependant, l'analyse de stabilité n'est réalisée qu'autour du point d'équilibre lorsque le terrain est plat. De plus, la robustesse du contrôleur n'est évaluée qu'en simulation.

D'autres approches utilisent la mesure de la vitesse du véhicule pour estimer la distance avec le terrain à partir du flot optique. Dans [Garratt 08], par exemple, un estimateur de distance basé sur la fusion des mesures inertielles et de la mesure du flot optique est réalisé et une loi de commande est élaborée pour contrôler la distance au terrain.

Cependant, à faible vitesse, le flot optique est faible et, par conséquent, l'estimateur de distance est très bruité. Ceci peut entrainer des instabilités dans le système.

3.3. Evitement d'obstacles

Le suivi de couloir et l'évitement d'obstacles sont des fonctions relativement proches et de nombreux travaux reprennent l'idée d'utiliser des capteurs latéraux pour éviter les obstacles [Zufferey 06, Green 08]. Cependant, les capteurs doivent également être dirigés vers l'avant pour détecter des obstacles frontaux. Le flot optique détecté à droite est comparé au flot optique détecté à gauche du véhicule et le contrôleur permet de diriger le véhicule du côté où il y a le moins de flot optique. La comparaison du flot optique à droite et à gauche revient en quelque sorte à estimer la divergence du flot optique. En effet, dans [Nelson 89], les auteurs montrent que le calcul d'une divergence directionnelle permet de déterminer la proximité d'obstacles. Dans [Zufferey 06], des mouvements rapides de rotation, appelés "saccades", sont déclenchés lorsque le flot optique dépasse un certain seuil. La direction des saccades est choisie en comparant le flot optique perçu à droite et à gauche du véhicule. Le principe de la saccade vient de l'observation des insectes [Wagner 82, Tammero 02]. On remarque que seule la composante du flot optique dépendant de la vitesse de translation peut être utilisée pour détecter la proximité des obstacles. Ainsi, lorsque les insectes changent leur direction de déplacement, le flot optique ne peut être utilisé, c'est pourquoi ils réalisent des mouvements

rapides de rotation. Une approche similaire est utilisée dans [Green 08]. Dans [Muratet 05], c'est un comportement de centrage qui est recherché en équilibrant le flot optique à droite et à gauche du véhicule. Pour compenser l'effet de la vitesse de rotation du véhicule sur la mesure du flot optique, des mesures inertielles sont utilisées. De plus, une estimation du temps à collision est utilisée pour détecter la présence d'un obstacle frontal. Lorsque le temps à collision devient inférieur à un certain seuil, le véhicule exécute un demi-tour. Dans [Beyeler 09b], un *OptiPilot* est développé. Le système de vision est constitué de plusieurs capteurs de souris et des gyroscopes sont utilisés pour compenser l'effet de la vitesse de rotation. Les capteurs sont disposés de manière optimale à 45° de la direction de déplacement du véhicule. Ils permettent de percevoir des mouvements à droite, à gauche, mais aussi en bas du véhicule. Ainsi, l'engin peut non seulement éviter des obstacles latéraux par des mouvements de roulis, mais aussi des obstacles frontaux par des mouvements de tangage. Notons que ce système permet également d'effectuer du suivi de sol et d'atterrir automatiquement.

4. Les applications en robotique mobile

Si l'utilisation du flot optique pour les robots manipulateurs se limite souvent à des tâches de positionnement ou de suivi de cibles, les applications potentielles en robotique mobile sont plus importantes en raison des nombreux défis à relever dans ce domaine. En effet, l'environnement dans lequel évolue un robot mobile est souvent peu ou mal connu, et souvent complexe. De plus, la mesure de l'état du robot est souvent imparfaite en raison du glissement des roues par exemple. Ces incertitudes posent problème pour la navigation sans collisions. En estimant le temps à collision ou le foyer d'expansion, et en équilibrant entre le flot optique détecté à droite et le flot optique détecté à gauche du robot, ce dernier peut élaborer des fonctions réactives qui garantissent sa sécurité pendant la navigation [Kahlouche 07 b].

4.1. Amarrage automatique

L'idée de freiner le robot en utilisant la mesure du temps à collision a été reprise en robotique mobile pour des applications d'amarrage automatique (*docking* en anglais).

Dans [SantosVictor 97], le temps à collision, mesuré à partir d'un modèle affine du mouvement, est utilisé pour le contrôle du déplacement face à une surface plane. On parle d'"*ego-docking*" si la caméra est embarquée sur l'engin et d'"*eco-docking*" si elle est fixée sur la plateforme d'amarrage. Dans [McCarthy 08], une approche plus robuste, basée sur la détection du foyer d'expansion, est proposée en partant du principe que la plateforme n'est

pas nécessairement alignée avec le robot et que la vitesse de rotation du véhicule peut perturber la détection du temps à collision.

4.2. Navigation dans un couloir

Une des utilisations majeures du flot optique en robotique mobile concerne le suivi de couloirs ou, plus généralement, l'évitement d'obstacles dans un environnement structuré.

Le défi principal est d'éviter la collision avec les murs des couloirs et ceci en empruntant le chemin correspondant au centre du couloir, ce comportement qui s'inspire du vol des insectes représente une navigation visuelle par reflexe, sans utilisation des modèles de l'environnement ou de planification des trajectoires.

La loi de contrôle est : $H = L - R$ .....(II.1)

Où L et R sont les moyennes des modules des vecteurs de flot optique calculés à droite et à gauche de l'image (figure 2.2)

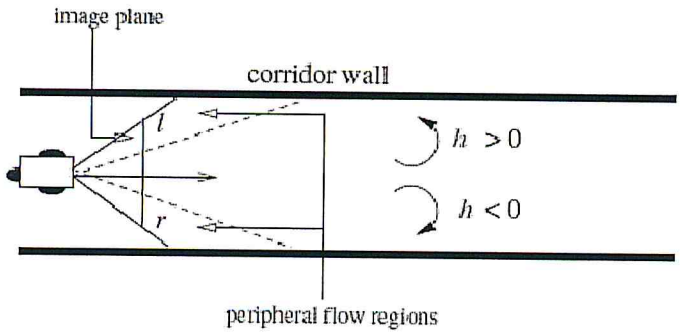


Fig.2.2 : Utilisation du flot pour la navigation dans un couloir

Dans [Kirchner 89], les auteurs montrent que les abeilles ont tendance à voler au centre d'un couloir étroit. Ils supposent alors que les insectes sont capables d'équilibrer le flot optique perçu par leurs deux yeux. Beaucoup de travaux utilisent ce principe de centrage pour la navigation des robots mobiles. Dans [Coombs 92, SantosVictor 95, Weber 97, Carelli 02, Hrabar 05], deux caméras latérales sont utilisées de façon à observer les deux murs simultanément. Le principe de la commande est alors d'équilibrer le flot optique perçu par les deux caméras comme le font les insectes. Dans [Dev 97, Duchon 98, Argyros 04, Humbert 10], un système grand angle ou panoramique est utilisé de façon à observer les deux murs avec une seule caméra.

Malgré les résultats encourageants obtenus avec l'approche par centrage, une autre technique a été étudiée suite à l'observation d'un autre comportement chez les abeilles. Dans [Serres 07, Ruffier 07], les auteurs montrent que les abeilles ne se centrent pas nécessairement, mais elles

préfèrent parfois suivre l'un des 2 murs du couloir, notamment lorsqu'il est large. L'observation de ce comportement chez l'abeille montre bien que les insectes ont besoin de percevoir du flot optique pour naviguer. Or lorsque le couloir est large, le flot optique perçu au centre est faible, l'insecte préfère alors se rapprocher de l'un des deux murs. On ne parle alors plus de suivi de couloirs mais de suivi de murs. On remarque également que si l'un des deux murs du couloir est retiré, les insectes continuent de suivre le mur opposé. Pour valider cette observation, un régulateur est proposé dans [Serres 08] pour la commande d'un aéroglisseur en utilisant deux capteurs latéraux. Le principe est de réguler le flot optique perçu sur l'un des deux murs à une valeur constante, le mur choisi étant celui qui est le plus proche à l'instant initial (celui sur lequel on perçoit le plus de flot optique). Ainsi, le flot optique étant fonction du rapport entre la vitesse du véhicule et la distance à l'objet observé, la distance au mur se stabilise à une valeur dépendant de la consigne de flot optique et de la vitesse d'avance de l'engin. De plus, on a observé que l'abeille a tendance à ralentir lorsque le couloir se rétrécit, un second contrôleur est donc utilisé pour réguler la vitesse d'avance en fonction du flot optique total perçu sur les deux murs.

En résumé, lorsque le couloir est étroit, le centrage semble plus adapté et lorsque le couloir est plus large, c'est le suivi de mur qui semble le plus adapté. Notons que les méthodes de centrage ou de suivi de murs peuvent très bien s'appliquer à des environnements plus complexes (labyrinthes, environnements encombrés). Dans ce cas, on parle plutôt d'évitement d'obstacles [Coombs 92, SantosVictor 95]. Cependant, contrairement au cas particulier du suivi de couloir, l'utilisation de deux capteurs latéraux ne suffit pas. En effet, pour éviter des obstacles frontaux, les capteurs doivent également pointer vers l'avant.

5. Comprendre et interagir avec l'environnement par le mouvement

Lorsqu'un point d'observation se déplace dans un environnement, la forme de la lumière réfléchie change d'une manière continue créant un flux optique. Ce flux optique véhicule de nombreuses informations comme : la disposition des obstacles dans l'environnement perçu par le robot, le Focus d'Expansion (FOE), et l'information de la profondeur, ainsi la structure tridimensionnelle de l'environnement peut être déterminée en analysant une séquence d'images monoculaire perçue par un observateur.

5.1. Calcul du Foyer d'expansion à partir du flot optique

Le foyer d'expansion (FOE) correspond à la projection du vecteur de translation sur le plan de l'image. C'est aussi le point à partir duquel les vecteurs du flot optique divergent (voir figure2.3).Une manière simple pour déterminer la position du FOE est de calculer le point d'intersection des lignes droites obtenues en prolongeant les vecteurs du flot optique.

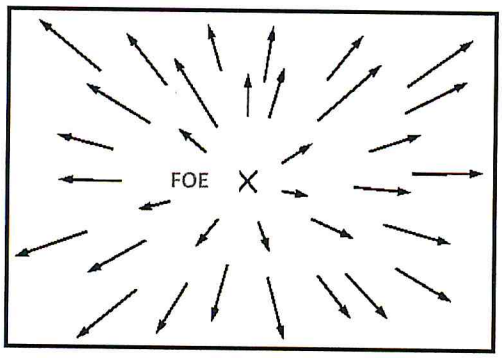


Figure2.3 : Focus d'expansion

Une autre manière pour déterminer la position du FOE, est d'employer la distribution des signes des composantes horizontales et verticales des vecteurs de flux optique (par exemple positif: à droite, négatif: à gauche) [Negahdaripour 89].

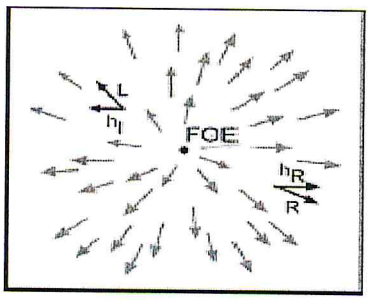


Figure2.4 : détermination de la position du FOE

D'après la figure2.4, nous constatons que le centre d'expansion FOE est calculé sur la base que les vecteurs du flot optique sont orientés dans des directions spécifiques relatives au FOE, la composante horizontale h_L d'un vecteur du flux optique L situé du coté gauche du FOE se dirige à gauche, tandis que la composante horizontale h_R d'un vecteur du flux optique R situé du coté droit du point FOE se dirige à droite. Donc, on peut dire que la position horizontale du FOE, correspondant à la position dans laquelle une majorité de composantes horizontales divergent, peut être estimée en utilisant un comptage simple pour retrouver les signes des composantes horizontales centrés en chaque point de l'image.

Au point où la divergence est maximale, la différence entre le nombre de composantes de h_L à gauche du FOE et le nombre des composantes de h_R à droite du FOE sera minimale. De même nous pouvons estimer l'endroit vertical du FOE en identifiant la position dans laquelle une majorité de composantes verticales (v_R et v_L) divergent.

La figure 2.5 montre le résultat de calcul du FOE, il est représenté sur l'image par un carré blanc encadré en rouge.

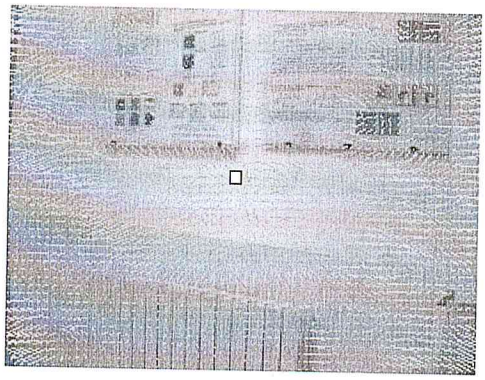


Figure2.5 : résultat de calcul du FOE

5.2. Calcul du de la profondeur à partir du flot optique

Dans ce qui suit nous allons monter deux cas de calcul de la profondeur d'un point par rapport à l'observateur en utilisant le flot optique. Dans le premier cas, aucune connaissance apriori sur la scène n'est disponible, tandis que le deuxième cas suppose la connaissance de la profondeur d'un point de la scène.

a. <u>Aucune connaissance sur la scène</u>

Nous pouvons calculer la profondeur d'un point en utilisant uniquement le vecteur du flot optique de la manière suivante [Verri 89]:

Soit *P(x, y, z)* un point se déplaçant avec une vitesse constante *(u, v, w) = (dx/dt, dy/dt,dz/dt)*.

L'équation (II.2) donne une information sur la profondeur suivant l'axe Z.

$$\frac{D}{V} = \frac{Z}{W} \quad \text{.....(II.2)}$$

La partie gauche de l'équation (II.2) se calcul en 2D (sur le plan image)

D: la distance entre le FOE et un point (x,y) de l'image.

V : le flot optique d'un point de l'image

La partie droite de l'équation se calcul en 3D

Z : la distance suivant l'axe z d'un point par rapport au plan image (la profondeur).

W : la vitesse *dz/dt* suivant l'axe Z.

Ainsi le rapport *distance/vitesse* d'un point de l'image est égal au rapport *distance/vitesse* d'un point 3D.

La figure 2.6 montre l'image de profondeur obtenue en utilisant l'équation II.2. Les images résultats sont codées en niveau de gris de façon à ce que les points les plus profonds ont la couleur noire tandis que les points les plus proches sont plus clairs.

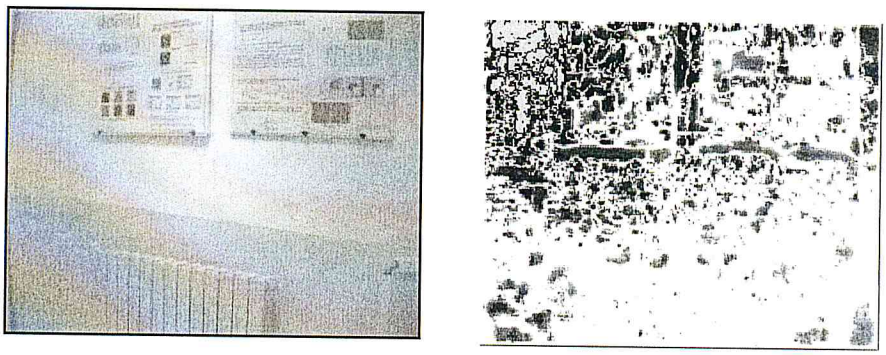


Fig.2.6 : Résultat du calcul de la profondeur sans aucune connaissance sur la scène

b. <u>En connaissant la profondeur d'un point</u>

Connaissant la profondeur de n'importe quel point, nous pouvons déterminer la profondeur de tous les autres points ayant la même vitesse de translation w, par la relation suivante [Verri 89]:

$$z_2(t) = \frac{z_1(t) V_1(t) D_2(t)}{D_1(t) V_2(t)}$$

$Z_1(t)$: distance connue

$Z_2(t)$: distance recherchée.

La figure 2.7 montre le résultat de la profondeur pour la même scène que la figure 2.6. Il est visiblement clair que la profondeur calculée connaissant celle d'un autre point est mieux que celle calculée sans aucune connaissance sur la scène.

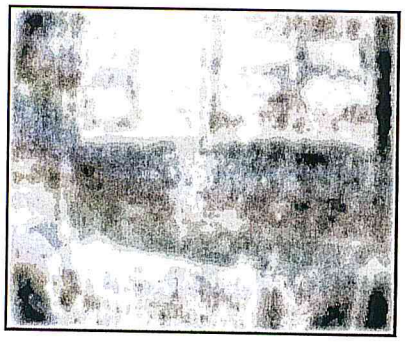


Fig.2.7 : Résultat du calcul de la profondeur en connaissant la profondeur d'un point de la scène

La figure 2.8 montre quelques résultats obtenus pour la détermination de la profondeur connaissant celle du premier point de l'image Dans cette expérience la profondeur de ce point est égale à 1000 mm.

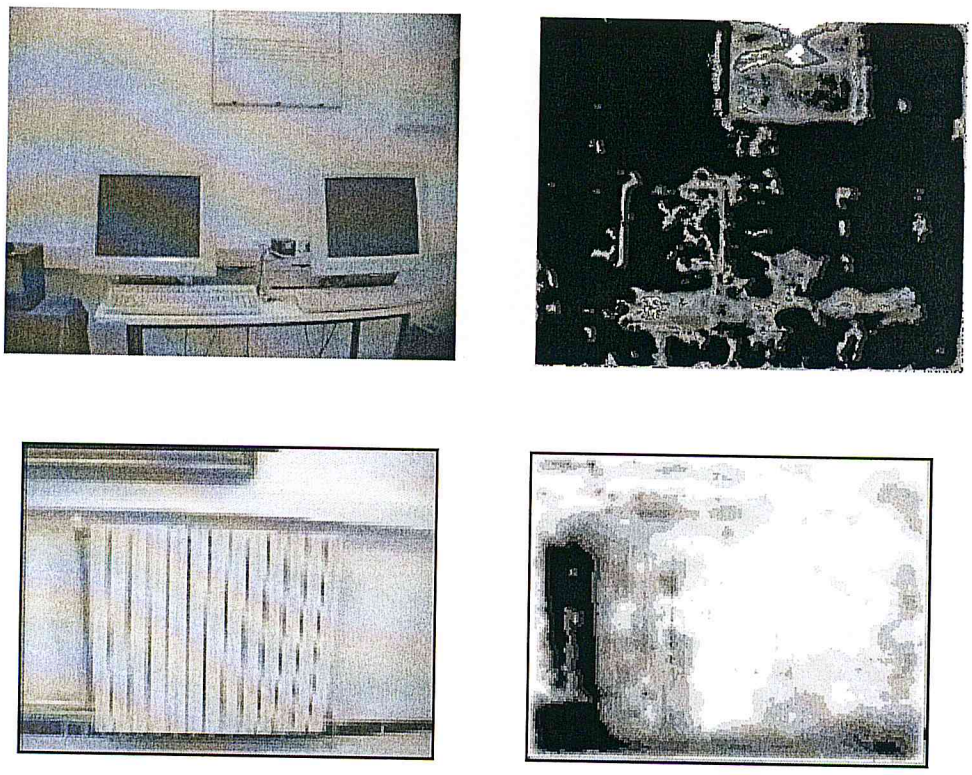


Fig.2.8 : Quelques résultats du calcul de la profondeur

5.3. La stratégie de la balance

C'est une règle de contrôle qui peut être utilisée par un robot pour éviter les obstacles, ou bien pour poursuivre une cible en mouvement. Pour ce faire, le robot devrait essayer d'équilibrer la quantité de changement d'intensité de l'image des côtés gauches et droits du centre de son champ visuel. Afin de faire ceci, le robot peut déterminer la direction de sa rotation à partir de la différence entre la somme des modules des vecteurs du flot optiques gauches et la somme des modules des vecteurs du flot optiques droits. Le robot tournera du côté opposé de la partie de l'image qui présente plus de mouvement, puisque ceci indique un obstacle stationnaire proche.

Cette règle de contrôle appelée stratégie de la balance est traduite par la formule suivante [Warren 88]:

$$\Delta(F_L - F_R) = \frac{\sum\|\vec{w}_L\| - \sum\|\vec{w}_R\|}{\sum\|\vec{w}_L\| + \sum\|\vec{w}_R\|} \quad \text{..... (2.3)}$$

Où, $\Delta(F_L - F_R)$ est la différence des forces des deux cotés du robot et $\sum\|\vec{w}\|$ est la somme des modules des vecteurs du flux optique en un coté du FOE.

Dans la figure 2.9, la somme des modules des vecteurs du flot optique du côté gauche (2588) est plus grande que celle du coté droit (2076). Par conséquent le robot devrait tourner à droite pour éviter l'obstacle le plus proche (le robot ATRV2).

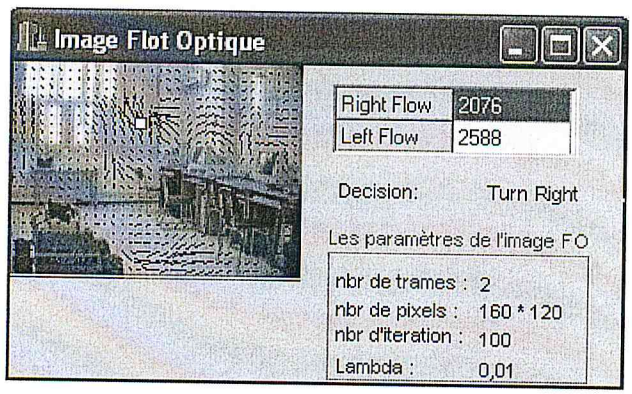


Fig.2.9 : Résultat de la stratégie de la balance

5.4. Reconstruction 3D des points par le mouvement

La figure 2.10 représente le modèle sténopé décrivant la projection perspective qui transforme un point de l'espace 3D en un point de l'image 2D.

Soit un point *O* dans le plan image appelé point principal et soit une droite perpendiculaire au plan image passant par *O*, l'axe optique. Soit un point *F* placé sur l'axe optique à une distance *f* du plan image. Le point *F* est le centre de projection et *f* est la distance focale. Un point P se projette dans le plan image le long d'une droite passant par P et *F*.

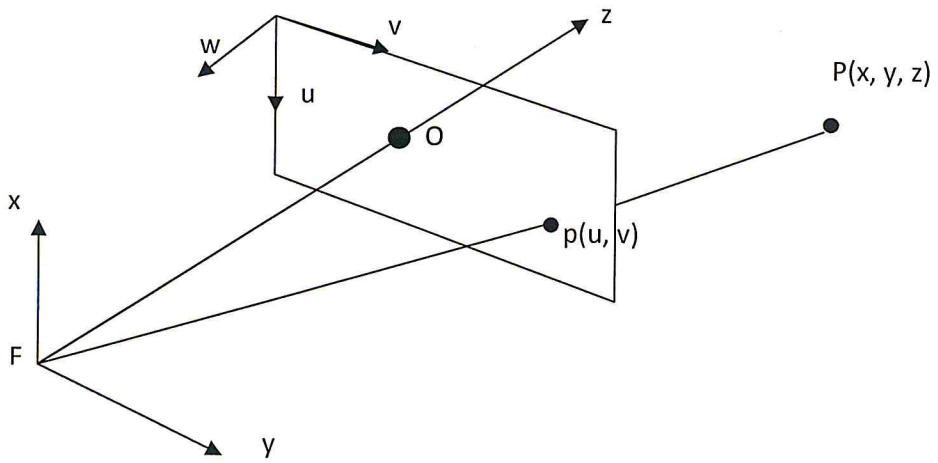


Fig.2.10 : Modèle de la caméra

Soit (Rc) le repère lié à la caméra, pris comme repère du monde, le plan *(xy)* de ce repère est parallèle au plan image et l'axe des *z* est confondu avec l'axe optique. L'origine de ce repère se trouve en *F*. Le mouvement de la caméra est caractérisé par les vecteurs vitesses translation *V* et rotation *R*. Par rapport au repère (Rc) attaché à la caméra, un point *P (X,Y,Z)* se déplace avec la vitesse suivante : $dP/dt = -V - R \times P$ (produit vectoriel)

Soit *p(x,y)* : la projection de *P(X,Y,Z)* sur l'image. Les coordonnées de *p* sont calculées comme suit :

$x = X/Z$

$y = Y/Z$

Le modèle interne linéaire de la caméra conduit à :

$(u-u_0) / K_u f = x$

$(v-v_0) / K_v f = y$

$K_u f$, $K_v f$ appelés paramètre interne de la caméra , obtenus par calibrage.

L'équation (II.4) lie les paramètres internes de la caméra ainsi le mouvement de cette dernière au vecteur flot optique, ce qui permet de déterminer la coordonnée Z d'un point 3D [Chavand 98].

$$\frac{dy}{dt} = \frac{1}{z^{-1}} \begin{pmatrix} -1 & 0 & x \\ 0 & -1 & y \end{pmatrix} T + \begin{pmatrix} xy & -(1+x^2) & y \\ (1+y^2) & -xy & -x \end{pmatrix} R \quad \text{.....(II.4)}$$

Tel que $\frac{dy}{dt}$: Exprime le flot optique.

Dans ce qui suit, nous allons appliquer cette méthode de reconstruction par le flot optique, pour reconstruire quelques points de la scène, et en utilisant les paramètres interne de la caméra : alpha u= -749,97, alpha v= 627,31, u0= 345,8 et v0= 398,6 (annexe B).

La figure 2.11 (à gauche) montre les 12 points choisis sur l'image de taille 600X550 de la mire de calibrage, dans le but de les reconstruire, ainsi que l'image du flot optique calculé par la méthode Horn et Schunk (à droite).

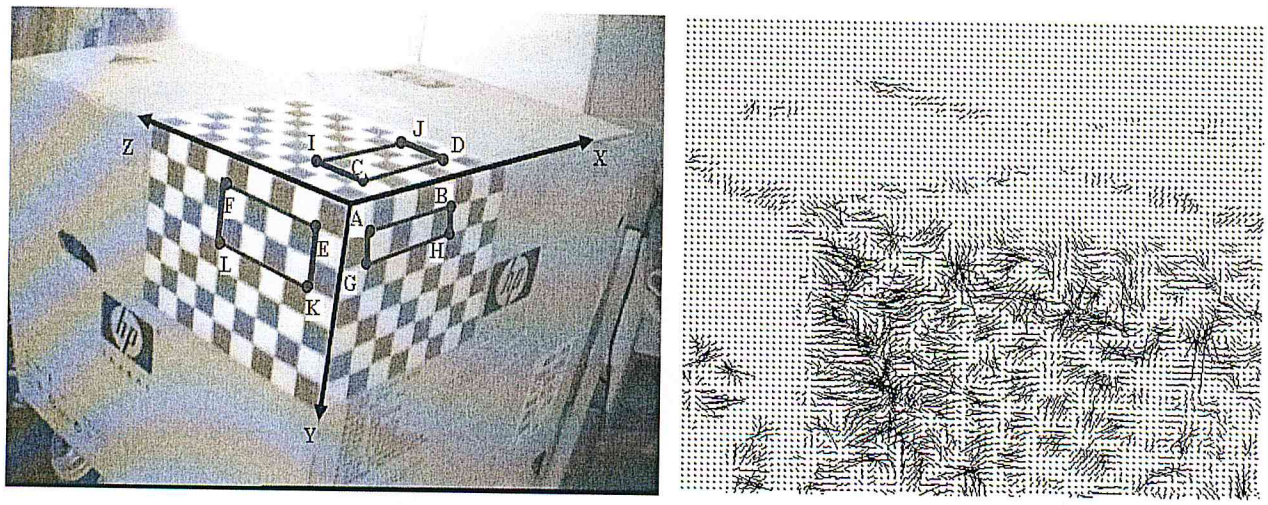


Fig.2.11 : A gauche les points choisi de la mire (A,B,C,D,E,F,G,H,I,J,K,L), a droite l'image du flot optique

Le résultat de la reconstruction des points (A,B,C,D,E,F,G,H,I,J,K,L) par rapport au repère lié à la caméra est illustré dans la figure 2.12 sous deux ongles de vue différents.

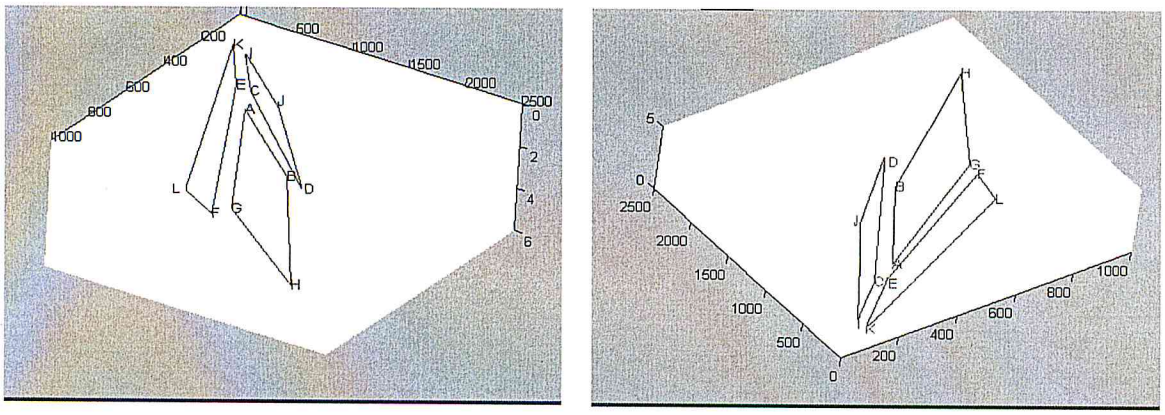


Fig.2.12 : Points reconstruits (A,B,C,D,E,F,G,H,I,J,K,L)

Nous remarquons que, dans cette reconstruction, les grandeurs réelles ne sont pas conservées tandis que les formes et le parallélisme sont pratiquement conservés. Notons que le résultat de la reconstruction 3D dépend fortement du résultat de calibrage de la caméra ainsi que, du résultat du calcul de flot optique. Les erreurs cumulées lors de ces deux étapes influent considérablement sur le résultat de la reconstruction 3D, et par conséquence l'amélioration de ces deux étapes conduira forcément à un résultat de reconstruction 3D meilleurs.

6. Conclusion

Dans ce chapitre, nous avons présenté un état de l'art sur l'intérêt du flot optique pour la robotique, tout d'abord pour des applications sur les robots manipulateurs, puis sur les robots aériens et enfin sur les robots mobiles, Nous remarquons que pour chaque domaine d'application, le flot optique est utilisé pour des fonctions différentes. Pour les robots manipulateurs, ce sont plutôt des tâches de suivi ou de positionnement qui sont visées. En robotique aérienne, on cherche avant tout à assurer la non-collision avec les obstacles et à atterrir automatiquement. En robotique mobile, ce sont le suivi de couloirs ou l'amarrage automatique qui sont utilisées. Néanmoins, nous remarquons que ces applications utilisent toutes des principes similaires, dans lesquelles les informations véhiculées par le vecteur du flot optique telle que la stratégie de balance, le foyer l'expansion et la profondeur sont calculées, ensuite nous avons montré pratiquement comment extraire du vecteur du flot optique ces informations. Le chapitre suivant portera sur la mise en œuvre des différentes techniques du flop optique. Une comparaison qualitative et quantitative des résultats sera donnée, ensuite l'application du flot optique dans une tâche d'évitement d'obstacle sera présentée et discutée.

Chapitre3

IMPLEMENTATION ET MISE EN ŒUVRE

Chapitre3 : Implémentation et mise en œuvre

Dans ce chapitre nous procédons à l'implémentation de quelques algorithmes d'estimation de mouvement présentés dans le premier chapitre notamment les méthodes différentielles telles que la méthode de Horn & Schunck, la méthode de Lucas et Kanade, et la méthode de Lucas et Kanade pyramidale, ainsi qu'une méthode de corrélation : le Bloc matching. Une comparaison entre les résultats de ces algorithmes, par rapport à la qualité des résultats d'une part et par rapport au temps de calcul d'une autre part, est présentée. Par la suite nous montrons l'application de ces algorithmes dans une tâche d'évitement d'obstacles appliquée au robot mobile B21r.

1 Implémentation de la méthode Horn et Schunck

Rappelons que la méthode de Horn et Schunck est une méthode différentielle basée sur l'hypothèse de la conservation de la luminosité entre les images consécutives. Dans cette méthode, une contrainte de lissage globale est imposée sur toute l'image, donnée par le terme de lissage λ dans le système d'équations (I.16) vu au chapitre 1:

$$\begin{cases} u^{n+1} = \bar{u}^n - \frac{E_x\bar{u}^n + E_x\bar{v}^n + E_x}{\lambda^2 + E_x^2 + E_y^2}\lambda E_x \\ v^{n+1} = \bar{v}^n - \frac{E_x\bar{u}^n + E_x\bar{v}^n + E_x}{\lambda^2 + E_x^2 + E_y^2}\lambda E_y \end{cases} \begin{cases} u^{n+1} = \bar{u}^n - \frac{E_x\bar{u}^n + E_x\bar{v}^n + E_x}{\lambda + (E_x^2 + E_y^2)}\lambda E_x \\ v^{n+1} = \bar{v}^n - \frac{E_x\bar{u}^n + E_x\bar{v}^n + E_x}{\lambda + (E_x^2 + E_y^2)}\lambda E_y \end{cases} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

(I.16)

Où E_x, E_y, E_t : sont les dérivées partielles de E respectivement par rapport à x,y et à t, calculées par approximation à des interpolateurs discrets en trois dimensions (horizontal, vertical et temporel), de la manière suivante:

$$\begin{cases} E_x(i,j,t) = \frac{1}{4}(E(i+1,j,t) + E(i+1,j,t+1) + E(i+1,j+1,t) + E(i+1,j+1,t+1)) \\ \qquad - \frac{1}{4}(E(i,j,t) + E(i,j,t+1) + E(i,j+1,t) + E(i,j+1,t+1)) \\ E_y(i,j,t) = \frac{1}{4}(E(i,j+1,t) + E(i,j+1,t+1) + E(i+1,j+1,t) + E(i+1,j+1,t+1)) \\ \qquad - \frac{1}{4}(E(i,j,t) + E(i,j,t+1) + E(i+1,j,t) + E(i+1,j,t+1)) \\ E_t(i,j,t) = \frac{1}{4}(E(i,j,t+1) + E(i,j+1,t+1) + E(i+1,j,t+1) + E(i+1,j+1,t+1)) \\ \qquad - \frac{1}{4}(E(i,j,t) + E(i,j+1,t) + E(i+1,j,t) + E(i+1,j+1,t)) \end{cases}$$

(III.1)

La figure 3.1 montre la relation temps-espace entre les mesures des dérivées partielles pour un point situé au centre d'un cube où chaque dérivée estimée est une moyenne des 4 premières différences prises sur les mesures adjacentes du cube.

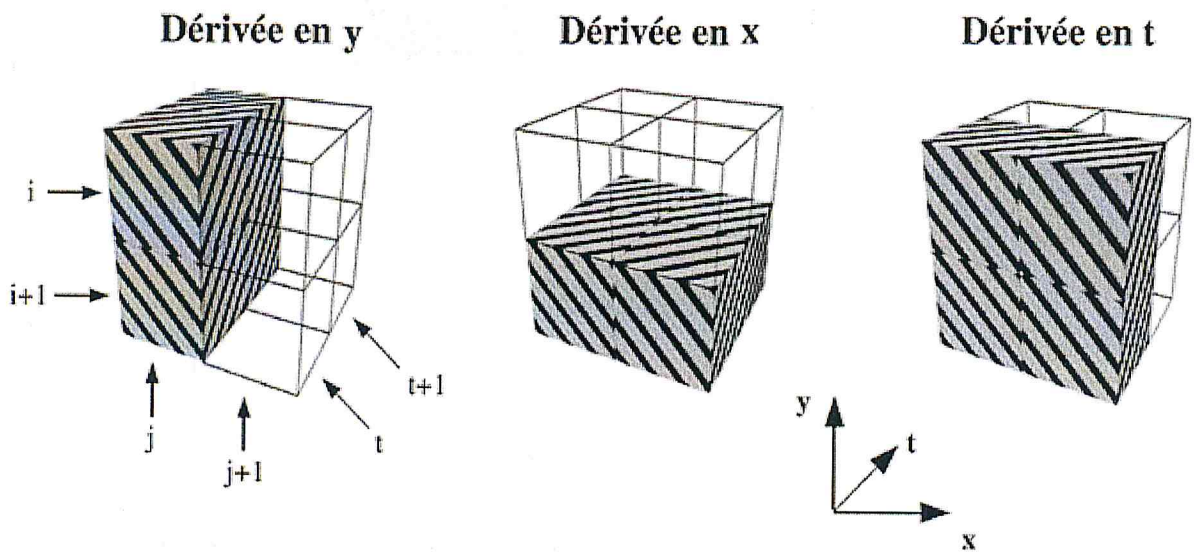


Fig.3.1 : La mesure des dérivées partielles au centre d'un cube.

La méthode de Horn et Shunck est présentée par le pseudo-code ci-après :

Début

Pour chaque pixel (i,j) de l'image **faire**

- calculer les valeurs E_x, E_y, et E_t ;
- initialiser $u(i, j)$ et $v(i, j)$ à u_0 et v_0 ;

Fin pour

- choisir une valeur lambda
- choisir un nombre $n_0 >= 1$ d'itérations;

$n:=1;$

Tant que $n <= n_0$ **faire**

Pour chaque pixel (i,j) de l'image **faire**

- Calculer $\bar{u}^n$ et $\bar{v}^n$ par le système (I.16).
- Calculer $\alpha := \dfrac{E_x \bar{u} + E_y \bar{v} + E_t}{\lambda^2 + E_x^2 + E_y^2}$;
- Calculer $u^{n+1} = \bar{u}^n - \alpha \cdot E_X$; et $v^{n+1} := \bar{v}^n - \alpha \cdot E_Y$

Finpour;

$n := n+1;$

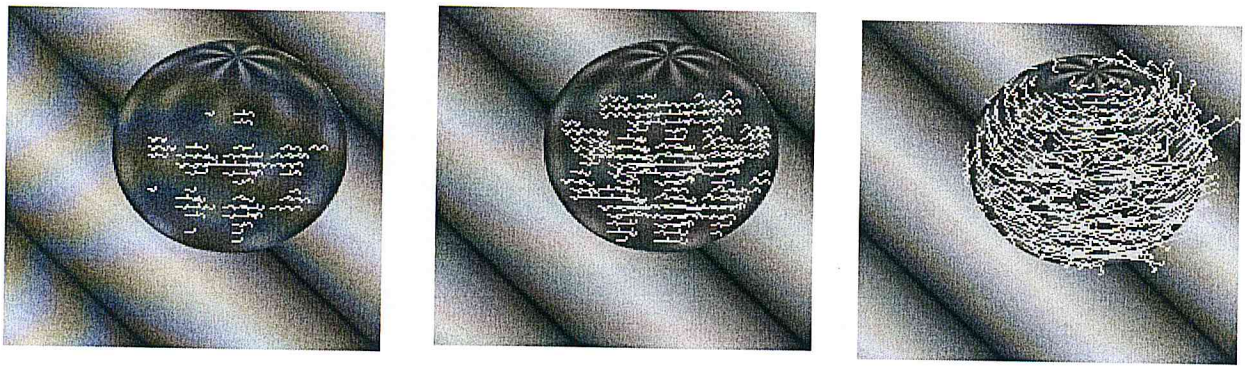
Fin tant que

Fin;

L'algorithme Horn et Schunck possède deux paramètres qui sont λ et le nombre d'itérations N, nous allons, dans ce qui suit, étudier l'influence de ses paramètres sur la qualité des résultats.

Influence du paramètre λ

Dans la figure3.2 nous étudions l'influence du paramètre de lissage λ, et donc nous fixons le nombre d'itération N, et nous faisons varierλ.



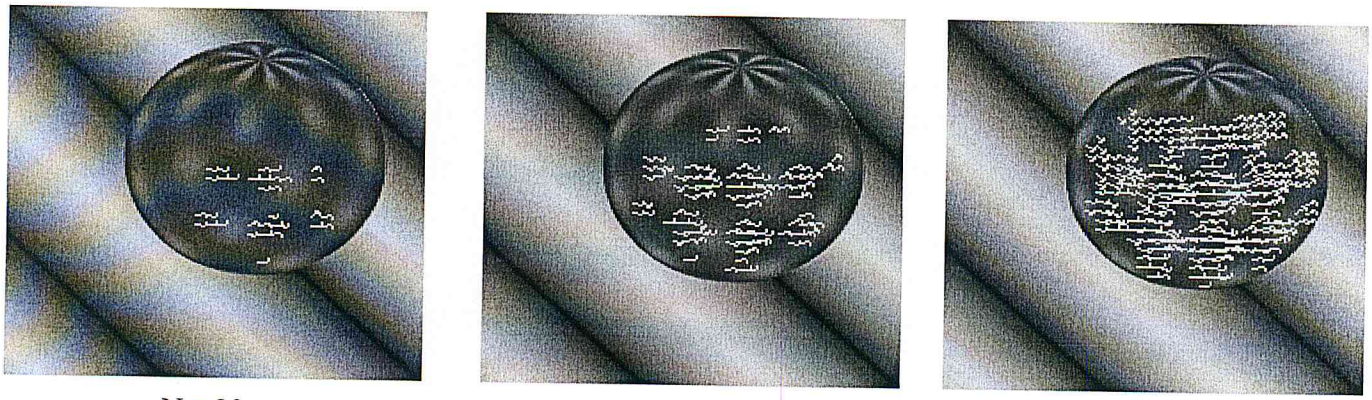
λ=0.005 λ=0.01 λ=1

Fig.3.2 : Influence du paramètre λ obtenu par l'algorithme de Horn et Schunck implémenté.

Nous remarquons que pour les valeurs de λ inférieures à 1, les résultats obtenus sont satisfaisants, où le champ de mouvement est dense, et fidèlement représenté par les vecteurs du flot optique. Cependant, plus λ augmente plus nombre de vecteurs erronés augmente.

Influence du paramètre N

Dans la figure3.3 nous étudions l'influence du paramètre nombre d'itérations N, et donc nous fixons le paramètre λ à 0.01 et nous faisons varier le nombre d'itérations N.



N = 20 N = 50 N = 100

Fig.3.3 : Influence du paramètre N obtenu par l'algorithme de Horn et Schunck implémenté.

Nous avons remarqué que plus le nombre d'itérations augmente, plus les résultats de l'estimation de mouvement sont précis et denses, mais pratiquement les résultats ne changent pas, ou changent peu, pour des valeurs de N supérieures à 100, et ceci au détriment du temps d'exécution.

Il est clair que l'ajustement des paramètres d'un algorithme est une tâche contraignante pour toutes les applications, car ces paramètres peuvent changer d'une application à une autre et d'une image à une autre dans une même application. L'ajustement des paramètres de l'algorithme de Horn et Schunck revient à trouver un compromis entre la qualité du résultat d'une part et le temps d'exécution d'une autre part.

Notons qu'il existe des bibliothèques spécialisées en vision artificielle, regroupant plusieurs algorithmes de traitement d'images. La plus connue et la plus utilisée par la communauté des chercheurs est la bibliothèque OpenCV. Elle intègre des algorithmes de vision améliorés en temps de calcul et optimisés en code C et C++. Les paramètres de convergence des algorithmes tels que les critères d'arrêts, les termes de lissage, les tailles des fenêtres, etc., sont gérés par cette bibliothèque d'une manière transparente. Aussi, des prétraitements et des post traitements sont rajoutés à beaucoup d'algorithmes de cette bibliothèque pour améliorer le résultat final, tout en laissant à l'utilisateur la possibilité de changer ces paramètres par défaut. La figure3.4 montre le résultat du calcul du flot optique pour la méthode Horn et Schunck en utilisant la bibliothèque OpenCV.

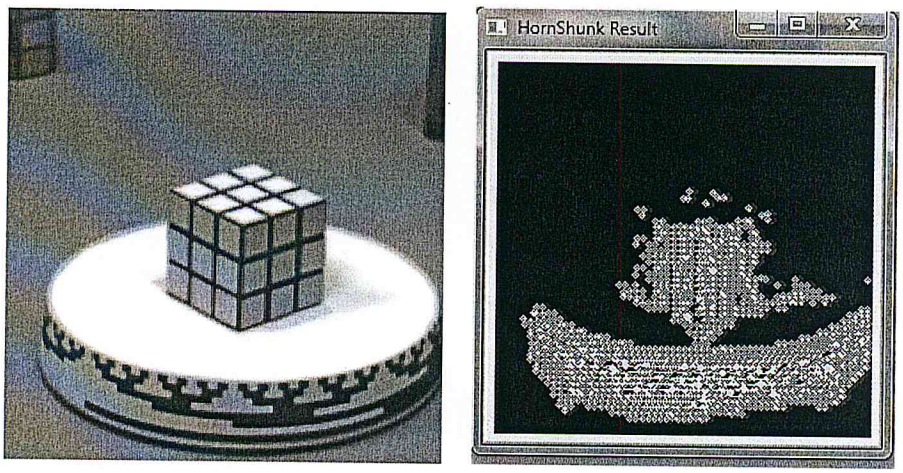


Fig.3.4 : Résultat du flot optique par la méthode Horn et Schunck

Bien que la méthode de Horn et Schunck soit sensible aux conditions d'éclairage, elle fournit un résultat précis dans la plus part des cas. Cependant, la violation des contraintes imposées par les méthodes différentielles, comme dans le cas d'un grand déplacement où la contrainte de conservation de la luminance n'est pas respectée, conduit à un résultat erroné.

Un autre inconvénient de cette méthode réside dans la non préservation des discontinuités des mouvements. La figure 3.5 montre un cas de deux objets adjacents avec des mouvements rotationnels dans des sens opposés, le champ de mouvement calculé dans ce cas, est lisse même au frontière du mouvement entre les deux objets en rotation, ceci est due à la contrainte de lissage imposée par la méthode Horn et Schunck.

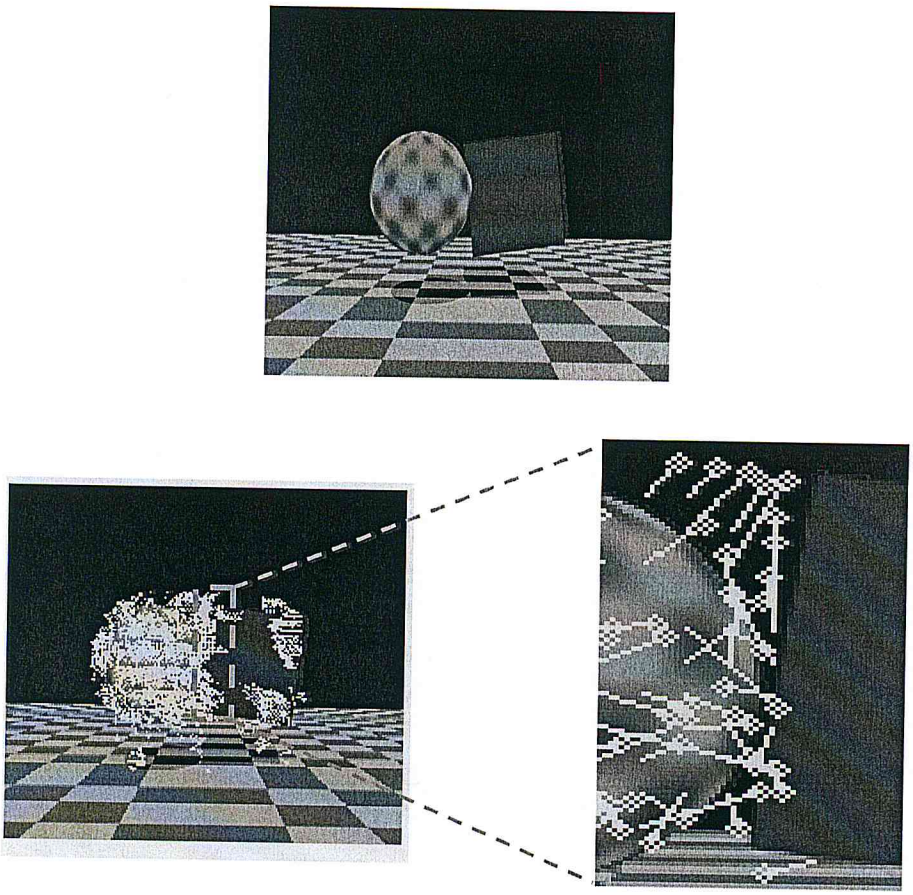


Fig. 3.5 : Cas du lissage de la discontinuité de mouvement

2. Implémentation de la méthode Lucas et Kanade

Dans le chapitre 1 nous avons vu que l'algorithme de Lucas-Kanade (LK), a apporté une solution à l'équation du flot optique (CFO), en se basant sur l'hypothèse de cohérence de mouvement localement, ce qui revient à résoudre le système d'équations suivant :

$$\underbrace{\begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} u \\ v \end{bmatrix}}_{d} = \underbrace{-\begin{bmatrix} \sum I_x I_t \\ \sum I_y I_t \end{bmatrix}}_{b} \quad \text{..... (III.2)}$$

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \sum I_x{}^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y{}^2 \end{bmatrix}^{-1} \begin{bmatrix} -\sum I_x I_t \\ -\sum I_y I_t \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \sum I_x{}^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y{}^2 \end{bmatrix}^{-1} \begin{bmatrix} -\sum I_x I_t \\ -\sum I_y I_t \end{bmatrix}$$

..... (III.3)

La solution du système d'équations sera :

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \frac{-\sum I_y{}^2 \sum I_x I_t + \sum I_x I_y \sum I_y I_t}{\sum I_x{}^2 \sum I_y{}^2 - \sum I_x I_y \sum I_y{}^2} \\ \frac{\sum I_x I_t \sum I_x I_t - \sum I_x{}^2 \sum I_y I_t}{\sum I_x{}^2 \sum I_y{}^2 - \sum I_x I_y \sum I_y{}^2} \end{bmatrix} \quad \text{.....(III.4)}$$

Le système (III.2) est resolvable si les conditions suivantes sont vérifiées [Gary 07]

- $(A^T A)$ est inversible : ne possède pas de valeurs propres nulles.

- $(A^T A)$ ne possède pas des valeurs propres trop petites.

- $(A^T A)$ est bien conditionnée : les valeurs propres λ_1 et λ_2 vérifient que le rapport λ_1 / λ_2 ne doit pas être trop grand ($\lambda_1 >> \lambda_2$).

Le pseudo code suivant résume la méthode de Lucas et Kanade pour le calcul du flot optique :

W : fenêtre de recherche

I_t: Image à l' instant *t*

Max : nombre d'itération maximum

Tant que *(nombre d'itération < Max) faire*

Pour tout pixel de l'image I_t faire

Pour tout pixel de la fenêtre W faire

- Calculer la matrice A= $\begin{bmatrix} \sum I_x{}^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y{}^2 \end{bmatrix} \begin{bmatrix} \sum I_x{}^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y{}^2 \end{bmatrix}$
- Calculer la matrice b = $\begin{bmatrix} -\sum I_x I_t \\ -\sum I_y I_t \end{bmatrix}$
- Calculer la solution $\Delta d = (A)^{-1} b$, du système (III.3)
- Calculer l'incrément du vecteur flot optique par les équations (III.4)

Fin Pour (boucle sur la fenêtre W)

Fin Pour (boucle sur l'image)

- Mettre à jours le vecteur flot optique $d = d + \Delta d$

Fin Tant que

Nous allons maintenant étudier le comportent de l'algorithme de Lucas et Kanade dans différents cas de figures :

- **Zone contenant un contour simple :** Dans une zone de contour simple comme celle de la figure 3.6 (zone 1), le gradient est orienté dans une seule direction, et la matrice $(A^T A)$ possède une grande valeur propre λ_1 mais une petite valeur propre λ_2.

- **Zone contenant une région homogène :** Dans une région homogène comme celle de la figure 3.6 (zone 2), la valeur du gradient est faible, et donc la matrice $(A^T A) \approx 0$, et possède des valeurs propres faibles ou nulles.

- **Zone contenant une région texturée** : Dans une zone de région texturée comme celle de la figure 3.6 (zone 3), le gradient est orienté dans les deux directions (x et y), et $(A^T A)$ possède deux grandes valeurs propres.

Fig.3.6 : Différents cas pour l'algorithme Lucas et Kanade

En résumé pour s'assurer de l'existence et la stabilité de la solution du système III.2, la matrice $(A^T A)$ doit avoir deux valeurs propres λ_1 et λ_2 importantes.

La figure 3.7 montre le résultat de calcul du flot optique par l'algorithme de Lucas et Kanade pour la même séquence de la figure 3.5. Nous remarquons que le fond de l'image, qui est une région homogène, contient beaucoup de vecteurs erronés, ce qui est en accord avec la théorie.

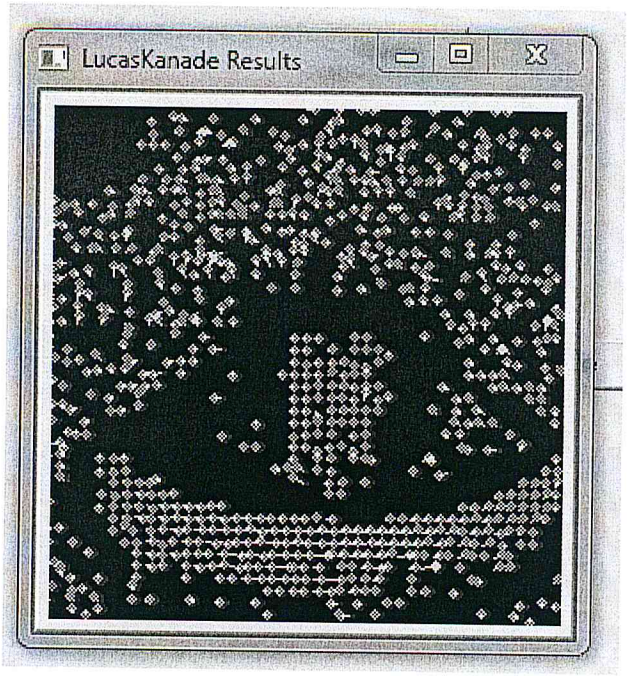


Fig. 3.7 : Résultat du flot optique par la méthode Lucas et Kanade

Cette méthode fournit un résultat dense de champ de mouvement, mais peu précis en particulier dans les régions homogènes. La méthode s'appuie uniquement sur les informations locales d'une fenêtre, et par conséquence, elle souffre de ne pouvoir mesurer que de faibles déplacements, car les grands mouvements peuvent déplacer les points à l'extérieur de la fenêtre locale ce qui rend leurs suivis impossibles par l'algorithme, un autre inconvénient réside dans la violation de la contrainte du mouvement constant dans une fenêtre, qui apparait dans le cas d'un mouvement non rigide où un pixel peut avoir un mouvement différent de ses voisins.

4. Implémentation de la méthode Lucas et Kanade pyramidale

C'est une méthode multi-résolutions qui consistent à estimer le flot optique sur des résolutions plus faibles et à revenir progressivement ver/s la résolution initiale afin d'affiner l'estimation.

Elle est donnée par le pseudo code suivant :

- Calcul du flot optique du niveau le plus haut par la méthode Lucas et Kanade itérative.

Pour chaque niveau *i de* la pyramide faire

- Soit *u(i-1), v(i-1)* le vecteur flot optique du niveau *i-1*
- Propager le flot du niveau *i* pour créer $u^*(i)$, $v^*(i)$, matrice de double résolution ; (figure3.8 : étape1)
- Multiplier $u^*(i)$, $v^*(i)$ par 2;
- Calculer l'image I_t à partir des blocs déplacés par $u^*(i)$, $v^*(i)$; (figure3.8 : étape 2)
- Appliquer Lucas et Kanade au niveau *i*, pour calculer *u'(i), v'(i)* ; (figure 3.8 : étape 3)
- Corriger le flot optique du niveau *i* (figure 3.8.étape 4): $u(i)=u^*(i)+u'(i)$, $v(i)=v^*(i)+v'(i)$;

Fin Pour

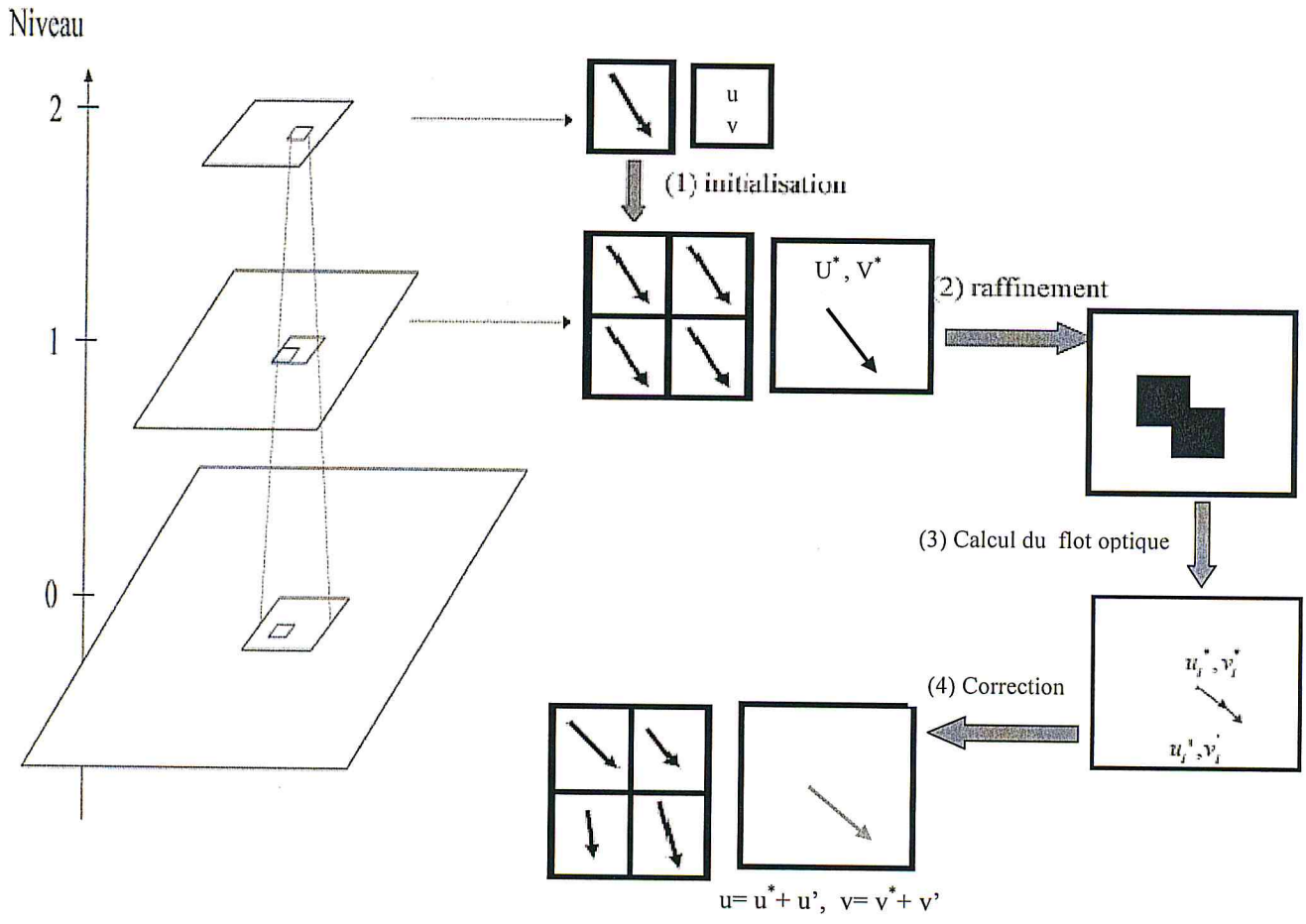


Fig.3.8 : Schéma illustrant l'algorithme de Lucas et Kanade pyramidal

D'après les constations faites concernant la méthode itérative de Lucas et Kanade, et pour éviter de calculer des vecteurs de flot optique erronés (ne représentent pas le mouvement d'une manière fidèle), il est intéressant de se limiter dans le calcul du flot optique aux points pertinents de l'image. Ces points doivent avoir un gradient fort dans des deux direction (x,y), ceci est vérifié pour les points appartenant aux régions de forte texture ou encore pour les points d'intérêts de type coins, cette technique est très utilisé dans les application de suivi [Tomasi 91, Shiand 94].

Nous montrons dans la figure 3.9 le résultat de calcul de flot optique pour les points d'intérêts de fort gradient.

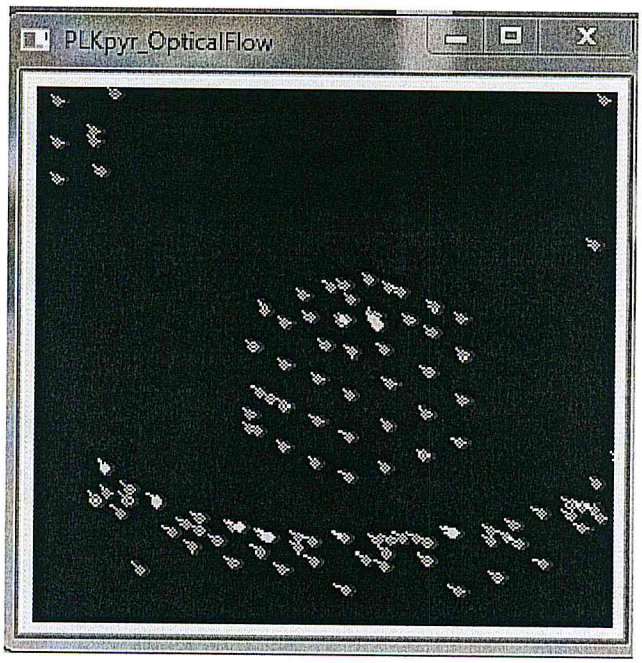


Fig3.9 : Résultat du calcul du flot optique avec l'algorithme de Lucas et Kanade pyramidal pour les points de fort gradient

5. Implémentation de la méthode de corrélation de bloc (Bloc matching)

C'est une méthode corrélative, dans laquelle l'image est devisée en petites ragions appelées bloc, qui sont généralement de forme carrées. Le calcul du vecteur mouvement revient à trouver la meilleure correspondance entre les blocs, soit en maximisant une mesure de similarité, telle que la corrélation normalisée ou en minimisant une métrique comme la Somme des Différences au Carré (*Sum of Squared Differences : SSD*).

L'algorithme de corrélation de blocs (Bloc Matching) utilisant le critère SSD est représenté ci-après :

N : taille du bloc ;

B : fenêtre de taille NxN

Pour chaque pixel de l'image I_t faire

Pour chaque pixel P=(x,y,t) du bloc faire

- Chercher le pixel P'=(x',y',t+1) qui lui ressemble le plus
- Calculer la corrélation : $SSD(P,P') = SSD + (B(x,y,t) - B(x,y,t+1))^2$;
- Score = SSD (P,P') ;

Fin pour (boucle sur le bloc)

- Régularisation du score = Moyenne des score dans le voisinage de P

Fin pour (boucle sur l'image)

La figure3.10 montre le résultat de calcul du flot optique par la méthode du bloc matching,

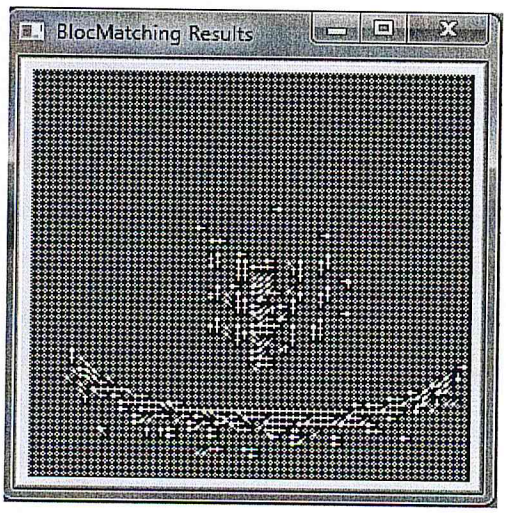


Fig.3.10 : Résultat du flot optique par la méthode de bloc matching

Contrairement aux méthodes différentielles qui fournissent une estimation subpixelique du mouvement, la méthode du Bloc Matching renvoie pour chaque bloc de l'image un vecteur flot optique associé, ce qui produit un résultat épars du mouvement calculé, de plus, elle suppose que le mouvement est constant dans un bloc, ce qui est vrai uniquement dans le cas d'un mouvement rigide

6. Comparaison et discussions

Dans la section précédente nous avons étudié les performances de chacun des algorithmes d'estimation de mouvement par rapport à la qualité des résultats obtenus. Dans ce qui suit nous allons voir leurs performances par rapport au temps d'exécution.

La comparaison s'effectuera sur trois séquences d'images différentes, dans les deux premières séquences (séquence1 et séquence2), les objets de la scène sont fixes, et la camera est en mouvement de translation vers l'avant, tandis que la séquence3, contient un objet en mouvement composé d'une rotation de droite à gauche et d'une translation de gauche à droite, alors que la camera est fixe (fig.3.11), les images de ces séquences sont de taille 160x120 pixels. Le tableau 3.1 montre les temps d'exécution pour les trois séquences avec les quatre algorithmes implémentés sur un processeur Intel Core 2 Duo (2.1 GHz, 800 HGz FSB), et 2 GB de RAM.

(a) Séquence 1 : camera mobile (b) Séquence 2 : camera mobile (c) : séquence 3 : camera fixe

Fig. 3.11 : Séquences de tests

Séquences / Algorithme	Séquence 1	Séquence 2	Séquence 3	Moyenne
Horn et Schunck	47	48	43	46
Lucas et Kanade	37	35	34	35,3
Lucas et Kanade Pyramidale	38	40	41	39,6
Bloc Matching	37	36	34	35,6

Tableau 3.1 : Temps de calcul pour les différents algorithmes en ms.

Les figures3.12, 3.13 et 3.14 montrent les résultats de calcul du flot optique pour les séquences 1,2 et 3 en appliquant les quatre algorithmes développés.

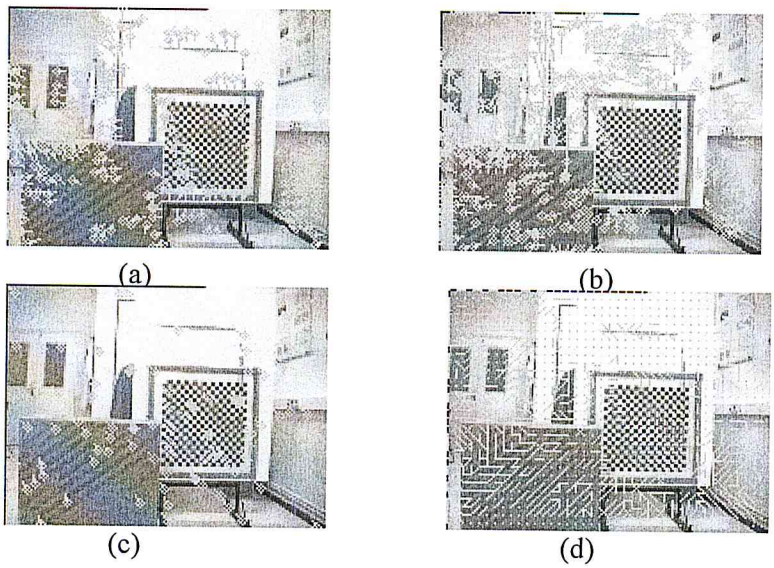


Fig. 3.12 : Résultat du calcul du flot optique pour séquence 1

(a) Horn et Schunck, (b) Lucas et Kanade, (c) Lucas et Kanade pyramidale, (d) Bloc Matching

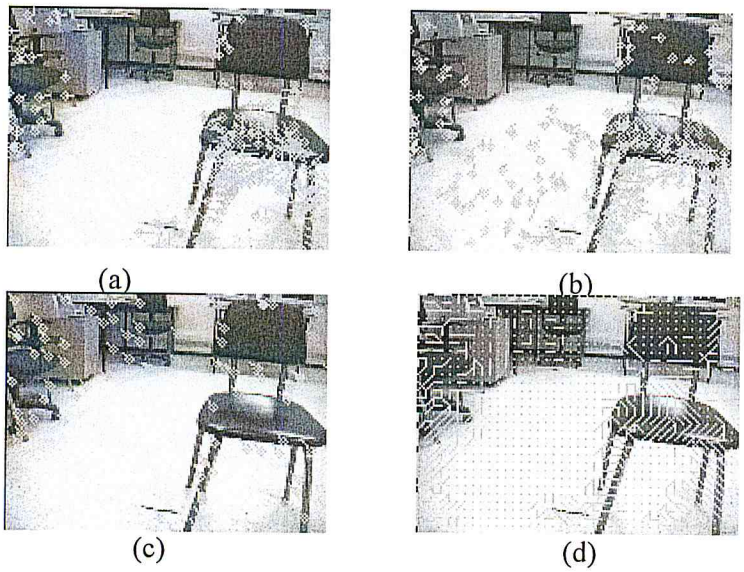
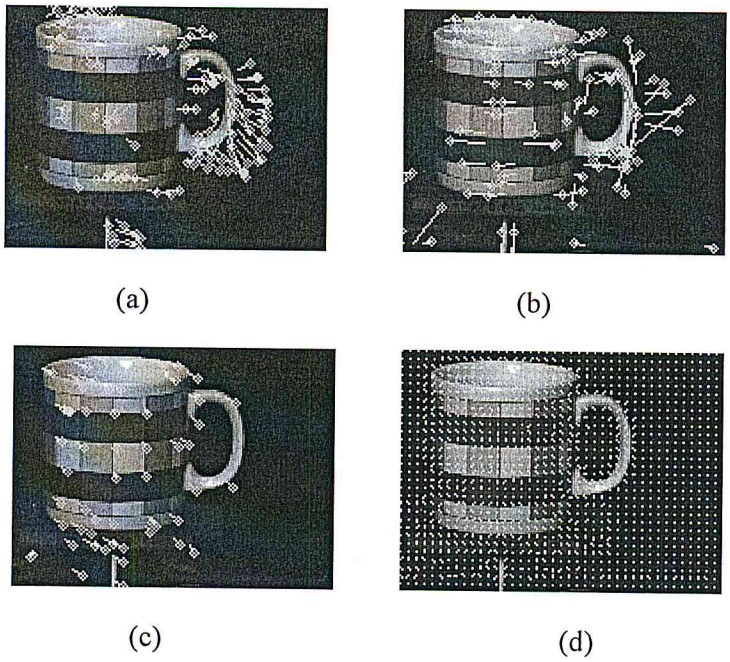


Fig. 3.13 : Résultat du calcul du flot optique pour séquence 2

(a) Horn et Schunck, (b) Lucas et Kanade, (c) Lucas et Kanade pyramidale, (d) Bloc Matching



(a) (b)

(c) (d)

Fig. 3.14 : Résultat du calcul du flot optique pour séquence 3

(a) Horn et Schunck, (b) Lucas et Kanade, (c) Lucas et Kanade pyramidale, (d) Bloc Matching

Nous constatons d'après les résultats expérimentaux que pour les trois séquences étudiées, le temps de calcul est très proche, sauf pour l'algorithme Horn et Schunck qui a un temps de calcul un peu plus élevé par rapport aux autres algorithmes, néanmoins le résultat obtenu avec cet algorithme est le plus précis.

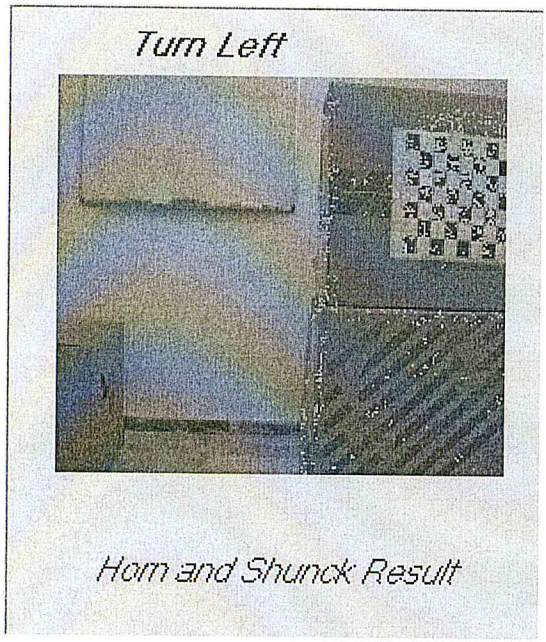
Dans ce qui suit, nous allons implémenter la stratégie de la balance décrite dans le chapitre précédent (§ 3.5) dans différentes situations de disposition d'obstacles dans l'environnement, et ceci en utilisant les deux algorithmes choisis pour leur densité et leur précision de calcul du flot optique : l'algorithme Horn et Schunck et l'algorithme Lucas et Kanade.

7. Implémentation de la stratégie de la balance

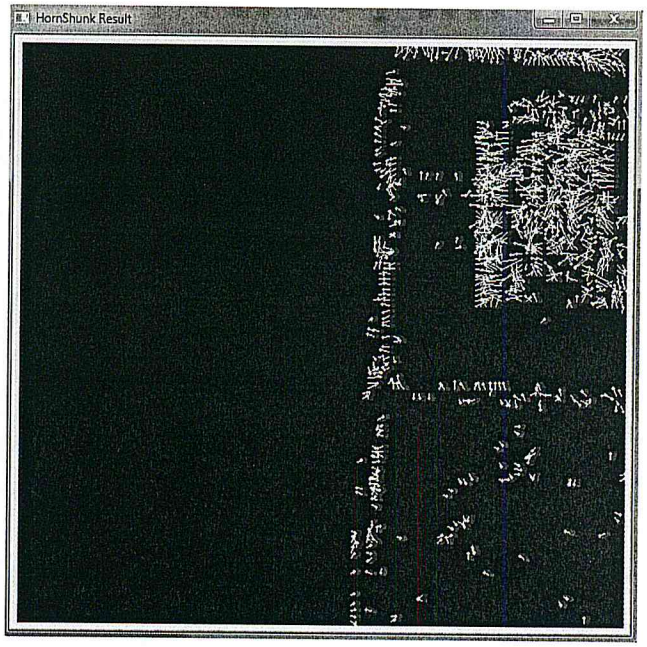
Rappelons que cette stratégie consiste à équilibrer entre la moyenne des modules des vecteurs du flot optique perçu à gauche et celle du flot optique perçu à droite de l'image. Le coté de l'image qui contient la moyenne des modules des vecteurs due flot optique la plus élevée, contient alors les obstacles les plus proches. Nous avons appliqué ce principe pour des séquences de scènes simples ensuite pour des séquences de scènes plus complexes.

7.1.Cas de scènes simples

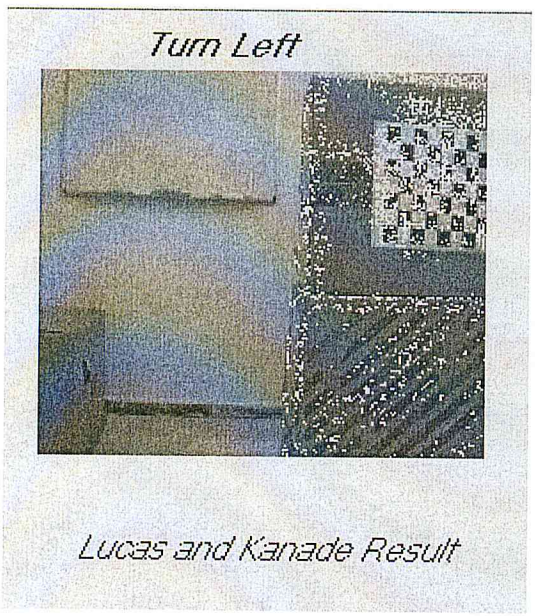
La séquence de la figure 3.15 représente une situation où l'obstacle se trouve à droite de l'image, nous avons déplacé l'obstacle dans la direction de la caméra, ensuite nous avons calculé le flot optique par les deux algorithmes choisis (Horn et Schunck, Lucas et Kanade). La décision prise par la stratégie de la balance est correcte avec les deux algorithmes car elle permet d'éviter l'obstacle.



(a) Décision par Horn et Schunck



(b) Résultat de Horn et Schunck

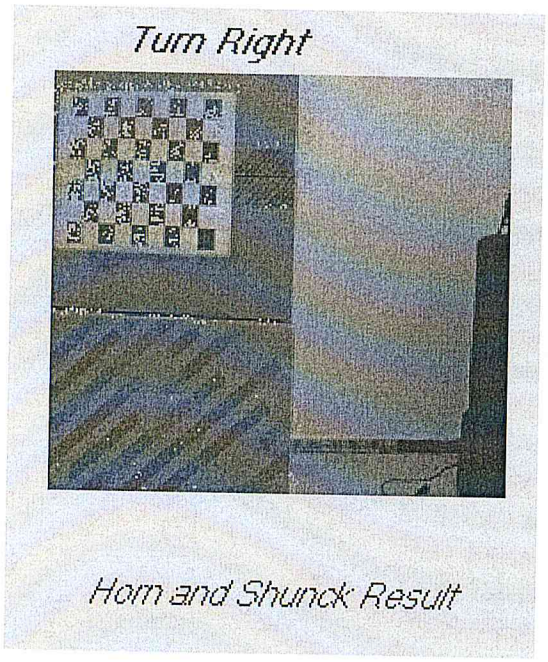


(d) Décision par Lucas et Kande

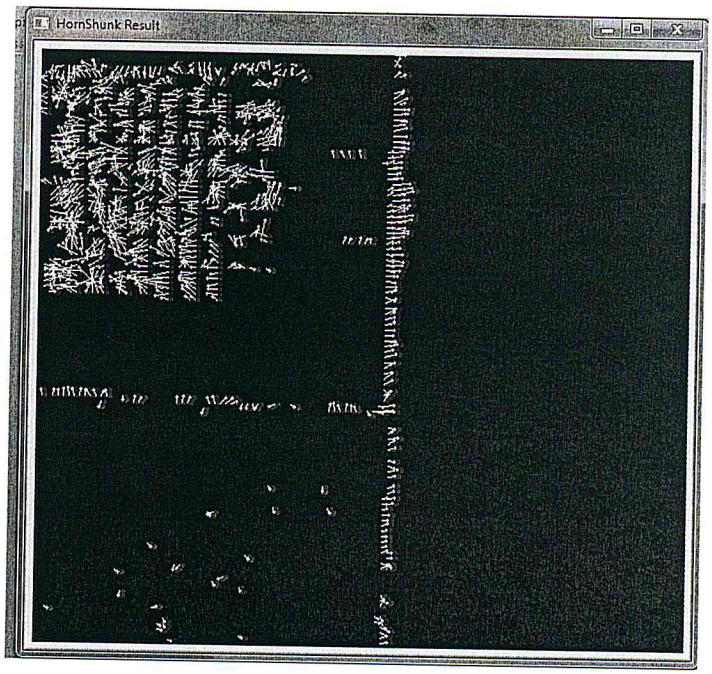
(b) Résultat de Lucas et Kanade

Fig 3.15 : Résultats de la stratégie de la balance pour un obstacle à droite du robot

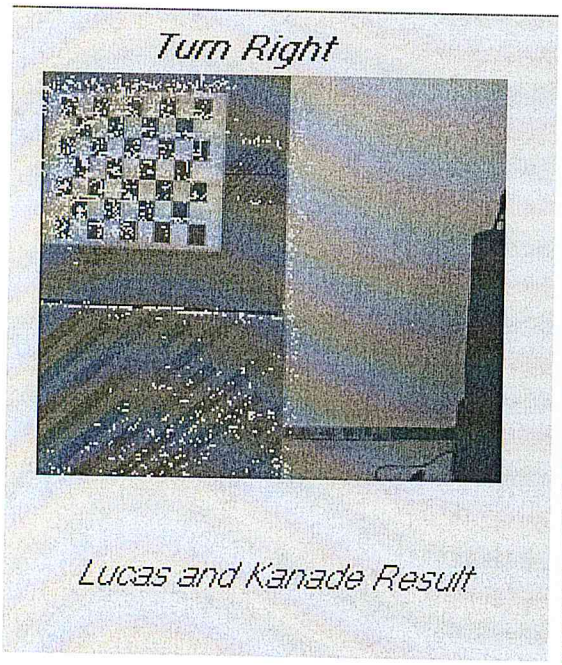
La figure 3.16 montre le résultat de la stratégie de la balance avec les mêmes conditions de l'expérience précédente, sauf que pour cette expérience l'obstacle se trouve à gauche de l'image.



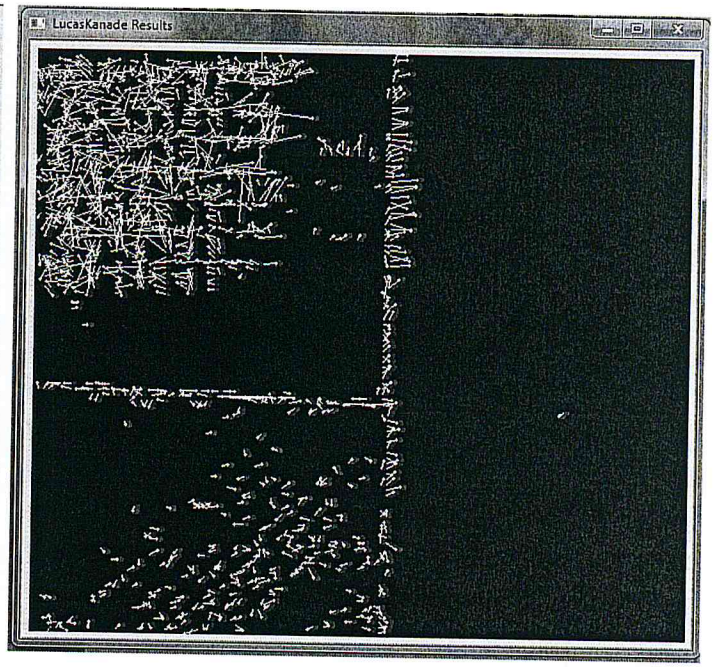
(a) Décision par Horn et Schunck



(b) Résultat de Horn et Schunck



(d) Décision par Lucas et Kande



(b) Résultat de Lucas et Kanade

Fig 3.16 : Résultats de la stratégie de la balance pour un obstacle à droite du robot

7.2.Cas de scènes complexes

Le tableau 3.2 montre les décisions prises à partir de la stratégie de la balance en appliquant les deux algorithmes de calcul du flot optique : l'algorithme Horn et Schunck et l'algorithme Lucas et Kanade.

Les expériences illustrées dans le tableau 3.2 montrent plusieurs configurations de disposition des obstacles, dans lesquelles, la caméra est en déplacement et les objets sont fixes.

- Séquence 1 : Obstacle à droite.
- Séquence 2 : Obstacle proche à gauche et obstacle loin à droite.
- Séquence 3 : Obstacle proche à droite et obstacle loin à gauche.
- Séquence 4 : Obstacle à gauche.
- Séquence 5 : Obstacle très proche à droite et obstacle loin à gauche.

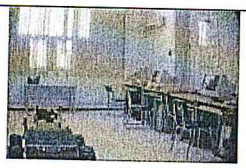
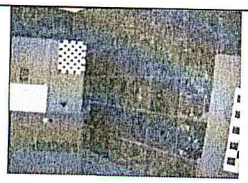
Séquences	***Méthodes***	***Décisions***
1 	Horn et Schunck	Tourner à gauche √
	Lucas et Kanade	Tourner à gauche √
2 	Horn et Schunck	Tourner à droite √
	Lucas et Kanade	Tourner à gauche X
3 	Horn et Schunck	Tourner à gauche √
	Lucas et Kanade	Tourner à droite X
4 	Horn et Schunck	Tourner à droite √
	Lucas et Kanade	Tourner à droite √
5 	Horn et Schunck	Tourner à gauche √
	Lucas et Kanade	Tourner à gauche √

Tableau 3.2 : Décision de la stratégie de la balance dans différentes situations

L'application de la stratégie de la balance avec l'algorithme Horn et Schunck a donné la bonne décision dans toutes les séquences testées, tandis que l'algorithme de Lucas et Kanade, sur cinq expériences, a donné deux décisions incorrectes, ce qui été prévisible d'après l'étude faite au paragraphe 3 sur les inconvénients de cet algorithme. De ce fait, nous avons choisi d'appliquer l'algorithme de Horn et Schunck dans l'application d'évitement d'obstacle pour le robot mobile B21r.

8. Evitement d'obstacles appliqué sur le robot B21r

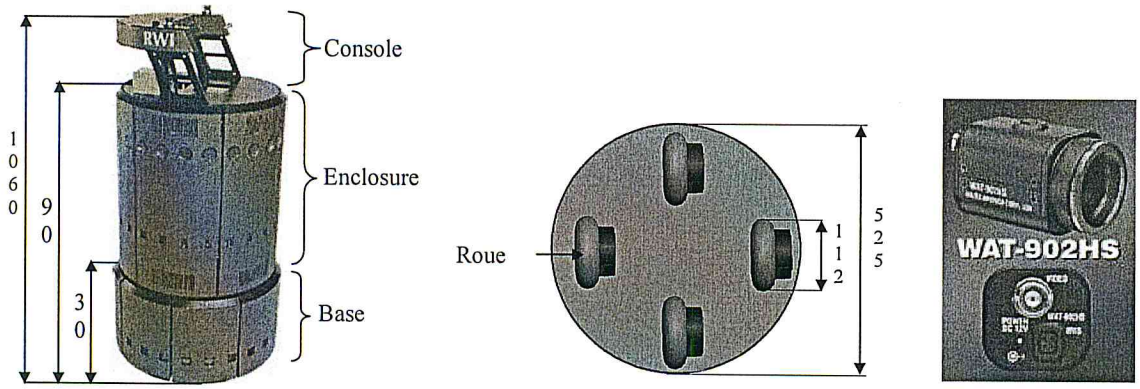
Pour valider les algorithmes implémentés, nous avons utilisé le robot mobile B21r, disponible au niveau de l'équipe Vision Artificielle et Analyse d'Images du CDTA.

8.1. Descriptif du robot mobile B21r

Le robot synchro drive B21r se déplace grâce à ses quatre roues disposées en forme carré. Il a une forme cylindrique de dimensions de 52.5x106 cm, pèse environ 122.5 kg, peut supporter une charge supplémentaire de 90 kg, et se déplace à une vitesse max de 90cm/s. Il est constitué de trois parties : la base, l'enclosure et la console (Figure 3.17 a et b).

L'enclosure et la console sont physiquement attachées, alors que l'enclosure et la base sont reliées par une articulation, ce qui permet une rotation libre de l'enclosure indépendamment de la base. L'enclosure comporte deux ceintures de capteurs, une ceinture de capteurs à ultrasons de type polaroid et une ceinture de capteurs infrarouges de type GP2D02, la base quant à elle comporte la deuxième ceinture de capteurs à ultrasons, et la console c'est la partie du robot qui supporte la caméra CCD (N/B) et l'interface rFLEX. Un télémètre laser est aussi embarqué sur le robot, celui-ci est placé dans la partie inférieure de l'enclosure, donc l'orientation du laser suivra celle du robot.

La caméra embarquée du robot mobile B21r est une '*WATEC LCL-902 HS*' (Figure 3.17 c).. les séquences vidéo sont acquises à travers une carte '*MATROX Imaging*' à un taux de 30 images par seconde installée sur un PC hôte et communique avec le robot B21r via une connexion sans fil. Les algorithmes ont été développés sous le système Windows en C++ Builder V.6.

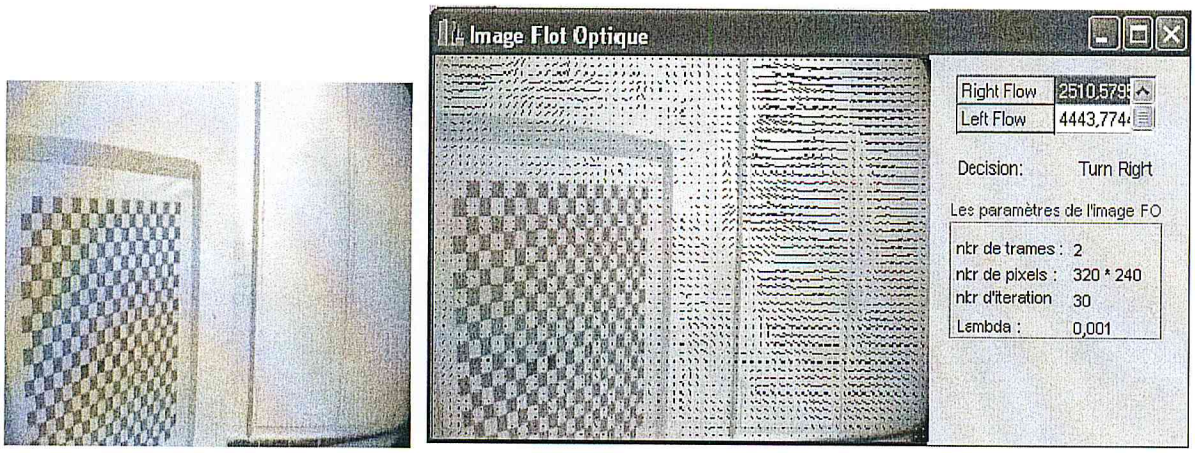


(a) vue de profil (b) vue de dessus (c) Caméra embarquée

Figure 3.17 : La géométrie du robot mobile B21r.

8.2. Scenario d'évitement d'obstacles

Dans le scénario suivant le robot doit se déplacer dans la salle en évitant les obstacles et ceci en se basant sur l'information du flot optique. La figure 3.18.a, montre la vue de la caméra au début du scénario ainsi que la moyenne du flot optique de gauche et celle de droite calculés sur l'image (figure 3.18. b). Le flot optique calculé à gauche (4443,77) du robot est plus grand que celui de droite (2510,57), et donc le robot doit tourner vers la droite en évitant le tableau du damier ce qui correspond à la position 1 dans la figure3.20.

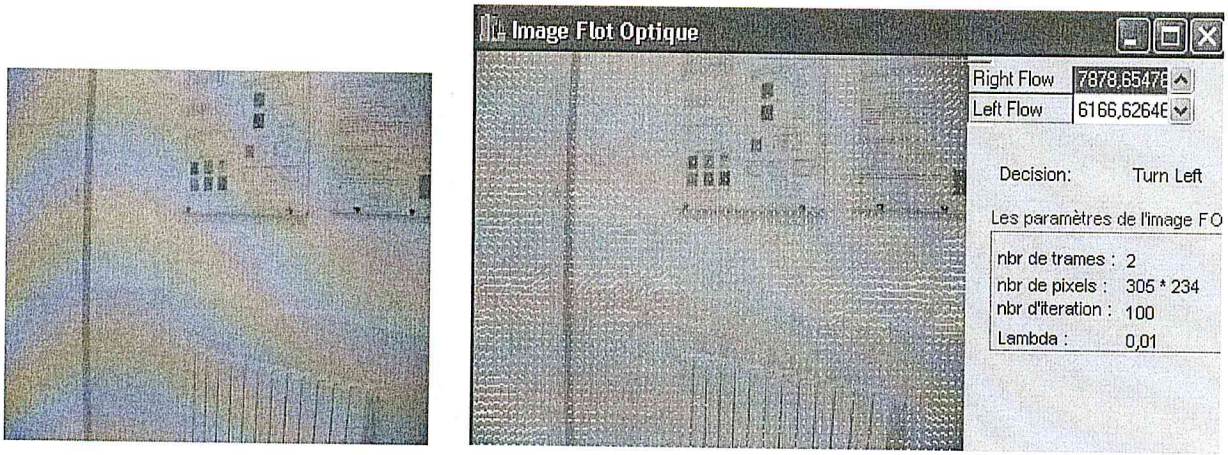


(a) Vue de la caméra (b) flot optique calculé

Fig. 3.18 : Première décision

Le robot ensuite continu dans le même sens jusqu'a la rencontre d'un autre obstacle qui va rendre le flot de droite (7878,65) plus élevé que celui de gauche (6156,62) et donc la décision

est de tourner à gauche pour éviter la collision avec le mur (figure 3.19), ce qui correspond à la position 2 dans la figure 3.20.



(a) Vue de la caméra (b) flot optique calculé

Fig. 3.19 : Deuxième décision

La figure3.20 montre le chemin entrepris par le robot durant son déplacement, dans lequel le robot rencontre deux points principaux où il doit changer son orientation, la position (1) correspond au tableau et la position (2) correspond au mur.

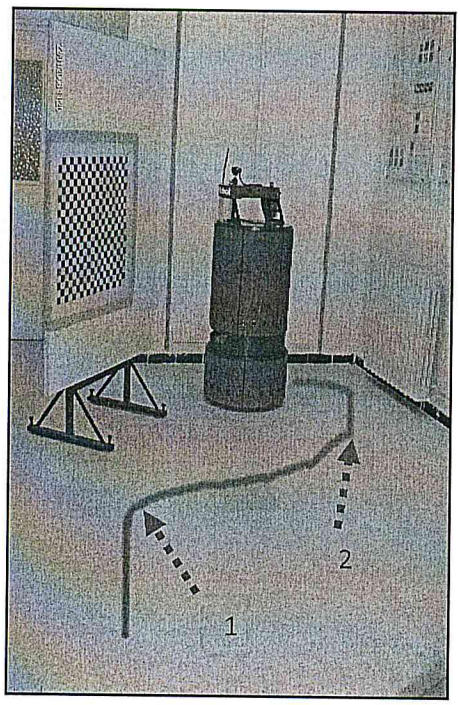


Fig.3.20 : Scenario de l'action d'évitement d'obstacle pour le robot B21r

9. Conclusion

Dans ce chapitre nous avons implémenté quatre algorithmes d'estimation de mouvement dont, trois algorithmes différentiels ; Horn et Schunck, Lucas et Kanade, et Lucas et Kanade Pyramidal et un algorithme de corrélation de blocs ou Bloc Matching. Ensuite, nous avons comparé les résultats par rapport à la qualité des flots optiques obtenus et par rapport au temps de calcul. Parmi toutes ces méthodes développées et testées, nous étions amenés à faire le choix de la méthode qui réalise le compromis entre la précision et la densité d'une part et la rapidité d'une autre part. Vu la nature de l'application qui est l'évitement d'obstacles pour un robot mobile, dans laquelle ce dernier doit prendre une décision en temps réel, notre choix s'est porté sur la méthode Horn et Schunck, car elle reste la plus robuste malgré sa sensibilité aux conditions d'éclairage.

Finalement nous avons validé les algorithmes développés par l'application de la stratégie de la balance pour la tâche d'évitement d'obstacles, son principe de base consiste à garder un équilibre entre le flot optique à gauche et à droite du robot, cette technique permet une navigation du robot sans aucune collision avec les obstacles, elle a été testé sur le robot B21r, durant laquelle le robot a pu éviter les obstacles présents dans la scène avec succès.

CONCLUSION GENERALE

Conclusion générale

Le travail présenté dans ce mémoire a pour objectif d'exploiter le flux des images acquises par une caméra embarquée sur un robot mobile, pour l'estimation du déplacement relatif 2D des objets ou de la caméra. Cet objectif requiert l'étude des différentes méthodes d'estimation de mouvement existantes dans la littérature, notamment les méthodes à modèles paramétriques, les méthodes de corrélation et les méthodes différentielles. Notons que les méthodes paramétriques sont généralement employées pour des classes de mouvement spécifiques, tandis que les méthodes de corrélations sont très utilisées en stéréovision, les méthodes différentielles quand à elles, permettent une estimation dense et sub-pixelique du mouvement, elles sont les plus prometteuses au niveau des résultats et les plus intéressantes à développer.

A cet effet, une synthèse de plusieurs méthodes d'estimation de mouvement dans une séquence d'images, a été faite, dans laquelle nous avons implémenté quatre algorithmes d'estimation de mouvement dont, trois algorithmes différentielles ; Horn et Schunck, Lucas et Kanade, et Lucas et Kanade Pyramidale, et un algorithme de corrélation de blocs ou Bloc Matching. Par la suite nous avons comparé les résultats par rapport à la qualité du flot optique d'une part et par rapport au temps de calcul d'une autre part. A partir de cette comparaison nous avons constaté que malgré la rapidité de la méthode Lucas et Kanade, le résultat du champ de mouvement qu'elle fournit manque de précision en particulier dans les régions homogènes, en ce qui concerne la méthode Lucas et Kanade Pyramidale, elle est plus lente par rapport à la précédente, mais prend en compte les grands mouvements, qui peuvent déplacer les points à l'extérieur de la fenêtre locale. La méthode de Bloc Matching quant à elle, suppose que le mouvement est constant dans un bloc, et fournit un résultat épars du mouvement. Nous avons remarqué que parmi les méthodes implémentées ; la méthode Horn et Schunck été la plus adaptée à notre application pour son compromis entre la qualité du résultat du champ de mouvement et le temps de calcul.

Nous avons aussi utilisé l'information du flot optique pour extraire des informations utiles pour une navigation basée vision. Ces informations fournissent une idée globale sur : la disposition des obstacles, la direction du robot, la profondeur, et les coordonnées 3D des points de la scène.

Les algorithmes d'estimation de mouvement développés, ont été implémentés et validés expérimentalement dans une application d'évitement d'obstacles sur le robot B21r. Cette application est une contribution au développement d'un algorithme pour la navigation visuelle d'un robot mobile, où l'information du flot optique est extraite à partir de la séquence d'images pour aider le robot à comprendre et à interagir avec son environnement.

La stratégie d'évitement d'obstacle utilisée calcule la moyenne du flot optique des côtés gauche et droite des images acquises de la scène par le robot mobile. Ce dernier prendra la direction du côté ayant la moyenne la plus faible. Cependant, cette stratégie soufre de sa sensibilité aux conditions d'éclairage, car le changement de l'intensité dans l'image n'est pas toujours dû à un mouvement.

Comme perspectives de ce travail, nous proposons:

- Réduire l'influence du changement d'éclairage par l'utilisation d'algorithmes de prétraitement (filtrage, filtrage adaptatif, etc.).

- Fusionner l'information du flot optique avec d'autres informations issues des capteurs (Ultrason, Infrarouge, Laser, GPS, etc.), dans le but de confirmer ou d'infirmer la décision prise lors du traitement de l'information issu de la caméra.

BIBLIOGRAPHIE

Bibliographie

[Argyros 04]: A.A. Argyros, D.P. Tsakiris, C. Groyer. – Biomimetic centering behavior for mobile robots with panoramic sensors. *IEEE roboticsand automation magazine: Vol. 11. Special issue on panoramic robotics*, pp. 21–30, 2004.

[Arkin 98]: Arkin. R - Behaviour-Based Robotics. MIT Press, Cambridge MA, USA, 1998.

[Aschwander 92]: P. Aschwander, W.Guggenbul, - Experimental results from a comparative study on correlation-type registration algorithms, In FORSTNER et RUWIEDEL, Robust computer vision, Wichmann, 1992, pages 268-282.

[Barber 05]: D. Barber, S. Griffiths, T. McLain, R. Beard. – Autonomous landing of miniature aerial vehicles. – *AIAA Infotech@Aerospace*, 2005.

[Barron 94]: J. L. Barron, D. J. Fleet, S. S. Beauchemin. – Performance of optical flow techniques. *International Journal of Computer Vision*, 12(1) :43–77, 1994.

[Beghdadi 03]: A. Beghdadi, M. Mesbah, J. Monteil. – A fast incremental approach for accurate measurement of the displacement field. *Image and vision computing*, 21:383–399, 2003.

[Bensalah 95]: F. Bensalah, F. Chaumette. – Compensation of abrupt motion changes in target tracking by visual servoing. – *IEEE/RSJ Int.Conf. on Intelligent Robots and Systems, IROS'95*, vol. 1, pp.181–187, Pittsburgh, Pennsylvania, August 1995.

[Beyeler 09a]: Antoine Beyeler, Jean-Christophe Zufferey, Dario Floreano. – OptiPilot: control of take-off and landing using optic flow. *Proceedings of the 2009 European Micro Air Vehicle conference and competition (EMAV '09)*, 2009.

[Beyeler 09b]: Antoine Beyeler, Jean-Christophe Zufferey, Dario Floreano. – Vision-based control of near-obstacle flight. *Autonomous Robots*, 27(3) :201–219, 2009.

[Bouguet 99]: J.Y. Bouguet. – Pyramidal Implementation of the Lucas Kanade Feature Tracker Description of the algorithm. – *Rapport de recherche, Intel Corporation, Microprocessor Research Labs, 1999.*

[Carelli 02]: R. Carelli, C. Soria, O. Nasisi, E. Freire. – Stable agv corridor navigation with fused vision-based control signals. – *Proceedings of the 28th conference of industrial electronics society*, 2002.

[Chaumette 93]: F. Chaumette, A. Santos. – Tracking a moving object by visual servoing. – *12th IFAC World Congress*, vol. 3, pp. 643–648, Sidney, Australia, July 1993.

[Chavant 98]: Chavant Florant, Colle Etienne – Perception de l'environnement en robotique, Rdition Hermes , Paris, 1998, ISBN: 2-866001-684-X, ISSN: 0981-7824.

[Cipolla 97a]: R. Cipolla, A. Blake. – Image divergence and deformation from closed curves. *Int. J. Rob. Res.*, 16(1):77–96, 1997.

[Cipolla 97b]: R. Cipolla, N. Hollinghurst. – Visually guided grasping in unstructured environments. *Robotics and Autonomous Systems*, 19(3- 4):337–346, 1997.

[Collett 75]: T.S. Collett, M.F. Land. – Visual control of flight behavior in the hoverfly, syritta pipiens. *Journal of Comparative Physiology*, 99:1–66, 1975.

[Collett 80]: T.S. Collett. – Some operating rules for the optomotor system of a hoverfly during voluntary flight. *Journal of Comparative Physiology A*, 138:271–282, 1980.

[Colombo 95]: C. Colombo, B. Allotta, P. Dario. – Affine visual servoing: A framework for relative positioning with a robot. – *IEEE Int.Conf. Robot. Automat.* Nagoya, Japan, 1995.

[Coombs 92]: D. Coombs, K. Roberts. – Bee-bot : using peripheral optical flow to avoid obstacles. – *Intelligent robots and computer vision XI*, vol. 1825, pp. 714–721. SPIE, 1992

[Crétual 01]: A. Crétual, F. Chaumette. – Visual servoing based on image motion. *Int. Journal of Robotics Research*, 20(11):857–877, November 2001.

[Crétual 01a]: A. Crétual, F. Chaumette. – Application of motion-based visual servoing to target tracking. *Int. Journal of Robotics Research*, 20(11):878–890, November 2001.

[Crétual 01b]: A. Crétual, F. Chaumette. – Visual servoing based on image motion. *Int. Journal of Robotics Research*, 20(11):857–877, November 2001.

[Dahmen 09]: H. Dahmen, A. Millers, H. Mallot. – Insect inspired odometry by optic flow recorded with optical mouse chips. *Flying insectsand robots*, éd. par D. Floreano, J.-C. Zufferey, M. Srinivasan, C. Ellington. – Berlin, Springer, 2009.

[Desachy 01]: Desachy Jacky, - Analyse d'images, *Edition Université de Antilles de la Guyane Pointe à pitre. France 2001.*

[Duchon 98]: A. Duchon, W. H. Warren, L. Kaelbling. – Ecological robotics. *Adaptive Behavior*, 6:473–507, 1998.

[Feddema 93]: J. T. Feddema, C. S. G. Lee, O. R. Mitchell. – Feature-based visual servoing of robotic systems. *Visual Servoing*, éd. par K. Hashimoto, pp. 105–138. – World Scientific, 1993.

[Garratt 08]: Matthew A. Garratt, Javaan S. Chahl. – Vision-based terrain following for an unmanned rotorcraft. *Journal of Field Robotics*, 25:284–301, 2008.

[Gary 07]: Gary Bradsky , Adrian Kachler – Learning computer vision with OpenCv Library, 2007, ISBN: 978-0-596-51613-0

[Green 08]: William E. Green, Paul Y. Oh. – Optic flow based collision avoidance. *IEEE Robotics & Automation Magazine*, 15(1):96–103, 2008.

[Grosso 96]: E. Grosso, G. Metta, A. Oddera, G. Sandini. – Robust visual servoing in third reaching tasks. *IEEE Transactions on Robotics and Automation*, 12(8):732–742, 1996.

[Horn 81]: B.K.P. Horn, B.G. Schunck. – Determining optical flow. *Artificial intelligence*, pp. 185–204, 1981.

[Hrabar 05]: S.E. Hrabar, P.I. Corke, G.S. Sukhatme, K. Usher, J.M. Roberts. – Combined optic-flow and stereo-based navigation of urban canyons for a uav. – *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems*, Edmonton, Canada, 2005.

[Humbert 05b]: J. Sean Humbert, R.M.Murray,M. H. Dickinson. – Pitch-altitude control and terrain following based on bio-inspired visuomotor convergence. – *AIAA Conference on Guidance, Navigation and Control*, San Francisco, CA, 2005.

[Humbert 06]: J. Sean Humbert. – *Bio-Inspired Visuomotor Convergence in Navigation and Flight Control Systems*. – Phd. thesis, California Institute of Technology, 2006.

[Humbert 10]: J. S. Humbert, A. M. Hyslop. – Bioinspired visuomotor convergence. *IEEE Transactions on Robotics*, 26(1) :121 – 130, 2010.

[Kahlouche 07a]: S.Kahlouche, O.Djekoune, D.Djebrouni, D.Meriche, -Segmentation by motion based on optical flow histogram, *JIG'2007, 3èmes journées internationales de l'information graphique, 29-30 Octobre 2007. Constantine- Algerie, pp. 52-55.*

[Kahlouche 07b] : S.Kahlouche, K.Achour, - Optical flow based robot obstacle avoidance, *International Journal of Advanced Robotics Systems, ISSN 1729-8806, Volume 4, Number 1, 2007, pp.13-16*

[Kirchner 89]: W. H. Kirchner, M. V. Srinivasan. – Freely flying honeybees use image motion to estimate object distance. *Naturwissenschaften*, 76:281–282, 1989.

[Lee 76]: D. N. Lee. – A theory of visual control of braking based on information about time to collision. *Perception*, 5(4):437–459, 1976.

[Lucas 81]:B. Lucas, T. Kanade. – An iterative image registration technique with an application to stereo vision. – *Proceedings of the Seventh International Joint Conference on Artificial Intelligence*, pp. 674–679, Vancouver, 1981.

[McCarthy 08]: C. McCarthy, N. Barnes, R. Mahony. – A robust docking strategy for a mobile robot using flow field divergence. *IEEE Transactions on Robotics*, 24(4):832–842, 2008.

[Muratet 05]: Laurent Muratet, Stéphane Doncieux, Yves Briere, Jean-Arcady Meyer. – A contribution to vision-based autonomous helicopter flight in urban environments. *Robotics and Autonomous Systems*, 50(4):195–209, 2005.

[Negahdaripour 89]: S. Negahdaripour and K.P. Horn -Direct method for locating the focus of expansion, *Comp. Vis. Graph. Imp. Process. 46(3), 303-326, 1989.*

[Nelson 89]: R. C. Nelson, J. Y. Alloimonos. – Obstacle avoidance using flow field divergence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(10):1102–1106, 1989.

[Netter 99]: T. Netter, N. Franceschini. – Neuromorphic optical flow sensing for nap-of-the-earth flight. – D.W. Gage, H.M. Choset (édité par), *Proceedings of the SPIE Conf. on Mobile Robots XIV*, vol. 3838, pp. 208–216, 1999.

[Odobez 95]: Odobez J., Bouthemy P. - MRF-based Motion Segmentation Exploiting a 2D Motion Model Robust Estimation, *Proceeding of IEEE International Conference on Image Processing ICIP'95, Washington DC, 628-631.*

[Questa 95]: P. Questa, E. Grossmann, G. Sandini. – Camera self orientation and docking maneuver using normal flow – *Proceedings of Spie -the International Society for Optical Engineering*, Orlando, USA, 1995.

[Richefeu 06]: M.Julien Richefeu, - Détection et analyse de mouvement sur système de vision à base de rétine numérique, Thèse de doctorat 2006, Université Paris 6.

[Ruffier 04]: F. Ruffier, N. Franceschini. – Visually guided micro-aerial vehicle: automatic take off, terrain following, landing and wind reaction. – *Proceedings of international conference on robotics and automation*, LA, New Orleans, April 2004.

[Ruffier 05]: F Ruffier, N. Franceschini. – Optic flow regulation: the key to aircraft automatic guidance. *Robotics and Autonomous Systems*, 50:177–194, 2005.

[Ruffier 07]: F. Ruffier, J. Serres, G.P. Masson, N. Franceschini. – A bee in the corridor: regulating the optic flow on one side. – *Proceedings of the 7th meeting of the German neuroscience society-31st Göttingen neurobiology conference*, Göttingen, Germany, 2007. – Abstract T14-7B.

[Ruffier 08]: F. Ruffier, N. Franceschini. – Aerial robot piloted in steep relief by optic flow sensors. – *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 1266–1273, Nice, France, September 2008.

[SantosVictor 95]: J. Santos-Victor, G. Sandini, F. Curotto, S. Garibaldi. – Divergent stereo in autonomous navigation: from bees to robots. *International Journal of Computers Vision*, 14:159–177, 1995.

[SantosVictor 97]: J. Santos-Victor, G. Sandini. – Visual behaviors for docking. *Computer Vision and Image Understanding*, 67(3):223–238, 1997.

[Serres 07]: J. Serres, F. Ruffier, G.P. Masson, N. Franceschini. – A bee in the corridor: centring or wall-following? – *Proceedings of the 7th meeting of the German neuroscience*

society-31st Göttingen neurobiology conference, Göttingen, Germany, 2007. – Abstract T14-8B.

[Serres 08]: J. Serres, D. Dray, F. Ruffier, N. Franceschini. – A vision-based autopilot for a miniature air vehicle: joint speed control and lateral obstacle avoidance. *Autonomous Robots, Springer*, 25(1-2) :103–122, August 2008.

[Shiand 94]: J.Shiand, C.Tomasi, - Good features to track, *In. Proc. Of IEEE Conference on Computer Vision and Pattern Recognition, Seattle, pages 593-600, June 1994.*

[Srinivasan 96]: M.V. Srinivasan, S.W. Zhang, M. Lehrer, T.S. Collett. – Honeybee navigation. *En route* to the goal: visual flight control and odometry. *Journal of Experimental Biology*, 199:237–244, 1996.

[Srinivasan 00]: M.V. Srinivasan, S.W. Zhang, J. S. Chahl, E. Barth, S. Venkatesh.– How honeybees make grazing landings on flat surfaces.*Biological Cybernetics*, 83 :171–183, 2000.

[Tammero 02]: L.F. Tammero, M.H. Dickinson. – The influence of visual landscape on the free flight behavior of the fruit fly drosophila melanogaster. *Journal of Experimental Biology*, 205:327–343, 2002.

[Tomasi 91]: C.Tomasi, T.Kanade, -Detection and tracking of point features, Technical report 91 132, Carnegie Mellon University – Robotics Institute, 1991.

[Verri 89]: Verri. A., Girosi, F. & V. Torre. 1989 - Mathematical Properties of the ID Motion Fields: from Singular Points to Motion Parameters. *Proceedings of the IEEE Workshop on Visual Motion, Irvine CA*

[Wagner 82]: H. Wagner. – Flow-field variables trigger landing in flies. *Nature*, 297:147–148, 1982.

[Warren Jr 88]: Warren Jr, - Action Models and laws of control for the visual guidance of action, in complex movement behaviour, *The Elsevier Science pubs. , B.V. (North-Holland), 1988.*

[Weber 97]: K. Weber, S. Venkatesh, M.V. Srinivasan. – Insect inspired behaviours for the autonomous control of mobile robots. *From living eyes to seeing machines*, éd. par M. V. Srinivasan, S. Venkatesh, pp. 226–248. – Oxford University Press, 1997.

[Wilson 94]: W. Wilson. – Visual servo control of robots using kalman filter estimates of robot pose relative to work-pieces. *Visual Servoing*, éd. Par K. Hashimoto, pp. 71–104. – World Scientific, 1994.

[Zufferey 06]: Jean-Christophe Zufferey, Dario Floreano. – Fly-inspired Visual Steering of an Ultralight Indoor Aircraft. *IEEE Transactions on Robotics*, 22(1):137–146, 2006.

Annexes

Annexe A

La méthode de Horn et Schunck

La méthode de Horn et Schunck s'appuie sur l'hypothèse de conservation de l'intensité, plus précisément, il est supposé que l'intensité ne varie pas au cours du temps sur les trajectoires des points dans l'image. Ceci s'exprime formellement de la manière suivante :

$$E(x+dx, y+dy, t+dt) = E(x, y, t) \qquad (1)$$

Où : $E(x+dx, y+dy, t+dt)$ et $E(x, y, t)$ sont respectivement les intensités lumineuses du projeté d'un point de la surface d'un objet à deux instant successifs, t et t+dt.

La contrainte de base : Le développement au premier ordre en série de Taylor de $E(x+dx, y+dy, t+dt)$ donne :

$$E(x+dx, y+dy, t+dt) = E(x, y, t) + dx \cdot \frac{\partial E}{\partial x} + dy \cdot \frac{\partial E}{\partial y} + dt \cdot \frac{\partial E}{\partial t} + 0(dx^2 + dy^2 + dt^2) \qquad (2)$$

Notons E_x, E_y et E_t les dérivées partielles de E respectivement par rapport à x, y et à t. En négligeant les termes du second ordre, en tenant compte de la relation (1) et en divisant le tout par dt, on obtient :

$$E_x \cdot \frac{dx}{dt} + E_y \cdot \frac{dy}{dt} + E_t = 0 \qquad (3)$$

En notant les deux composantes du vecteur vitesse par :

$$u = \frac{dx}{dt} \quad , \quad v = \frac{dy}{dt}$$

La relation (3) peut s'écrire :

$$E_x \cdot u + E_y \cdot v + E_t = 0 \qquad (4)$$

Soit u et v les composantes de vecteurs vitesse $V = (u, v)^t$ et ∇E le gradient spatial de E au point (x, y) et au temps t. Donc l'équation (4) peut être réécrite de la façon suivante :

$$\nabla E \cdot V = -E_t$$

Cette équation est connue sous le nom d'équation de contrainte du flot optique que l'on note par **CFO**. Elle contient deux inconnues u et v.

Si l'on se place dans l'espace des vitesses (u, v), on peut tracer la droite d'équation $E_x \cdot \frac{dx}{dt} + E_y \cdot \frac{dy}{dt} + E_t = 0$ (Figure1), la solution du flot optique au pixel considéré est n'importe quel point de cette droite.

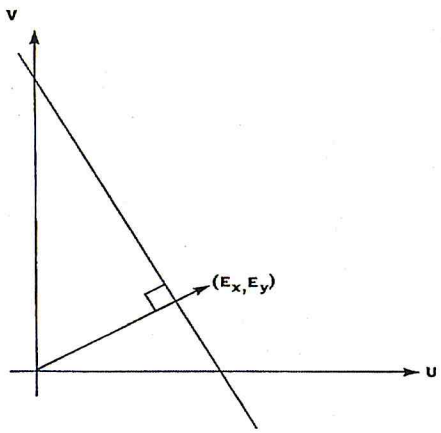


Figure1 : la droite de contrainte du mouvement

Pour résoudre une équation avec deux inconnues (les composantes u et v du vecteur vitesse), Horn et Schunck ont proposé une méthode d'estimation du flot optique, on ajoutant à l'équation de contrainte du flot optique, une contrainte supplémentaire appelée contrainte de lissage (CL).

La contrainte de lissage :

La contrainte de lissage introduite par Horn et Schunck est basée sur l'hypothèse que des points voisins ont des vecteurs vitesses similaires et que le flot optique vari progressivement dans une image, c'est à dire que le champ de vitesse recherché est régulier. Mathématiquement cette contrainte est exprimée en minimisant la différence entre le flot optique d'un pixel et la moyenne des flots optiques de ses proches voisins, ce qui revient à minimiser la somme des carrés du Laplacien des composantes du flot optique selon x et y.

$$\nabla^2 u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \quad \text{et} \quad \nabla^2 v = \frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}$$

$\nabla^2 u$, $\nabla^2 v$ les Laplaciens de u et v respectivement.

Minimisation :

Le problème se résume à minimiser les erreurs sur l'équation de contrainte du flot optique.

$$\varepsilon_b = E_x u + E_y v + E_t. \qquad (5)$$

Et la contrainte de lissage du flot optique : $\varepsilon_c^2 = \left(\nabla^2 u\right)^2 + \left(\nabla^2 v\right)^2$ (6)

Cette contrainte est approximée par :

$$\nabla^2 u \approx k\left(\overline{u_{i,j,k}} - u_{i,j,k}\right) \quad \text{et} \quad \nabla^2 v \approx k\left(\overline{v_{i,j,k}} - v_{i,j,k}\right) \qquad [7]$$

En rapprochant ∇^2 par le masque suivant :

$$\begin{bmatrix} \frac{1}{12} & \frac{1}{6} & \frac{1}{12} \\ \frac{1}{6} & -1 & \frac{1}{6} \\ \frac{1}{12} & \frac{1}{6} & \frac{1}{12} \end{bmatrix}$$

Nous avons les moyennes $\overline{u_{ij}}$ *et* $\overline{v_{ij}}$ des voisinages définis comme suit :

$$\bar{u}_{ij} = \frac{1}{6}\left(u_{i-1,j} + u_{i,j+1} + u_{i+1,j} + u_{i,j-1}\right) + \frac{1}{12}\left(u_{i-1,j-1} + u_{i-1,j+1} + u_{i+1,j+1} + u_{i+1,j-1}\right)$$

$$\bar{v}_{ij} = \frac{1}{6}\left(v_{i-1,j} + v_{i,j+1} + v_{i+1,j} + v_{i,j-1}\right) + \frac{1}{12}\left(v_{i-1,j-1} + v_{i-1,j+1} + v_{i+1,j+1} + v_{i+1,j-1}\right)$$

En remplaçant (7) dans (6) on obtient : $\varepsilon_c^2 = \left(\bar{u} - u\right)^2 + \left(\bar{v} - v\right)^2$

Dans la réalité les mesures d'intensité sont faussées par les erreurs du processus d'acquisition et par le bruit, donc l'erreur ε_b ne peut être nulle, et son amplitude est proportionnelle à la quantité de bruit. De ce fait un facteur λ appelé *paramètre de lissage* est introduit. Ce facteur joue un rôle signifiant dans les surfaces où le gradient de la luminosité est faible et son ajustement permet de réduire l'erreur en lissant l'image.

La formule globale des erreurs à minimiser est donnée par : $\varepsilon^2 = \iint_{\Omega} \lambda^2 \varepsilon_c^2 + \varepsilon_b^2$

Où Ω est le support d'estimation et correspond en général à toute l'image.

En dérivant la formule globale des erreurs, respectivement par rapport à u et v on obtient les équations suivantes: $\frac{\partial \varepsilon^2}{\partial u} = -2\lambda^2\left(\bar{u} - u\right) + 2\left(E_x u + E_Y v + E_t\right) \cdot E_x$

$$\frac{\partial \varepsilon^2}{\partial v} = -2\lambda^2\left(\bar{v} - v\right) + 2\left(E_x u + E_Y v + E_t\right) \cdot E_y$$

On posant ces deux dérivées partielles égales à zéro, on obtient un système de deux équations et deux inconnus u et v.

$$\begin{cases}\left(\lambda^2+E_x^2\right)u+E_xE_yv=\left(\lambda^2\bar{u}-E_xE_t\right)\\ E_xE_yu+\left(\lambda^2+E_y^2\right)=\left(\lambda^2\bar{v}-E_yE_t\right)\end{cases} \qquad (8)$$

Résolution du système (8)

Le déterminant de ce système est :

$$\Delta=\lambda^2\left(\lambda^2+E_x^2+E_y^2\right)$$

$$\det_{ij}=\Delta=\begin{vmatrix}\lambda^2+E_x^2 & E_xE_y\\ E_xE_y & \lambda^2+E_y^2\end{vmatrix}=\left(\lambda^2+E_x^2\right)\left(\lambda^2+E_y^2\right)-E_x^2E_y^2$$

En utilisant la méthode de Cramer, on obtient :

$$u=\frac{\Delta_1}{\Delta}$$

$$\Delta_1=\begin{vmatrix}\lambda^2\bar{u}-E_xE_t & \mathrm{E_xE_y}\\ \lambda^2\bar{v}-E_yE_t & \lambda^2+E_y^2\end{vmatrix}=\left(\lambda^2\left(\lambda^2+E_y^2\right)\bar{u}-E_xE_y\bar{v}-E_xE_t\right)$$

$$v=\frac{\Delta_2}{\Delta}$$

$$\Delta_2=\begin{vmatrix}\lambda^2+E_x^2 & \lambda^2\bar{\mathrm{u}}-\mathrm{E_xE_t}\\ E_xE_y & \lambda^2\bar{v}-E_yE_t\end{vmatrix}=\lambda^2\left(-E_xE_y\bar{u}+\left(\lambda^2+E_x^2\right)\bar{v}-E_yE_t\right)$$

Donc la solution du système d'équation est :

$$\begin{cases}u=\left(\left(\lambda^2+E_y^2\right)\bar{u}-E_xE_y\bar{v}-E_xE_t\right)/\lambda^2+E_x^2+E_y^2\\ v=\left(-E_xE_y\bar{u}+\left(\lambda^2+E_x^2\right)\bar{v}-E_yE)_t/\lambda^2+E_x^2+E_y^2\right.\end{cases}$$

Forme simplifiée :

$$\begin{cases}u=\dfrac{\lambda^2+E_y^2}{\lambda^2+E_x^2+E_y^2}\bar{u}-\dfrac{E_xE_y}{\lambda^2+E_x^2+E_y^2}\bar{v}-\dfrac{E_xE_t}{\lambda^2+E_x^2+E_y^2}\\ v=-\dfrac{E_xE_y}{\lambda^2+E_x^2+E_y^2}\bar{u}+\dfrac{\lambda^2+E_x^2}{\lambda^2+E_x^2+E_y^2}\bar{v}-\dfrac{E_yE_t}{\lambda^2+E_x^2+E_y^2}\end{cases} \qquad \ldots\ldots\ldots\ldots\ldots\ldots (*)$$

Solution itérative :

Une solution directe pour chaque cordonnée du système précédent est possible mais pas pratique. Si une matrice de coefficients est construite pour chaque composante du vecteur de vitesse, certains aspects de cette matrice se manifestent; La matrice sera très grande, deux fois le nombre de pixels de l'image (chaque vecteur comprend deux composantes) et la matrice contiendra beaucoup de valeurs nulles. A cause de ces aspects, des méthodes itératives sont utilisées. Une solution au problème d'estimation de mouvement est obtenue par plusieurs étapes de relaxation de la méthode de Gauss-Seidel, nous pouvons donc calculer une estimation du flot optique (u^{n+1},v^{n+1}) à partir des dérivées estimées et la moyenne de l'estimation précédente du flot optique (u^n,v^n).

- **Principe de la méthode itérative :**

Le principe de cette méthode est de rechercher les racines d'un système linéaire par des méthodes numériques approchées.

Soit un système de la forme suivante : A.x= b

$$A=\begin{bmatrix} a_{11}..a_{1n} \\ \cdot \quad \cdot \\ \cdot \quad \cdot \\ a_{n1}..a_{nn} \end{bmatrix} \qquad x=\begin{bmatrix} x_1 \\ \cdot \\ \cdot \\ x_n \end{bmatrix} \qquad b=\begin{bmatrix} b_1 \\ \cdot \\ \cdot \\ b_n \end{bmatrix}$$

Mettons ce système sous forme $x = \beta + \alpha \quad x$ avec

$$\beta=\frac{b_i}{a_{ii}} \qquad \text{pour i= j =1...n}$$

$$\alpha=\frac{a_{ij}}{a_{ii}} \qquad \text{pour i=1...n ; j=1...n}$$

$$\beta=\begin{bmatrix} \beta_1 \\ \cdot \\ \cdot \\ \beta_n \end{bmatrix} \qquad \alpha=\begin{bmatrix} \alpha_{11}..\alpha_{1n} \\ \cdot \quad \cdot \\ \cdot \quad \cdot \\ \alpha_{n1}..\alpha_{nn} \end{bmatrix}$$

La solution itérative de ce système s'écrira alors comme suit, avec k représentant le nombre d'itérations : $x^{k+1}=\beta+\alpha x^k$ (k=0,1,2,...)

Donc, par identification avec la méthode de Gauss-Seidel, le système (*), admet la solution itérative suivante :

$$\begin{cases} u^{n+1} = \dfrac{\lambda^2 + E_y^2}{\lambda^2 + E_x^2 + E_y^2}\bar{u}^n - \dfrac{E_x E_y}{\lambda^2 + E_x^2 + E_y^2}\bar{v}^n - \dfrac{E_x E_t}{\lambda^2 + E_x^2 + E_y^2} \\ v^{n+1} = -\dfrac{E_x E_y}{\lambda^2 + E_x^2 + E_y^2}\bar{u}^n + \dfrac{\lambda^2 + E_x^2}{\lambda^2 + E_x^2 + E_y^2}\bar{v}^n - \dfrac{E_y E_t}{\lambda^2 + E_x^2 + E_y^2} \end{cases}$$

Forme simplifiée :

$$\begin{cases} u^{n+1} = \bar{u}^n - \dfrac{E_x\left(E_x\bar{u}^n + E_y\bar{v}^n + E_t\right)}{\lambda^2 + E_x^2 + E_y^2} \\ v^{n+1} = \bar{v}^n - \dfrac{E_y\left(E_x\bar{u}^n + E_y\bar{v}^n + E_t\right)}{\lambda^2 + E_x^2 + E_y^2} \end{cases}$$

A chaque itération une nouvelle estimation des composantes du vecteur vitesse est obtenu, chaque vecteur vitesse V_{ij}^{n+1} calculé, dépend de la moyenne des vecteurs de vitesse de l'itération précédente $\bar{V}_{ij}^n$. Avec $V_{ij} = (u_{ij}, v_{ij})$, $\bar{V}_{ij} = (\bar{u}_{ij}, \bar{v}_{ij})$

Estimation des dérivées partielles :

Les deux équations du système linéaire des deux composantes u et v sont écrites pour chaque point sur le plan image. Ce système exige l'évaluation des dérivées partielles et temporelles E_x, E_y et E_t, dans leur présentation, Horn et Schunck ont proposé des approximation des dérivées basées sur des interpolateurs discrets en trois dimension (horizontal, vertical et temporel), pour un point situé au centre d'un cube de 8 mesures. La relation temps-espace entre ces mesures est montrée dans la figure 2. Chaque estimée est une moyenne des 4 premières différences prises sur les mesures adjacentes du cube.

$$E_x(i,j,t) = \frac{1}{4}\left(E(i+1,j,t) + E(i+1,j,t+1) + E(i+1,j+1,t) + E(i+1,j+1,t+1)\right)$$
$$- \frac{1}{4}\left(E(i,j,t) + E(i,j,t+1) + E(i,j+1,t) + E(i,j+1,t+1)\right)$$
$$E_y(i,j,t) = \frac{1}{4}\left(E(i,j+1,t) + E(i,j+1,t+1) + E(i+1,j+1,t) + E(i+1,j+1,t+1)\right)$$
$$- \frac{1}{4}\left(E(i,j,t) + E(i,j,t+1) + E(i+1,j,t) + E(i+1,j,t+1)\right)$$
$$E_t(i,j,t) = \frac{1}{4}\left(E(i,j,t+1) + E(i,j+1,t+1) + E(i+1,j,t+1) + E(i+1,j+1,t+1)\right)$$
$$- \frac{1}{4}\left(E(i,j,t) + E(i,j+1,t) + E(i+1,j,t) + E(i+1,j+1,t)\right)$$

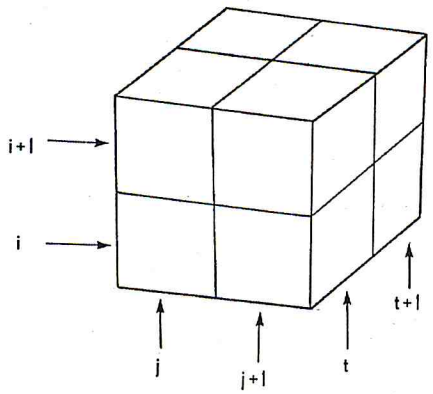


Figure 2 : la mesure des dérivées partielles au centre d'un cube.

Un résumé de la méthode Horn et Schunck est donné par le pseudo l'algorithme suivant :

Début

pour j =1 à N **faire**

pour i=:1 à M **faire**

calculer les valeurs E_x, E_y, et E_t;

initialiser $u(i, j)$ et $v(i, j)$ avec zéro

Fin pour

choisir une valeur lambda {par exemple lambda=0.1}

choisir un nombre n_0>=*1* d'itérations; $\{n_0 = 8\}$

n:=*1*;

Tant que n<= n_0 **faire début**

Pour j: = 1 à N **faire**

pour i:=1 à M **faire**

Calculer $\bar{u}(i, j)$ et $\bar{v}(i, j)$ en fonction des points voisins.

$$\alpha := \frac{E_X \bar{u} + E_y \bar{v} + E_t}{\lambda^2 + E_x^2 + E_y^2} \; ;$$

$u(i,j) := \bar{u} - \alpha \cdot E_X$; $\quad v(i,j) := \bar{v} - \alpha \cdot E_Y$;

Finpour;

n :=n+*1*;

Fin {tant que}

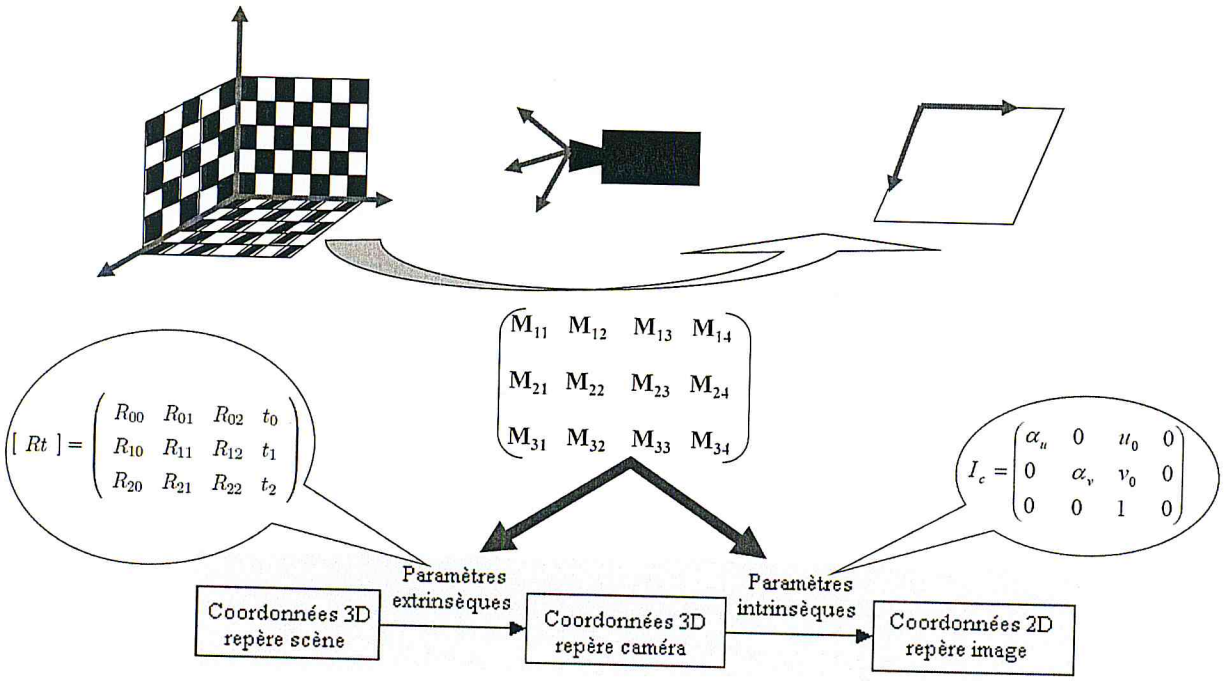
Fin;

Annexe B

Calibrage de caméra

Calibrer une caméra consiste à déterminer la fonction qui associe à un point de l'espace tridimensionnel, sa projection dans l'image, ce qui permet de retrouver les positions des points dans l'espace, connaissant uniquement leurs projections sur les images.

La figure ci-après montre le processus de calibrage de la caméra, dans lequel nous plaçons devant la caméra une mire dont les points sont parfaitement connus dans un repère de la mire qui est différent du repère caméra. Chaque point de la mire se projette dans l'image et on mesure ses coordonnées dans le repère image. La transformation mire /image se décompose donc en une transformation mire /caméra suivie d'une transformation caméra/image.



1. Transformation mire/camera

La transformation mire/caméra se compose d'une rotation et d'une translation, cette matrice est appelé aussi matrice des paramètres extrinsèques.

$$\begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix}$$

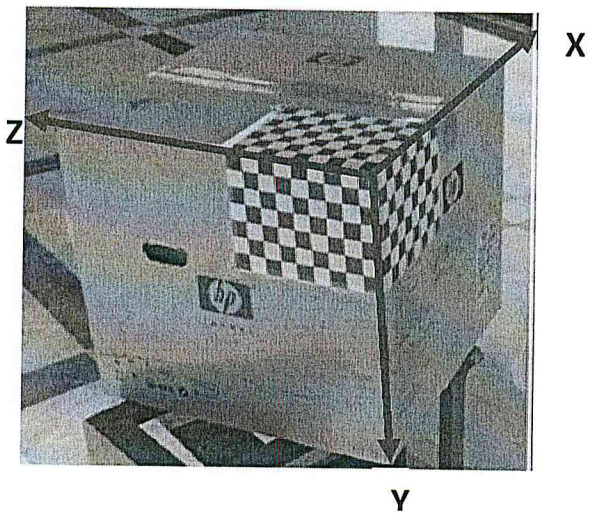
2. Transformation caméra /image

La transformation du repère caméra au repère image est appelée matrice des paramètres intrinsèques :

$$\begin{pmatrix} su \\ sv \\ s \end{pmatrix} = \begin{pmatrix} \alpha_u^i & 0 & u_0^i \\ 0 & \alpha_v^i & v_0^i \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x^C \\ y^C \\ z^C \end{pmatrix}$$

Calibrage de la caméra du robot B21r (Méthode : Faugeras-Toscani)

Le calibrage a été fait avec une mire sous forme de trois damiers dont les plans sont perpendiculaires deux a deux, et dans lesquelles chaque case est un carré de 28 mm. Le repère de la mire a été choisi comme le montre la figure ci-dessous :



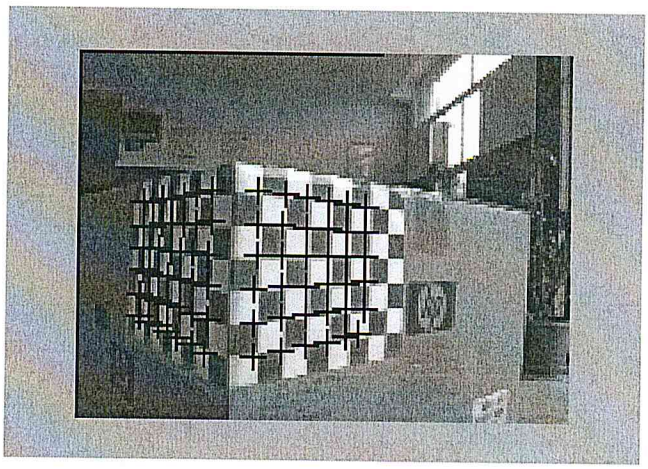
Le calibrage a été fait avec 78 points. Le tableau suivant représente les coordonnées de quelques points, ou (X,Y,Z) sont les coordonnées 3D du point dans le repère de la mire, et (V,U) sont les coordonnées 2D du même point dans le repère image.

X	Y	Z	V	U
28	28	0	60	46
56	28	0	68	47
84	28	0	75	48
112	28	0	83	49
140	28	0	88	49
168	28	0	94	50
28	56	0	59	57

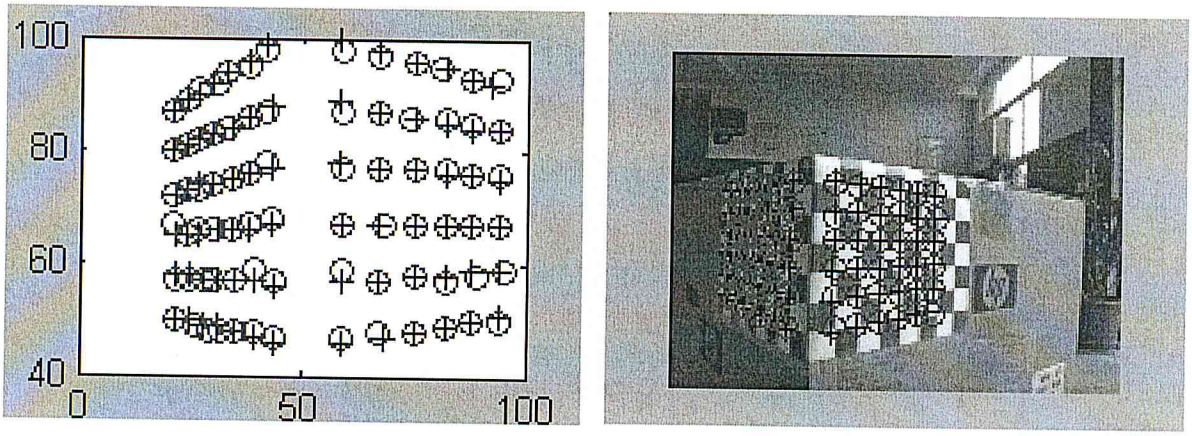
Pour vérifier notre calibrage on projette les points 3D de la mire (colonne X, Y, Z du tableau) sur le plan de l'image en utilisant l'équation :

$$\begin{pmatrix} su \\ sv \\ s \end{pmatrix} = M \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

Les points 2D de la mire calculés avec la matrice M sont montrés par un « + » bleu



Les points calculés sont affichés en même temps que les points mesurés manuellement pour voir s'ils correspondent bien aux vrais points :



Les paramètres intrinsèques à la caméra calculés par calibrage sont :

u0 = 3.4582972 e+ 02

v0 = *3.9561875 e+ 02*

alphu = *-7.3052412 e+ 02*

alphv = *6.0016980 e +02*