

Université de Montréal

**Une approche globale à l'estimation du mouvement de caméra pour
des scènes encombrées**

par
Vincent Chapdelaine-Couture

Département d'informatique et de recherche opérationnelle
Faculté des arts et des sciences

Mémoire présenté à la Faculté des études supérieures
en vue de l'obtention du grade de Maître ès sciences (M.Sc.)
en Informatique

Octobre, 2004

© Vincent Chapdelaine-Couture, 2004.



QA

76

U54

2005

v. 014

AVIS

L'auteur a autorisé l'Université de Montréal à reproduire et diffuser, en totalité ou en partie, par quelque moyen que ce soit et sur quelque support que ce soit, et exclusivement à des fins non lucratives d'enseignement et de recherche, des copies de ce mémoire ou de cette thèse.

L'auteur et les coauteurs le cas échéant conservent la propriété du droit d'auteur et des droits moraux qui protègent ce document. Ni la thèse ou le mémoire, ni des extraits substantiels de ce document, ne doivent être imprimés ou autrement reproduits sans l'autorisation de l'auteur.

Afin de se conformer à la Loi canadienne sur la protection des renseignements personnels, quelques formulaires secondaires, coordonnées ou signatures intégrées au texte ont pu être enlevés de ce document. Bien que cela ait pu affecter la pagination, il n'y a aucun contenu manquant.

NOTICE

The author of this thesis or dissertation has granted a nonexclusive license allowing Université de Montréal to reproduce and publish the document, in part or in whole, and in any format, solely for noncommercial educational and research purposes.

The author and co-authors if applicable retain copyright ownership and moral rights in this document. Neither the whole thesis or dissertation, nor substantial extracts from it, may be printed or otherwise reproduced without the author's permission.

In compliance with the Canadian Privacy Act some supporting forms, contact information or signatures may have been removed from the document. While this may affect the document page count, it does not represent any loss of content from the document.

Université de Montréal
Faculté des études supérieures

Ce mémoire intitulé:

**Une approche globale à l'estimation du mouvement de caméra pour
des scènes encombrées**

présenté par:

Vincent Chapdelaine-Couture

a été évalué par un jury composé des personnes suivantes:

Max Mignotte, président-rapporteur
Sébastien Roy, directeur de recherche
Douglas Eck, membre du jury

Mémoire accepté le: 8 décembre 2004

RÉSUMÉ

Ce mémoire s'inscrit dans le domaine de la vision par ordinateur, et plus précisément dans celui de la recherche automatique de la trajectoire de caméra à partir d'une séquence vidéo. La résolution de ce problème peut mener à plusieurs applications dont la navigation autonome.

Nous analysons des vidéos de scènes encombrées. Ce type de scènes, composé d'une multitude d'objets à plusieurs profondeurs, survient fréquemment en milieu naturel (forêt, plantes, etc.) et en milieu urbain (câbles, salle remplie de meubles, etc.) et sont propices aux occlusions et au problème d'échelle. Par conséquent, les méthodes traditionnelles de flux optique et de suivi de points saillants ne peuvent être appliquées sur ce type de scènes.

La séquence vidéo est transformée dans le domaine de Fourier spatio-temporel, puis analysée pour retrouver la trajectoire de la caméra. Une nouvelle méthode est établie pour l'analyse du mouvement latéral. Cette méthode est ensuite adaptée au mouvement global par rectification de la séquence vidéo. Son analyse ne nécessite ni décomposition en régions ni suivi de points saillants.

Mots-clés : vision par ordinateur, neige optique, flux optique, problème d'occlusions, problème d'échelle, analyse des composantes principales.

ABSTRACT

This thesis is a research in the field of computer vision and deals with egomotion estimation from a video sequence. Solving egomotion can lead to many applications, like automated navigation for instance.

We analyse videos of cluttered scenes, composed of many objects at different depths (forest, plants, wires, etc.). Since scale and occlusion problems are severe in these scenes, traditional optical flow methods cannot be applied.

A video sequence is transformed in the Fourier domain and analysed to estimate the camera motion. First, a new method to analyse lateral motion is presented. Then, the video sequence is rectified to estimate the global 3D motion. This analysis does not require the use of patches or tracking.

Keywords : computer vision, optical flow, occlusion problem, scale problem, principal components analysis (PCA).

TABLE DES MATIÈRES

RÉSUMÉ	iv
ABSTRACT	v
TABLE DES MATIÈRES	vi
LISTE DES FIGURES	ix
LISTE DES TABLEAUX	x
DÉDICACE	xi
REMERCIEMENTS	xii
INTRODUCTION	1
CHAPITRE 1 : TRAVAUX PRÉCÉDENTS	4
CHAPITRE 2 : NEIGE OPTIQUE	7
2.1 Algorithme pour analyser la neige optique	9
2.1.1 Cas particulier : absence de rotation	9
2.1.2 Cas général	10
CHAPITRE 3 : (ARTICLE) PRINCIPAL COMPONENTS ANALYSIS OF OPTICAL SNOW	12
3.1 Introduction	12
3.2 Previous work	13
3.2.1 Optical Snow	13
3.2.2 Bowtie axis estimation using wedge filter	16
3.3 Bowtie Axis Estimation with PCA	18
3.3.1 Comparison of the two best fit plane methods	19

3.3.2	Finding the bowtie axis	21
3.3.3	Results	22
3.3.4	Comparing the eigenvalues	23
3.4	Conclusion	25
CHAPITRE 4 : ESTIMATION DE LA TRAJECTOIRE 3D DE LA		
CAMÉRA		26
4.1	Méthode par décomposition de la vidéo en pièces	27
CHAPITRE 5 : (ARTICLE) A GLOBAL ANALYSIS OF OPTICAL		
SNOW FOR ARBITRARY CAMERA MOTIONS		30
5.1	Introduction	31
5.2	Previous works	31
5.2.1	Optical Snow	31
5.3	Motion Field	34
5.3.1	Motion Field Rectification	34
5.3.2	Finding rotation ω_z	36
5.3.3	Finding translation t_z	36
5.4	Experimental results	36
5.5	Comparing the eigenvalues	37
5.6	Conclusion	38
CHAPITRE 6 : DISCUSSION		39
6.1	Algorithme biologiquement plausible pour l'analyse de la neige optique	39
6.2	Distribution d'énergie uniforme	39
6.3	Une approche hiérarchique	40
6.4	Analyse dans le domaine de Fourier des séquences rectifiées	42
6.5	Mesure robuste de la présence d'un noeud papillon	44
6.6	Distribution d'énergie des profondeurs	44
6.7	Interprétation des deux plans limites	45

CONCLUSION	48
BIBLIOGRAPHIE	49
ANNEXE I : PRÉSENCE D'ÉNERGIE SUR L'AXE DU NOEUD PAPILLON	55
ANNEXE II : CODE DE L'ANALYSE DE COMPOSANTES PRIN- CIPALES AVEC CISAILLEMENT TEMPOREL .	57

LISTE DES FIGURES

1	Exemples de flux observé lors d'un mouvement de caméra	2
2.1	Distribution d'énergie d'un noeud papillon dans le domaine de Fourier	8
2.2	Noeud papillon et prisme utilisé pour l'estimation de son orientation	10
2.3	Cisaillement temporel itératif	11
3.1	Wedge used to estimate the orientation of a bowtie signature	15
3.2	Eigenbasis of a bowtie in the frequency domain	18
3.3	Sequences used to compare the Wedge method and the PCA method	23
3.4	Fitness model of optical snow	25
5.1	Eigenbasis of bowtie signature	33
5.2	Motion rectification	35
5.3	Sequences used to evaluate motion rectification	37
5.4	Fitness model of forward and lateral motions	38
6.1	Distribution d'énergie avant et après normalisation	41
6.2	Exemple de l'effet de normalisation	41
6.3	Séquence test créée par professeur Richard Mann	42
6.4	Exemple de l'effet d'un processus hiérarchique	43
6.5	Estimation de la distribution d'énergie des plans du noeud papillon	45
6.6	Estimation des plans limites du noeud papillon	46
I.1	Séquence Holly utilisée pour mesurer la concentration d'énergie sur l'axe du noeud papillon	55

LISTE DES TABLEAUX

3.1	Comparison between the PCA method and the Wedge method	24
5.1	Motion rectification : results for synthetic and real scenes	37
6.1	Étapes de l'analyse globale de scènes encombrées	47
I.1	Base propre en fonction du poids des énergies	56

À mes parents...

REMERCIEMENTS

Je veux tout d'abord remercier Sébastien Roy, mon directeur de recherche, pour m'avoir aidé à orienter ce projet et à l'approfondir par des discussions enrichissantes.

Je remercie Michael Langer, professeur à l'université McGill, et Richard Mann, professeur à l'université de Waterloo, pour leur disponibilité et leur précieuse collaboration ; ainsi que tous les membres du laboratoire d'informatique pour leur soutien.

Je remercie également mes parents pour leur encouragement, et ma mère pour la lecture et la correction du texte de ce mémoire.

INTRODUCTION

Par ce qu'il voit, un observateur humain peut estimer son propre mouvement. Par exemple, en regardant par la fenêtre d'un train en mouvement, il peut évaluer sa trajectoire relative. L'objet de ce mémoire est d'estimer la trajectoire d'une caméra filmant l'espace environnant encombré d'une multitude d'objets à plusieurs profondeurs.

La trajectoire de la caméra et le mouvement d'objets dans une scène sont deux composantes importantes du mouvement observé dans une séquence vidéo. Cependant, dans ce mémoire, nous ne considérons que le mouvement de caméra dans des scènes statiques.

Le mouvement 3D de caméra génère un flux de mouvement 2D dans l'image (voir Fig. 1). Même si ce flux est connu, l'estimation du mouvement de caméra reste difficile puisque chaque mouvement observé dans la séquence dépend des paramètres du mouvement de caméra et de la profondeur de chaque point visible dans la scène.

Plusieurs approches existent pour l'estimation de la trajectoire de caméra. L'une de celles-ci estime directement le flux optique à chaque pixel à partir du gradient dans une séquence. Le flux optique est considéré depuis plus d'un demi-siècle comme un indice important utilisé par le système visuel humain ^[4]. Cette approche suppose l'existence d'une vitesse unique à chaque point dans le champ visuel ^[3,44]. Cette supposition n'est cependant valide que s'il y a continuité dans l'espace, comme la surface d'un objet lisse. En effet, si un observateur se déplace devant une scène 3D encombrée, une forêt par exemple, cette supposition n'est plus valide ; les feuilles et les branches situées à plusieurs profondeurs causent beaucoup d'occlusions et des discontinuités du mouvement observé. Les frontières des nombreux objets présents dans la scène rendent donc difficile le calcul du flux optique. Or, on observe qu'une personne peut se promener facilement à travers des environnements 3D encombrés ; des études psychologiques confirment d'ailleurs qu'un observateur humain peut s'orienter dans un nuage de points ^[43].

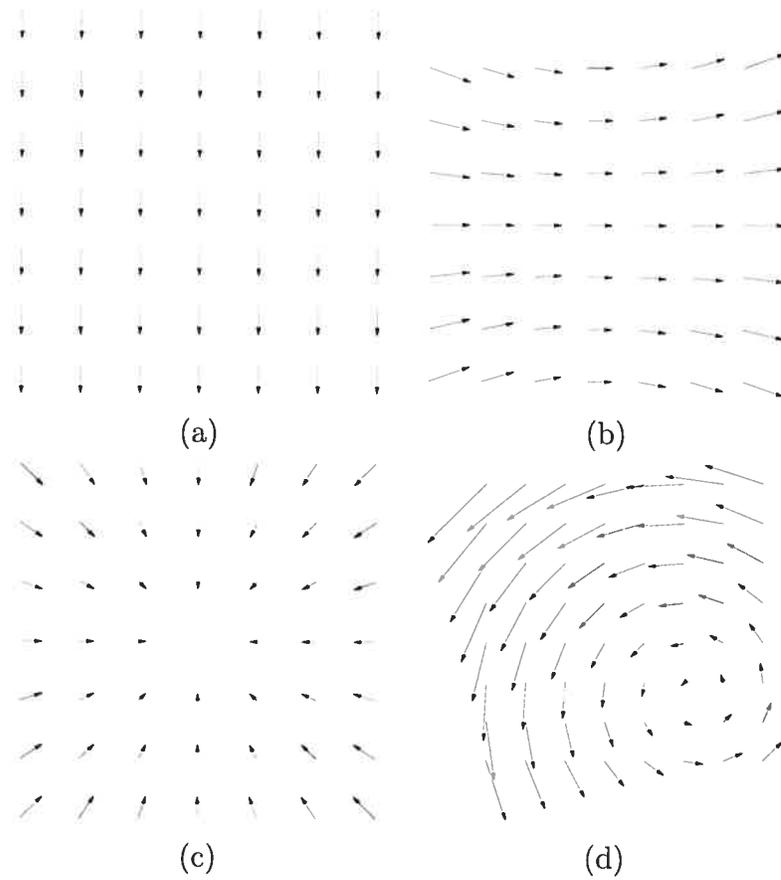


FIG. 1 – Exemples de flux observé lors d'un mouvement de caméra (a) Translation vers le haut (a) Rotation vers la gauche (a) Translation vers l'arrière (a) Trajectoire complexe.

Une autre approche se base sur le suivi de points saillants (coins, frontières, etc.). Cette approche robuste et rapide devient par contre difficile à utiliser pour des scènes encombrées dans lesquelles il y a une multitude de petits objets difficiles à suivre à cause des occlusions.

Récemment, Langer et Mann ^[9] ont introduit une nouvelle catégorie de mouvements appelée neige optique. La neige optique généralise le flux optique en abandonnant toute contrainte de continuité spatiale. Les scènes de neige optique (ou encombrées) sont composées par définition de petits objets situés à une multitude de profondeurs. Comme mentionné précédemment, si l'on abandonne la contrainte de continuité des profondeurs, les méthodes traditionnelles de flux optique ne peuvent plus être utilisées. La neige optique fournit d'importants indices sur la profondeur et le mouvement, et s'avère particulièrement utile au calcul de la trajectoire de caméra.

Nous estimerons, dans ce mémoire, la trajectoire de caméra en utilisant les principes de la neige optique, sans calculer le flux et sans suivi de points saillants. Au chapitre 1, nous ferons une revue des principaux travaux s'inscrivant dans le domaine de l'estimation du mouvement de caméra. Puis, au chapitre 2, il sera question des principes de la neige optique établies dans les travaux des professeurs Langer et Mann. Nous présenterons, au chapitre 3, un algorithme pour résoudre rapidement le modèle de neige optique en utilisant l'analyse des composantes principales (PCA). Au chapitre 4, nous résumerons les travaux reliés à l'estimation de la trajectoire 3D de la caméra pour des scènes encombrées. Soulignons que ces travaux proposent une analyse des mouvements latéraux par régions, alors que nous présentons, au chapitre 5, une rectification de la séquence originale qui permet une estimation rapide et globale du mouvement arbitraire de caméra pour des scènes encombrées. Cette approche ne nécessite aucune décomposition en régions. Nous terminerons par une discussion au chapitre 6, qui sera suivie d'une conclusion.

CHAPITRE 1

TRAVAUX PRÉCÉDENTS

Ce chapitre résume les travaux reliés au problème de l'estimation du mouvement de caméra, excepté les articles ayant servi de base pour ce mémoire et qui feront l'objet des chapitres 2 et 4.

Plusieurs méthodes de vision par ordinateur ont été développées pour analyser la trajectoire du mouvement de caméra. Des hypothèses sont habituellement associées au modèle de la trajectoire de caméra ou au modèle de l'environnement. Par exemple, Adiv ^[14] a supposé un modèle simplifié du monde, composé de surfaces planaires; Horn et Weldon ^[15] ont limité l'espace des mouvements possibles (translation pure, rotation pure, ou trajectoire générale avec profondeur connue); Fleet et Jepson ^[16] ont supposé la constance temporelle d'un mouvement sur une longue séquence. D'autres méthodes ^[17-19] utilisent une analyse spatio-temporelle pour estimer les vitesses locales d'une image avec de larges régions temporelles.

Le mouvement 3D de la caméra est souvent estimé à l'aide du flux optique ou du flux normal (flux optique dans le sens du gradient de l'image) dérivé à partir de deux images ^[14, 20, 22-30], ou à l'aide de la correspondance entre des points saillants préalablement détectés d'images successives ^[31-33]. Ces deux approches dépendent de la texture dans l'image et de la précision des étapes de prétraitement ^[34]. Le calcul du flux normal est plus précis que celui du flux optique, mais son analyse est plus ambiguë. De plus, le calcul du flux optique dépend de la continuité spatiale dans la scène. Cette hypothèse n'est plus valide pour des scènes encombrées. Quant aux points saillants, ils peuvent être détectés sur un plus long intervalle de temps en utilisant des filtres temporels récurrents ^[16]. Mais le problème de repérage de points saillants utiles et celui des occlusions persistent.

Plusieurs auteurs ^[15, 35, 36] proposent de calculer la trajectoire de caméra directement à partir de l'intensité des pixels. Horn et Weldon ^[15] analysent seulement des cas particuliers, comme décrit précédemment. Hanna ^[35] suggère une méthode

itérative pour des cas généraux, mais requiert une approximation initiale raisonnable. Taalebinezhad ^[36] a dérivé une formulation mathématique complète, mais s'appuie sur des calculs à partir de points d'ancrage de même intensité entre deux images, ce qui peut être très instable.

Les rotations et les translations 3D induisent des images 2D similaires ^[37-39], ce qui cause des ambiguïtés d'interprétation, particulièrement lorsqu'il y a présence de bruit. Retrouver le mouvement de caméra 3D à partir du flux optique est un problème mal posé puisque de petites erreurs dans le flux optique 2D peuvent causer de larges perturbations dans le mouvement 3D obtenu^[37]. Par contre, il est beaucoup plus facile de distinguer les effets de rotations et de translations 3D aux discontinuités de profondeur, puisque le mouvement 2D de pixels rapprochés à des profondeurs différentes a des composantes rotationnelles très similaires, mais des composantes translationnelles très différentes ^[40]. À partir de cette observation, des méthodes utilisant la parallaxe de mouvement ont été construites pour obtenir la trajectoire 3D de la caméra ^[20,33,40-42]. Il est intéressant de noter que les discontinuités de profondeur donnent beaucoup d'information pour distinguer les composantes translationnelles et rotationnelles, mais que le flux optique y est plus difficile à calculer. Dans ^[20,40], différents vecteurs, calculés à partir du flux optique, annulent les effets des rotations, et le point d'expansion (FOE : focus of expansion, l'intersection entre le plan image et la direction de la composante translation du mouvement de caméra et l'image) est obtenu à l'intersection de ces vecteurs. Cependant, les discontinuités de profondeur rendent le flux optique très imprécis, affectant ainsi le calcul de la parallaxe de mouvement.

Heeger ^[12] utilise des filtres de Gabor pour une analyse dans le domaine fréquentiel. La méthode repose sur la propriété qu'une région à profondeur constante subissant une translation produit un plan d'énergie dans le domaine fréquentiel.

Récemment, Langer et Mann ^[7,9] ont introduit une nouvelle catégorie de mouvements appelée neige optique dans laquelle aucune continuité spatiale n'est supposée. Nous utilisons, dans ce mémoire, les principes de la neige optique pour estimer le mouvement de caméra. Notre méthode ne requiert aucun suivi de points saillants

et aucune décomposition en régions. Puisque l'analyse se fait dans le domaine de Fourier, nous supposons une constance temporelle du mouvement de caméra d'une longueur choisie par l'utilisateur, c'est-à-dire que le mouvement de caméra doit demeurer constant pour un nombre T d'images consécutives.

CHAPITRE 2

NEIGE OPTIQUE

Ce chapitre introduit le modèle de neige optique permettant l'analyse de scènes encombrées. Dans de telles scènes, aucune continuité spatiale n'est supposée et plusieurs vitesses peuvent être perçues aux environs d'un pixel. Les méthodes traditionnelles d'estimation du flux optique ne sont donc plus valides. De plus, les scènes encombrées contiennent un grand nombre d'occlusions, ce qui rend le suivi de points saillants très difficile, voire impossible.

Les principes de la neige optique ^[7,9] sont une extension de la propriété d'un plan en mouvement ^[5] qui énonce qu'une région de la scène de profondeur constante subissant une translation latérale (c'est-à-dire perpendiculaire à l'axe optique) de vitesse $(v_x, v_y, 0)$ produit un plan dans le domaine de Fourier.

De façon plus formelle, soit $I(x,y,t)$ une séquence d'images. Si une région de la séquence subit une translation pure de vitesse (v_x, v_y) , nous savons de ^[45] que cette vitesse est contrainte par

$$v_x \frac{\partial I}{\partial x} + v_y \frac{\partial I}{\partial y} + \frac{\partial I}{\partial t} = 0 \quad (2.1)$$

La transposition de cette contrainte dans le domaine de Fourier produit l'équation suivante :

$$-2\pi(v_x f_x + v_y f_y + f_t) \hat{I}(f_x, f_y, f_t) = 0 \quad (2.2)$$

où $\hat{I}(f_x, f_y, f_t)$ est la transformée de Fourier de $I(x, y, t)$. L'équation 2.2 implique que toutes les fréquences non nulles de $\hat{I}(x, y, t)$ sont sur le plan $v_x f_x + v_y f_y + f_t = 0$.

Dans ^[7,9], ce modèle a été étendu à la situation où un ensemble de vitesses ont un paramètre libre à l'intérieur d'une région, c'est-à-dire où les vitesses varient

selon l'équation :

$$(v_x, v_y) = (u_x + \alpha t_x, u_y + \alpha t_y) \quad (2.3)$$

où u_x, u_y, t_x et t_y sont des constantes et α est une variable inversement proportionnelle à la profondeur visible au point (x, y) . Comme décrit plus en détails au chapitre 4, u_x, u_y sont reliées à la rotation de la caméra et t_x, t_y , à sa translation.

Ce nouveau modèle porte le nom de neige optique et produit plusieurs plans dans le domaine fréquentiel. Ces plans forment un noeud papillon (voir figure 2.1) qui respecte les deux propositions suivantes :

Proposition 1 : Les plans du noeud papillon s'intersectent en une ligne, appelée l'axe du noeud papillon, passant par l'origine.

Proposition 2 : L'axe du noeud papillon suit la direction $(-t_y, t_x, u_x t_y - u_y t_x)$. En normalisant le vecteur (t_x, t_y) , cette direction devient $(-t_y, t_x, |U| \sin(\theta))$, où θ est l'angle entre les vecteurs (t_x, t_y) et (u_x, u_y) .

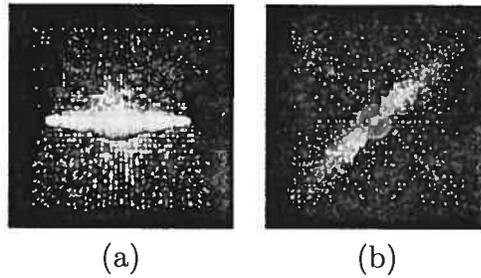


FIG. 2.1 – (a) Distribution d'énergie dans le domaine de Fourier (a) d'un point de vue perpendiculaire à l'axe du noeud papillon (b) d'un point de vue parallèle à l'axe du noeud papillon.

La neige optique ne suppose aucune continuité spatiale, contrairement aux modèles standards de mouvement comme le flux optique ou le mouvement en couches. Le modèle de neige optique permet donc de grandes discontinuités de profondeur. En fait, l'ouverture du noeud papillon s'agrandit en fonction de α qui varie proportionnellement à la quantité de profondeurs dans une région. Par contre,

s'il n'y a qu'une profondeur visible, le noeud papillon devient un simple plan et la notion d'axe ne tient plus.

Puisque les deux premières composantes de l'axe du noeud papillon donnent la direction de la translation, cet axe permet donc de séparer les composantes rotationnelles et translationnelles du mouvement. Sans cet axe, ces composantes sont confondues.

2.1 Algorithme pour analyser la neige optique

Langer et Mann ^[9] ont tout d'abord développé un algorithme pour trouver l'orientation du noeud papillon pour une trajectoire de caméra sans composantes rotationnelles u_x et u_y . Cet algorithme a ensuite été modifié dans ^[6] pour permettre la présence des composantes rotationnelles.

2.1.1 Cas particulier : absence de rotation

Langer et Mann ^[9] ont développé un algorithme pour trouver l'orientation du noeud papillon dans le cas où $u_x = u_y = 0$ (c'est-à-dire aucune rotation), ou lors du suivi d'un point dans la scène, c'est-à-dire lorsque le vecteur $(-u_y, u_x)$ est parallèle au vecteur (t_x, t_y) . Dans ces deux cas, l'axe du noeud papillon repose dans le plan de pente $(f_x, f_y, 0)$ en direction de $(-t_y, t_x, 0)$.

À partir du spectre des puissances $|\hat{I}_{xyt}|^2$, l'algorithme développé dans ^[9] profite de la géométrie du noeud papillon. L'orientation de l'axe du noeud papillon correspond à l'angle qui accumule le moins d'énergie dans un prisme à base triangulaire de pente v_{max} (voir Fig. 2.2-b). L'énergie minimale, nulle en théorie, se trouve en pratique affectée par les effets de frontières (exemple d'effet de frontières).

Plus précisément, pour un angle θ , Langer et Mann ^[9] définissent f_θ comme suit :

$$f_\theta \equiv \cos\left(\theta + \frac{\pi}{2}\right)f_x + \sin\left(\theta + \frac{\pi}{2}\right)f_y$$

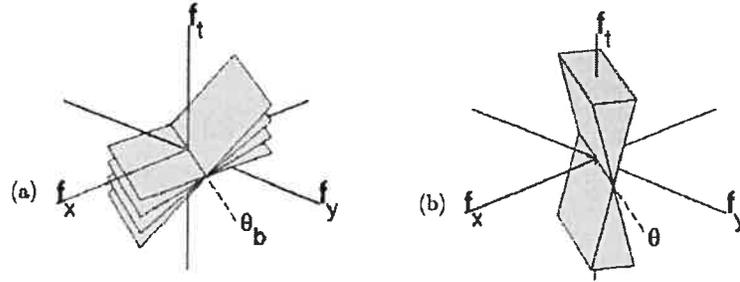


FIG. 2.2 – (a) Noeud papillon dans le domaine fréquentiel. L'axe du noeud papillon est dans la direction θ_b . (b) Prisme utilisé pour l'estimation de l'orientation du noeud papillon.

Soit T le nombre d'images dans la séquence et N la taille des images. L'énergie est accumulée suivant la condition $|\frac{f_t}{f_\theta}| < v_{max} \frac{T}{N}$.

De plus, pour s'assurer que l'énergie accumulée minimale soit de 0, Langer et Mann ^[9] n'utilisent que les fréquences qui satisfont $\|(f_\theta, f_t)\|_2 > \tau$ en raison de l'imprécision de la vitesse des basses fréquences.

Avant d'effectuer la transformée de Fourier, l'intensité moyenne de la séquence est soustraite de la séquence originale pour enlever la composante continue (le DC). Un fenêtrage Gaussien de la séquence originale, d'un écart type d'un sixième de la taille de l'image, est aussi appliqué à la séquence pour réduire le bruit dû aux effets de bord dans le domaine de Fourier.

2.1.2 Cas général

Dans le cas où il y a présence de rotation, l'axe du noeud papillon ne repose plus dans le plan de pente $(f_x, f_y, 0)$. Mann et Langer ^[6] utilisent une compensation de mouvement ^[46] pour annuler l'effet de la rotation. Pour ce faire, il faut trouver un plan $\pi = (m_x, m_y, 1)$ qui minimise la somme des différences :

$$\sum_{(f_x, f_y, f_t) \in N \times N \times M} f_t^2 |\hat{I}(f_x, f_y, (f_t - m_x f_x - m_y f_y) \bmod M)|^2.$$

Une fois ce plan estimé, les puissances sont déplacées suivant l'axe f_t d'une distance $f_x m_x + f_y m_y$. Ce cisaillement est itératif jusqu'à ce que π corresponde au

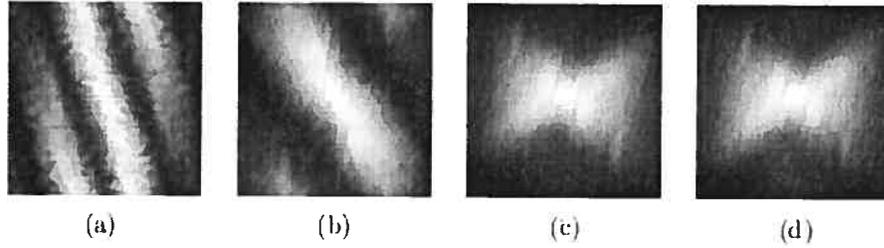


FIG. 2.3 – Processus du cisaillement temporel itératif, des puissances originales en (a) aux puissances complètement cisillées en (d). On peut clairement voir en (a) l’aliassage temporel causé par les pentes trop abruptes des plans.

plan de pente $(f_x, f_y, 0)$, c’est-à-dire ayant comme normale $(0, 0, 1)$ (voir Fig. 2.3). Puisque tous les plans du noeud papillon contiennent l’axe du noeud papillon, le plan π contient aussi l’axe du noeud papillon et l’axe repose donc dans le plan $(f_x, f_y, 0)$. L’axe du noeud papillon peut ensuite être trouvé à l’aide de la méthode décrite à la section 2.1.1.

La composante rotationnelle normale \vec{u} est ensuite calculée selon l’équation :

$$\vec{u} = \mathbf{m} - (\mathbf{m} \cdot \vec{\tau})\vec{\tau}$$

où $\vec{\tau}$ est la direction translationnelle retrouvée à partir de l’axe du noeud papillon et $\mathbf{m} = (m_x, m_y)$.

Nous présentons dans l’article qui suit une nouvelle méthode basée sur l’analyse des composantes principales permettant l’analyse rapide de la neige optique dans le cas général et ne nécessitant aucun paramètre.

CHAPITRE 3

(ARTICLE) PRINCIPAL COMPONENTS ANALYSIS OF OPTICAL SNOW

Cet article ^[63] a été publié comme l'indique la référence bibliographique

V. C.-Couture, S. Roy, M. S. Langer et R. Mann. "Principal Components Analysis of Optical Snow", dans *British Machine Vision Conference 2004 (BMVC'04)*, Kingston (UK), septembre 2004, vol. 2, pages 799-808.

L'article est présenté ici dans sa version originale.

Many applications in computer vision use Principal Components Analysis (PCA), for example, in camera calibration, stereo, localization and motion estimation. We present a new and fast PCA-based method to analyze optical snow. Optical snow is a complex form of visual motion that occurs when an observer moves through a highly cluttered 3D scene. For this category of motion field, no spatial or depth coherence can be assumed. Previous methods for measuring optical snow have used a wedge filter in a spatiotemporal frequency domain. The PCA method is also based on the spatiotemporal frequency domain analysis, but examines a different geometry property of the spectrum. We compare the results of the PCA method to the previous methods using both real and synthetic sequences.

3.1 Introduction

Recently, Langer and Mann ^[7] considered a category of visual motion called *optical snow* which generalizes optical flow by abandoning classical assumptions of spatial continuity. Optical snow arises when an observer moves relative to a rigid 3-D cluttered scene or object (falling snow, forest, plants). Optical snow is characterized by small spatial features and a dense set of depth discontinuities. It produces a highly discontinuous motion field. Traditional optical flow methods

cannot be expected to recover an image velocity field from optical snow since these methods typically assume local smoothness in the velocity field [3], or a small number of well-isolated discontinuities e.g. [53–55]. By definition, smoothness constraints do not apply for optical snow. Layered motion models such as [13, 57–60] also do not apply, since these models assume smoothness within layers[56], and the number of layers is typically very small (2 or 3). In optical snow, the number of “layers” can be in the hundreds or more.

To overcome the dense depth discontinuity problem that arises in measuring optical snow, Langer and Mann [7] introduced an analysis of the motion which was based in the spatio-temporal frequency domain. In [8,9], they presented an algorithm to find the direction of the motion for the special case of parallel optical snow, which arises for example in the case of a lateral camera motion [8,61]. This algorithm was then generalized in [6] to handle the case of optical snow that arises from general observer motion, including both non-lateral motion and camera roll.

The previous method for measuring optical snow uses a wedge-filter to estimate the distribution of power in the spatio-temporal Fourier power spectrum. This filter was motivated by the geometric properties of optical snow in the spatio-temporal frequency domain (see Sec. 3.2). In this paper, we present a new and fast algorithm which uses PCA to estimate the properties of optical snow. The PCA method is complementary to the wedge filter method, in that it is based on a geometrical property of the power spectrum that the wedge method ignores. We show the effectiveness of the PCA method by comparing experimental results of the PCA method to that of the wedge method of [9].

3.2 Previous work

3.2.1 Optical Snow

Optical snow is produced by an observer moving relative to a rigid 3D cluttered scene. It is a special case of *motion parallax* in which the density of depth discontinuities is high. For motion parallax in general (and optical snow in particular), the

image velocity field can be closely approximated as a sum of two fields – one that is due to camera translation and the other due to camera rotation ^[2]. As a result, the image velocity vectors in a local image patch satisfy the relation

$$(v_x, v_y) = (u_x + \alpha t_x, u_y + \alpha t_y) \quad (3.1)$$

where u_x, u_y, t_x, t_y are constants which depend on the patch and α depends on position x, y in the patch and on the depth of the point visible at (x, y) ^[6]. Thus, optical flow yields a one parameter family of velocities (a line!) in any local image patch.

One can examine the power spectrum properties of optical flow by using Eq. (3.1) to extend the classical motion plane property ^[5, 12]. The motion plane property states that an image translating with uniform image velocity produces a plane of energy in the 3D spatio-temporal frequency domain. Formally, let $I(x, y, t)$ be a time varying image which is defined by an image patch translating with velocity (v_x, v_y) . From ^[45], this velocity is constrained by

$$v_x \frac{\partial I}{\partial x} + v_y \frac{\partial I}{\partial y} + \frac{\partial I}{\partial t} = 0 \quad . \quad (3.2)$$

This constraint, transposed in the Fourier domain, yields

$$(v_x f_x + v_y f_y + f_t) \hat{I}(f_x, f_y, f_t) = 0 \quad (3.3)$$

where $\hat{I}(f_x, f_y, f_t)$ is the Fourier transform of $I(x, y, t)$. Eq. (3.3) implies that all frequencies (f_x, f_y, f_t) for which $\hat{I}(f_x, f_y, f_t) \neq 0$ lie on the plane

$$v_x f_x + v_y f_y + f_t = 0 \quad . \quad (3.4)$$

Substituting Eq. (3.1) into Eq. (3.4) yields a family of planes in the frequency domain,

$$(u_x + \alpha t_x) f_x + (u_y + \alpha t_y) f_y + f_t = 0$$

This set of planes forms a *bowtie* (see Fig. 3.1a) and follows the two following propositions ^[9] :

Proposition 1 : The planes of the bowtie intersect at a common line, called the *axis of the bowtie*, that passes through the origin.

Proposition 2 : The axis of the bowtie is in direction $(-t_y, t_x, u_x t_y - u_y t_x)$ in the 3D frequency domain. By normalizing (t_x, t_y) , this direction becomes $(-t_y, t_x, |U| \sin(\phi))$, where ϕ is the angle between vectors (t_x, t_y) and (u_x, u_y) . By assuming that (t_x, t_y) is perpendicular to (u_x, u_y) , we get $(-t_y, t_x, \sqrt{u_x^2 + u_y^2})$.

For simplicity, all equations in this paper assume image sequences of equal dimension in space and time, i.e. the size N of image region equals the number T of frames. In practice, the case of unequal dimensions can be easily be accounted for using factors of $\frac{N}{T}$.

We emphasize that optical snow is a very general model of visual motion. It describes the velocities of *any* rigid 3D scene as seen by a moving observer ^[6,9]. The model applies whether the surfaces are partly transparent, smooth, layered, or densely cluttered. The model does not assume any spatial continuity. This is in sharp contrast to conventional optical flow estimation which relies on spatial coherence of the motion fields or uses motion layers model.

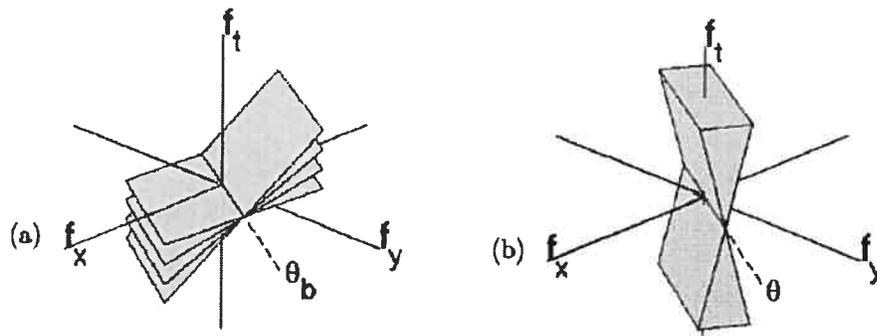


FIG. 3.1 – (a) Bowtie signature in the frequency domain. The axis of bowtie is in direction θ_b . (b) Wedge used to estimate the orientation of the bowtie. The dotted line at angle θ in the (f_x, f_y) is denoted l_θ .

3.2.2 Bowtie axis estimation using wedge filter

Langer and Mann ^[7,9] introduced an algorithm to find the bowtie axis in the case that $(u_x, u_y) = (0, 0)$, where no rotation is allowed. In this case, the bowtie axis lies in the $(f_x, f_y, 0)$ plane and the direction of the bowtie axis is $(-t_y, t_x, 0)$. To find the bowtie axis, a *wedge filter* is used. The wedge filter is defined by two motion planes, of slope $\pm v_{max}$, respectively. These two planes intersect at a line in the (f_x, f_y) plane, which is oriented at an angle θ (see figure 3.1). We let l_θ denote this line. It corresponds exactly to the axis of the bowtie when $\theta = \theta_b$. If all motion planes in the bowtie have slopes of magnitude that is less than v_{max} then, for this θ , the intersection of the wedge filter and the bowtie will be restricted entirely to the bowtie axis.

To find the bowtie axis, the wedge filter is rotated (θ is varied) and the power that falls within the wedge filter is measured as a function of θ . In particular – and this is an important detail – a small cylinder containing the line l_θ is removed from the wedge filter. This removes all power from the wedge filter for the case that $\theta = \theta_b$. The angle θ_b of the bowtie axis is estimated to be the angle for which the power in the wedge filter is a minimum.

The more general case is that $(u_x, u_y) \neq (0, 0)$. In this case, the bowtie axis does not lie in the (f_x, f_y) plane (see Proposition 2). To estimate the bowtie axis in this case, a method was presented in ^[6]. This method has two steps. The first step is to find the best fit motion plane for the bowtie and then shear the power spectrum in the f_t direction to bring this motion plane in correspondence with the (f_x, f_y) plane. This shear amounts to *motion compensation* ^[46] of the image sequence (within the patch in question) such that the mean velocity in the patch becomes 0. In particular, since the bowtie axis is contained in all motion planes within the bowtie, this best fit plane also contains the bowtie axis. The second step of the method for finding the bowtie is to find the θ that best aligns the wedge filter to the bowtie, as described above for the case that the bowtie axis happens to already lie in the (f_x, f_y) plane.

To perform motion compensation, Mann and Langer ^[6] find the vector (v_x, v_y) that minimizes the sum of squares :

$$g(v_x, v_y) = \sum_{f_x, f_y, f_t} |\hat{I}(f_x, f_y, f_t)|^2 (f_t - (v_x f_x + v_y f_y) \bmod N)^2$$

We re-express the motion compensation step, in a way that will allow us to compare it to the PCA method, which we introduce in this paper.

Let \mathbf{F} be the $N^3 \times 3$ matrix which lists the N^3 possible frequency triplets (f_x, f_y, f_t) , where each f_x, f_y, f_t value is in $\{0, \dots, N-1\}$. Let \mathbf{W} be an $N^3 \times N^3$ diagonal matrix of weights, with diagonal elements,

$$W(f_x, f_y, f_t) = |\hat{I}(f_x, f_y, f_t)|.$$

Consider the matrix product

$$\mathbf{F}^T \mathbf{W}^2 \mathbf{F} = \begin{pmatrix} \sum f_x^2 W(f_x, f_y, f_t)^2 & \sum f_x f_y W(f_x, f_y, f_t)^2 & \sum f_x f_t W(f_x, f_y, f_t)^2 \\ \sum f_x f_y W(f_x, f_y, f_t)^2 & \sum f_y^2 W(f_x, f_y, f_t)^2 & \sum f_y f_t W(f_x, f_y, f_t)^2 \\ \sum f_x f_t W(f_x, f_y, f_t)^2 & \sum f_t f_y W(f_x, f_y, f_t)^2 & \sum f_t^2 W(f_x, f_y, f_t)^2 \end{pmatrix}$$

where each of the summations is a triple summation over f_x, f_y, f_t , namely over the entire 3D frequency domain. In the case that there is no temporal aliasing, the modulus N can be ignored and the summation is equivalent to

$$g(v_x, v_y) \equiv (v_x, v_y, -1) \mathbf{F}^T \mathbf{W}^2 \mathbf{F} (v_x, v_y, -1)^T$$

This method thus finds the best fitting motion plane by minimizing the weighted squared distance in the f_t direction.

With this background, let us now turn to the original contribution of this paper which is to introduce a new, simpler and faster PCA-method for finding the bowtie axis.

3.3 Bowtie Axis Estimation with PCA

The basic idea of our PCA method is to find a line in the frequency domain that passes through the origin and contains a concentration of power. It does so by computing the eigenvectors of a 3×3 covariance matrix which is defined in the (f_x, f_y, f_t) domain. This strategy is justified by the fact that the energy on the bowtie axis is the superposition of the energies (not merely the intersection) of the planes that contribute to the bowtie (see Fig. 3.2).

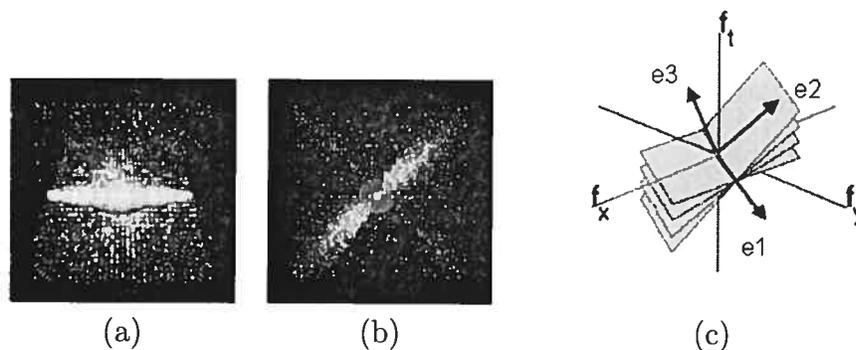


FIG. 3.2 – (a) Energy distribution in the Fourier domain from a viewpoint perpendicular to the axis of the bowtie (b) Energy distribution in the Fourier domain from a viewpoint parallel to the axis of the bowtie (c) Eigenbasis of bowtie signature.

The PCA method uses classical *orthogonal distance regression* [1], which finds the best fit line and best fit plane to a set of 3D points. In our problem, the 3D space is the frequency domain and so the coordinates are (f_x, f_y, f_t) . The lines and planes we are interested in pass through the origin. Let

$$n_x f_x + n_y f_y + n_t f_t = 0$$

be the equation of a plane that passes through the origin, where

$$\mathbf{n}^T = (n_x, n_y, n_t)$$

is a vector normal to the plane. We wish to minimize the *sum of squared orthogonal*

weighted distances of each (f_x, f_y, f_t) to this plane,

$$f(\mathbf{n}) = \frac{1}{\mathbf{n}^T \mathbf{n}} \sum_{f_x, f_y, f_t} W(f_x, f_y, f_t)^2 |n_x f_x + n_y f_y + n_t f_t|^2 .$$

The function $f(\mathbf{n})$ can be expressed as the matrix product

$$f(\mathbf{n}) = (\mathbf{n}^T \mathbf{F}^T \mathbf{W}^2 \mathbf{F} \mathbf{n}) / (\mathbf{n}^T \mathbf{n})$$

where \mathbf{F} is the same matrix as in Sec. 3.2. We will discuss our choice of $W(f_x, f_y, f_t)$ below.

The function $f(\mathbf{n})$ is a *Rayleigh quotient*. This quotient is maximized and minimized by the eigenvectors corresponding to the largest smallest eigenvalues of $\mathbf{F}^T \mathbf{W}^2 \mathbf{F}$, respectively. The minimum eigenvector defines the normal of the best fit plane π . The maximum eigenvector defines to the best fit line l which is considered to be the bowtie axis. The line l always lies in the plane π , since $\mathbf{F}^T \mathbf{W}^2 \mathbf{F}$ is a symmetric real 3×3 matrix and so it has an orthogonal set of eigenvectors.

3.3.1 Comparison of the two best fit plane methods

Both the wedge filter method of [6,9] and our new PCA method compute a minimization which finds a motion plane that best fits a bowtie. However, there are several differences between the methods.

Let us first briefly mention two minor differences. One is that $f(\mathbf{n})$ measures the weighted distance orthogonal to a plane, whereas $g(v_x, v_y)$ measures the weighted distance in the f_t direction only. One can make arguments for either of these distance choices. Indeed similar non-orthogonal [12,62] vs. orthogonal [50,52] choices were made in classical optical flow. A second difference is that the $f(\mathbf{n})$ minimization normalizes by the length of \mathbf{n} , whereas the $g(v_x, v_y)$ minimization does not normalize by the length of $(v_x, v_y, -1)$. The effect of normalizing is to explicitly penalize for high speeds. Again, arguments can be made for or against.

A more significant difference between the two methods is the choice of $W(f_x, f_y, f_t)$.

For wedge filter method ^[6], the weighting function was the amplitude spectrum $|\hat{I}(f_x, f_y, f_t)|$. The amplitude spectrum turns out to be inappropriate for the PCA formulation, however.

To understand how problems can arise, consider an illustrative example. Suppose that an image sequence is defined by summing two translating image sequences

$$I(x, y, t) = I_1(x, y, t) + I_2(x, y, t) .$$

The Fourier transform of the summed image sequence is

$$\hat{I}(f_x, f_y, f_t) = \hat{I}_1(f_x, f_y, f_t) + \hat{I}_2(f_x, f_y, f_t)$$

with power spectrum

$$\begin{aligned} |\hat{I}(f_x, f_y, f_t)|^2 &= |\hat{I}_1(f_x, f_y, f_t)|^2 + |\hat{I}_2(f_x, f_y, f_t)|^2 \\ &\quad + 2 \operatorname{Re}(\hat{I}_1(f_x, f_y, f_t) \overline{\hat{I}_2(f_x, f_y, f_t)}) . \end{aligned}$$

Let's consider the expected value. The phases of $\hat{I}_1(f_x, f_y, f_t)$ and $\hat{I}_2(f_x, f_y, f_t)$ can be assumed independent, so we have

$$E \{ \hat{I}_1(f_x, f_y, f_t) \overline{\hat{I}_2(f_x, f_y, f_t)} \} = 0$$

and so

$$E \{ |\hat{I}(f_x, f_y, f_t)|^2 \} = E \{ |\hat{I}_1(f_x, f_y, f_t)|^2 \} + E \{ |\hat{I}_2(f_x, f_y, f_t)|^2 \} .$$

The expected power spectra on the bowtie axis is thus the sum of the expected power spectrum of the two component image sequences. Thus there is a concentration of power on the bowtie axis, which is where the two planes superimpose.

Unfortunately, there is no reason why the PCA method should detect this concentration of power, if we choose $W(f_x, f_y, f_t) = | \hat{I}(f_x, f_y, f_t) |$ as does the wedge method. To illustrate, take the counter-example in which the first frame (and hence all frames) of the two translating images have the same amplitude spectra, $| \hat{I}_1(f_x, f_y) | = | \hat{I}_2(f_x, f_y) |$, but they have independent phase spectra. Further suppose that two translating images sequences have equal but opposite velocities. Then one can show that two of the eigenvectors of $\mathbf{F}^T \mathbf{W}^2 \mathbf{F}$ lie in the plane (f_x, f_y) and the third eigenvector is parallel to f_t . Moreover, the direction of the two eigenvectors in (f_x, f_y) depends on only on the amplitude spectra, $| \hat{I}_1(f_x, f_y) | = | \hat{I}_2(f_x, f_y) |$. These eigenvector directions would be independent of the direction of image motion. Thus, for this counterexample, the PCA method would *not* be able to recover the bowtie axis, even though there *is* a concentration of power there. We have done experiments to show that this difficulty arises for more general sequences (voir l'annexe I).

3.3.2 Finding the bowtie axis

The difficulty that we just described arises because the power spectrum is a second order property, whereas the concentration of power that occurs on the bowtie axis is a higher order property. This difficulty is avoided by using a higher order power of the Fourier coefficients, namely :

$$W(f_x, f_y, f_t) = | \hat{I}(f_x, f_y, f_t) |^2$$

Running a similar argument as above on the expected values yields :

$$\begin{aligned}
& E \{ | \hat{W}(f_x, f_y, f_t) |^2 \} \\
= & E \{ | \hat{I}(f_x, f_y, f_t) |^4 \} \\
= & E \{ (| \hat{I}_1(f_x, f_y, f_t) |^2 + | \hat{I}_2(f_x, f_y, f_t) |^2 \\
& + 2 \operatorname{Re}(\hat{I}_1(f_x, f_y, f_t) \overline{\hat{I}_2(f_x, f_y, f_t)}))^2 \} \\
= & E \{ | \hat{I}_1(f_x, f_y, f_t) |^4 \} + E \{ | \hat{I}_2(f_x, f_y, f_t) |^4 \} \\
& + E \{ 4 (\operatorname{Re}(\hat{I}_1(f_x, f_y, f_t) \overline{\hat{I}_2(f_x, f_y, f_t)}))^2 \} \\
& + E \{ 2 | \hat{I}_1(f_x, f_y, f_t) |^2 | \hat{I}_2(f_x, f_y, f_t) |^2 \} \\
& + E \{ 2 \operatorname{Re}(\hat{I}_1(f_x, f_y, f_t) \overline{\hat{I}_2(f_x, f_y, f_t)}) (| \hat{I}_1(f_x, f_y, f_t) |^2 + | \hat{I}_2(f_x, f_y, f_t) |^2) \} .
\end{aligned}$$

The first four terms in the summation are all positive, whereas the last term vanishes, since $\operatorname{Re}(\hat{I}_1(f_x, f_y, f_t) \overline{\hat{I}_2(f_x, f_y, f_t)})$ is equally likely to be positive as negative.

In particular, we observe the desired interaction of the two motion planes along the bowtie axis. There, the third and fourth terms in the summation are positive since both $\hat{I}_1(f_x, f_y, f_t)$ and $\hat{I}_2(f_x, f_y, f_t)$ are simultaneously non-zero and each of the terms involves squared (and hence positive) quantities only. These terms add a large extra weight to the minimization, which causes the first eigenvector e_1 of $\mathbf{F}^T \mathbf{W}^2 \mathbf{F}$ to align with the bowtie axis.

3.3.3 Results

To evaluate our method we rendered several synthetic image sequences of scenes containing lambertian spheres (see Fig. 3.3). Image motion was generated by moving a camera (45° field of view) through the scene with various translation and rotation parameters. For *lateral* camera motion (axis of translation and axis of rotation both perpendicular to the optical axis) and a small field of view, the motion satisfies Eq. (1) and a bowtie should be present.

Table 3.1 shows the results for the PCA method and the wedge detector. Except

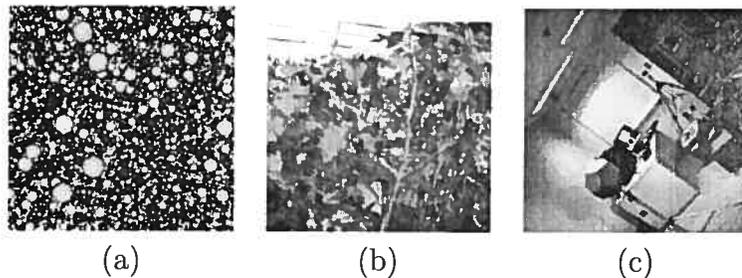


FIG. 3.3 – (a) Synthetic scenes consisted of small balls at a wide range of depths (64 frames of size 128x128 pixels). (b) Holly bush taken from a horizontally translating camera (32 frames of size 64x64 pixels). (c) Lab sequence taken from a camera translating diagonally in the image plane (40 frames of 128x128 pixels).

for the real sequences in the last two rows, each row shows the average over ten runs of each method. We report both the error and the standard deviation σ of estimated bowtie parameters θ and $U = \sqrt{u_x^2 + u_y^2}$. We use both synthetic and real scenes to compare the accuracy of the PCA method to the wedge-based method of [6, 7, 9]. The results are very similar. Temporal shearing (see Sec. 3.2.2) is used as a preprocessing step of the PCA method to account for temporal aliasing. We also perform a power normalization^[12] for each spatial frequency (f_x, f_y) to ensure uniform power distribution. Note that the PCA method is parameter-free. The wedge method needs two parameter, the slope of the wedge v_{max} and the cylinder radius r (see Sec. 3.2.2). For all sequences, v_{max} was set to 1.0, except for the holly sequence, where 2.0 was used. A radius r of 4 pixels was used in all cases. These values are somewhat image dependent, which emphasizes the advantage of getting rid of parameters.

The average running times for the wedge method was 30 seconds¹. Under similar conditions, the PCA method runs in around 1.5 seconds.

3.3.4 Comparing the eigenvalues

Although scenes with a wide range of depths often produce optical snow, there are certain conditions in which optical snow does not occur. The first example

¹The machine was an AMD Athlon 1600MHz.

TAB. 3.1 – Comparison between the PCA method and the Wedge method

Ground truth		PCA method		Wedge method	
θ	$ U $	$err(\theta)$ ($\sigma(\theta)$)	$err(U)$ ($\sigma(U)$)	$err(\theta)$ ($\sigma(\theta)$)	$err(U)$ ($\sigma(U)$)
90	0	0.24 (1.51)	0.00 (0.00)	1.16 (4.17)	0.03 (0.02)
90	0	0.58 (0.88)	0.01 (0.00)	0.59 (3.63)	0.03 (0.05)
0	0	0.33 (1.18)	0.00 (0.00)	0.14 (5.00)	0.03 (0.03)
135	0	0.08 (1.12)	0.00 (0.00)	0.35 (1.85)	0.02 (0.02)
60	0	1.48 (1.30)	0.01 (0.01)	3.92 (4.13)	0.07 (0.04)
225	0.71	0.93 (1.73)	0.02 (0.01)	0.88 (1.91)	0.02 (0.01)
90	1	2.45 (1.16)	0.05 (0.02)	3.13 (5.09)	0.06 (0.04)
135	1.41	1.11 (2.83)	0.04 (0.01)	0.04 (2.53)	0.04 (0.03)
Real Sequence		θ	$ U $	θ	$ U $
Holly		92.78	0.01	88.95	0.16
Lab		28.58	0.02	26.01	0.05

is if the objects are all distant (a mountain seen by a walking observer) then the translation component of the image velocities vanishes, since the translation component depends on inverse depth [2]. Second, even if the objects are nearby, if the observer is moving directly forward and the field of view is small, then the translation component of image motion is small. The reason is that the translation component varies directly with the angular distance from the direction of heading. Third, if the camera motion is pure rotation (no translation), then the image motion field is independent of depth. In all three of these cases, the “bowtie” in fact consists only of a single motion plane.

When only a single motion plane is present, it is the best fitting plane to the “bowtie”. This plane π is spanned by the eigenvectors e_1 and e_2 . The eigenvalues would depend on the spatial orientation structure in the (translating) image, rather than on a bowtie axis. Moreover, the eigenvalue for the e_3 direction would be near zero.

With this issue in mind, we would like to estimate how well the analyzed motion corresponds to optical snow, i.e. if a bowtie is present. We define a bowtie “fitness measure” by comparing the two largest eigenvalues. There are two extreme cases to consider. The first is that optical snow is present and there is a wide range of image speeds. In this case, the first two eigenvalues should correspond to the bowtie axis and a perpendicular vector in the best fitting plane π , respectively.

The second extreme case is that a single motion plane is present. Assuming that spatial content of the image is roughly evenly distributed over all orientations, the two corresponding eigenvalues should be equal.

Figure 3.4 shows a plot of the ratio of the first two eigenvalues λ_2 and λ_1 as a function of the range of depths. As predicted, the ratio falls off as the range of depths decreases.

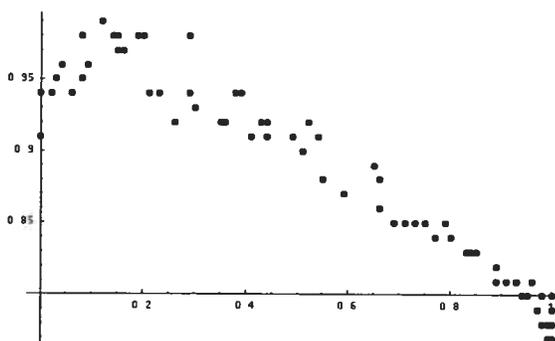


FIG. 3.4 – $\frac{\lambda_2}{\lambda_1}$ as a function of depth range (from a single depth to a large interval). As expected, it starts at 1 (pure plane) and decreases as a bowtie signature takes shape.

3.4 Conclusion

We presented a new simple PCA-based method to analyze optical snow. The performance of the method in estimating the bowtie axis is similar to that of the previous method, which is based on a wedge filter. As the results demonstrated, the PCA method has many advantages such as its simplicity, its efficiency and its absence of parameters.

CHAPITRE 4

ESTIMATION DE LA TRAJECTOIRE 3D DE LA CAMÉRA

Ce chapitre traite du problème de l'estimation du mouvement 3D de caméra pour des scènes encombrées. Le mouvement de caméra est régi de façon générale par six paramètres, soit les trois composantes translationnelles t_x, t_y, t_z et les trois composantes rotationnelles $\omega_x, \omega_y, \omega_z$. Pour une scène rigide, le mouvement perçu par la caméra dépend de sa trajectoire et des profondeurs dans la scène. Le modèle de neige optique, décrit au chapitre 2, est en mesure d'analyser des scènes encombrées caractérisées par une multitude de petits objets à différentes profondeurs. Comme décrit précédemment, les méthodes standards de flux optique et de suivi de points saillants (tracking) sont inadéquates pour de telles scènes. Nous avons présenté, au chapitre 3, une nouvelle méthode d'analyse des composantes principales du modèle de la neige optique.

Pour estimer le mouvement de caméra 3D dans un cadre général à l'aide de ce modèle, deux approches sont proposées. Premièrement, l'approche locale se base sur le fait que plus une région est petite et plus le mouvement, au départ 3D, devient approximativement latéral et 2D. Pour une région i , l'équation 2.3 de la neige optique devient :

$$(v_{i_x}, v_{i_y}) = (\omega_{i_x} + \alpha t_{i_x}, \omega_{i_y} + \alpha t_{i_y}) \quad (4.1)$$

où $\omega_{i_x}, \omega_{i_y}$ est l'approximation locale de la rotation de caméra et t_{i_x}, t_{i_y} , l'approximation locale de la translation. À l'opposé, le modèle de neige optique nécessite une région assez grande pour contenir suffisamment de profondeurs. De plus, comme nous ne savons pas à l'avance le rapport entre la grosseur des objets et celle d'un pixel (problème d'échelle), la taille des régions est sujette à discussion. Nous présentons à la section 4.1 l'algorithme développé par Mann et Langer ^[6] basé sur

cette approche.

Deuxièmement, l'approche globale divise les paramètres du mouvement de caméra en deux ensembles A et B . L'ensemble A comprend les paramètres $t_x, t_y, \omega_x, \omega_y$ et l'ensemble B , les paramètres t_z, ω_z . Pour l'analyse de l'ensemble A , nous supposons donc qu'il n'y a pas de mouvement vers l'avant ($t_z = 0$) et pas de rotation par rapport à l'axe optique ($\omega_z = 0$). La translation de la caméra génère une vitesse (t_x, t_y) inversement proportionnelle à la profondeur et, en supposant un champ de vue restreint ($\pm 20^\circ$), la rotation génère approximativement une vitesse linéaire $(-\omega_y, \omega_x)$ ^[9]. L'équation 2.3 correspond alors à :

$$(v_x, v_y) = (-\omega_y + \alpha t_x, \omega_x + \alpha t_y) \quad (4.2)$$

Une rectification est ensuite appliquée à la séquence vidéo pour retrouver les paramètres de l'ensemble B , t_z et ω_z , les paramètres de l'ensemble A étant supposés nuls. Cette approche globale nécessite un moyen de détecter de façon robuste la présence d'un noeud papillon, mais ne requiert aucune décomposition en régions. Nous la présentons au chapitre 5. Il est à noter qu'une méthode hiérarchique pourrait éventuellement unifier ces deux approches.

4.1 Méthode par décomposition de la vidéo en pièces

Récemment, Mann et Langer ^[6] ont estimé la trajectoire 3D de la caméra en subdivisant la séquence en régions. Ils ont tout d'abord observé la continuité du champ translationnel (sauf aux environs du FOE) et du champ rotationnel à partir de l'équation des vitesses instantanées de Longuet-Higgins^[2] :

$$\begin{aligned} \begin{bmatrix} v_x \\ v_y \end{bmatrix} &= \frac{T_z}{Z(x,y)} \begin{bmatrix} x - x_T \\ y - y_T \end{bmatrix} \\ &+ \frac{T_z}{Z(x,y)} \begin{bmatrix} xy/f & -f - x^2/f & y \\ f + y^2/f & -xy/f & -x \end{bmatrix} \frac{T_z}{Z(x,y)} \begin{bmatrix} \Omega_x \\ \Omega_y \\ \Omega_z \end{bmatrix} \end{aligned}$$

Ces deux champs sont donc localement constants et les vitesses reposent sur une ligne dans l'espace des vitesses (appelée la ligne de la parallaxe de mouvement). Cette équation est modélisée suivant l'équation 4.1.

La séquence est divisée en régions *suffisamment* petites pour que le mouvement devienne approximativement latéral et 2D. Pour une séquence d'images de 256×256 pixels, des régions de 64×64 pixels ont été utilisées. Pour chaque région i , l'axe du noeud papillon donne la direction locale de la translation \vec{t}_i ainsi que la composante rotationnelle normale à la translation $\vec{\omega}_i$. Mann et Langer ^[6] estiment la translation de la caméra T en minimisant l'équation :

$$\arg \min_{\mathbf{T}} \sum_i \|\text{unit}(\vec{t}_i) \times p_i\|^2.$$

avec $|\mathbf{T}| = 1$ et où p_i est le centre de la région i et \vec{t}_i sa direction translationnelle.

La rotation 3D Ω est résolue en utilisant les composantes normales de la rotation $\vec{\omega}_i$ et en minimisant l'équation :

$$\arg \min_{\Omega} \sum_i (\|\vec{\omega}_i\| - \text{unit}(\vec{\omega}_i) \cdot \mathbf{B}_i \Omega)^2$$

où \mathbf{B}_i est la matrice de rotation de la région i , c'est-à-dire :

$$\begin{bmatrix} xy/f & -f - x^2/f & y \\ f + y^2/f & -xy/f & -x \end{bmatrix}$$

Langer et Mann ont donc établi une méthode pour estimer le mouvement 3D complet de la caméra. Cette méthode nécessite une décomposition de la séquence en régions. La taille des régions reste toutefois mal définie puisqu'elle doit être suffisamment petite pour satisfaire l'hypothèse du mouvement latéral, mais suffisamment grande pour contenir assez de profondeurs.

Nous présentons, dans l'article qui suit, une approche permettant l'analyse de la neige optique (c'est-à-dire de scènes encombrées) sans décomposition en régions. Cette approche introduit une rectification de la séquence et se veut un moyen rapide d'effectuer une première estimation du mouvement de caméra.

CHAPITRE 5

(ARTICLE) A GLOBAL ANALYSIS OF OPTICAL SNOW FOR ARBITRARY CAMERA MOTIONS

Cet article ^[64] a été publié comme l'indique la référence bibliographique

V. Chapdelaine-Couture et Sébastien Roy. "A Global Analysis of Optical Snow for Arbitrary Camera Motions", dans *Irish Machine Vision and Image Processing Conference 2004 (IMVIP 2004)*, Dublin (Irlande), septembre 2004, pages 210-215.

L'article est présenté ici dans sa version originale.

Optical snow, introduced in ^[7], is a new category of motion for highly cluttered scenes in which no spatial continuity can be assumed. Since no smoothness constraint can be imposed on the velocity field, traditional optical flow methods can no longer be used ^[3]. However, a model of optical snow has been proposed in ^[9] and algorithms based on this model were suggested using an analysis in the spatio-temporal frequency domain ^[9,63]. This model assumes lateral motion and can be used to solve the 3D camera motion problem by decomposing sequences in sufficiently small patches ^[6]. We would like to use the same model to find arbitrary camera motions globally instead of using patches. In the present paper, we introduce a complementary model for purely non-lateral optical snow. The standard optical snow model and this complementary form could lead to a new global approach for solving the general egomotion problem. We show how non-lateral optical snow sequences can be rectified such that standard methods to analyze optical snow can be applied. The effectiveness of the method is shown for both real and synthetic sequences.

5.1 Introduction

Most studies assume a unique velocity at each point in the visual field [3]. This assumption is only valid if the depth map is continuous. If an observer moves in a 3D highly cluttered scene, a forest for instance, this assumption no longer holds; branches and leaves at many depths cause discontinuities in the motion field. Such cases can be solved by a human observer [43]. However, these scenes are hard to solve since feature points cannot be tracked and since traditional optical flow methods cannot be expected to recover an image velocity.

Recently, Langer and Mann [7] introduced a new category of movement called optical snow which generalizes optical flow by abandoning assumptions of spatial continuity. A model to analyse optical snow induced by all one-parameter set of velocities has been proposed in [9]. Taking the whole sequence in consideration, this model allows for *lateral observer motions* only [9]. Since the image velocity field can locally be approximated as a sum of two fields, a parallel field due to camera translation and a constant field due to camera rotation [6,21], it was showed in [6] that this model could be used to recover 3D egomotion by subdividing the image sequence in sufficiently small patches.

We suggest a more global approach without the use of patches. In this paper, we analyze purely *non-lateral observer motions* and present a method to rectify these sequences such that standard optical snow methods can be applied. Finally, we give some experimental results.

5.2 Previous works

5.2.1 Optical Snow

The model of optical snow defined in [7,9] is an extension of the motion plane property [5] which states that an image pattern translating with uniform image velocity produces a plane of energy in the frequency domain. Formally, let $I(x, y, t)$ be a time varying image. If an image patch is translating by (v_x, v_y) , we know from

[45] that this velocity is constrained by

$$v_x \frac{\partial I}{\partial x} + v_y \frac{\partial I}{\partial y} + \frac{\partial I}{\partial t} = 0 \quad (5.1)$$

This constraint, transposed in the Fourier domain, yields the following equation :

$$-2\pi(v_x f_x + v_y f_y + f_t) \hat{I}(f_x, f_y, f_t) = 0 \quad (5.2)$$

where $\hat{I}(f_x, f_y, f_t)$ is the Fourier transform of $I(x, y, t)$. As noted in [9], equation 5.2 implies that all frequencies $\hat{I}(x, y, t) \neq 0$ lie on the plane

$$v_x f_x + v_y f_y + f_t = 0. \quad (5.3)$$

This model was extended in [7] for the case in which there is a one-parameter set of velocities within an image region, i.e. where velocities vary according the following equation :

$$(v_x, v_y) = (u_x + \alpha t_x, u_y + \alpha t_y) \quad (5.4)$$

where u_x, u_y, t_x, t_y are constants and α depends on the visible depth at point (x, y) [9].

Substituting Eq. 5.4 into Eq. 5.3 yield a family of planes in the frequency domain,

$$(u_x + \alpha t_x) f_x + (u_y + \alpha t_y) f_y + f_t = 0. \quad (5.5)$$

Thus, this model produces a set of planes in the frequency domain. This set of planes forms a *bowtie* (see Fig. 5.1) and follows the two following propositions :

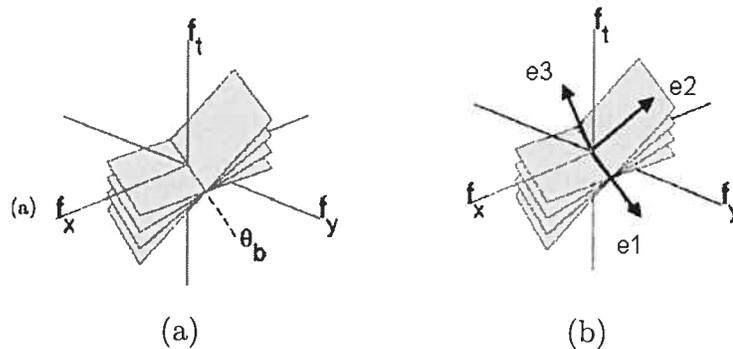


FIG. 5.1 – (a) Bowtie signature in the frequency domain (b) Eigenbasis of bowtie signature

Proposition 1 : The planes of the bowtie intersect at a common line, called the *axis of the bowtie*, that passes through the origin.

Proposition 2 : The axis of the bowtie is in direction $(-t_y, t_x, u_x t_y - u_y t_x)^1$. By normalizing (t_x, t_y) , this direction becomes $(-t_y, t_x, |U| \sin(\phi))$, where ϕ is the angle between vectors (t_x, t_y) and (u_x, u_y) .

Applied to the egomotion problem, it was noted in ^[9] that Eq. (5.4) corresponds to

$$(v_x, v_y) = (-\omega_y + \alpha t_x, \omega_x + \alpha t_y) \quad (5.6)$$

In other words, camera rotation generates a constant velocity component $(-\omega_y, \omega_x)$ for small field of views ($\pm 20^\circ$) and lateral translation (t_x, t_y) generates a velocity inversely proportional to depth. Components ω_z and t_z were assumed to be 0. Therefore, this model cannot recover arbitrary camera motions.

¹For simplicity, all equations in this paper assume image sequences of equal dimension in space and time.

The main contribution of this paper is to show that complementary camera motions, i.e. following components ω_z and t_z (all other components assumed to be 0), produce non-lateral optical snow sequences that can be rectified to be analyzed by existing optical snow methods. We will describe how to find components ω_z and t_z .

5.3 Motion Field

The motion field equation contains 7 variables which are the depth P_z at each pixel, the translation vector (t_x, t_y, t_z) and the rotation vector $(\omega_x, \omega_y, \omega_z)$.

More precisely, the velocity field for a pixel (x,y) , as defined in [2], is :

$$(v_x, v_y)^T = \begin{pmatrix} p_x p_y \omega_x - (1 + p_x^2) \omega_y + p_y \omega_z + \frac{p_x t_z}{P_z} - \frac{t_x}{P_z} \\ (1 + p_y^2) \omega_x - p_x p_y \omega_y - p_x \omega_z + \frac{p_y t_z}{P_z} - \frac{t_y}{P_z} \end{pmatrix}$$

By assuming that only P_z , t_z and ω_z are non-zero, we are left with :

$$(v_x, v_y)^T = \begin{pmatrix} p_y \omega_z + \frac{p_x t_z}{P_z} \\ -p_x \omega_z + \frac{p_y t_z}{P_z} \end{pmatrix}$$

Note that rotation component ω_z is perpendicular to translation component t_z for each pixel (x,y) , and that ω_z is independent of depth. From this observation, rectification is performed on image sequences to obtain optical snow motion $(0, \omega_z) + \alpha(t_z, 0)$.

5.3.1 Motion Field Rectification

The rectification is a polar transformation around the FOE [65] which is the image center in our case. First, we rectify the motion field induced by pure ω_z rotation. Consider a point p located in the image at $(1,0)$ on the camera projection plane (see Fig. 5.2-a). The path followed through time by p for a pure ω_z movement is a circle of radius 1, and its speed is exactly equal to $|\omega_z|$. The rectification is done by “unfolding” images such that this circle becomes straight and vertical, as shown

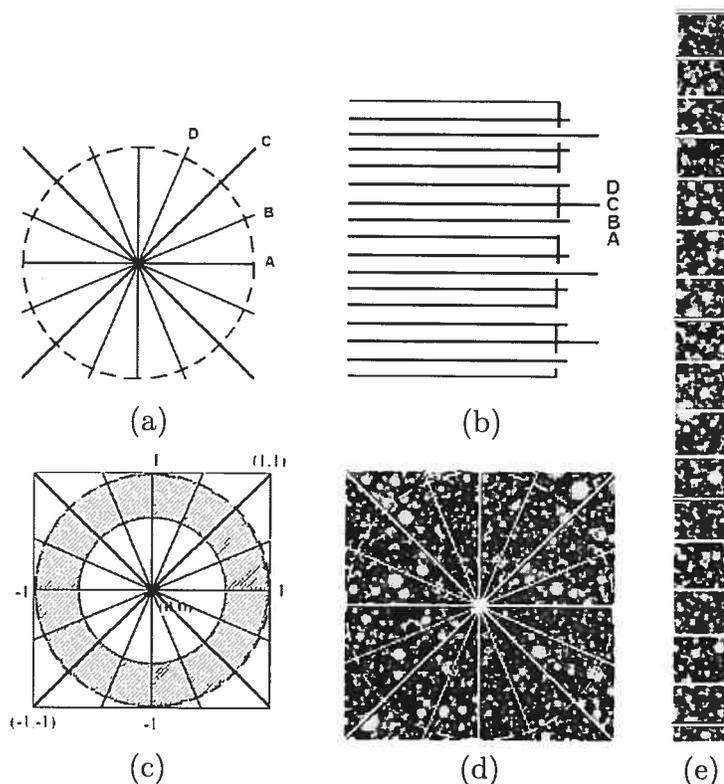


FIG. 5.2 – (a) Original motion field (b) Rectified motion field (c) Rectified region of the original sequence (d) Original frame (e) Rectified frame

in Fig. 5.2-b. For a field of view of 90° and images of size $N \times N$, the vertical length of the rectified image is $2\pi\frac{N}{2} = \pi N$ pixels. Note that flow lines of a pure forward motion become horizontal. The velocities in the rectified motion field correspond to $(v'_x, v'_y)^T = (\alpha t_z \sqrt{p_x^2 + p_y^2}, \omega_z)^T$.

Standard optical flow produces velocities that only depend on depth. However, the velocities in rectified non-lateral motion sequences vary according to depth and to image position. Therefore, horizontal lines in the rectified sequence must be resampled according to factor $\frac{1}{\sqrt{p_x^2 + p_y^2}}$, where (p_x, p_y) is the position of a pixel on the camera projection plane in the original sequence. In theory, we get infinite sampling at center $(0,0)$. In practice, we do not rectify near the image center as illustrated in Fig. 5.2-c. Notice that image corners are cut to remove empty spaces in the rectified sequence.

5.3.2 Finding rotation ω_z

The bowtie axis can be computed from the rectified sequence using Principal Components Analysis as described in [63]. For standard optical snow, the angle ϕ in the bowtie axis equation (see Proposition 2) is unknown. For non-lateral optical snow, however, we know that $\phi = 90^\circ$. Hence, the bowtie axis equation becomes $(0, 1, \omega_z)$. Thus, the third component of the bowtie axis gives us ω_z directly with speed given in pixels/frame in the rectified sequence. The rotation given in degrees, for a field of view of 90° , is then $360 \frac{\omega_z}{\pi N}$.

The bowtie axis can also be found using the best fit plane [63]. Let (n_x, n_y, n_z) be the normal of the best fit plane π . Since t_z is the only component affected by depth, the bowtie axis is the line on the best fit plane in direction $(0, 1, -\frac{n_y}{n_z})$.

5.3.3 Finding translation t_z

From Eq. 5.5, and since t_z generates only horizontal velocities and ω_z only vertical velocities, planes forming the bowtie have equation $(\alpha t_z, \omega_z, 1)$. The best fit plane π is defined as $(\frac{n_x}{n_z}, \frac{n_y}{n_z}, 1) = (\bar{\alpha} t_z, \omega_z, 1)$, where $\bar{\alpha}$ is the weighted ‘‘average’’ of the slopes of the motion planes that compose the bowtie. The weights depend on depth distribution in the scene as well as image contrast contributed by each object. Since this information is unknown, we can only compute t_z up to a scale factor $\bar{\alpha}$.

5.4 Experimental results

To evaluate our method, we rendered several synthetic image sequences of scenes containing lambertian spheres (see Fig. 5.3-a). Image motion was generated by moving a camera (90° field of view) through the scene with various t_z and ω_z parameters.

Table 5.1 shows results for various rectified sequences. The last two rows correspond to real image sequences, respectively the lab sequence (see Fig. 5.3-b) and the plants sequence (see Fig. 5.3-c). The ω_z component is found almost exactly. The

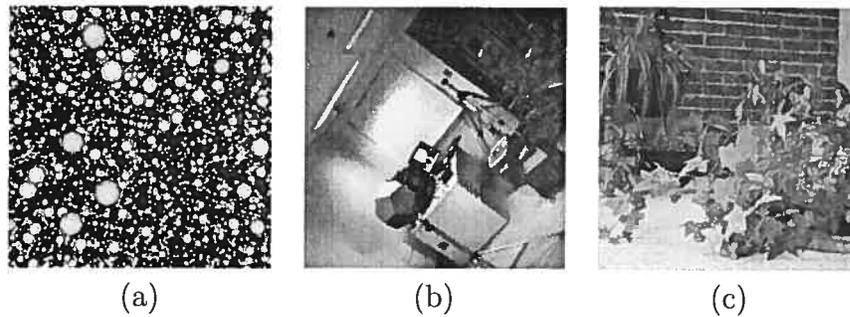


FIG. 5.3 – (a) Synthetic scenes were constituted of small balls at different depths (64 frames of size 128x128 pixels). (b) Lab sequence taken from a camera rotating around the z-axis (40 frames of 128x128 pixels). (c) Plants sequence taken from a camera making a forward motion (32 frames of 128x128 pixels).

running time is about 1.6 seconds for 128x128x32 sequences on a 1.3GHz AMD Athlon machine.

TAB. 5.1 – Results for synthetic and real scenes

True Rotation (degrees/frame)	True Translation (pixel/frame)	Rotation Found	Translation Found (up to a scale factor)
0.00	1.00	0.00	0.15
1.80	0.00	1.78	0.00
1.80	1.00	1.81	0.22
≈ 0.50	≈ 0.00	0.35	-0.05
≈ 0.00	≈ 0.80	0.00	0.31

5.5 Comparing the eigenvalues

When analyzing image sequences globally, we would like to estimate how well the motion fits the optical snow model, i.e. if a bowtie is present. For instance, an unrectified forward motion or a rectified lateral motion do not produce a bowtie signature. For such cases, the motion field features velocities oriented in all directions. Detecting these situations would be a great benefit.

In ^[63], depth range is evaluated by comparing the two largest eigenvalues of the Principal Components Analysis method. We can adapt this measure to detect the presence of a bowtie. The ratio of the first two eigenvalues λ_2 and λ_1 has a

maximum of 1 and should decrease as we get closer to a bowtie signature. In fact, the presence of a bowtie creates a high power concentration along its axis which increases the first eigenvalue. The absence of a bowtie is characterized by a uniform power distribution which makes λ_1 and λ_2 almost equal. Fig. 3.4-a shows a plot of $\frac{\lambda_2}{\lambda_1}$ as a function of $\frac{|t_z|}{|T|}$ for rectified sequences. As expected, the ratio falls off as the bowtie takes shape in the frequency domain. Fig. 3.4-b shows $\frac{\lambda_2}{\lambda_1}$ as a function of $\frac{|t_z|}{|T|}$ for non-rectified sequences. As expected, it increases linearly up to 1 as bowtie signature disappears. The sequences used for these graphs have random translation vectors of unit length.

Notice that the curve in Fig. 3.4-a decreases non-linearly while the other one is linear. This might be caused by the subsampling during rectification which accentuates the effect of any lateral motion. This is still under investigation. These curves, if modelled correctly, would allow merging non-lateral and lateral motions analysis into a general egomotion estimation algorithm.

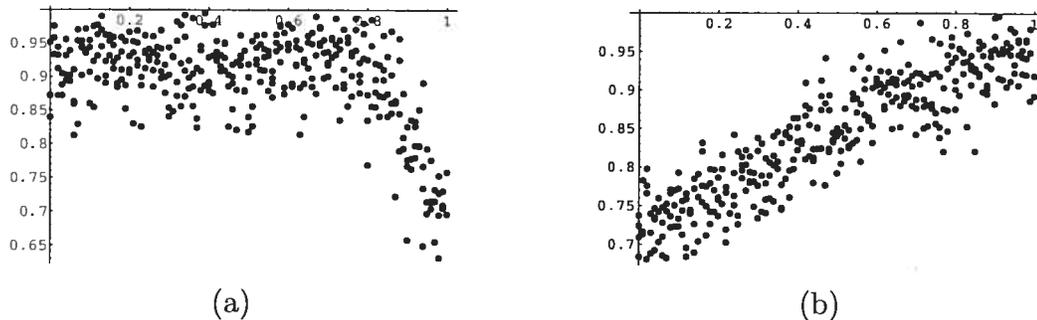


FIG. 5.4 – (a) $\frac{\lambda_2}{\lambda_1}$ as a function of $\frac{|t_z|}{|T|}$ for rectified sequences. It starts at 1 and decreases as a bowtie signature takes shape. (b) $\frac{\lambda_2}{\lambda_1}$ as a function of $\frac{|t_z|}{|T|}$ for non-rectified sequences. It increases linearly up to 1 as bowtie signature disappears.

5.6 Conclusion

This paper presented a method to analyze purely non-lateral optical snow by introducing a rectification process. Results show its accuracy and efficiency. We hope to solve the general egomotion problem in a global approach using this scheme.

CHAPITRE 6

DISCUSSION

Une nouvelle méthode PCA d'analyse du modèle de neige optique est présentée au chapitre 3 (voir le code à l'annexe II). Cette méthode trouve une ligne d'énergie dans la transformée de Fourier 3D de la séquence. De plus, une rectification de la séquence vidéo, introduite au chapitre 5, permet l'estimation du mouvement de caméra vers l'avant (t_z) et de la rotation par rapport à l'axe optique (ω_z). Par contre, les analyses de la séquence rectifiée et non rectifiée sont interdépendantes. Un modèle robuste permettant de mesurer la contribution des composantes ciblées dans la séquence originale ($t_x, t_y, \omega_x, \omega_y$) et celles ciblées dans la séquence rectifiée (t_z, ω_z) reste à établir. Plusieurs autres observations sont aussi nécessaires, dont certaines peuvent servir de pistes de recherche.

6.1 Algorithme biologiquement plausible pour l'analyse de la neige optique

L'algorithme PCA recherche en fait une ligne dans le domaine fréquentiel. Plusieurs méthodes ont déjà été proposées pour trouver de façon biologiquement plausible un plan dans le domaine fréquentiel à l'aide de filtres de Gabor [50-52]. Il serait intéressant d'adapter ces algorithmes pour trouver non pas un plan, mais une ligne.

6.2 Distribution d'énergie uniforme

La méthode du prisme à base triangulaire de Langer et Mann [9] et la méthode PCA présentée au chapitre 3 sont complémentaires, en ce sens que la première mesure l'absence d'énergie et la deuxième, la présence d'énergie. Cette observation peut laisser croire que la méthode du prisme est plus robuste pour des scènes où les objets et le contraste dans l'image sont non uniformes.

En effet, la méthode PCA suppose une distribution uniforme de l'énergie dans chaque plan du noeud papillon. En pratique, cette distribution est rarement uniforme. En effet, la texture des objets affecte grandement la distribution spatiale d'énergie. Par exemple, le contraste d'un objet blanc sur fond noir produit de plus grandes puissances dans le domaine fréquentiel qu'un objet gris. Une séquence peut aussi comporter beaucoup plus de hautes fréquences que de basses fréquences. Un zèbre plus rayé qu'un autre produit des puissances de plus hautes fréquences¹. Cependant, nous utilisons une normalisation telle que décrite par Heeger^[12] pour uniformiser la distribution d'énergie (voir la Fig. 6.1). Cette normalisation consiste à diviser la puissance d'une fréquence (f_x, f_y, f_t) par la somme des puissances ayant la même fréquence spatiale (f_x, f_y) et compense pour les différences de contraste et les différences de fréquences inhérentes aux textures. Cette normalisation correspond à l'équation :

$$\hat{I}^*(f_{x_i}, f_{y_j}, f_{t_k}) = \frac{\hat{I}(f_{x_i}, f_{y_j}, f_{t_k})}{\sum_{l \in f_t} \hat{I}(f_{x_i}, f_{y_j}, f_{t_l})}$$

Ce processus de normalisation assure aussi une plus grande puissance sur l'axe du noeud papillon. En effet, la géométrie de l'axe du noeud papillon fait en sorte que la puissance sur l'axe devient 1 et décroît à mesure que l'on s'en éloigne. On peut voir à la figure 6.1-b que le surplus d'énergie sur l'axe est conservé.

La figure 6.2 montre l'importance de la normalisation d'énergie sur un mouvement de caméra vers l'avant. La séquence utilisée est composée de plusieurs carrés, chacun ayant une orientation différente de gradient de texture (voir 6.3).

6.3 Une approche hiérarchique

Plus il y a présence de hautes vitesses, plus l'aliasage temporel peut mener à des imprécisions en dépit du cisaillement temporel itératif tel que décrit à la section 2.1.2 (voir Fig. 2.3). Lorsque la taille de l'image N est beaucoup plus grande

¹Il est à noter que les scènes réelles tendent à avoir plus de basses fréquences.

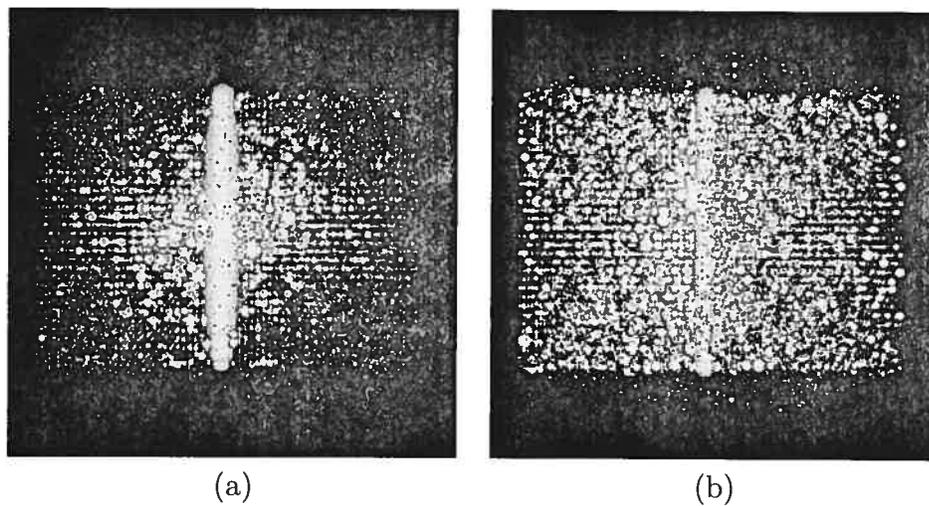


FIG. 6.1 – Noeud papillon (de la séquence Holly ^[9]) vu de l'axe f_t (a) sans normalisation (b) avec normalisation.

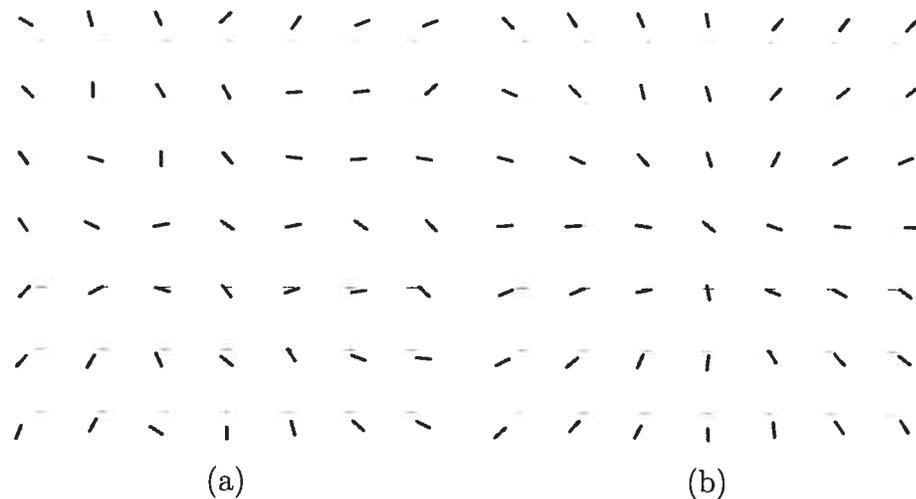


FIG. 6.2 – (a) Exemple de flux (mouvement vers l'avant) (a) sans normalisation (b) avec normalisation.

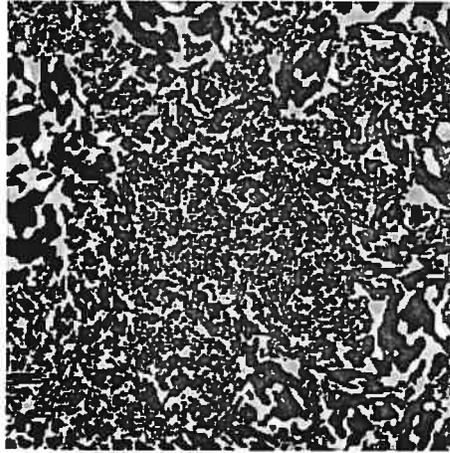


FIG. 6.3 – Séquence créée par Richard Mann, professeur à l’université de Waterloo. La scène est composée de plusieurs carrés texturés à plusieurs profondeurs.

que le nombre d’images T (de l’ordre de 4, 8 fois ou plus), la pente des plans devient plus faible et les hautes vitesses causent ainsi moins d’imprécision. Pour une décomposition en régions de petite taille, telle que décrite par Langer et Mann [9], nous proposons une approche hiérarchique qui consiste à utiliser comme plan de départ du cisaillement temporel itératif le plan trouvé lors de l’analyse d’une région plus grande. À la figure 6.4-a, on peut voir que la rotation vers la droite génère de trop grandes vitesses (> 8 pixels par image) dans plusieurs régions, lesquelles sont mal évaluées (estimation > 100 pixels par image) à cause de l’aliassage temporel. À la figure 6.4-b, on peut voir l’amélioration apportée par ce processus hiérarchique.

Il est à noter que plusieurs méthodes suggèrent déjà un processus hiérarchique [10–13], c’est-à-dire une décomposition de la séquence en régions plus ou moins grandes selon le niveau de la hiérarchie. Cependant, plus la région devient grande et plus l’hypothèse du mouvement latéral définie au chapitre 4 ne tient plus. Une approche globale, comme celle décrite au chapitre 5, évite un tel problème.

6.4 Analyse dans le domaine de Fourier des séquences rectifiées

Jusqu’à présent, l’analyse de la neige optique se fait dans le domaine de Fourier. Puisque la séquence rectifiée a des dimensions inégales (pour une séquence originale

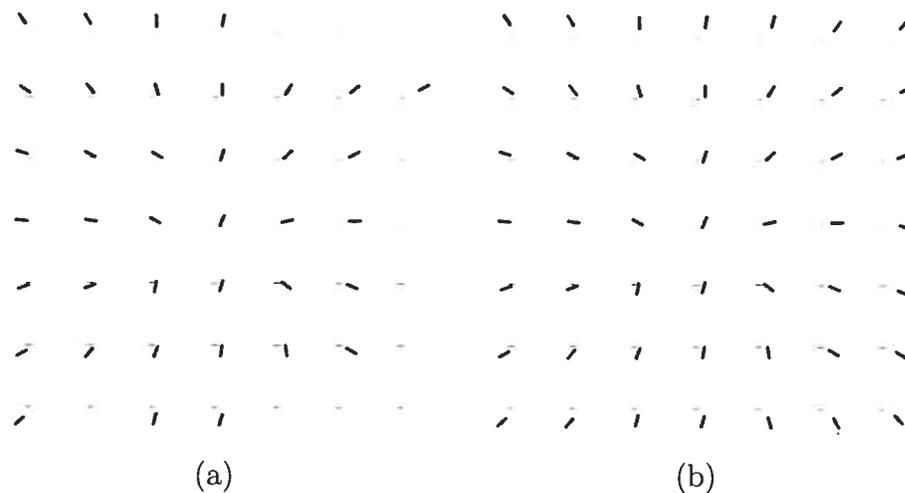


FIG. 6.4 – (a) Flux translationnel par régions d’une séquence comportant de hautes vitesses (a) sans processus hiérarchique. Les pièces sans translation ont été mal évaluées (b) avec processus hiérarchique.

de $128 \times 128 \times 32$, on obtient une séquence rectifiée de $32 \times 401 \times 32$), celle-ci est décomposée en blocs de $N \times N \times T$ ($32 \times 32 \times 32$ par exemple). Puisque le même noeud papillon doit en principe se retrouver dans chaque bloc, la moyenne des transformées de Fourier de chaque bloc peut être considérée. Pour un mouvement vers l’avant t_z , par exemple, le noeud papillon de chaque bloc correspond à un mouvement vers la gauche. Quant à un mouvement latéral, le noeud papillon de chaque bloc est différent (dû à la rectification), et leur moyenne donne une signature qui ne devrait correspondre à aucun mouvement précis.

Une approche similaire pourrait être utilisée sur des séquences non rectifiées de grande taille pour accélérer l’étape de la transformée de Fourier. Par contre, la décomposition d’une séquence en petits blocs peut augmenter l’aliasage temporel ; il faut donc s’assurer de maintenir le rapport N/T suffisamment grand (> 1 par exemple).

6.5 Mesure robuste de la présence d'un noeud papillon

À la section 5.5, nous avons établi une vraisemblance de la présence d'un noeud papillon. Cette mesure visait à mesurer la contribution des composantes t_z et ω_z dans les séquences rectifiée et non-rectifiée. À la Fig. 5.4-a, on peut voir que la mesure de la présence d'un noeud papillon pour la séquence non rectifiée est linéaire et constitue donc un bon indice de la contribution du mouvement vers l'avant t_z . Cependant, l'intervalle des valeurs est difficile à évaluer et dépend du contraste et du contenu de la scène. Une mesure robuste aurait comme intervalle $[0,1]$, peu importe le contenu de la scène. Une piste de recherche serait d'intégrer la mesure du prisme à base triangulaire (construit par Langer et Mann ^[9] pour épouser la forme du noeud papillon) à la méthode des valeurs propres λ_1 et λ_2 décrite à la section 5.5.

6.6 Distribution d'énergie des profondeurs

Il serait intéressant de déduire la distribution des profondeurs à partir du noeud papillon.

Pour obtenir la distribution d'énergie des plans passant par l'axe du noeud papillon, et ainsi estimer les profondeurs visibles dans la scène, un cylindre de rayon infini ayant comme axe celui du noeud papillon est échantillonné (voir Fig. 6.5). L'énergie est ensuite accumulée pour chaque θ , auquel correspond un plan $[\omega_x + \alpha t_x, \omega_y + \alpha t_y, 1]$ (voir Fig. 6.6).

Les normales des plans sont comprises entre les plans limites $\Omega + \frac{T}{Pz_{max}}$ et $\Omega + \frac{T}{Pz_{min}}$, où Pz_{min} est la profondeur visible la plus proche et Pz_{max} , la profondeur visible la plus éloignée. Il est à noter que si une région ne contient qu'une seule profondeur, le noeud papillon se réduit en un plan et l'histogramme devient un pic d'énergie.

Pour trouver ces deux plans limites, les deux frontières les plus probables sont détectées à partir de l'histogramme d'énergie des plans en utilisant un détecteur de frontières ^[47-49]. Pour s'assurer de trouver les limites correspondant à Pz_{min} et

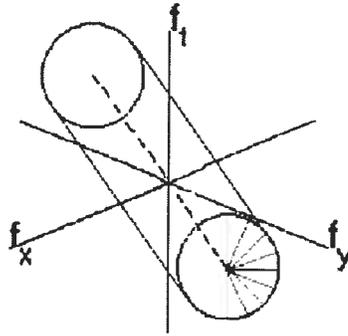


FIG. 6.5 – Estimation de la distribution d'énergie des plans du noeud papillon

Pz_{max} , un filtre Gaussien de variance proportionnelle à la distance maximale entre les profondeurs dans la scène est appliqué à l'histogramme d'énergie.

Les effets de frontières affectent l'histogramme des vitesses. Pour obtenir les vraies pentes des plans, changées lors du cisaillement temporel itératif, on introduit un cisaillement temporel inverse. Soit le plan $\pi = (n_x, n_y, n_t)$, l'accumulation de tous les plans du cisaillement temporel itératif. Pour chaque fréquence, on obtient un décalage en f_t de $d(x, y) = n_x * f_x + n_y * f_y$. À partir du spectre des puissances cisailées, ce décalage peut être rajouté (sans modulo) pour obtenir les pentes réelles des plans du noeud papillon, sans l'effet modulo associé à l'aliassage temporel.

6.7 Interprétation des deux plans limites

Les plans limites d'équations $\Omega + \frac{T}{Pz_{max}}$ et $\Omega + \frac{T}{Pz_{min}}$ présentent cependant l'ambiguïté suivante : lequel des deux plans correspond à la profondeur la plus rapprochée de la caméra ? Une piste de recherche serait de comparer la distribution d'énergie de ces deux plans. En effet, un plan composé d'objets plus rapprochés devrait avoir une proportion plus grande de basses fréquences.

En supposant cette ambiguïté résolue, on obtient les 4 équations suivantes :

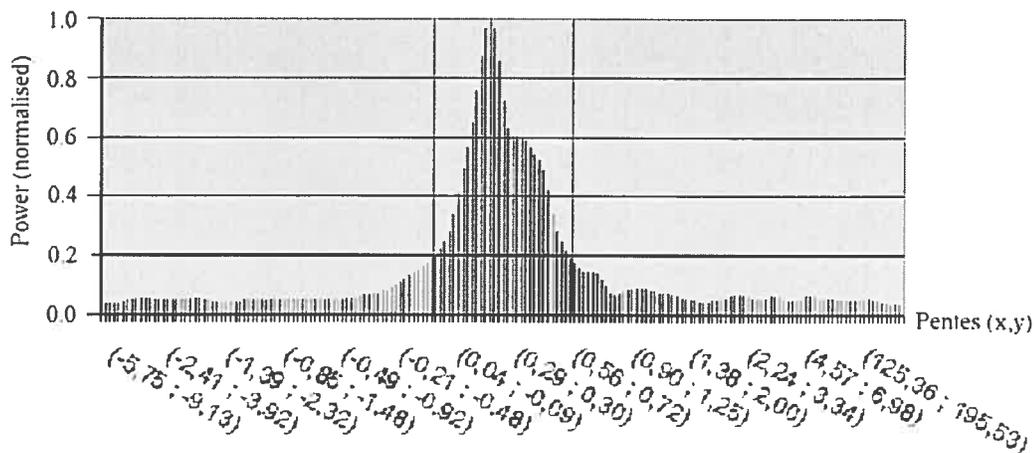


FIG. 6.6 – Histogramme d'énergie des plans du noeud papillon. Les deux lignes verticales représentent les deux plans limites trouvés

$$\begin{cases} \omega_x + \frac{t_x}{P_{z_{max}}} = k_1 \\ \omega_y + \frac{t_y}{P_{z_{max}}} = k_2 \\ \omega_x + \frac{t_x}{P_{z_{min}}} = k_3 \\ \omega_y + \frac{t_y}{P_{z_{min}}} = k_4 \end{cases}$$

En supposant que la profondeur la plus éloignée soit à l'infini ($\frac{1}{P_{z_{max}}} = 0$), la rotation peut être extraite directement. De plus, si l'utilisateur connaît $P_{z_{min}}$ et $P_{z_{max}}$, le mouvement complet exact peut être retrouvé car l'ambiguïté d'échelle entre la translation et la rotation disparaît.

Nous présentons au tableau 6.1 les différentes étapes de l'analyse globale de scènes encombrées, de l'acquisition de la séquence vidéo à l'interprétation des plans limites.

TAB. 6.1 – Étapes de l'analyse globale de scènes encombrées

1.	Soustraction de l'intensité moyenne des pixels
2.	Fenêtrage de la séquence
3.	Transformée de Fourier
4.	Normalisation d'énergie
5.	Cisaillement temporel itératif en utilisant le plan moyen estimé avec la méthode PCA
6.	Estimation de l'axe du noeud papillon avec la méthode PCA
7.	Cisaillement temporel inverse
8.	Histogramme d'énergie des plans
9.	Estimation et interprétation des plans limites

CONCLUSION

Ce mémoire s'est servi du modèle de neige optique qui permet l'analyse du mouvement de caméra pour des scènes encombrées. Les méthodes de flux optique traditionnelles ne sont plus adéquates puisqu'aucune continuité spatiale ne peut être supposée, et celles du suivi de points saillants deviennent difficiles, voire impossibles, à appliquer à cause des occlusions.

Nous avons introduit un nouvel algorithme, rapide et sans paramètre, basé sur l'analyse des composantes principales de ce modèle. Cette approche pourrait mener à l'élaboration d'un algorithme biologiquement plausible d'analyse de scènes encombrées.

Nous avons de plus introduit une approche globale pour retrouver le mouvement 3D de caméra ne requérant ni décomposition en régions, ni suivi de points saillants. L'approche globale, telle que présentée dans ce mémoire, comporte deux analyses, celle de la séquence originale et celle de la séquence rectifiée. Ces deux analyses sont cependant interdépendantes, les composantes ciblées dans la séquence originale $(t_x, t_y, \omega_x, \omega_y)$ affectant les composantes ciblées dans la séquence rectifiée (t_z, ω_z) , et vice-versa. L'introduction de plusieurs autres rectifications serait une piste intéressante afin d'acquérir plus d'information sur le mouvement.

Le modèle de neige optique est complémentaire aux méthodes de flux optique en ce sens que le flux optique s'applique à des zones ayant localement une seule profondeur, tandis que la neige optique s'applique à des zones ayant plusieurs profondeurs. Nous considérons que les concepts de flux optique et de neige optique doivent ultimement être unifiés. Cette unification impliquerait une analyse de la distribution des profondeurs dans la scène, ce qui permettrait de choisir le modèle le plus approprié.

BIBLIOGRAPHIE

- [1] K. Pearson. *On lines and planes of closest fit to systems of points in space*. Philosophical Magazine, 2(11) :559-572, 1901.
- [2] Longuet-Higgins, H.C. et Prazdny, K. *The Interpretation of a Moving Retinal Image*. Proceedings of the Royal Society of London B, B-208 :385-397, 1980.
- [3] Barron, J.L. et Fleet, D.J. and Beauchemin, S.S. *Performance of Optical Flow Techniques*. International Journal of Computer Vision, 12(1) :43-77, février 1994.
- [4] J. J. Gibson. *The perception of the visual world*. Houghton Mifflin, Boston, 1950.
- [5] Watson, A.B. et Ahumada, A.J. *Model of Human Visual-Motion Sensing*. Journal of the Optical Society of America A, 2(2) :322-342, 1985.
- [6] R. Mann et M. S. Langer. *Estimating camera motion through a 3D cluttered scene*. Canadian Conference on Computer and Robot Vision, London, Canada, mai 2004.
- [7] M. S. Langer et R. Mann. *Dimensional analysis of image motion*. IEEE International Conference on Computer Vision, 2001, pp. 155-162.
- [8] M.S. Langer et R. Mann. *Tracking through optical snow*. Lecture Notes in Computer Science 2525 : Biologically Motivated Computer Vision, 2nd International Workshop, Springer, pages 181-188, novembre, 2002.
- [9] M. S. Langer et R. Mann. *Optical Snow*. International Journal of Computer Vision, 55(1) : 55-71, 2003.
- [10] F. C. Glazer. *Computation of optical flow by multilevel relaxation*. Rapport Technique COINS-TR-87-64, Université du Massachusetts, 1987.
- [11] P. Anandan. *A computational framework and an algorithm for the measurement of visual motion*. International Journal of Computer Vision, 2 :283-312, 1989.

- [12] D. J. Heeger. *Optical flow using spatiotemporal filters*. International Journal of Computer Vision, 1(4) :279-302, 1987.
- [13] J. R. Bergen, P. J. Burt, R. Hingorani, and S. Peleg, *Three-frame algorithm for estimating two-component image motion*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 14(9), 1992.
- [14] G. Adiv. *Determining three-dimensional motion and structure from optical flow generated by several moving objects*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 7(4) :384-401, juillet 1985.
- [15] B.K.P Horn et E.J. Weldon. *Direct methods for recovering motion*. International Journal of Computer Vision, 2(1) :51-76, juin 1988.
- [16] R. Deriche et O. Faugeras. *Tracking line segments*. 1st European Conference on Computer Vision, pages 259-268, Antibes, avril 1990.
- [17] D.J. Fleet et A.D. Jepson. *Computation of component image velocity from local phase information*. International Journal of Computer Vision, 5(1) :77-104, 1990.
- [18] D.J. Heeger. *Optical flow using spatiotemporal filters*. International Journal of Computer Vision, 1 :279-302, 1988.
- [19] M. Shizawa et K. Mase. *Simultaneous multiple optical flow estimation*. International Conference on Pattern Recognition, pages 274-278, Atlantic City, New Jersey, juin 1990.
- [20] D.T. Lawton et J.H. Rieger. *The use of difference fields in processing sensor motion*. DARPA IU Workshop, pages 78-83, juin 1983.
- [21] J.H. Rieger et D.T. Lawton. *Processing Differential Image Motion*. Journal of the Optical Society of America A, volume 2, pages 254-260, 1985.
- [22] D.J. Heeger et A. Jepson. *Simple method for computing 3d motion and depth*. International Conference on Computer Vision, pages 96-100, 1990.
- [23] R. Guissin et S. Ullman. *Direct Computation of the focus of expansion from velocity fields measurements*. IEEE Workshop on Visual Motion, pages 146-155, Princeton, NJ, octobre 1991.

- [24] A.D. Jepson et D.J. Heeger. *A fast subspace algorithm for recovering rigid motion*. IEEE Workshop on Visual Motion, pages 124-131, Princeton, NJ, octobre 1991.
- [25] V. Sundareswaran. *Egomotion from global flow field data*. IEEE Workshop on Visual Motion, pages 140-145, Princeton, NJ, octobre 1991.
- [26] S. Negahdaripour et S. Lee. *Motion recovery from image sequences using first-order optical flow information*. IEEE Workshop on Visual Motion, pages 132-139, Princeton, NJ, octobre 1991.
- [27] L. Huang et Y. Aloimonos. *Relative depth from motion using normal flow : an active and purposive solution*. IEEE Workshop on Visual Motion, pages 156-162, Princeton, NJ, octobre 1991.
- [28] M. Tistarelli et G. Sandini. *Direct estimation of time-to-impact from optical flow*. IEEE Workshop on Visual Motion, pages 226-233, Princeton, NJ, octobre 1991.
- [29] Y. Aloimonos et Z. Duric. *Active egomotion estimation : a qualitative approach*. European Conference on Computer Vision, pages 497-510, Santa Margarita Ligure, mai 1992.
- [30] Z. Duric, A. Rosenfeld, et L. Davis. *Egomotion analysis based on the Frenet-Serret motion model*. International Conference on Computer Vision, pages 703-712, Berlin, mai 1993.
- [31] F. Lustman, O.D. Faugeras et G. Toscani. *Motion and structure from motion from point and line matching*. 1st International Conference on Computer Vision, pages 25-34, Londres, 1987.
- [32] B.K.P. Horn. *Relative orientation*. International Journal of Computer Vision, 4(1) : 58-78, juin 1990.
- [33] R. Chipolla, Y. Okamoto et Y. Kuno. *Robust structure from motion using motion parallax*. International Conference on Computer Vision, pages 374-382, Berlin, mai 1993.

- [34] J. Weng, T.S. Huang et N. Ahuja. *Motion and structure from two perspective views : Algorithms, error analysis, and error estimation*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 11(5), 1989.
- [35] K. Hanna. *Direct multi-resolution estimation of egomotion and structure from motion*. IEEE Workshop on Visual Motion, pages 156-162, Princeton, NJ, octobre 1991.
- [36] M.A. Taalebinezhad. *Direct recovery of motion and shape in the general case by fixation*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 14 :847-853, août 1992.
- [37] G. Adiv. *Inherent ambiguities in recovering 3D motion and structure from a noisy flow field*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 11 :477-489, mai 1989.
- [38] K. Daniilidis et H.-H. Nagel. *The coupling of rotation and translation in motion estimation of planar surfaces*. IEEE Conference on Computer Vision and Pattern Recognition, pages 188-193, New York, juin 1993.
- [39] R. Dutta, R. Manmatha, E.M. Riseman et M.A. Snyder. *Issues extracting motion parameters and depth from approximate translational motion*. DARPA IU Workshop, pages 945-960, 1988.
- [40] H.C. Longuet-Higgins et K. Prazdny. *The interpretation of a moving retinal image*. Proceedings of The Royal Society of London B, 208 :385-397, 1980.
- [41] C.H. Lee. *Structure and motion from two perspective views via planar patch*. International Conference on Computer Vision, pages 158-164, 1988.
- [42] J. Lawn et R. Chipolla. *Epipole estimation using affine motion parallax*. Rapport Technique CUED/F-INFENG/TR-138, Cambridge, juillet 1993.
- [43] Warren, W.H. et Hannon, D.J. *Eye movements and optical flow*. Journal of the Optical Society of America A, 7(1) :160-169, 1990.
- [44] Koenderink, J.J. *Optical flow*. Vision Research, 26(1), pages 161-179, 1986.
- [45] Horn, B. et Schunck, B. *Determining optical flow*. Artificial Intelligence, 17 :185-203, 1981.

- [46] Lucas, B.D. and Kanade, T. *Optical Navigation by the Method of Differences*. International Joint Conference on Artificial Intelligence, 1985, pages 981-984.
- [47] W.K. Pratt. *Digital Image Processing*. Wiley, New York, 1991.
- [48] R.C. Gonzalez et R.E. Woods. *Digital Image Processing*. Addison-Wesley, Readings (MA), 1992.
- [49] R.M. Haralick et L.G. Shapiro. *Computer and Robot Vision*. Volume 1, Addison-Wesley, 1992.
- [50] N. Grzywacz et S. Yuille. *A model for the estimate of local image velocity by cells in the visual cortex*. Proceedings of the Royal Society of London B. 239 :129-161, 1990.
- [51] E.P. Simoncelli et D.J. Heeger. *A model of neural responses in visual area MT*. Vision Research, 38(5) :743-761, 1998.
- [52] C. Huang. et Y. Chen. *Motion estimation method using a 3D steerable filter*. Image and Vision Computing, 13 :21-32, 1995.
- [53] K.M. Mutch et W.B. Thompson. *Analysis of Accretion and Deletion at Boundaries in Dynamic Scenes*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 7(2) :133-138, mars, 1985.
- [54] H.H. Nagel. *On the Estimation of Optical Flow : Relations between Different Approaches and Some New Results*. Artificial Intelligence, 33(3) :299-324, novembre, 1987.
- [55] M.J. Black et D.J. Fleet. *Probabilistic detection and tracking of motion discontinuities*. International Conference on Computer Vision, Corfu, Grèce, pages 551-558, septembre, 1999.
- [56] Y. Weiss. *Smoothness in Layers : Motion Segmentation Using Nonparametric Mixture Estimation*. IEEE Conference on Computer Vision and Pattern Recognition, pages 520-526, 1997.
- [57] M. Irani, B. Rousso et S. Peleg. *Computing Occluding and Transparent Motions*. International Journal of Computer Vision, 12(1) :5-16, février, 1994.

- [58] A.D. Jepson et M.J. Black. *Mixture Models for Optical Flow Computation*. IEEE Conference on Computer Vision and Pattern Recognition, pages 760-761, 1993.
- [59] M.J. Black et P. Anandan. *The Robust Estimation of Multiple Motions : Parametric and Piecewise-Smooth Flow-Fields*. Computer Vision and Image Understanding, 63(1) :75-104, janvier, 1996.
- [60] T. Darrell et A.P. Pentland. *Cooperative Robust Estimation Using Layers of Support*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 17(5) :474-487, mai, 1995.
- [61] M. Lappe et J. P. Rauschecker. *A neural network for the processing of optical flow from egomotion in man and higher mammals*. Neural Computation, volume 5, pages 374-391, 1993.
- [62] E.P. Simoncelli. *Distributed Analysis and Representation of Visual Motion*. Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, 1993.
- [63] V. Chapdelaine-Couture, S. Roy, M.S. Langer et R. Mann. *Principal Components Analysis of Optical Snow*. dans British Machine Vision Conference, Kingston (UK), septembre 2004, vol. 2, pages 799-808.
- [64] V. Chapdelaine-Couture et Sébastien Roy. *A Global Analysis of Optical Snow for Arbitrary Camera Motions*, dans Irish Machine Vision and Image Processing Conference, Dublin (Irlande), septembre 2004, pages 210-215.
- [65] M. Pollefeys, R. Koch et L. Van Gool. *A simple and efficient rectification method for general motion*. International Conference on Computer Vision, Corfu, Grèce, pages 496-501, 1999.

ANNEXE I

PRÉSENCE D'ÉNERGIE SUR L'AXE DU NOEUD PAPILLON



FIG. I.1 – Séquence Holly : la caméra subit une translation horizontale (32 images de 64x64 pixels).

À la section 3.3.1, un argument mathématique démontre que la magnitude des énergies dans le domaine de Fourier ne fournit pas un indice suffisant pour détecter l'axe du noeud papillon. Par contre, les puissances donnent une concentration d'énergie supplémentaire sur cet axe. Dans cette annexe, nous présentons les résultats de l'analyse de la séquence Holly (voir Fig. I.1) en utilisant d'abord les magnitudes et ensuite les puissances. L'axe retrouvé devrait être $(0.0, 1.0, 0.0)$. Au tableau I.1, on peut voir que les puissances donnent clairement une valeur propre maximale correspondant à l'axe, alors que les magnitudes d'énergie donnent les deux plus grandes valeurs propres presque égales ; il y a donc ambiguïté : quelle est la bonne direction de l'axe ? Dans ce cas-ci, l'axe correspondant à la plus grande valeur propre est relativement dans la bonne direction, ce qui est probablement dû au fait qu'il y a généralement plus de basses fréquences (fréquences proches de l'axe du noeud papillon) dans une séquence réelle.

TAB. I.1 – Base propre en fonction du poids des énergies

Utilisation des Magnitudes		
Valeur propre	Vecteur propre	
209170.48	(-0.15, 0.98, -0.25)	Axe
185281.42	(-0,98, -0,15, -1,72)	
31717.10	(-1.75, -0.01, 1.00)	Plan moyen
Utilisation des Puissances		
Valeur propre	Vecteur propre	
320485.50	(-0.00, 0.99, 0.01)	Axe
10852.66	(0,99, 0,02, 1,94)	
2080.58	(-1.95, 0.00, 1.00)	Plan moyen

ANNEXE II

CODE DE L'ANALYSE DE COMPOSANTES PRINCIPALES AVEC
CISAILLEMENT TEMPOREL

```

int i,j,k,index;
double fx,fy,ft,len;
int NB_FREQUENCIES=SIZE_T * SIZE_Y * SIZE_X;

Vector bowtie_axis(0.0,0.0,0.0); //bowtie axis direction
Vector best_fit_plane(0.0,0.0,1.0); //normal of best fit plane
//normal of current plane fit used iteratively to find best fit plane
Vector curr_fit(0.0,0.0,1.0);
//stores shearing offset for each spatial frequency fx,fy
Matrix Shearing(SIZE_Y,SIZE_X);
Matrix A(NB_FREQUENCIES,3); //stores sheared power
Matrix Covariance(3,3);
int iter=0;

Normalize(Power,SIZE_T,SIZE_Y,SIZE_X);

do {
  for (j=0;j<SIZE_Y;j++) {
    if (j>SIZE_Y/2) fy=j-SIZE_Y;
    else fy=j;
    for (k=0;k<SIZE_X;k++) {
      if (k>SIZE_X/2) fx=k-SIZE_X;
      else fx=k;
      Shearing(j,k)+=(-fx*curr_fit.x-fy*curr_fit.y)/curr_fit.t;
      for (i=0;i<SIZE_T;i++) {
        ft=i-Shearing(j,k); //sheared ft

```

```

        //put ft back within frequency range
        ft=fmod(ft,SIZE_T);
        if (ft<0) ft+=SIZE_T;
        if (ft>SIZE_T/2) ft-=SIZE_T;

        index=i*SIZE_Y*SIZE_X+j*SIZE_X+k;
        A(index,0)=fx*Power(i,j,k);
        A(index,1)=fy*Power(i,j,k);
        A(index,2)=ft*Power(i,j,k);
    }
}
}
Covariance = (Transpose(A) * A) / NB_FREQUENCIES;
SVD(Covariance, U, D, V); //eigenvalues in D sorted in desc. order
//bowtie axis direction is eigenvector having biggest eigenvalue
bowtie_axis.x=V[0][0];
bowtie_axis.y=V[1][0];
bowtie_axis.t=V[2][0];
//current plane fit is eigenvector having smallest eigenvalue
curr_fit.x=V[0][2];
curr_fit.y=V[1][2];
curr_fit.t=V[2][2];

//normalize current plane fit to (x,y,1)
if (curr_fit.t) curr_fit.scale(1/curr_fit.t);
else curr_fit.set(0.0,0.0,1.0); //this case should not occur!

best_fit_plane.x+=curr_fit.x;
best_fit_plane.y+=curr_fit.y;
iter++;
len=sqrt(curr_fit.x*curr_fit.x+curr_fit.y*curr_fit.y);
}while(len>TRESHOLD && iter<MAX_NB_ITER); //while curr. plane != (0,0,1)

```