

N° d'ordre : 1377

THESE

Présentée à

L'UNIVERSITE DES SCIENCES ET TECHNOLOGIES DE LILLE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITE

Spécialité : Electronique

par

Isabelle DEHON



**"ETUDE DE L'UTILISATION DE LA TRANSFORMEE
DE WIGNER-VILLE A L'ESTIMATION DE MOUVEMENT
DANS UNE SEQUENCE D'IMAGES REELLES "**

Soutenu le 18 OCTOBRE 1994 devant la commission d'examen

Membres du Jury :	MM. Y.LEROY	Président
	H.VU.THIEU	Rapporteur
	J.THIEL	Rapporteur
	J.P.DUBUS	Directeur de thèse
	A.HACKETT	Examineur
	M.KERDRANVAT	Examineur
	V.DEVLAMINCK	Examineur

Ce travail a été effectué dans les laboratoires de traitement d'images de Thomson Consumer Electronics d'Illkirch-Graffenstaden, en Alsace.

Je tiens à remercier les personnes dont la collaboration, à des degrés divers, m'a été précieuse pour la réalisation de cette étude.

Je citerai en premier lieu Messieurs J .C. Guillon et A. Hackett qui m'ont permis d'effectuer ma thèse au sein de leur laboratoire. Ensuite, Monsieur J.P. Dubus, mon directeur de thèse, qui m'a accordé une grande liberté et une confiance totale. Monsieur Y. Leroy qui me fait l'honneur de présider le jury, ainsi que Messieurs les professeurs J. Thiel et H. VU. Thien pour avoir accepté d'être rapporteurs de ma thèse. V. Devlaminck pour sa participation en temps qu'examineur. M. Kerdranvat, également, ingénieur chez Thomson, qui m'a apporté une aide précieuse à la réalisation de ce travail.

Je n'oublierai pas non plus les membres de Thomson pour la sympathie qu'ils m'ont témoigné, et plus particulièrement François Albin, Werner Boie, François Cau, Jean-Claude Chevet, Larry Fitzgerald, Philippe Galaup, Catherine Heimburger, Pascal Maetz, Jean-Louis Monasson, Laurent Perdrieau, Laurence Piquet et Bertrand Wendling.

Et bien sur mes parents, mon frère Eric, et mes deux soeurs Eliane et Sophie.

Que tous soient assurés de mon extrême reconnaissance.

Je tiens particulièrement à citer les personnes suivantes pour leur réelle amitié qu'elles m'ont plus d'une fois témoigné.

Carine Ferber, Ioana Stan, Catherine Laugas trio de choc sur les pistes, et ailleurs., Akim Tekiouk pour sa patience et son soutien permanent, Benoît Trinquet son acolyte, Bertrand Riedin pour les souvenirs de voyage...et les chutes de VTT, Catherine et Philippe Fuch, Mohamed Elhag, John Sadi et ses coup de foudres perpétuels, Nathalie Greffier et ses remèdes contre les maux de gorge, Fabienne Duhamel, ma binôme de choc. Nadine Christ, Isabelle et Sophie Menard ainsi que tous les étudiants du Deust "Gestion de l'entreprise industrielle", Thomas l'autrichien et Thomas l'allemand, ainsi que la famille Boutoubat et leurs nombreux amis.

SOMMAIRE

CHAPITRE I DIFFERENTES METHODES.D'ESTIMATION DE MOUVEMENT

1.GRADIANT SPATIO-TEMPOREL.....	2
1.1.Principe	2
1.2.Algorithme itératif.....	4
1.3.Contraintes de lissage.....	6
1.4.Conclusion	6
2.METHODES PROBABILISTES	7
2.1.Principe	7
2.2.Méthode de calcul	7
2.3.Conclusion	9
3.METHODES MULTIRESOLUTION.....	10
3.1.Introduction	10
3.2.Exemples de décomposition multirésolution	10
3.3.Conclusion	14
4.METHODES DE MISE EN CORRESPONDANCE	15
4.1.BLOCK MATCHING.....	15
4.1.1.Principe	15
4.1.2.Différentes méthodes de calcul	16
4.1.3.Utilisation de blocs de tailles variables	19
4.1.4.Conclusion.....	20
4.2.METHODES UTILISANT L'EXTRACTION DE CONTOURS	20
4.2.1.Détection de points caractéristiques	20
4.2.2.Détection de contours.....	21
4.2.3.Détection des points anguleux.....	22
4.2.4.Calcul des vecteurs déplacement.....	22
4.2.5.Conclusion.....	23
4.3.TRANSFORMEE DE HOUGH.....	23
4.3.1.Principe	23
4.3.2.Méthode de caractérisation de droite.....	23
4.3.3.Recherche des valeurs de différents paramètres de mouvement	25
4.3.4.Conclusion :	25
5.METHODES PAR TRANSFORMEE.....	26
5.1.Méthodes utilisant les propriétés de la transformée de Fourier	26
5.1.1.Introduction	26
5.1.2.Théorie	26
5.1.3.Conclusion.....	27
5.2.Corrélation de phase.....	27
5.2.1.Théorie	27
5.2.2.Calculs.....	28
5.2.3.Interprétation de la surface de corrélation	30
5.2.4.Recherche des coordonnées (x,y) du vecteur déplacement	33
5.2.5.Conclusion.....	33
5.3.Transformées temps-fréquence	34
5.3.1.Introduction	34
5.3.2.Utilisation de la transformée de Wigner-Ville	36
5.3.2.1.Exemple de représentation de la fonction test.....	38
5.4.Conclusion	39
6.Conclusion	39

CHAPITRE II TRANSFORMEE DE WIGNER-VILLE : THEORIE

1.CAS MONODIMENSIONNEL	44
1.1.Definition et Propriétés	44
1.1.1.Définition	44
1.1.2.Propriétés.....	44
1.2.Transformée de Wigner-Ville discrète (TWVD)	46
1.3.Calculs du suréchantillonnage du signal	46
1.3.1.Calcul du Signal analytique	47
1.4.Terms d'interférences	49
1.4.1.Calcul de la transformée de Wigner-Ville.....	51
2.CAS TRIDIMENSIONNEL	53
2.1.Définition	53
2.2.Interprétation.....	53
2.3.Calcul de la transformée de Wigner-Ville 3D.....	55
2.3.1.Calcul de la FFT 3D.....	55
2.3.2.Calcul du signal analytique 3D	55
3.TRANSFORMEE DE WIGNER-VILLE 3D ET DEPLACEMENT	57
3.1.DWV d'une image en translation	57
3.2.DWV d'une image en rotation.....	60
3.3.Calcul de l'inclinaison du plan	61

CHAPITRE III CALCULS ET RESULTATS

1.DESCRPTION DES CALCULS	61
1.1.Théorie.....	61
1.2.Méthode de calcul.....	61
1.3.Calcul théorique de V_x, V_y	64
2.TESTS PRELIMINAIRES	66
2.1.Introduction	66
2.2.Séquences utilisées	66
2.3.Description des calculs	66
2.3.1.Filtrage de Sobel.....	67
2.3.2.Affectation d'une valeur de (V_x, V_y) aux points non calculés	68
2.3.3.Tests sur le calcul de (V_x, V_y)	69
2.4.Tests sur les différents algorithmes	70
2.5.Conclusion	72
3.CALCUL SUR UN POINT FICTIF	73
4.OPTIMISATION DES PARAMETRES DE CALCUL	76
4.1.Introduction	76

4.2. Tests sur la taille du cube : N_1, N_2, N_b -Im.....	76
4.2.1. Importance de la taille du cube	76
4.2.2. Modification de l'algorithme de calcul	78
4.2.3. Description des tests	81
4.2.4. Conclusion	82
4.3. Tests sur la manière de calculer (V_x, V_y)	82
4.3.1. Tests sur les points réels	83
4.3.2. Tests sur les points fictifs	84
5. SURECHANTILLONNAGE DU SIGNAL.....	85
5.1. Introduction	85
5.2. Utilisation de la FFT.....	85
5.3. Calcul de la transformée de Hilbert par l'intermédiaire d'un filtre	85
5.4. Préfiltrage de l'image originale.....	86
6. INFLUENCE DU BRUIT	89
6.1. Calcul sur les points réels de l'image.....	89
6.2. Test sur les points réels.....	90
6.3. Test sur les images suréchantillonnées	91
7. PRECISION DU DEMI-PIXEL.....	91
8. METHODES DE CALCUL DE L'INCLINAISON DES PLANS	92
8.1. Méthode directe	92
8.2. Calcul de (V_x, V_y) par régression	93
9. VISUALISATION DES PLANS ET DE LA FONCTION TEST	95
9.1. Description de la visualisation de plans	95
9.2. Description de la visualisation de la fonction test.....	96
9.3. Visualisation des plans et de la fonction test	97
10. CONCLUSION	106

CHAPITRE IV APPLICATION DE L'ALGORITHME AUX IMAGES DE T.V

1. Le système de télévision.....	108
1.1. Généralités	108
1.2. Les défauts du 50Hz entrelacé.....	109
1.2.1. Le papillotement des grandes plages (Large area flicker)	110
1.2.2. Le scintillement interligne (line flicker)	111
1.2.3. Le défilement de lignes (line crawling)	112
2. La conversion 100Hz.....	113
2.1. Passage au 100Hz sans compensation de mouvement	113
2.1.1. Le mode "AABB"	113
2.1.2. Le mode "ABAB"	115

2.2. Passage au 100Hz avec estimation de mouvement.....	116
2.2.1. Passage au 50Hz progressif	116
2.3. Passage au 100Hz	120
3. PRINCIPAUX RESULTATS OBTENUS.....	122
3.1. Test sur la manière de calculer le vecteur déplacement.....	125
4. Calculs sur les points fictifs.....	126
4.1. Tests sur la taille des blocs de calcul.....	128
CONCLUSION	133

INTRODUCTION

La transmission ou l'amélioration des images animées nécessite souvent le calcul des vecteurs déplacement.

A titre d'exemple, nous citerons deux cas où celui-ci constitue une étape importante à l'analyse d'images en mouvement.

Le premier emploi est le calcul des vecteurs déplacement pour comprimer les images de la séquence étudiée.

En effet, une séquence d'image représente une telle quantité d'information qu'il est impossible de la transmettre telle quelle par les réseaux de communications usuels. Cependant, dans une séquence d'images animées, il existe de nombreuses informations redondantes telles que les zones fixes d'une image à l'autre. Afin de réduire la quantité d'information à envoyer, on estime alors les vecteurs déplacement de chaque pixel, ou de zones spécifiques de l'image afin de n'envoyer que ces vecteurs et l'erreur d'estimation. C'est d'ailleurs cette méthode couplée à la transformée en cosinus discret suivie d'un codage adéquat qui est utilisée pour le visiophone.

Le deuxième exemple concerne l'amélioration de la qualité des images à transmettre, et plus particulièrement en ce qui concerne les images de télévision. En effet, le mode de visualisation actuel n'est pas sans présenter certains défauts. Une solution consiste alors à modifier la fréquence d'apparition des images à l'écran. Mais pour que les résultats soient réellement acceptables, la connaissance des vecteurs déplacement entre deux images (ou trames) consécutives s'avère nécessaire.

Dans le cadre de cette thèse, nous nous proposons d'étudier le moyen de calculer les vecteurs déplacement d'une image réelle dans le domaine transformé. Notre choix d'étude s'est porté sur la transformée de Wigner-Ville qui contrairement aux autres méthodes d'estimation de mouvement par transformée permet d'estimer directement le vecteur déplacement de n'importe quel pixel de l'image.

Dans un premier chapitre, nous détaillerons les différentes méthodes d'estimation de mouvement en soulignant leur principaux avantages et inconvénients.

Dans le second chapitre, nous décrirons les principales propriétés de la transformée de Wigner-Ville ainsi que la manière de l'appliquer sur une séquence d'images. Nous y détaillerons aussi le comportement de cette transformée suivant le mouvement à déterminer.

Le troisième chapitre sera consacré au calcul de la transformée de Wigner-Ville 3D sur des séquences fictives d'images réelles dont nous connaissons parfaitement les vecteurs déplacement de chaque point de l'image ; ce qui nous a permis d'effectuer les premiers tests quant à la possibilité de calculer les vecteurs déplacement à partir de cette transformée.

Le dernier chapitre porte sur l'application de notre algorithme sur des images de télévision.

Auparavant nous aurons décrit le mode de visualisation du système actuel ainsi que ses principaux inconvénients.

CHAPITRE I

DIFFERENTES METHODES D'ESTIMATION DE MOUVEMENT

Dans ce chapitre, nous allons passer en revue les différentes méthodes d'estimation de mouvement en soulignant leurs principaux avantages et inconvénients.

L'une des premières méthodes de calcul des vecteurs déplacement est basée sur la résolution de l'équation de contrainte du mouvement que nous décrirons dans le premier chapitre. Bien que celle-ci permette effectivement d'estimer les vecteurs déplacement des pixels de l'image, elle ne traite pas correctement les problèmes d'occlusion.

Afin d'y remédier, Bouthemy et Heitz [HeB 90] proposent une méthode basée sur l'étude statistique du mouvement que nous étudierons dans le second chapitre.

Cependant, le cas des mouvements de forte amplitude n'est pas résolu de manière satisfaisante.

La méthode multirésolution, développée dans le troisième chapitre, permet de résoudre ce problème. Elle consiste à calculer le vecteur déplacement des pixels de l'image sur plusieurs niveaux de résolutions.

Plutôt que de résoudre l'équation de contrainte du mouvement une autre solution consiste à rechercher les pixels, ou objets les plus semblables entre deux images consécutives. C'est ce que l'on appelle la méthode de mise en correspondance que nous décrirons dans le chapitre quatre. Dans ce chapitre sont regroupés la méthode du "Block-Matching", les méthodes basées sur l'extraction des contours, et la transformée de Hough.

Jusqu'à présent, les méthodes décrites effectuent les calculs des vecteurs déplacement directement sur l'image d'origine mais il est tout à fait possible d'effectuer ces calculs sur des images transformées.

Dans le dernier chapitre, nous décrirons les méthodes de calcul des vecteurs dans un domaine autre que le domaine temporel. Trois transformées y sont regroupées : la transformée de Fourier, la corrélation de phase, basée elle aussi sur les propriétés de la transformée de Fourier, et pour en finir les transformées temps-fréquences qui permettent de palier au principal défaut des méthodes précédentes à savoir la perte d'information sur le temps.

1. GRADIENT SPATIO-TEMPOREL

1.1. Principe

Cette méthode est basée sur le calcul du gradient de luminance en chaque point de l'image. On suppose que la variation d'intensité entre deux images, ou trames, successives est une fonction linéaire du déplacement.

Si on désigne par x et y les coordonnées d'un pixel d'une image ; $f(x,y,t)$ représente la fonction intensité du pixel de coordonnées (x,y) à l'instant t .

Soit DFD (Displacement Frame Différence, en anglais) tel que :

$$DFD(x,y,D) = f(x, y, t) - f(x - dx, y - dy, t - dt) \quad (1)$$

où :

- dx et dy sont les composantes du vecteur déplacement.
- $(f(x, y, t) - f(x - dx, y - dy, t - dt))$ représente la différence entre l'intensité du pixel de coordonnées (x,y) situé dans la trame (ou l'image t) et celle du pixel de coordonnées $(x + dx, y + dy)$, situé dans l'image (ou trame) $t+1$.

Si l'on suppose que l'intensité du pixel reste constante au cours du déplacement, on peut écrire:

$$DFD(x,y) = f(x,y,t) - f(x + dx, y + dy, t + dt) \quad (2)$$

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

$$f(x + dx, y + dy, t + dt) = f(x, y, t) + \left[\frac{\partial f(x, y, t)}{\partial x} \right] dx + \left[\frac{\partial f(x, y, t)}{\partial y} \right] dy + \left[\frac{\partial f(x, y, t)}{\partial t} \right] dt + \varepsilon \quad (3)$$

avec ε terme d'ordre supérieur que l'on néglige.

En injectant l'équation (3) dans (2), puis en divisant par dt, on obtient un système d'équation que l'on résout par régression linéaire afin d'obtenir les valeurs de dx et de dy.

La résolution de ce système donne les solutions suivantes

$$dx = \frac{\sum DFD \delta y_i \cdot \sum \delta x_i \delta y_i - \sum DFD \delta x_i \cdot \sum \delta y_i^2}{\sum \delta x_i^2 \cdot \sum \delta y_i^2 - (\sum \delta x_i \delta y_i)^2} \quad (4)$$

$$dy = \frac{\sum DFD \delta x_i \cdot \sum \delta y_i \delta x_i - \sum DFD \delta y_i \cdot \sum \delta x_i^2}{\sum \delta x_i^2 \cdot \sum \delta y_i^2 - (\sum \delta x_i \delta y_i)^2} \quad (5)$$

La figure 1 représente de façon schématisée le déplacement d'un point P1 entre deux trames successives. Ce point se retrouve, d'après les hypothèses posées, avec la même intensité en P2.

Le vecteur déplacement est représenté par la flèche en pointillés.

Si l'on projette P1 sur l'image qui apparaît au temps t + dt, le vecteur déplacement est alors représenté par la flèche reliant P'1 à P2.

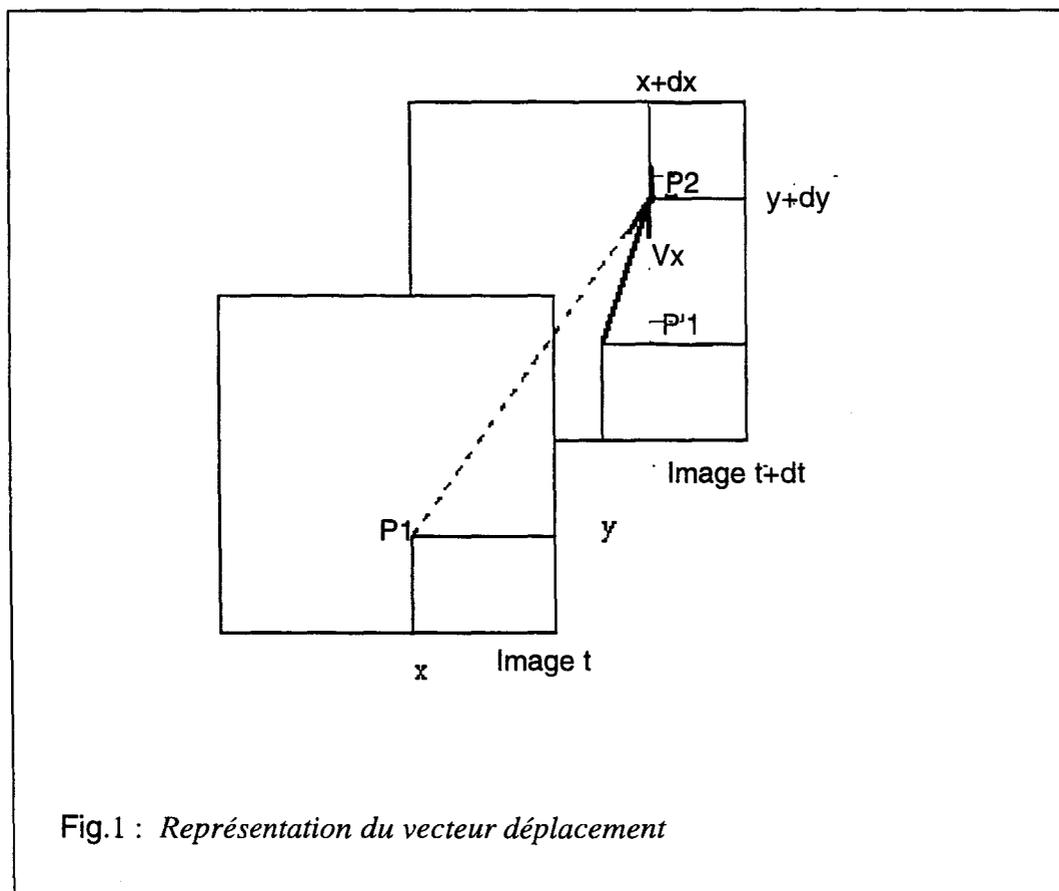


Fig.1 : Représentation du vecteur déplacement

Mais ce moyen de calculer dx , dy n'est valable que pour de faibles déplacements.

Afin d'étendre les calculs aux cas des mouvements de grandes amplitudes, Netravali et Robbins [NET 79] ont proposé d'utiliser une méthode récursive.

1.2.Algorithme itératif

Si l'on considère que D_i est l'estimation initiale du vecteur déplacement à l'instant i , et V_i un terme de correction à l'itération i .

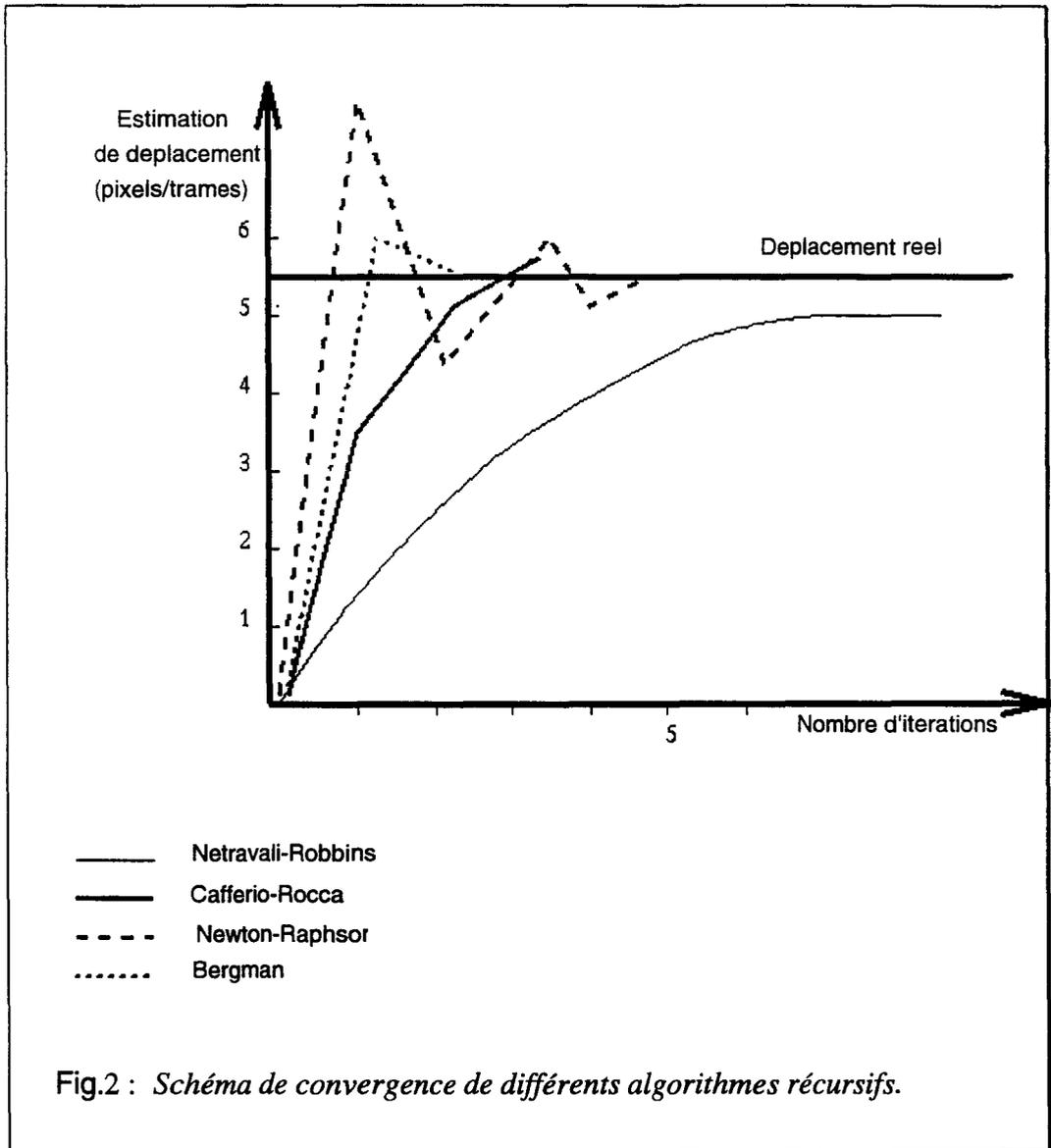
On peut écrire que l'estimation du vecteur à l'instant $i+1$, notée D_{i+1} est égal à :

$$D_{i+1} = D_i + V_i \quad (6)$$

L'algorithme itératif procède de la manière suivante : si la différence d'intensité entre les deux images, DFD, est négative, on se déplacera dans le sens des intensités croissantes, et inversement si DFD est positive. Et ceci de manière à minimiser le carré de la différence.

Le choix du terme de mise à jour, V_i , dépendra de la complexité de l'algorithme et de la vitesse de convergence désirée.

La figure 2 est un schéma comparatif des méthodes itératives les plus connues.[Mus 85].



Pour les différents algorithmes nous avons représenté la nombre d'itérations nécessaires pour aboutir au déplacement horizontal qui est de 5,5 pixels par trame dans l'exemple de la figure 2.

L'un des problèmes majeur de ce mode de recherche du vecteur déplacement (V_x, V_y) est que l'on ne peut estimer les deux composantes du vecteur vitesse. En effet, seule la composante parallèle au gradient d'intensité est calculable.

Afin d'y remédier, il est donc nécessaire d'ajouter d'autres contraintes aux calculs telles que les contraintes de lissage.

1.3. Contraintes de lissage

Horn et Schunk [Hos 81] ont introduit l'hypothèse que le champ des vitesses varie lentement et proposent alors de minimiser une intégrale étendue sur toute l'image.

On obtient alors deux équations que l'on résout par itération.

Mais cette méthode, très lourde en calculs, s'est avérée incapable de traiter correctement les discontinuités du champ de vitesse. Ceci est principalement dû au fait que la double intégration s'étend à toute l'image.

Il apparaît donc nécessaire d'introduire une contrainte locale supplémentaire, traitant les problèmes liés aux transitions de luminance entre deux zones de gradients différents.

Ainsi, Nagel [Nag 87] proposa de minimiser l'intégrale précédente, mais en poussant le développement de Taylor à l'ordre deux.

Cela permet un adoucissement du lissage dans les zones de discontinuités et donc une meilleure estimation de la vitesse.

1.4. Conclusion

La méthode spatio-temporelle est relativement simple à mettre en œuvre, mais de par sa récursivité, cet algorithme est délicat à implémenter ; et le traitement des discontinuités, qui est souvent nécessaire, alourdit encore les calculs.

L'obtention des vecteurs déplacement par le calcul du gradient spatio-temporel donne de bons résultats pour les faibles déplacements, avec la possibilité d'avoir une bonne précision sub-pixel.

Par contre, elle fait défaut dans le cas de grands mouvements car on ne peut plus supposer que l'intensité est constante pendant le temps dt , de même que pour les mouvements non translationnels.

Les performances se dégradent dès que l'approximation perd de sa validité c'est à dire dans les zones correspondant à des occlusions, dans les régions homogènes étendues ou fortement texturées.

Bouthemy et Heitz [HeB 90] proposent une solution en introduisant la notion d'approche bayésienne de l'analyse de mouvement.

2.METHODES PROBABILISTES

2.1.Principe

Cette approche consiste à solutionner les problèmes tels que les phénomènes d'occlusion par des méthodes statistiques.

Bouthemy et Heitz [HeB 90] décrivent le mouvement à l'aide de deux sortes d'étiquettes. Les étiquettes continues ω_s décrivant le champs des vecteurs vitesse au point s , et les étiquettes binaires, γ , décrivant les discontinuités (valant 1 en présence d'un contour d'occlusion).

2.2.Méthode de calcul

A partir de l'observation de la variation temporelle de l'intensité au point s , $\partial f/\partial t$, noté $f_t(s)$, et de la détection de contours, on obtient deux critères ; l'un caractérisant la vraisemblance des gradients temporels ; l'autre lié à la présence ou non de contours spatio-temporels.

Si on suppose que $f_t(s)$ est défini de la manière suivante :

$$f_t(s) = f_x(s).u_s + f_y(s).v_s + n(s)$$

avec :

$n(s)$: bruit blanc de variance σ^2

$\omega_s = (u_s, v_s)$: vecteur déplacement du point s .

La vraisemblance de ces observations est donnée par

$$p(f_t(s) | \omega_s) = 1/Z \frac{(f_x(s) u_s + f_y(s) v_s + f_t(s))^2}{2\sigma^2_s}$$

avec Z constante de normalisation.

La loi de probabilité est définie comme suit :

$$\begin{aligned} p(\theta = 0 | \gamma = 0) &= 0.5 \\ p(\theta = 0 | \gamma = 1) &= \varepsilon \\ p(\theta = 1 | \gamma = 0) &= 0.5 \\ p(\theta = 1 | \gamma = 1) &= 1 - \varepsilon \end{aligned}$$

θ est égal à 1 en présence de contours, dans le cas contraire, θ vaut 0.

L'estimation du champs de vecteurs vitesse se fera par le critère du M.A.P (Maximum A Priori), soit :

$$MAX(\omega, \gamma) = p(f_t, \theta | \omega, \gamma) p(\omega, \gamma). \quad (10)$$

Si on suppose en outre que la vraisemblance des observations se simplifie de la manière suivante :

$$p(f_t, \theta | \omega, \gamma) = \prod_{s \in S} p(ft(s) | \omega) \times \prod_{d \in D} p(\theta | \gamma) \quad (11)$$

le critère MAX devient :

$$MAX = \prod_{s \in S} p(ft(s) | \omega) \times \prod_{d \in D} p(\theta | \gamma). p(\omega, \gamma) \quad (12)$$

Si on modélise le problème de l'estimation de mouvement par un champs de Markov cela permet de ne plus considérer un pixel comme un point isolé, mais d'y associer les interactions existant avec ses voisins. A chaque site particulier s , on lui associe alors un voisinage V_{sk} .

L'hypothèse markovienne suppose que la probabilité conditionnelle n'est fonction que du voisinage immédiat du site considéré, ce qui donne :

$$Prob(x_i | X_j) = Prob(x_i | V_i) \quad (13)$$

où

V_i est le voisinage du site i .

x_i est la valeur de l'intensité lumineuse au site i .

X_j est l'image moins le site i en question.

Un système markovien peut être décrit par la loi de Gibbs, soit donc :

$$Prob(x_i | X_j) = k e^{-U(x)} \quad (14)$$

avec :

U : énergie de l'image

k : constante de normalisation appelée fonction de répartition.

On peut donc écrire :

$$p(\omega, \gamma) = 1/Z e^{-U(x)} \quad (15)$$

$$\text{avec } U(x) = \sum_{c \in C} V_c(\omega, \gamma).$$

Le critère du MAP revient donc à maximiser la fonction d'énergie suivante :

$$U(x) = \sum_{s \in S} \{1/\sigma^2 (f_x(s) u_s + f_y(s) v_s + f_f(s))^2\} - \sum_{d \in D} \log p(\theta | \gamma) + \sum_{c \in C} V_c(\omega, \gamma). \quad (16)$$

2.3.Conclusion

Les méthodes probabilistes s'affranchissent de l'hypothèse très restrictive de la constance de la luminance. Elles peuvent par conséquent traiter les mouvements plus complexes que la translation ou la rotation pure. Un autre avantage de cette méthode par rapport à la méthode basée sur le calcul du gradient spatio-temporel est que les contours d'occlusion (ou discontinuités de mouvement) sont beaucoup mieux traités. Ces derniers ont en effet tendance à être trop lissés lors de l'utilisation de contraintes supplémentaires des méthodes différentielles.

Mais le principal inconvénient de ces méthodes est la détermination des paramètres définissant les distributions du champ de Markov car ceux-ci changent d'une séquence à l'autre.

Cependant, le problème des mouvements de fortes amplitudes subsiste encore. Une solution consiste à étudier les images de la séquence à divers niveaux de résolution.

3.METHODES MULTIRESOLUTION

3.1.Introduction

Cette technique consiste à réaliser des estimations successives du vecteur déplacement sur des représentations des images à des échelles et des niveaux de résolution de plus en plus fins. Les principales différences existant entre les solutions proposées résident dans la façon d'obtenir ces différents niveaux d'images, ainsi que dans la façon de propager l'estimation aux travers de ces niveaux.

L'obtention de ces derniers se ramène en général à des opérations de filtrage passe-bas avec sous-échantillonnage mis en œuvre dans les structures pyramidales. Les filtres utilisés sont parfois choisis pour leur simplicité de réalisation mais peuvent aussi être dérivés de techniques plus générales. C'est notamment le cas lorsque l'on utilise l'approche multirésolution qui permet de prendre en compte certains paramètres plus spécifiquement liés à des applications particulières. On peut citer par exemple le cas des problèmes de compression d'images dans lesquels un critère de décorrélation entre les niveaux est un atout supplémentaire afin d'obtenir un taux de compression élevé.

3.2.Exemples de décomposition multirésolution

La décomposition multirésolution permet d'appliquer le calcul des vecteurs déplacement à différents niveaux de résolutions de l'image.

Enkelmann [Enk 86] utilise une décomposition multirésolution pour le calcul des vecteurs déplacement.

Le vecteur de l'image de niveau $k+1$ est obtenu à partir de celui du niveau k et d'un terme correcteur D_u soit donc :

$$u^{(k+1)}(x) = u^{(k)}(x) + D_u^{(k)}(x) \quad (17)$$

Le filtre de décomposition est un filtre gaussien, symétrique et circulaire.

Le passage du vecteur de niveau k à celui de niveau $k+1$ s'effectue par l'opérateur suivant :

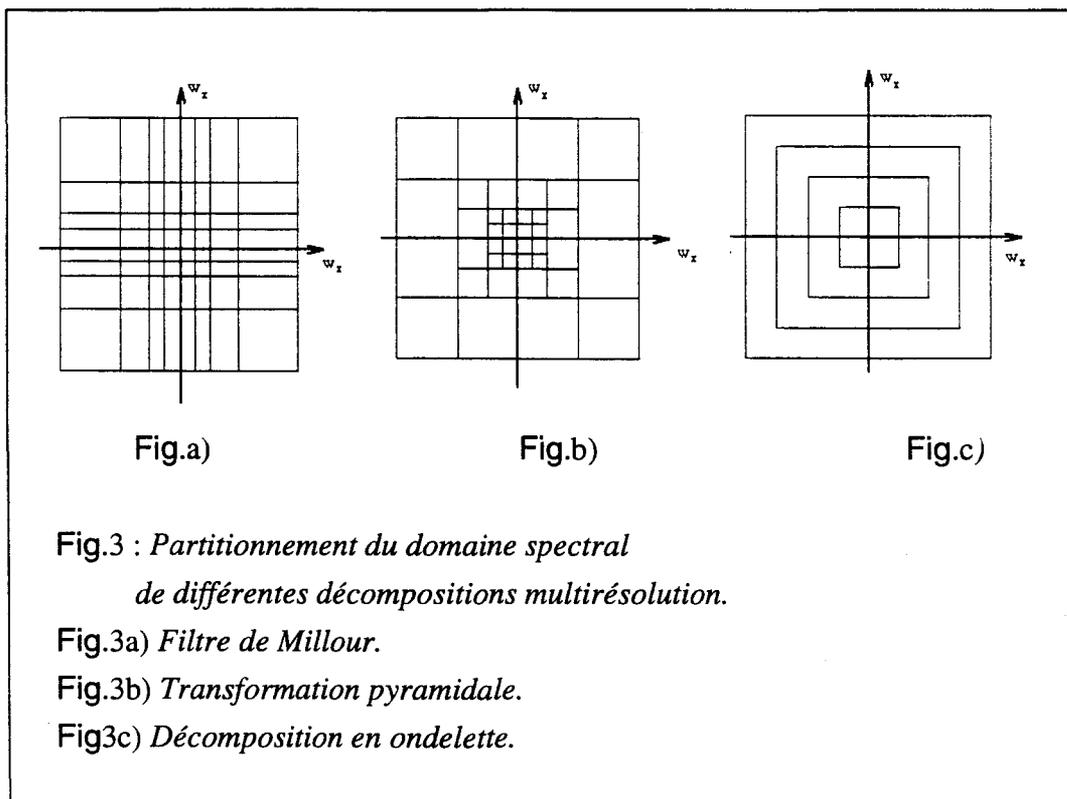
$$I = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Millour [Mill 89] utilise une méthode dérivée de la méthode différentielle sur des images de divers niveaux de résolutions obtenues par filtrages successifs.

Aux équations de mouvement, il ajoute deux coefficients λ et μ qui sont adaptés en fonction de la fréquence centrale du filtre passe bande. A chaque bande, on estime les vecteurs, qui serviront d'initialisation pour le calcul des vecteurs de mouvement de la bande voisine.

Cela permet d'affiner les calculs à chaque niveau de décomposition. A chaque fois, Millour utilise un petit groupe d'images élémentaires ce qui donne un système surdimensionné que l'on résout par une méthode de pseudo-inverse.

Le filtre utilisé par Millour partitionne l'image dans le domaine spectral de la manière représentée sur la figure 3.a.). A titre de comparaison les décompositions spectrales pour une transformation pyramidale et en ondelettes ont aussi été schématisées (Fig.3b et Fig.3c)



Comme on peut le remarquer sur la figure 3 suivant la décomposition utilisée, le domaine spectrale sera partitionné différemment.

Dernièrement, on peut noter un vif intérêt des chercheurs pour une décomposition particulière, la décomposition en ondelettes, et notamment celle développée par S.Mallat [Mallat 88] [Mallat 89]

Cette méthode consiste dans son principe à réaliser une décomposition des fonctions de carrés intégrables ($L^2(\mathbb{R})$) sur une succession de sous-espaces vectoriels emboîtés de ($L^2(\mathbb{R})$) au moyen d'un opérateur de projection orthogonale. Les sous-espaces vectoriels sont alors constitués par l'ensemble des approximations des fonctions de ($L^2(\mathbb{R})$) aux différents niveaux de résolution

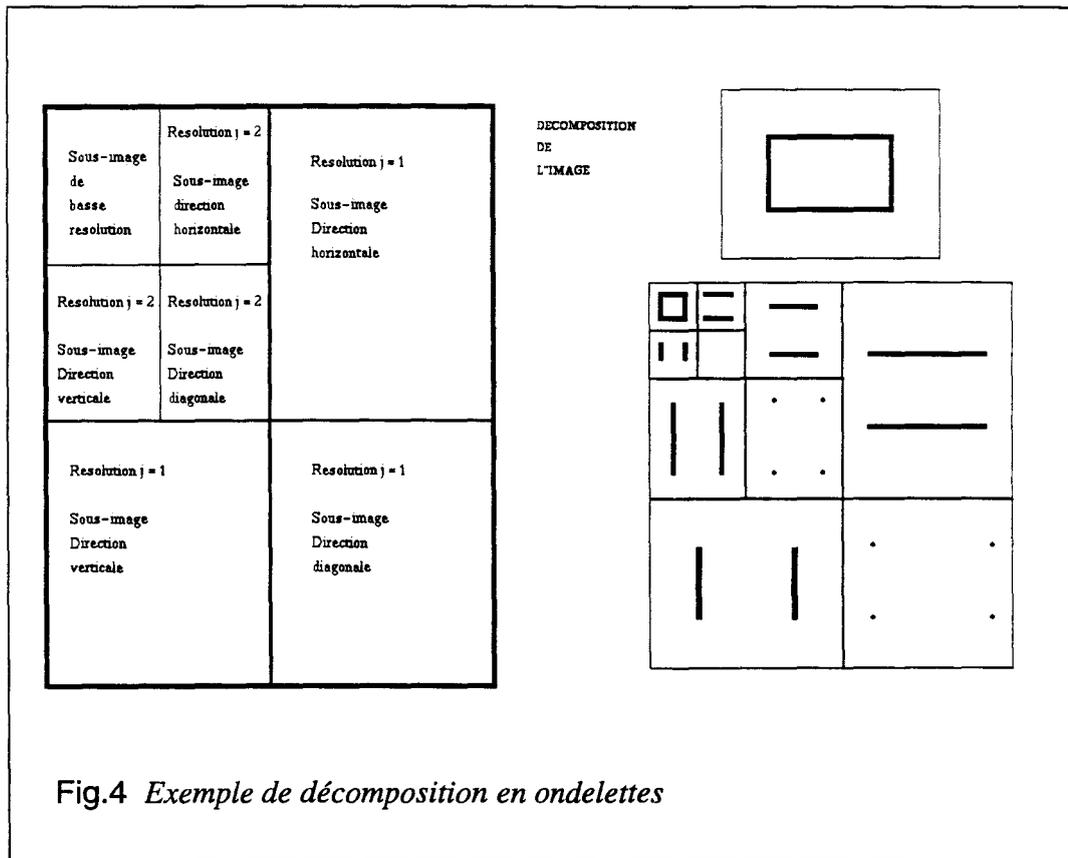
On suppose de plus, lors de la construction de ces sous-espaces, l'existence d'une fonction ϕ dite d'échelle telle que l'ensemble des translations entières et des dilatations par un facteur α de cette fonction constitue des bases orthonormées de ces sous-espaces vectoriels.

A la pyramide constituée de l'ensemble des projections orthogonales d'une fonction sur les sous-espaces vectoriels emboîtés, il est possible d'adjoindre une deuxième pyramide, dite de détails, obtenue par projection orthogonale de cette fonction sur les sous-espaces orthogonaux des précédents. On montre alors que ces derniers sous-espaces admettent pour base des translations entières et des dilatations d'un facteur α d'une fonction ψ appelée ondelette conformément à la théorie développée par Y.Meyer [Meyer 87]

Les filtres associés aux fonctions ϕ et ψ sont de type quadrature en miroir dans la solution proposée par S.Mallat.

Cette décomposition fournit des sous-images à différents niveaux de résolution et pour différentes orientations de contours : verticales, horizontales, et autre, que l'on appelle souvent diagonales. A chaque décomposition d'une image de taille ($N*N$), on obtient donc quatre sous-images de taille ($N/2*N/2$).

La figure 4 représente une image originale, un carré en l'occurrence, et sa représentation en ondelettes à trois niveaux de décomposition.



Outre son formalisme, l'apport de l'analyse par ondelettes en multirésolution, réside dans la possibilité de contrôle de certains paramètres des filtres lors de leur construction tel que l'orthogonalité, et surtout la régularité.

La possibilité de prise en compte des images de détails afin d'améliorer la qualité des estimations notamment sur les contours semble donner des résultats intéressants surtout lorsqu'elle se combine avec des méthodes destinées à traiter le problème des occlusions.

La décomposition en ondelettes est souvent utilisée pour la compression d'images animées : il suffit d'envoyer les coefficients d'ondelettes, correspondant à chaque sous-image, après les avoir codés. Mais cette décomposition peut aussi servir à l'estimation de mouvement.[Niky 93] [Chup 91]

Le principe consiste à calculer les vecteurs déplacements, sur l'image de plus basse résolution qui serviront ensuite à l'initialisation du calcul des vecteurs sur l'image de résolution supérieure et ainsi de suite jusqu'à l'image originale. Le calcul initial se faisant sur une image de dimensions réduites ce qui apporte un gain considérable par rapport aux calculs effectués directement sur l'image originale.

La correspondance des vecteurs entre les images de résolutions différentes posent les problèmes suivants :

- Nécessité de multiplier le module des vecteurs trouvés à la décomposition (J-1) afin de tenir compte du changement d'échelle lié à la décomposition.
- Affecter des vecteurs à des zones d'images n'existant pas lors des calculs, du fait de l'expansion.

3.3.Conclusion

La décomposition en ondelettes est encore rarement utilisée pour déterminer les vecteurs déplacements d'une image. Sa principale utilisation restant tout de même la compression.

Comme toutes les méthodes multirésolution, elle permet le calcul des grands déplacements. La rapidité des calculs est augmentée du fait de l'initialisation à chaque décomposition.

Cependant les problèmes de correspondance entre deux images de niveaux de décompositions différents sont assez délicats.

Plutôt que de calculer les vecteurs déplacement, on peut aussi rechercher deux objets semblables entre deux images consécutives. C'est ce que l'on appelle les méthodes de mise en correspondance.

4.METHODES DE MISE EN CORRESPONDANCE

Ces méthodes consistent à retrouver dans l'image 2, à un instant $t + dt$, un objet de la trame 1, à l'instant t , afin d'obtenir le vecteur déplacement de l'objet.

Il existe deux grandes catégories de mise en correspondance : les méthodes travaillant sur des zones quelconques de l'image, tel que le "Block Matching", et celles dont l'étape principale consiste à rechercher les points caractéristiques de l'image, pour ensuite effectuer la correspondance de ces points entre les images.

4.1.BLOCK MATCHING

4.1.1.Principe

Le Block Matching (mot à mot : correspondance de blocs) est une méthode bien connue qui consiste à faire correspondre certains éléments entre deux images et à les comparer. Ces éléments peuvent être soit des pixels, soit des blocs de pixels.

Pour estimer le vecteur déplacement (D_x, D_y) au point (x, y) dans la trame $(k + 1)$, on prend un bloc centré sur le point (x, y) dans la trame $(k + 1)$ et on le compare avec tous les blocs centrés au points $(x - D_x, y - D_y)$, et ceci à l'intérieur d'une zone fixée. Le vecteur déplacement correspondra aux deux blocs les plus semblables.

Sur la figure 5, nous avons visualisé le principe de la méthode de calcul par Block-Matching.

On utilise le bloc de taille (M, N) situé dans la trame K que l'on compare à un bloc de même taille situé dans la trame $K-1$.

Le bloc de la trame K se déplace à l'intérieur de la zone de recherche, de dimensions $(N+2d_n, M+2d_m)$.

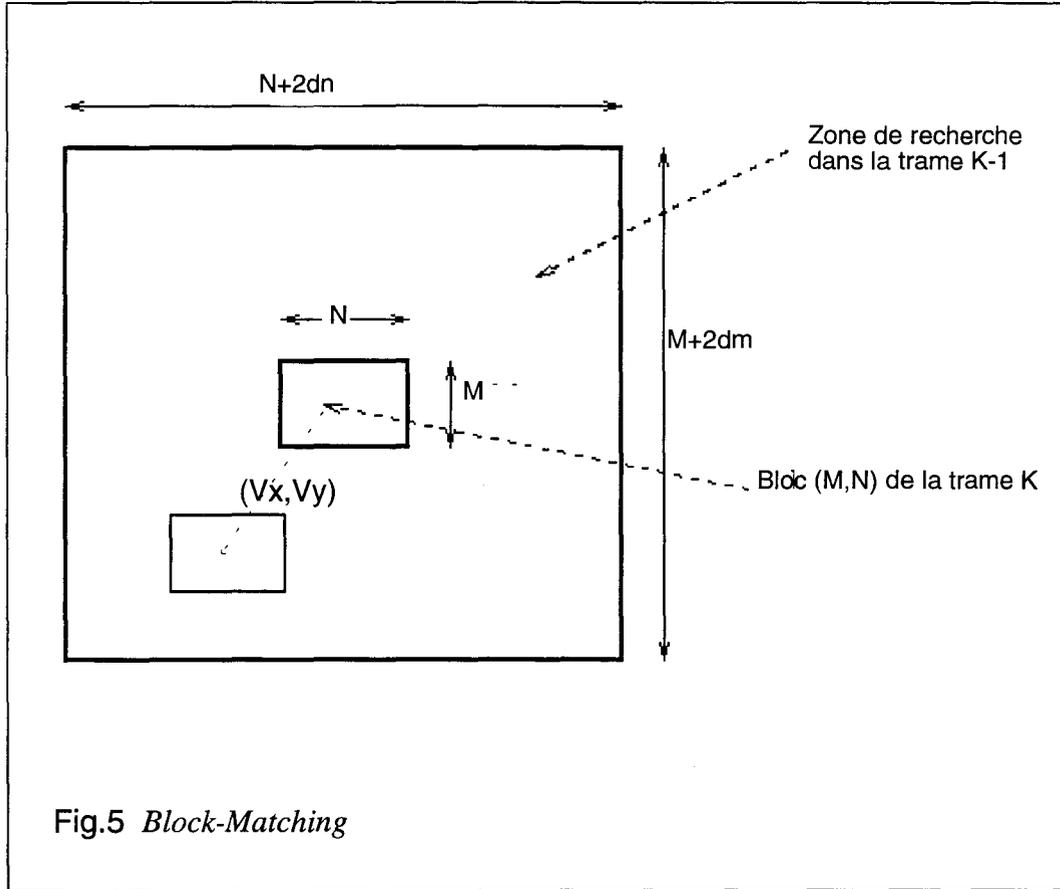


Fig.5 Block-Matching

4.1.2. Différentes méthodes de calcul

Les critères les plus souvent utilisés sont l'erreur quadratique moyenne, MSE : mean square error et la moyenne de la différence, MAD : mean absolute difference dont nous rappelons ici la définition.

$$MSE(i, j) = \frac{1}{MN} \sum_{m=1}^M \sum_{n=1}^N [I_k(m, n) - I_{k-1}(m+i, n+j)]^2 \quad (17)$$

$$MAD(i, j) = \frac{1}{MN} \sum_{m=1}^M \sum_{n=1}^N |I_k(m, n) - I_{k-1}(m+i, n+j)| \quad (18)$$

Avec :

- $I_k(x, y)$: Intensité du pixel de coordonnées (x, y) situé dans l'image k .
- $I_{k-1}(x, y)$: Intensité du pixel de coordonnées (x, y) situé dans l'image $k-1$.
- M, N : taille du bloc.

Pour chaque bloc, on calcule donc un vecteur déplacement. Il est à noter que cette méthode ne calcule pas directement le vecteur déplacement, mais recherche les deux blocs les plus similaires à l'intérieur d'une zone définie. Comme on ne connaît rien en ce qui concerne le vecteur à trouver, il faudrait tester tous les déplacements possibles à l'intérieur de la zone de calculs, ce qui serait beaucoup trop long.

Afin d'y remédier, J.Jain et A.Jain [Jain 81] ont mis au point la méthode de recherche logarithmique, en utilisant la MSE comme critère. Elle est basée sur l'hypothèse que le critère de comparaison décroît lorsque l'on se rapproche du bloc similaire à celui de référence. Cela permet donc de réduire de façon appréciable le nombre de calculs car dans ce cas on ne teste plus toutes les valeurs possibles de déplacement.

Par exemple, pour un bloc de taille (17 * 17) la recherche du vecteur déplacement ne nécessite que 26 tests au lieu de 289.

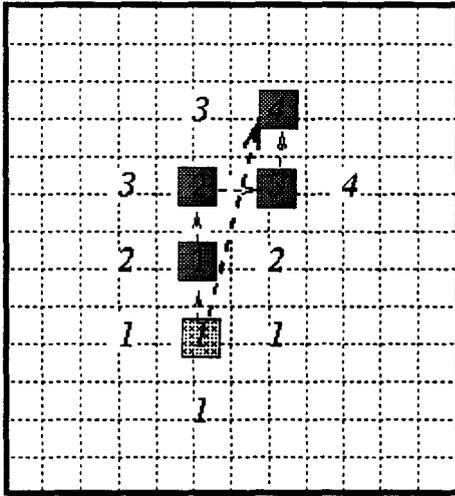
Une méthode similaire a été développée [Kog 81].

Au départ huit points, situés assez loin du point central, sont testés. A l'aide du critère MSE on recherche le point correspondant à un minimum du critère, on recommence alors l'opération centrée sur ce point, mais en réduisant l'espacement entre les points.

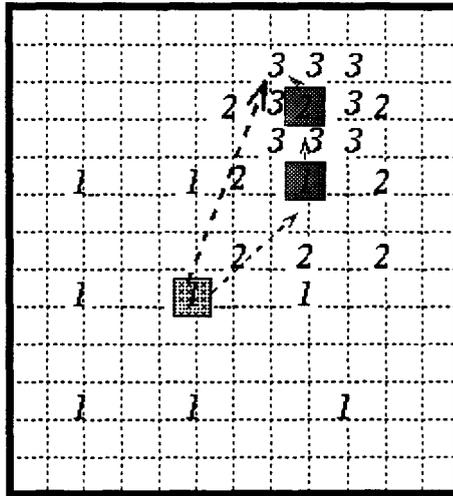
Une autre variante est la méthode conjuguée (conjugate direction search) [Sri 81].

Le critère utilisé est le MAD. On travaille tout d'abord dans une seule direction, horizontale par exemple. On calcule les MAD des points adjacents (de même ordonnée) au point de départ, on retient comme nouveau point de départ celui ayant le critère le plus faible, et ainsi de suite. On change de direction lorsque la valeur du critère du point central est entourée de deux valeurs de MAD qui lui sont supérieures.

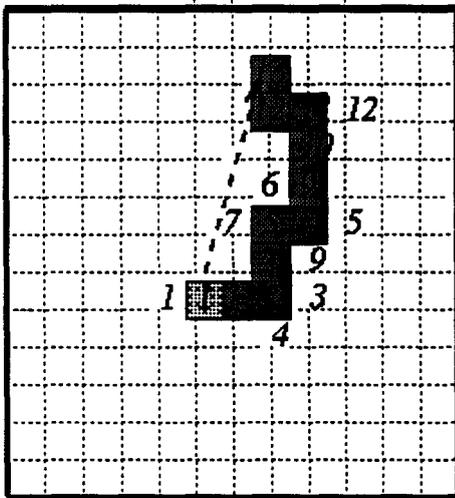
Nous avons regroupé ci-dessous les trois méthodes citées précédemment afin de faciliter leur comparaison.



Recherche logarithmique



Menu search



Recherche conjuguée

-  Point de départ
-  Point ayant la valeur maximale pour le critère

Dans le cas de mouvement d'amplitude importante, le nombre d'itérations à effectuer reste encore trop élevé car dans ce cas il est nécessaire d'utiliser des blocs de grandes tailles. De plus, on risque d'avoir plusieurs objets d'intensités différentes à l'intérieur de ce bloc.

Afin d'y remédier, une solution consiste à utiliser des blocs de taille variables.

4.1.3. Utilisation de blocs de tailles variables

Les calculs débutent avec de larges blocs afin d'obtenir un vecteur représentatif du mouvement principal. Il est ensuite utilisé comme vecteur initial afin d'affiner les calculs, sur des blocs de plus faibles tailles. Lors des premiers calculs, l'image est sous-échantillonnée afin de réduire le nombre de calculs.

La figure 6 représente de manière simplifiée le principe que nous venons de décrire.

Les calculs commencent en utilisant le bloc (1). On obtient alors le vecteur V_1 .

A la seconde étape, on recherche le vecteur V_2 à l'intérieur du bloc (2). Celui-ci est centré sur le point P_1 qui est le centre du bloc 1 décalé de V_1 ; et ainsi de suite.

Le vecteur final est égal à la somme des vecteurs obtenus à chaque itération.

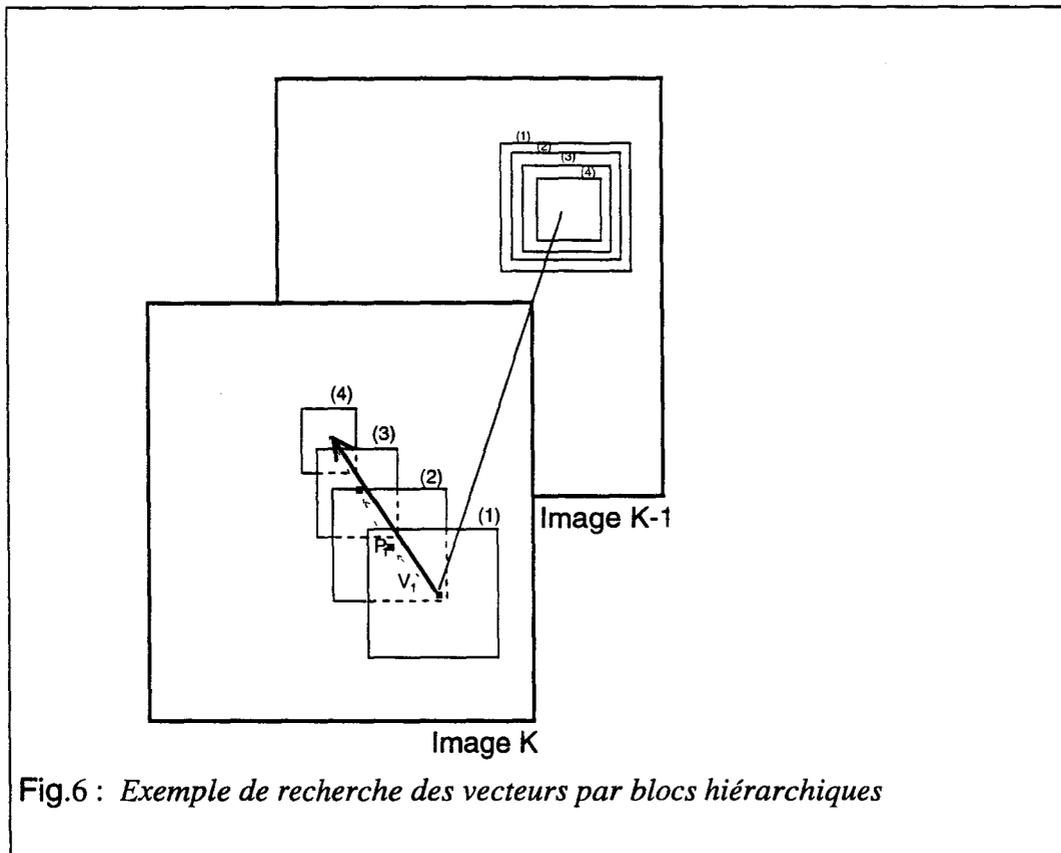


Fig.6 : *Exemple de recherche des vecteurs par blocs hiérarchiques*

4.1.4.Conclusion

La méthode du "Block Matching" est facile à mettre en oeuvre, et donne des résultats satisfaisants. De plus, les calculs sont facilement parallélisables. Cependant, suivant la taille des blocs, ceux-ci peuvent être visibles sur l'image.

L'algorithme fonctionne mal sur les structures homogènes ainsi que sur les zones d'ombre.

Cette méthode reste cependant tout à fait exploitable, et est d'ailleurs utilisée pour le visiophone.

Une autre méthode de mise en correspondance consiste à n'estimer les vecteurs déplacement que sur les points caractéristiques de l'image.

Dans le paragraphe suivant nous nous proposons de décrire brièvement les principales méthodes permettant d'extraire ces points caractéristiques.

4.2.METHODES UTILISANT L'EXTRACTION DE CONTOURS

Afin de diminuer les temps de calculs des algorithmes d'estimation de mouvement, de nombreux chercheurs ont essayé d'extraire les points caractéristiques des images et de n'estimer les vecteurs déplacement que sur ces points [Mon 90]

4.2.1.Détection de points caractéristiques

La première approche dans ce domaine de "segmentation" de l'image en zones d'intérêts, fut d'essayer d'extraire les points caractéristiques de l'image. D'où la naissance de plusieurs opérateurs simples détectant "l'activité" d'un pixel dont nous nous contenterons de ne citer que les plus connus.

L'opérateur de Moravec consiste à sommer pour chaque point de l'image sur une fenêtre centrée sur le point les carrés des différences de tous les couples de pixels voisins, horizontaux ou verticaux.

Le pixel sera considéré comme point caractéristique si le résultat de la somme obtenue est supérieur à un certain seuil.

L'opérateur de Hannah utilise les résultats de l'opérateur de Moravec en quatre points adjacents, et recherche la valeur minimale des ratios suivants : $Dv1/Dv3$, $Dv3/Dv1$, $Dv2/Dv4$, $Dv4/Dv2$ avec Dvi évaluation de Moravec au point i .

Cependant, ces opérateurs ne sont vraiment efficaces que sur des images binaires simples. L'étape suivante fut donc d'essayer d'estimer les contours des objets présents dans la scène.

4.2.2. Détection de contours

Elle s'effectue classiquement de deux manières :

- par le calcul des extrema locaux de la norme du gradient.
- par le passage à zéro du laplacien.

Il s'agit donc d'estimer le gradient ou le Laplacien.

Soit $G(I)$ le gradient d'un signal bruité I et $L(I)$ le Laplacien de ce même signal.

$G(I)$ et $L(I)$ se calculent de la manière suivante :

$$G(I) = (dI/dx_1, dI/dx_2, \dots, dI/dx_n) \quad (19)$$

$$L(I) = (d^2I/dx^2_1, d^2I/dx^2_2, \dots, d^2I/dx^2_n) \quad (20)$$

Un des premiers filtres permettant de détecter des contours dans des images bidimensionnelles fut introduit par Prewitt [Mon 90]

$$I = \begin{vmatrix} 1 & 1 \\ 1 & 1 \end{vmatrix}$$

$$\frac{dL}{dx} = \begin{vmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{vmatrix}$$

$$\frac{dL}{dy} = \begin{vmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{vmatrix}$$

Masque de Prewitt

Sobel introduisit le premier opérateur utilisant un filtrage séparable, c'est à dire décomposable en un filtre suivant X et un filtre suivant Y .

$$H1 = \begin{vmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{vmatrix}$$

$H1$: gradient en X ,

$$H2 = \begin{vmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{vmatrix}$$

$H2$: gradient en Y .

Après les calculs du Gradient ou du Laplacien, il faut ensuite effectuer un seuillage afin de déterminer les points de contours.

Il existe d'autres filtres plus élaborés tels que les filtres de Sehn, de Deriche ou des filtres gaussiens qui sont des filtres récurrents.

$$\text{filtre de Sehn} : s(x) = c \cdot e^{-\alpha |x|} \quad (21)$$

Le paramètre α définit la largeur du filtre ; plus il est petit, plus le lissage effectué par le filtre est important. c est choisi de manière à obtenir un filtre normalisé.

$$\text{filtre de Deriche} : s(x) = (c \cdot |x| + 1) \cdot e^{-\alpha |x|} \quad (22)$$

$$\text{filtre gaussien} : s(x) = c \cdot e^{-\alpha x^2} \quad (23)$$

Le problème majeur lorsque l'on se contente d'estimer les vecteurs déplacement des contours d'une image est qu'il est très difficile d'estimer le mouvement perpendiculaire à ces contours. Une solution est de rechercher non seulement les contours d'une image, mais surtout les points anguleux présents dans cette image.

4.2.3. Détection des points anguleux

D'autres opérateurs comme ceux de Zuniga-Haralick, Kitchen-Rosenfeld ou Deschler-Nagel sont basés sur la modélisation par un polynôme de la luminance et d'une estimation des coefficients par les moindres carrés en ce qui concerne les deux premiers ; et sur un développement de Taylor d'ordre 2 de la fonction luminance pour le dernier.

Ces algorithmes ne se contentent plus de retenir les points intéressants d'une image, mais on peut cette fois estimer les points anguleux de l'image.

4.2.4. Calcul des vecteurs déplacement

A partir de certains points caractéristiques de l'image, Vasquez et Mayora [VaM 91] proposent d'utiliser le filtre de Kalman afin de mettre à jour l'estimation des paramètres des vecteurs mouvement. Le filtre utilisé est un filtre de Sobel.

Bhanu et Burger [Bhanu 88] proposent de calculer les vecteurs déplacement d'une image par l'intermédiaire de la détermination de régions uniformes. Mais l'algorithme décrit ne fonctionne correctement que sur des images ayant des contours simples.

4.2.5.Conclusion

Ces méthodes donnent de bons résultats au point de vue de l'estimation de mouvement mais cela suppose une bonne détermination des primitives de l'image. Or la complexité des calculs a tendance à croître avec le nombre de primitives à extraire. Ces méthodes semblent donc assez mal appropriées aux images de télévision.

4.3.TRANSFORMEE DE HOUGH

La transformée de Hough peut être classée parmi les méthodes de mise en correspondance car elle constitue un moyen efficace pour détecter des formes et les mettre en correspondance d'une image à la suivante ; et cela même si ces objets ont subi au cours du temps des déformations géométriques telles que des homothéties ou des rotations.[KrC 87] [Lavin 86] [Lopez 88] [Tsai 90] [Zhan 90].

4.3.1.Principe

La transformée de Hough consiste à passer d'un espace image à un autre appelé " espace des paramètres ". Elle est issue de la reconnaissance des formes et est généralement utilisée pour détecter des droites dans une image simple. On peut aussi s'en servir pour caractériser des courbes, mais son emploi devient alors plus complexe

Si la forme à détecter est représentée par N paramètres, la transformée de Hough associée sera de dimension N.

4.3.2.Méthode de caractérisation de droite

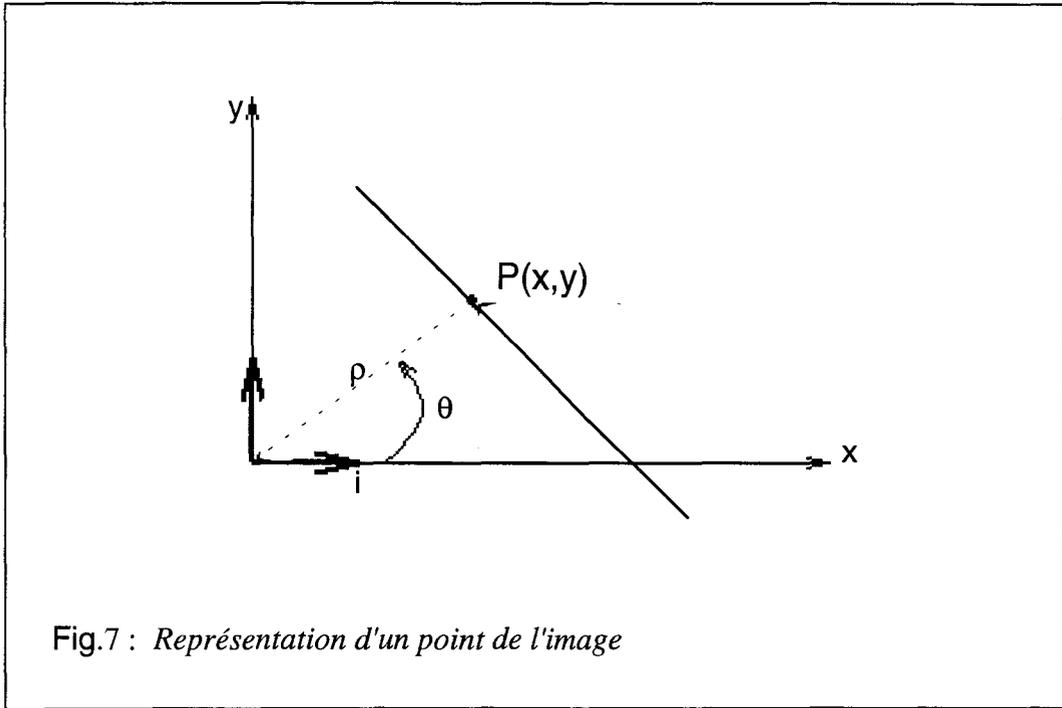
La transformée de Hough travaille sur les points caractéristiques de l'image (contours, coins). Leur extraction constitue donc une étape importante du calcul.

Elle permet, à partir de ces points, de déterminer les caractéristiques des droites présentes dans l'image.

Chaque ligne droite peut être décrite par un vecteur unitaire $a(\rho, \theta)$ (Fig.7).

θ est l'angle que fait a avec l'axe horizontal ; ρ représente la distance entre l'origine et la droite à caractériser.

Tous les points caractéristiques appartenant à la même droite seront repérés par le même angle θ (Fig.7).

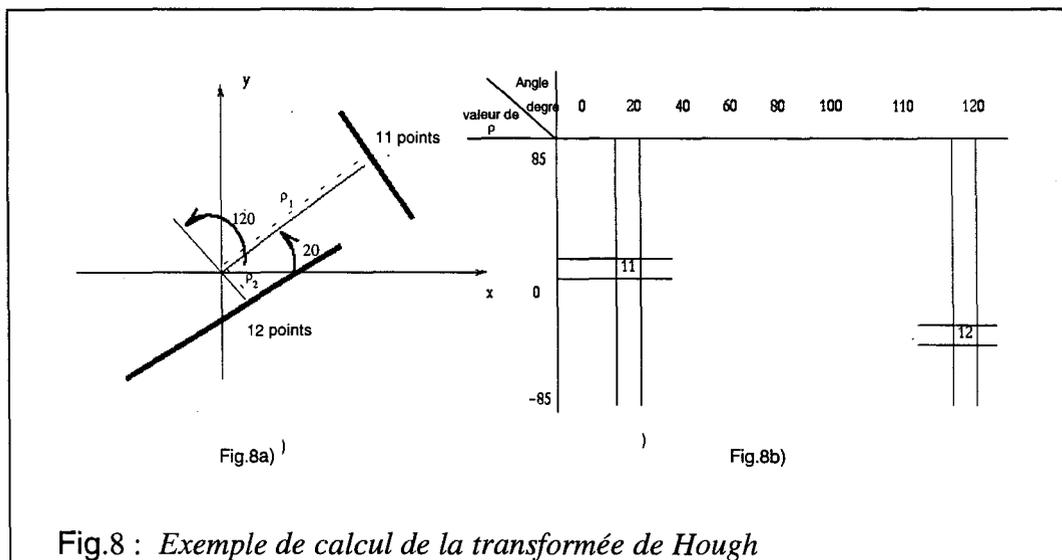


Les points sont donc décrit par l'équation suivante :

$$x \cos \theta + y \sin \theta = \rho \quad (24)$$

Un compteur est incrémenté chaque fois qu'un point $P(x,y)$ est sur la droite paramétrée en (ρ, θ) . La transformée de Hough est donc représentée par un tableau $H(\rho, \theta)$.

La figure 8 représente une image pour laquelle on recherche le nombre de droites présentes (Fig.8a) et la transformée de Hough associée (Fig.8b).



Le schéma de la figure 8a) représente de deux droites à caractériser, l'une constituée de 11 points et faisant un angle de 20 degrés avec l'horizontale, l'autre constituée de 12 points et faisant un angle de 120 degrés avec l'horizontale.

On fait varier θ par pas de 10 degrés, par exemple, on calcule le module puis on incrémente les éléments du tableau $H(\rho, \theta)$ (Fig.8b). On obtient bien la valeur 11 pour un angle de 20 degrés et la valeur 12 pour un angle de 120 degrés, ce qui correspond aux deux droites à caractériser.

L'étape suivante consiste ensuite à estimer le déplacement de ces droites.

4.3.3. Recherche des valeurs de différents paramètres de mouvement

Soit un mouvement décrit par les paramètres s caractérisant un changement d'échelle, par ψ angle de rotation et par t décrivant la translation.

Si l'on désire estimer un tel déplacement entre deux images, il faudra calculer les paramètres s , t , ψ tels que si l'on translate les valeurs d'une de deux transformées suivant l'axe θ ou ρ (suivant le paramètre considéré), on retrouve la seconde transformée de Hough.

En pratique, on recherche en premier lieu la valeur de l'angle de rotation. Pour cela, on translate le tableau $H(\rho, \theta)$ suivant l'axe θ de toutes les valeurs de ψ possibles.

L'angle ψ qui donnera une valeur maximale en terme de corrélation sera considéré comme angle de rotation entre les deux images. Pour cette valeur de ψ , on recherche ensuite les valeurs de t et de s .

Ce calcul est très long et assez hasardeux. De plus l'utilisation de la transformée de Hough au calcul des vecteurs déplacement n'a été utilisée que dans le cas d'images simples où un seul objet, de forme connue, est en mouvement [KriC87].

4.3.4. Conclusion :

La transformée de Hough possède de nombreux avantages. Elle présente une bonne immunité au bruit, elle est insensible aux occlusions partielles et est applicable dans le cas de rotations et d'éloignement.

Mais l'estimation des vecteurs déplacement par le biais de la transformée de Hough dépend essentiellement de l'extraction des points caractéristiques de l'image. L'emploi de cette transformée en vue d'estimer les vecteurs d'une image n'est donc applicable que dans le cas d'images simples.

Les méthodes de calcul des vecteurs déplacement décrites jusqu'à présent s'effectuaient dans le domaine temporel, donc directement sur l'image d'origine.

Notre étude est surtout axée sur les méthodes d'estimation de mouvement par transformée d'images que nous développerons dans le chapitre suivant.

5.METHODES PAR TRANSFORMEE

Le calcul du vecteur ne se fait plus dans l'espace temporel c'est à dire sur l'image originale, mais à partir d'une (ou plusieurs) image(s) modifiée(s) par transformation.

5.1.Méthodes utilisant les propriétés de la transformée de Fourier

5.1.1.Introduction

Cette méthode, assez ancienne, est basée sur la propriété suivante : une translation dans le domaine temporel correspond à un déplacement de phase dans le domaine fréquentiel.[BalB 89a] [BalB 89b] [Castro 87]

Il suffit donc de calculer les transformées de Fourier de l'image à l'instant (t) et à l'instant (t + 1) et à estimer le déplacement de phase pour en déduire le déplacement entre (t) et (t + 1).

5.1.2.Théorie

Si g est l'image de f translatée, on a :

$$g(x,y) = f(x+dx, y+dy). \quad (25)$$

La transformée de Fourier de g(x,y), notée G(u,v) est égale à :

$$G(u,v) = F(u,v) \exp [- 2 \Pi j (u * dx + v * dy)] \quad (26)$$

avec F(u,v), transformée de Fourier de f(x,y).

d'où une différence de phase de :

$$- 2 \Pi j (u * dx + v * dy). \quad (27)$$

5.1.3. Conclusion

Une première difficulté de cette méthode résulte de l'indétermination de la phase à $2k\pi$ près, ce qui peut introduire des erreurs.

De plus cette méthode n'est applicable que dans le cas de mouvements simples.

5.2. Corrélation de phase

5.2.1. Théorie

Si l'on considère F et G les transformées de Fourier bidimensionnelles de deux images successives $f(x,y)$ et $g(x,y)$; pour chaque fréquence spatiale on a :

$$TF [f(x,y)**g(x,y)] = F(u,v).G(u,v)^* = |F(u,v).G(u,v)^*| \exp(j\theta). \quad (28)$$

avec :

TF : transformée de Fourier.

"**" : opérateur de convolution.

"*" : opérateur complexe conjugué.

On a donc :

$$\exp(j\theta) = \frac{F(u,v).G(u,v)^*}{|F(u,v).G(u,v)^*|} \quad (29)$$

La corrélation de phase $P(d)$ se calcule de la manière suivante :

$$P(d) = F^{-1} \left\{ \frac{F(u,v).G(u,v)^*}{|F(u,v).G(u,v)^*|} \right\} \quad (30)$$

avec : F^{-1} : transformée de Fourier inverse.

5.2.2.Calculs

Le calcul s'effectue sur des blocs en vis-à-vis de taille égale à une puissance de 2 afin d'utiliser l'algorithme de FFT rapide (Fast Fourier Transform). La grandeur des blocs doit être assez importante pour que la FFT soit représentative du signal ; mais assez faible pour ne contenir qu'un nombre limité d'objets en déplacement à l'intérieur du bloc de calcul.

En général, la corrélation de phase se calcule sur des blocs (64*64) à travers une fenêtre de Hamming. On obtient ainsi un tableau (ou surface de corrélation) dont les pics (valeurs maximales) permettent de déterminer le ou les vecteur(s) déplacement présents dans le bloc de calcul.

Cette transformée est comparable à la fonction de corrélation, mais les pics sont plus distincts, ce qui permet d'obtenir des résultats plus exploitables.[Grotze]

La figure 9 permet de comparer les résultats obtenus avec la fonction de corrélation et avec la corrélation de phase.

La figure 9a) représente trois objets en mouvement sur un fond constant. La figure 9b) est la fonction de corrélation entre deux images successives, et la figure 9c) la corrélation de phase entre ces deux même images. On remarque que les pics sont bien distincts dans la figure 9c), ce qui n'est pas le cas de la figure 9b).

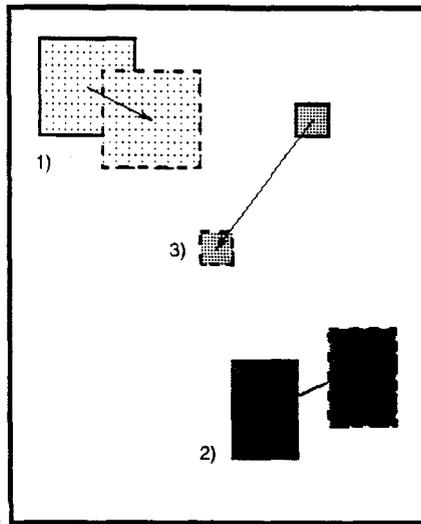


Fig.9a)

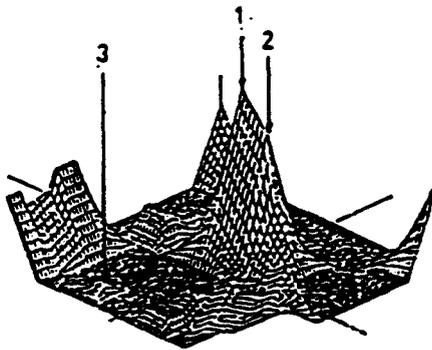


Fig.9b) : *fonction de corrélation*

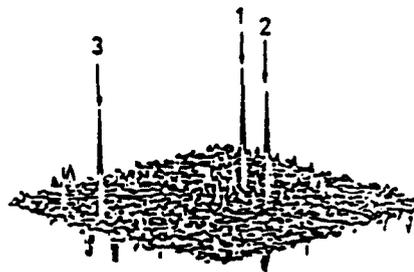


Fig.9c) : *corrélation de phase.*

Fig.9 : *Comparaison entre la corrélation de phase et la fonction de corrélation*

5.2.3. Interprétation de la surface de corrélation

La distance d'un pic à l'origine représente le vecteur déplacement de l'objet associé à ce pic. On prendra comme origine le centre du bloc. La hauteur du pic est proportionnelle à la taille de l'objet : plus un objet sera important, plus son pic sera élevé.

Le décalage du pic par rapport à l'origine permet d'évaluer le vecteur déplacement du pic caractérisant un objet.

La surface de corrélation est représentée de la manière décrite par la figure 10.

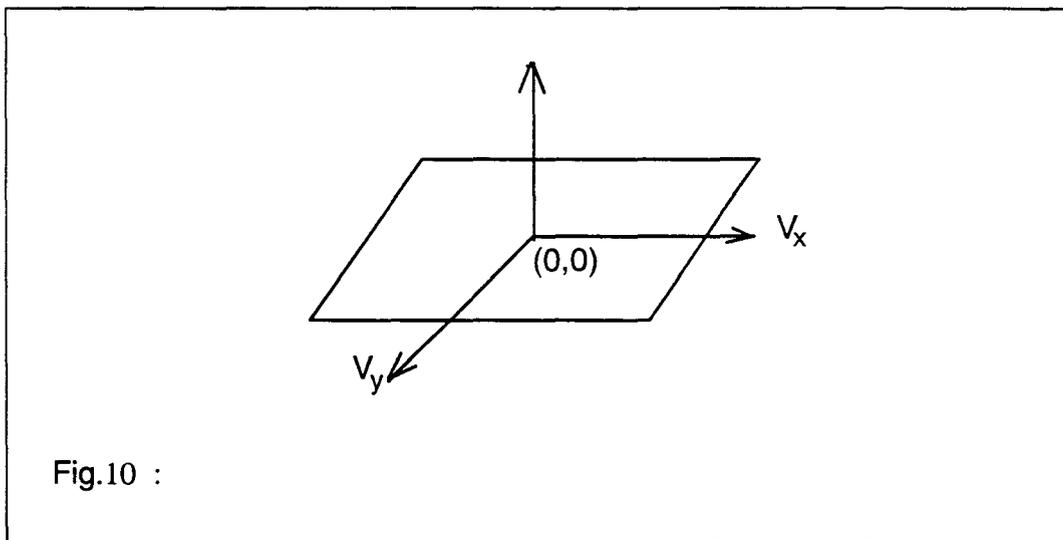


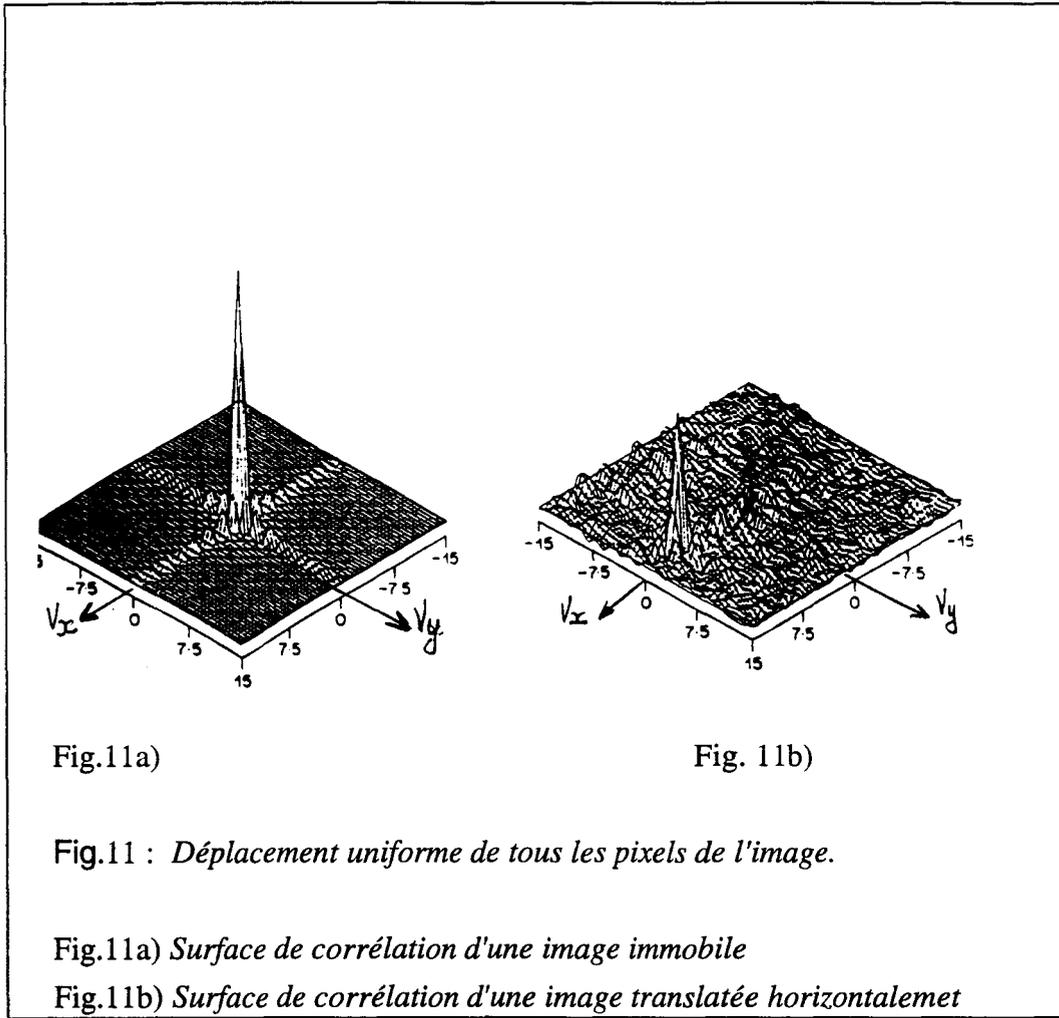
Fig.10 :

Dans son article Thomas [Thom 87] décrit le comportement de la corrélation de phase suivant différents types de mouvements.

Nous nous proposons de reprendre les exemples les plus significatifs afin de mieux comprendre de quelle manière la surface de corrélation est modifiée suivant le type de déplacement.

La surface de corrélation d'une image fixe soit donc deux fois la même image est un pic unique de coordonnées (0,0). La position de ce pic en (0,0) signifie qu'il n'y a pas de déplacement entre deux images consécutives ((Fig 11a)).

La surface de corrélation d'une image traduite de (V_x, V_y) est un pic se trouvant à la position (V_x, V_y) par rapport à l'origine. Il est à noter que dans ce cas, tous les pixels de l'image ont le même mouvement uniforme.

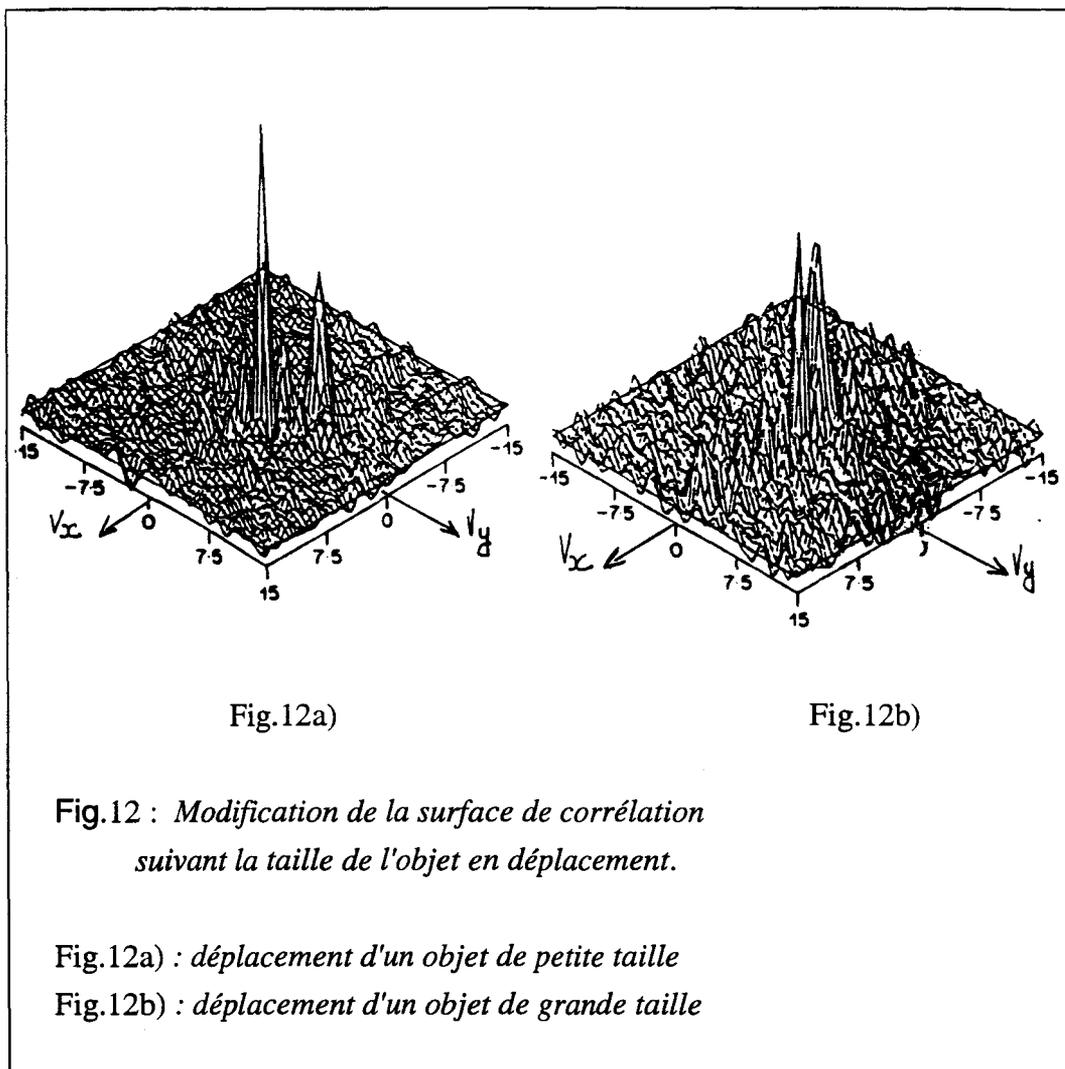


Observons le cas où un objet est en translation sur un fond fixe.

La surface de corrélation est représentée par deux pics ; l'un en (0,0) pour le fond fixe, l'autre décalé de (V_x, V_y) par rapport à l'origine (Fig.12).

La taille du second pic est proportionnelle à la taille de l'objet par rapport au fond.

La figure 12a) représente une surface de corrélation où l'objet, caractérisé par le pic n°3 est de petite taille par rapport au fond, pic 1, tandis que pour la figure 12b) la taille de l'objet, pic n°2 en mouvement est aussi importante que celle du fond, pic n°1. De plus un troisième pic apparaît (surtout visible sur la figure 12a)) caractérisant les zones de découvrment et de recouvrement.



On peut donc ainsi déterminer les vecteurs déplacement du bloc.

Cependant, nous n'avons pris comme exemple que des cas très simples.

Dans le cas d'images réelles (de télévision par exemple) nous aurons une multitude de pics, étant donné le nombre d'objets en mouvement dans le bloc.

Comme on ne peut sélectionner tous les pics comme vecteur déplacement seuls les plus importants seront assimilés à des déplacements (les autres pouvant être dus au bruit ou aux occlusions). En général, on ne conservera que trois ou quatre déplacements par bloc.

Après avoir sélectionné les pics les plus représentatifs, il reste à déterminer leur point d'application.

5.2.4. Recherche des coordonnées (x,y) du vecteur déplacement

Une solution consiste à translater le bloc (64*64) de la valeur des vecteurs déplacement trouvée, puis à calculer la différence en valeur absolue entre chaque pixel des deux blocs. Pour chaque vecteur on calcule en quelque sorte une surface d'erreur. Le ou les pixels correspondant au minimum d'erreur déterminent le point d'application du vecteur déplacement. Les régions qui ne donnent pas une erreur faible correspondent soit à un mauvais calcul de vecteur déplacement, soit à une zone de recouvrement ou de découverture d'objets.

De plus, si un objet "traverse" le bloc on ne peut estimer son vecteur déplacement. Pour y remédier, la corrélation de phase s'effectue sur des blocs de taille (64*64) avec recouvrement de 16 pixels par lignes.

5.2.5. Conclusion

La corrélation de phase est une méthode qui possède de nombreux avantages tels que une bonne immunité au bruit et la possibilité de détecter de larges déplacements ; de plus, les valeurs trouvées semblent tout à fait correctes.

Mais son principal défaut est que l'on ne connaît pas le point d'application des vecteurs déplacement.

Un autre problème est la différenciation des pics les uns par rapport aux autres et l'estimation correcte de leur position par rapport à l'origine. Par exemple, si un objet de faible taille se déplace peu par rapport à l'origine, son pic sera quasiment confondu avec celui du fond, il sera alors très difficile de calculer le vecteur déplacement de cet objet. Plus le bruit est important, plus la taille des pics diminue.

Cette méthode semble être tout à fait adaptée aux cas des images de télévision, et notamment en ce qui concerne les problèmes de conversions. [FeP 87] [Grotze] [Thom 90]

Les deux méthodes décrites précédemment permettent de calculer des vecteurs déplacement, mais sans savoir exactement où ces derniers s'appliquent.

Les transformées temps-fréquence permettent de résoudre ce problème dans la mesure où elle conservent la notion de temps.

5.3. Transformées temps-fréquence

5.3.1. Introduction

Si on calcule la transformée de Fourier, notée $F(\omega_x, \omega_y, \omega_t)$, d'une image en translation constante, notée $f(x, y, t)$; on peut en déduire l'amplitude et le sens du mouvement [JaW 86]

La démonstration est assez simple :

Si on écrit la formule de la transformée de Fourier $F(\omega_x, \omega_y, \omega_t)$, on a :

$$F(\omega_x, \omega_y, \omega_t) = \iiint_{dx, dy, dt=-\infty}^{\infty} f(x, y, t) \exp[-j(\omega_x x + \omega_y y + \omega_t t)] dx dy dt \quad (31)$$

Pour une valeur fixée de $\omega_x = \omega_{x0}$, $\omega_y = \omega_{y0}$, et $\omega_t = \omega_{t0}$, $(\omega_{x0}, \omega_{y0}, \omega_{t0})$ représente une raie du spectre et s'écrit :

$$F(\omega_{x0}, \omega_{y0}, \omega_{t0}) = \delta(\omega_x - \omega_{x0}, \omega_y - \omega_{y0}, \omega_t - \omega_{t0}) \quad (32)$$

A cette raie correspond donc une onde sinusoïdale qui se propage dans le plan (x, y) .

Nous avons représenté sur la figure 13, le plan de phase ainsi que les droites correspondant à des lieux de phase constante à un instant donné.

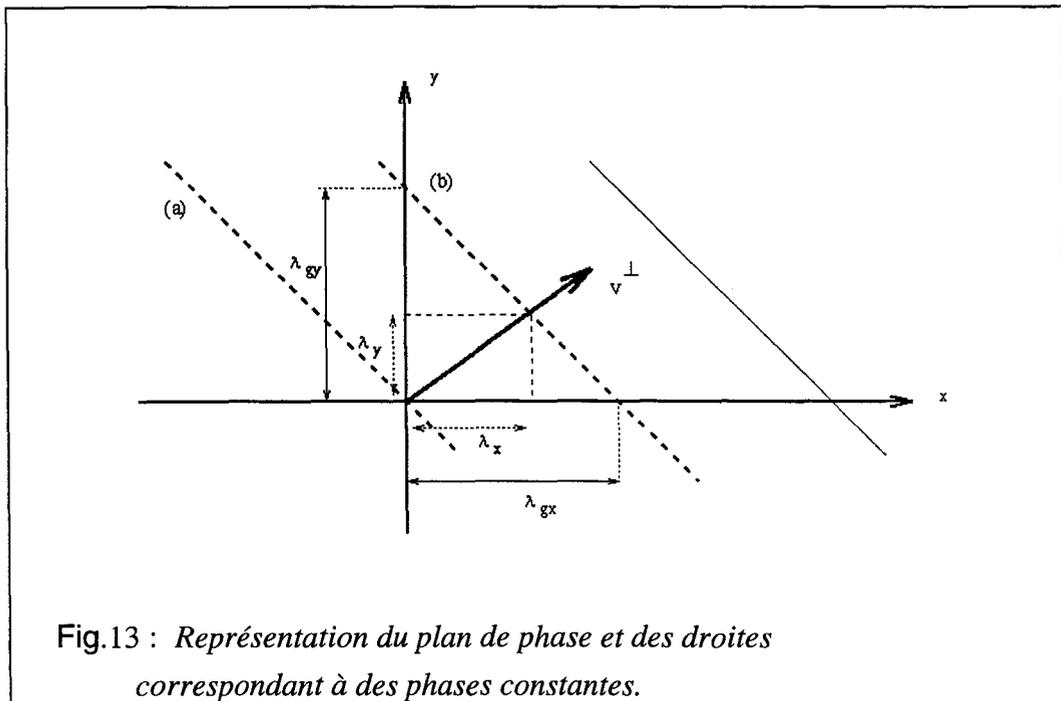


Fig.13 : Représentation du plan de phase et des droites correspondant à des phases constantes.

La droite (a) est le lieu des points où la phase est nulle ; elle passe par le point 0 et s'écrit :

$$\omega_{x0}x + \omega_{y0}y + \omega_{t0}t = 0 \quad (33).$$

La droite (b) est le lieu des points où la phase vaut 2π .

Son équation est alors égale à : $\omega_{x0}x + \omega_{y0}y = 2\pi$. (34)

On a $\omega_{t0} = 2\pi f$ avec f fréquence de l'onde.

Une onde sinusoïdale se déplace à une vitesse v égale à $f \cdot \lambda$.

Soit :

$$v = \lambda \cdot \omega_{t0} / 2\pi \quad (35)$$

A partir de la valeur de la longueur d'onde, on peut en déduire le vecteur vitesse.

Comme on peut le voir sur la figure 13 les valeurs de λ_x et de λ_y se calculent en recherchant le point d'intersection entre la droite (b) et l'axe (Oz), perpendiculaire à (b).

L'équation de cet axe est donnée par :

$$y - (\omega_{y0} / \omega_{x0}) \cdot x = 0 \quad (36)$$

On obtient alors le système suivant :

$$(37) \quad \begin{cases} \omega_{x0}x + \omega_{y0}y = 2\pi \\ y - \frac{\omega_{y0}}{\omega_{x0}}x = 0 \end{cases}$$

De la résolution de ce système on extrait les valeurs de x et y , soit de λ_x et de λ_y .

$$x = \lambda_x = \frac{2\pi\omega_{x0}}{\omega_{x0}^2 + \omega_{y0}^2} \quad \text{et} \quad y = \lambda_y = \frac{2\pi\omega_{y0}}{\omega_{x0}^2 + \omega_{y0}^2} \quad (38)$$

Le front d'onde se déplace à la vitesse v telle que : $v = \lambda \cdot \omega_{t0} / 2\pi$. (39)

On obtient ainsi les valeurs de v_x et de v_y à savoir :

$$v_x^\perp = \frac{\omega_{t0}\omega_{x0}}{\omega_{x0}^2 + \omega_{y0}^2} \quad \text{et} \quad v_y^\perp = \frac{\omega_{t0}\omega_{y0}}{\omega_{x0}^2 + \omega_{y0}^2} \quad (40)$$

On obtient donc bien le vecteur déplacement, mais les résultats obtenus après le calcul de la transformée de Fourier dépendent fortement de la fenêtre utilisée.

Afin de s'affranchir de cette dépendance, on peut employer la transformée de Wigner_Ville.

5.3.2. Utilisation de la transformée de Wigner-Ville

Le spectre de la transformée de Wigner-Ville tridimensionnelle, notée DWV3D, associée à un point d'une image en translation constante est nul partout sauf sur un plan dont l'orientation permet de déterminer le vecteur déplacement du pixel de l'image.

La distribution de Wigner-Ville est relativement simple à mettre en oeuvre et ses propriétés laissent à supposer que l'on peut estimer facilement les vecteurs déplacement d'une séquence.

Nous allons donc décrire plus en détails comment on peut estimer le mouvement d'une séquence d'images à partir de la transformée de Wigner-Ville tridimensionnelle (DWV 3D)

Transformée de Wigner-Ville et estimation de mouvement :

Si l'on considère une séquence d'images animées, pour chaque pixel de l'image, la distribution de Wigner se calcule de la manière suivante :

$$W_f (x, y, t, \omega_x, \omega_y, \omega_t) = \iiint_{\alpha, \beta, \delta = -\infty}^{\infty} R_f (x, y, t, \alpha, \beta, \delta) \exp [-j (\alpha \omega_x + \beta \omega_y + \delta \omega_t)] d\alpha d\beta d\delta \quad (41)$$

avec

$$R_f (x, y, t, \omega_x, \omega_y, \omega_t) = f (x + \alpha / 2, y + \beta / 2, t + \delta / 2) f^* (x - \alpha / 2, y - \beta / 2, t - \delta / 2)$$

On peut aussi exprimer la transformée de Wigner-Ville par la transformée de Fourier.

$$Wf (x, y, t, \omega_x, \omega_y, \omega_t) = 1 / 8\pi \iiint_{\eta, \xi, \epsilon = -\infty}^{\infty} S_f (\eta, \xi, \epsilon, \omega_x, \omega_y, \omega_t) \exp [j (\eta x + \xi y + \epsilon t)] d\eta d\xi d\epsilon$$

$$S_f (\eta, \xi, \epsilon, \omega_x, \omega_y, \omega_t) = F (\omega_x + \eta / 2, \omega_y + \xi / 2, \omega_t + \epsilon / 2) F^* (\omega_x - \eta / 2, \omega_y - \xi / 2, \omega_t - \epsilon / 2) \quad (42)$$

La distribution de Wigner attribue donc à chaque pixel un spectre spatio-temporel 3D, alors qu'avec la transformée de Fourier on obtient un spectre pour l'image entière.

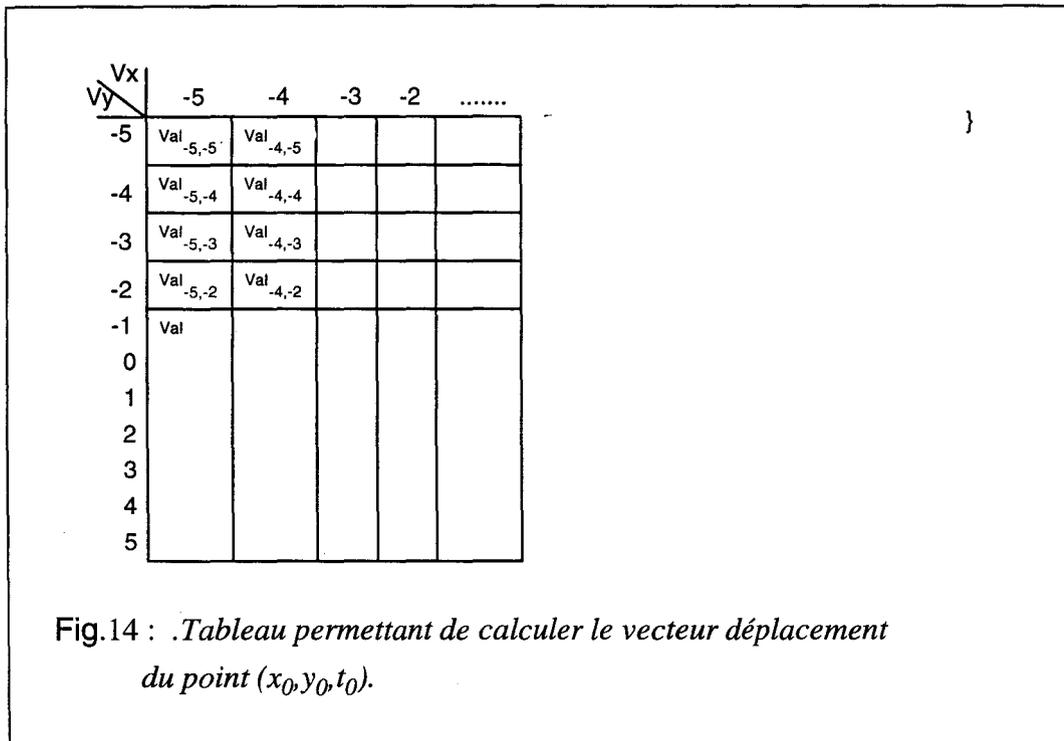
Grâce à la transformée de Wigner-Ville, on peut estimer un vecteur déplacement par pixel, et non plus sur l'image entière comme le font les méthodes utilisant la transformée de Fourier.

Tout comme le spectre de Fourier, celui de Wigner d'une image en translation constante de vitesse (V_{x0}, V_{y0}) est nul partout sauf sur un plan défini par :

$$\{\omega_x, \omega_y, \omega_t / V_{x0} \omega_x + V_{y0} \omega_y + V_{ot0} = 0\} \quad (43)$$

Si un pixel est en translation constante au cours du temps, on peut calculer son vecteur déplacement en procédant de la manière suivante :

Pour chaque couple (V_{x0}, V_{y0}) possible, on somme toutes les valeurs de la transformée de Wigner-Ville appartenant au plan décrit par V_{x0} et V_{y0} , et ce pour chaque pixel P. On obtient ainsi un tableau, que nous appellerons "Fonction test" ; dont la valeur maximale permet de déterminer la valeur du vecteur déplacement du pixel.P.



$$Val_{-5, -5} = \sum_{\omega_x, \omega_y, \omega_t} DWV(\omega_x, \omega_y, \omega_t) / -5\omega_x - 5\omega_y + \omega_t = 0 \quad (44)$$

En ce qui concerne le tableau, les valeurs de V_x et de V_y varient de -5 à 5 par pas de 1.

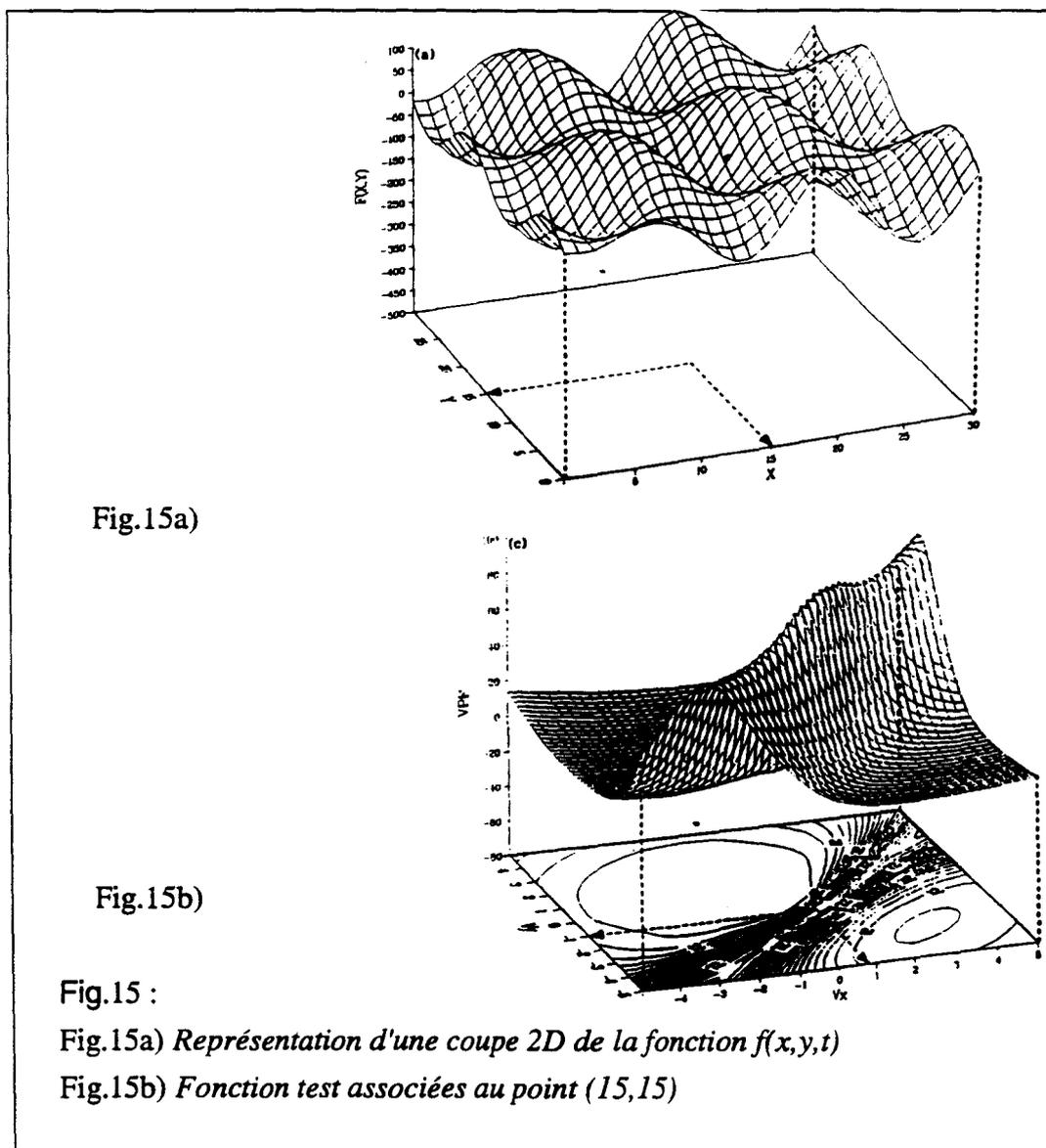
5.3.2.1. Exemple de représentation de la fonction test

Nous allons reprendre les résultats obtenus par Jacobson et Weschler.

Il est à noter que ces derniers ont utilisé leur algorithme sur une fonction variant au cours du temps et non sur une séquence d'images.

Soit donc la fonction $f(x,y,t)$ décrite de la manière suivante :

$$f(x,y,t) = \cos\left(\frac{2\pi}{5\sqrt{5}\sqrt{(0.75)^2 + (0.75)^2 + (-1.5)^2}} * (0.75x + 0.75y - 1.5t)\right) \quad (45)$$



En ce qui concerne la figure 15b), V_x varie de -5 à 5 ainsi que V_y ; valeurs qui sont fixées arbitrairement. Le maximum de la "Fonction test" est obtenu pour $V_x = 1$ et $V_y = 1$. La valeur du vecteur déplacement du pixel (15,15) est donc égale à (1,1).

5.4.Conclusion

L'atout majeur de la transformée de Wigner-Ville par rapport à la transformée de Fourier est que l'on ne perd pas l'information concernant la position du pixel ; on calcule le vecteur déplacement pour chaque pixel de l'image. De plus, la distribution de Wigner-Ville est moins sensible aux problèmes de fenêtrage que la transformée de Fourier.

Le principal inconvénient de cette méthode est le temps de calcul. En ce qui concerne la "Fonction test", par exemple, l'utilisateur fixe les différentes valeurs de V_x, V_y, V_t . Plus ces valeurs seront nombreuses, plus le calcul des vecteurs déplacement sera précis. Mais cela augmente considérablement les calculs.

6.CONCLUSION

Les méthodes différentielles se basent sur une ou plusieurs contraintes liant les dérivées spatiales et temporelles. Elles reposent sur l'hypothèse, restrictive, que la variation lumineuse d'un même point en mouvement est faible d'une image à l'autre. De plus les problèmes d'occlusion ne sont pas résolus de manière satisfaisante malgré l'introduction de contraintes supplémentaires. Celles-ci ont tendance soit à trop lisser les discontinuités, soit à être trop lourdes à implémenter.

L'utilisation d'une décomposition multirésolution présente l'avantage de pouvoir estimer les mouvements de grandes amplitudes ; cependant le problème du passage d'un niveau à un autre n'est pas résolu de manière satisfaisante.

Les méthodes de mises en correspondance ne peuvent donner de bons résultats que dans le cas où les objets en déplacement ne subissent pas de déformation. En effet, elles consistent à repérer le même objet d'une image à l'autre. Les calculs s'effectuent soit en minimisant un terme d'erreur, dans le cas du Block Matching, soit en repérant les points caractéristiques de l'image dont la complexité algorithmique de cette méthode croît exponentiellement avec le nombre de primitives à extraire.

La transformée de Hough possède une bonne immunité au bruit, peut traiter les problèmes d'occlusion partielle. Cependant, elle repose sur l'extraction des points caractéristiques. Elle ne peut s'appliquer que sur des images aux contours simples, car le traitement des lignes courbes augmente considérablement les calculs.

La corrélation de phase permet une bonne précision sub-pixel. De plus elle présente une bonne immunité au bruit et ne nécessite pas de prétraitement sur l'image ; par contre elle fait

défaut lorsque les déplacements à estimer sont faibles. Il est alors très difficile de séparer les pics les uns des autres. Un autre inconvénient de cette méthode est que l'on ne connaît pas le point d'application du vecteur trouvé.

La transformation en ondelette permet de déterminer rapidement les contours verticaux et horizontaux de l'image. Elle est cependant rarement utilisée pour le calcul des vecteurs déplacements ; son application principale reste la compression.

La distribution de Wigner permet le calcul direct du vecteur en chaque point. De par sa définition, elle tient compte des pixels adjacents au pixel considéré. Son principal inconvénient est le temps de calcul élevé.

Nous avons regroupé dans le tableau suivant les principaux avantages et inconvénients des différentes méthodes citées afin de faciliter leur comparaison.

METHODES	AVANTAGES	INCONVENIENTS
Gradient	Simple	Lourd en calculs Traitement des discontinuités Suppose luminance constante
Méthodes probabilistes	Pas de constance de luminance Traite mouvements complexes	Calculs des paramètres.
Multirésolution	Mouvements de forte amplitude	Correspondance entre niveaux Changement d'échelle
Block Matching	Facile à mettre en oeuvre	Blocs visibles
Hough	Bonne immunité au bruit Possible si occlusion partielle	Calculs des points caractéristiques Long en calculs Calcul possible sur images simples
Corrélation de phase	Précision sub-pixel Pas de prétraitement sur l'image	Mouvement faible de petits objets Point d'application du vecteur non connu
Wigner-Ville	Point d'application du vecteur Utilisation des pixels adjacents Peu de prétraitement sur l'image	Temps de calculs

Notre choix d'étude s'est porté sur l'utilisation de la transformée de Wigner-Ville car cette méthode, originale, permet de calculer directement le vecteur déplacement de n'importe quel pixel, et ce en prenant en compte son voisinage proche.

CHAPITRE II

TRANSFORMEE DE WIGNER-VILLE : THEORIE

La transformée de Fourier est souvent utilisée pour l'analyse des signaux, mais elle ne fournit qu'une information moyennée sur la durée d'observation, et ne peut rendre avec exactitude les caractéristiques instantanées du signal.

Une première solution fut d'étudier les signaux non stationnaires à l'aide de plusieurs transformées de Fourier calculées sur une fenêtre d'observation adéquate. Malheureusement ce système se heurte au compromis entre la résolution fréquentielle et temporelle. Il s'avéra donc nécessaire de trouver une nouvelle méthode non directement issue de celles existants pour l'analyse des signaux stationnaires. Wigner par le biais de la mécanique quantique, et Ville par la théorie du signal définirent la transformée de Wigner-Ville.

Cette transformée semble beaucoup mieux adaptée aux calculs temps-fréquence des signaux non stationnaires (pas de permanence temporelle du contenu spectral du signal). Elle s'apparente à une répartition d'énergie du signal dans le plan temps-fréquence.

Mais malgré les avantages qu'elle présente, elle n'en reste pas moins un outil de traitement du signal dont l'interprétation reste délicate. En effet plusieurs problèmes interviennent : la transformée de Wigner-Ville peut prendre des valeurs négatives, interdisant toute interprétation locale comme densité énergétique. De plus, la création de termes d'interactions pose des problèmes lors de l'analyse de cette fonction.

Les propriétés de la transformée de Wigner-Ville étant assez proches de celle de la transformée de Fourier, nous nous contenterons de les citer brièvement dans le cas monodimensionnel, ainsi que les caractéristiques de la transformée de Wigner-Ville liés à la discrétisation du signal.

Nous expliquerons la façon de transposer les calculs de la transformée de Wigner-Ville monodimensionnelle, que nous noterons DWV1D, au calcul de la transformée de Wigner-Ville tridimensionnelle, notée DWV3D.

Dans le dernier chapitre, nous expliquerons de quelle manière on peut calculer le vecteur déplacement d'un pixel d'une image à partir de la transformée de Wigner-Ville tridimensionnelle.

1.CAS MONODIMENSIONNEL

1.1.DEFINITION ET PROPRIÉTÉS

1.1.1.Définition

La transformée de Wigner-Ville, $W_f(t,u)$, d'une fonction $f(t)$ s'écrit de la manière suivante

$$W_f(t,u) = \int_{-\infty}^{+\infty} f(t + \tau/2) f^*(t - \tau/2) \exp(-2i\pi u\tau) d\tau \quad (1)$$

avec

"*" : opérateur de conjugaison complexe.

La transformée de Wigner-Ville de la fonction $f(t)$ dépend donc de deux variables u et t .

Elle peut aussi s'exprimer à l'aide de la transformée de Fourier :

$$W_f(t,v) = \int_{-\infty}^{+\infty} F(v + \eta/2) F^*(v - \eta/2) \exp(2i\pi\eta t) d\eta \quad (2)$$

avec

F : transformée de Fourier de $f(t)$.

1.1.2.Propriétés

Dans ce chapitre, nous nous proposons de passer en revue les principales caractéristiques de la transformée de Wigner-Ville.

Afin de simplifier les écritures, nous noterons $W_f(t,u)$ la transformée de Wigner-Ville de la fonction $f(t)$.

0) La transformée de Wigner-Ville est réversible à une constante près.

$$\int W_f(t/2,u) \exp(2iu\pi t) du = f(t)f^*(0)$$

de même en fréquentiel :

$$\int W_f(t,u/2) \exp(-2iu\pi t) = F(u)F^*(0)$$

1) La transformée de Wigner-Ville est une fonction réelle.

Les projections de cette transformée respectivement sur l'axe des temps et l'axe des fréquences donnent les valeurs instantanées de la puissance et de la densité spectrale.

$$\int W_f(t, u) \cdot du = |f(t)|^2$$

$$\int W_f(t, u) \cdot dt = |F(u)|^2$$

2) La transformée d'une fonction translatée dans le temps en temporel donne une fonction translatée selon la variable t.

$$W_f(t-t_0) \text{ est la TWV de } f(t-t_0)$$

3) La transformée d'une fonction paire est une fonction paire par rapport à la variable fréquentielle.

$$\text{Si } f(t) = f(-t), \text{ alors } W_f(t, u) = W_f(t, -u).$$

4) On obtient une translation selon u en multipliant f(t) par $\exp(2i\pi ut)$.

$$W_f(t-t_0) \text{ est la transformée de Wigner-Ville de } f(t) \exp(2i\pi u t)$$

5) La transformée de Wigner-Ville du produit (respectivement du produit de convolution) de deux signaux est égale au produit de convolution de la transformée de Wigner-Ville des deux signaux par rapport à la fréquence (respectivement par rapport au temps)

$$W_{f \cdot g}(t, u) = \int W_f(t, u) \cdot W_g(t, u-v) dv$$

$$W_{f * g}(t, u) = \int W_f(\tau, u) \cdot W_g(t-\tau, u) d\tau$$

6) La transformée de Wigner-Ville conserve les support temporels et fréquentiels :

$$\text{si } |f(t)| = 0 \text{ pour } |t| > T \text{ alors } W_f(t, -u) = 0 \text{ pour } |t| > T$$

$$\text{si } |X(u)| = 0 \text{ pour } |u| > B \text{ alors } W_f(t, -u) = 0 \text{ pour } |u| > B$$

7) La transformée de f^* , fonction complexe conjuguée de f(t) est égale à la transformée de f(t) à la fréquence -u :

$$TWV[f^*(t)] = W_f(t, -u).$$

$$8) TWV[f(-t)] = W_f(-t, -u).$$

$$9) TWV[f(at)] = (1/|a|) \cdot W_f(at, u/a).$$

Il existe d'autres propriétés spécifiques à la transformée de Wigner-Ville mais qui rendent son interprétation difficile, notamment le fait qu'elle puisse prendre des valeurs négatives et que sa structure bilinéaire crée des phénomènes d'interférences.

$$9) TWV[f(at)] = (1/|a|).Wf(at, u/a).$$

Il existe d'autres propriétés spécifiques à la transformée de Wigner-Ville mais qui rendent son interprétation difficile, notamment le fait qu'elle puisse prendre des valeurs négatives et que sa structure bilinéaire crée des phénomènes d'interférences.

1.2. TRANSFORMÉE DE WIGNER-VILLE DISCRÈTE (TWVD)

La transformée de Wigner-Ville de la fonction échantillonnée, f_{ech} , est égale à :

$$W_{f_{ech}}(t, u) = \sum_{n=-\infty}^{+\infty} \sum_{k=-\infty}^{+\infty} f(kT) f^*((n-k)T) \cdot \exp(-2iu\pi(2k-n)T) \cdot \delta(t - nT/2) \quad (3)$$

Si le signal d'origine est échantillonné à un pas ΔT , la transformée de Wigner-Ville de ce même signal sera, elle, échantillonnée à un pas $\Delta T/2$; c'est à dire à une période deux fois plus petite que la période de répétition du spectre du signal discret original. Cette caractéristique de périodicité de la TWV est à l'origine de tous les recouvrements éventuellement présents dans la TWV calculée à partir du signal temporel réel.

Donc, contrairement à la transformée de Fourier discrète (TFD) simple, dans laquelle il suffit d'échantillonner le signal temporel à la fréquence de Shannon, l'utilisation de la TWVD nécessite un échantillonnage du signal réel à une fréquence au moins deux fois plus grande que celle de Shannon.

1.3. CALCULS DU SURÉCHANTILLONNAGE DU SIGNAL

En pratique, pour obtenir une TWVD ne souffrant pas de recouvrement, il existe trois solutions :

- suréchantillonner le signal temporel réel $f(t)$ à une fréquence d'échantillonnage au moins deux fois plus grande que celle de Shannon.
- introduire le signal analytique associé au signal réel temporel $f(t)$ et calculer la TWV de ce signal analytique ; le signal réel peut alors être échantillonné à la fréquence de Shannon.
- interpoler le signal temporel. Dans ce cas, il suffit d'échantillonner le signal réel de départ à la fréquence de Shannon.

L'utilisation du signal analytique permet de supprimer une grande partie des termes apparaissant lors d'éventuels recouvrements de spectre, nous allons décrire le moyen de la calculer.

1.3.1. Calcul du Signal analytique

Il est souvent plus avantageux de prendre la TWV du signal analytique associé au signal réel de départ que celle du signal réel pour les raisons suivantes :

- L'introduction du signal analytique dans le calcul de la TWV permet d'enlever l'exigence de suréchantillonnage du signal temporel réel.

- La TWV d'un signal analytique (qui permet de supprimer les fréquences négatives) ne présente pas de termes d'interaction entre les fréquences positives et les fréquences négatives, ce qui rend la TWV plus simple et plus facile à interpréter.

- La TWV du signal analytique conserve toutes les informations utiles contenues dans la TWV du signal réel correspondant.

- La TWV d'un signal réel est une fonction paire par rapport à la variable fréquence. La redondance d'informations présentes dans la TWV du signal réel à cause de cette symétrie est éliminée, si l'on introduit le signal analytique dans la TWV.

Un signal analytique est un signal complexe dont la partie réelle et la partie complexe sont chacune transformée de Hilbert l'une de l'autre. Le signal analytique $z(t)$ associé au signal réel $f(t)$ est défini par :

$$z(t) = f(t) + i H(f(t)) \quad (4)$$

où H représente la transformée de Hilbert définie par :

$$H(t) = \frac{1}{\pi} \int_{-\infty}^{+\infty} f\left(\frac{\tau}{t - \tau}\right) d\tau \quad (5)$$

Le spectre $Z(v)$ de $z(t)$ est donné par les relations suivantes :

$$(6) \begin{cases} 2F(v) \text{ pour } v > 0 \\ F(0) \text{ pour } v = 0 \\ 0 \text{ si } v < 0 \end{cases}$$

avec F : transformée de Fourier de $f(t)$.

Cependant, le fait de mettre brutalement les fréquences négatives à zéro introduit une transition brutale du signal (équivalent à un filtre dont la réponse en fréquence possède une pente verticale) ce qui n'est pas réaliste.

Il existe un filtre permettant de calculer la transformée de Hilbert d'un signal $f(t)$.

Valeur des coefficients du filtre de Hilbert

$H(1) = 0.16026190E-01 = -H(20)$	$H(6) = 0.55333299E-0.1 = -H(15)$
$H(2) = 0.14173287E-01 = -H(19)$	$H(7) = 0.78542752E-0.1 = -H(14)$
$H(3) = 0.20452437E-01 = -H(18)$	$H(8) = 0.11823755E-0.1 = -H(13)$
$H(4) = 0.28736882e-0.1 = -H(17)$	$H(9) = 0.20664125E-0.1 = -H(12)$
$H(5) = 0.39852581E-0.1 = -H(16)$	$H(10) = 0.63475619E-0.1 = -H(11)$

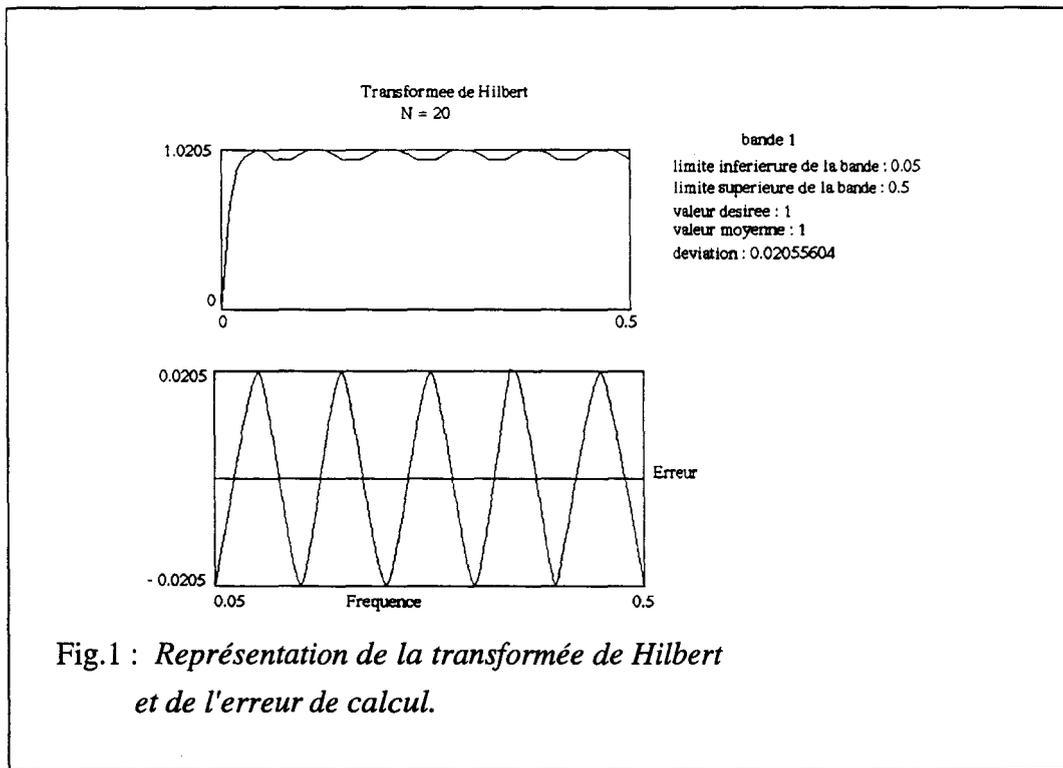


Fig.1 : Représentation de la transformée de Hilbert et de l'erreur de calcul.

1.4. TERMES D'INTERFÉRENCES

La transformée de Wigner-Ville de la somme de deux signaux n'est pas égale à la somme des transformées de chacun des signaux.

Soit un signal composé de la somme de deux signaux $x(t)$ et $y(t)$.

La transformée de Wigner-Ville de $(x(t) + y(t))$ notée $W_{x+y}(t, \nu)$ s'écrit :

$$W_{x+y}(t, \nu) = W_x(t, \nu) + W_y(t, \nu) + 2 \operatorname{Re}\{W_{xy}(t, \nu)\} \quad (7)$$

$$\text{avec } W_{xy}(t, \nu) = \int_{-\infty}^{+\infty} x\left(t + \frac{\tau}{2}\right) y^*\left(t - \frac{\tau}{2}\right) e^{-2i\pi\nu\tau} d\tau$$

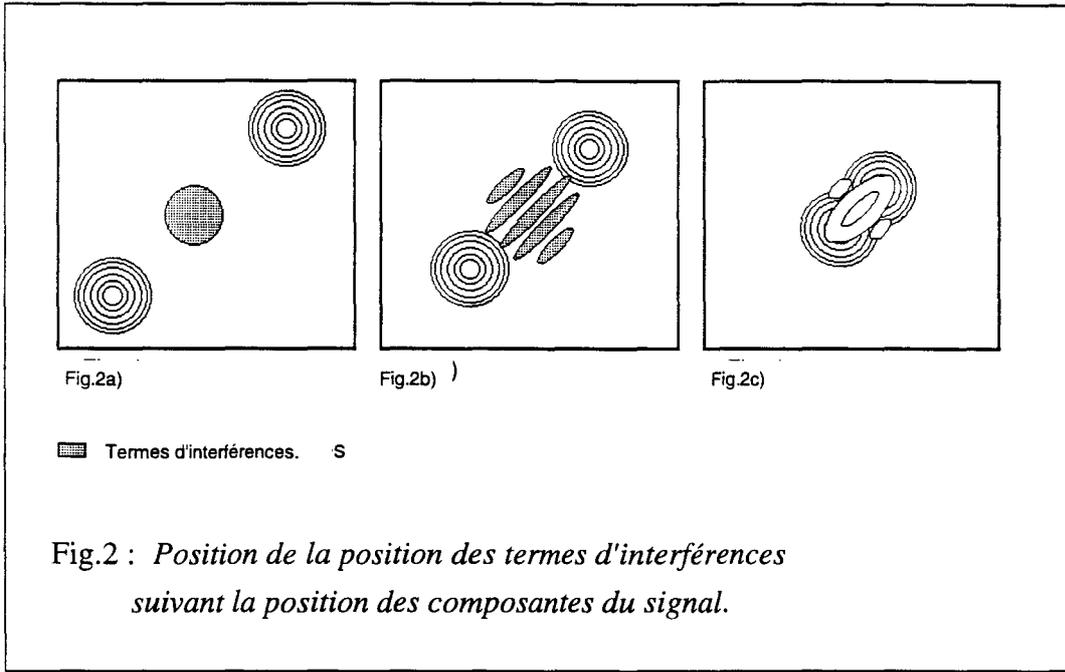
On remarque l'apparition du terme caractérisant les interférences : $\operatorname{Re}\{W_{yx}(t, \nu)\}$

Ceux-ci ont une structure oscillante caractérisée par un terme en cosinus [Fland 91], et dépendent de la distance en temporel et en fréquentiel séparant les composantes du signal. Ces oscillations sont d'autant plus rapides que les composantes sont éloignées.

De plus, les termes d'interférences sont localisés sur l'axe séparant les deux composantes en "interaction".

la figure 2 illustre la position des termes d'interférences suivant les positions des composantes du signal.

Dans le cas où les composantes sont assez éloignées les unes des autres, les termes d'interférences sont bien localisés. Ceux-ci sont situés au milieu de la droite joignant les deux composantes (Fig.2a)). Plus les composantes sont rapprochées, moins les termes d'interférences sont distincts. Il devient alors très difficile de les séparer du signal original (Fig.2b et Fig.2c))



Sur la figure 2a) les composantes du signal sont bien éloignées, les termes d'interférences sont bien localisés.

Plus les composantes du signal sont proches, moins les termes d'interférences sont distincts (Fig. 2b et Fig. 2c).

Ces interférences nuisent à une bonne interprétation de la transformée de Wigner-Ville. Plusieurs solutions existent afin de supprimer ou de diminuer ces termes parasites.

Une méthode consiste à décomposer le signal $x(t)$ sur une base adéquate ce qui permet d'éliminer les termes d'interférences. Cependant, cela nécessite soit une bonne connaissance du signal étudié, soit de posséder un ensemble de bases possibles. Cette méthode semble difficilement applicable au cas des images réelles dont on ne connaît rien à priori sur la nature de l'information.

Une autre solution est de tenir compte de la forme oscillatoire des termes d'interférences afin d'utiliser un lissage adéquat.

La première idée est d'utiliser un spectrogramme, c'est à dire une version lissée en temps et en fréquence de la transformée de Wigner-Ville. Mais si l'on privilégie une direction (temps ou fréquence) pour le lissage, on perd certaines propriétés de la transformée dans l'autre direction.

Une solution consiste à employer un lissage séparable en temps et en fréquence.

Une autre méthode est de modifier le noyau de la transformée de Wigner-Ville, aboutissant ainsi à d'autres transformées temps-fréquence.

Cependant, si ces dernières éliminent plus ou moins totalement ces termes parasites, elles ne conservent pas toujours toutes les propriétés de la transformée de Wigner-Ville.

1.4.1. Calcul de la transformée de Wigner-Ville

Il faut noter que le calcul de la transformée de Wigner nécessite la prise en compte de deux fois plus d'échantillons qu'un spectre classique. Si l'on part d'un signal réel décrit par M échantillons, l'utilisation du signal analytique nous fait travailler sur un signal complexe dont les parties imaginaires et réelles ont chacune M échantillons.

La transformée de Hilbert du signal réel, nécessaire au calcul du signal analytique associé au signal d'entrée, se fera par l'utilisation du filtre décrit plus haut.

La figure suivante représente le schéma d'architecture d'un transformateur de Wigner-Ville pour un signal évolutif discret, le calcul étant effectué à partir du signal analytique. Il est à noter qu'il faut un espace mémoire double pour stocker les échantillons obtenus après le calcul de la transformée de Hilbert du fait que ces échantillons sont maintenant complexes. Mais on ne multiplie pas par quatre l'espace mémoire en raison du fait que la TWV d'un signal réel ou complexe est toujours une fonction réelle.

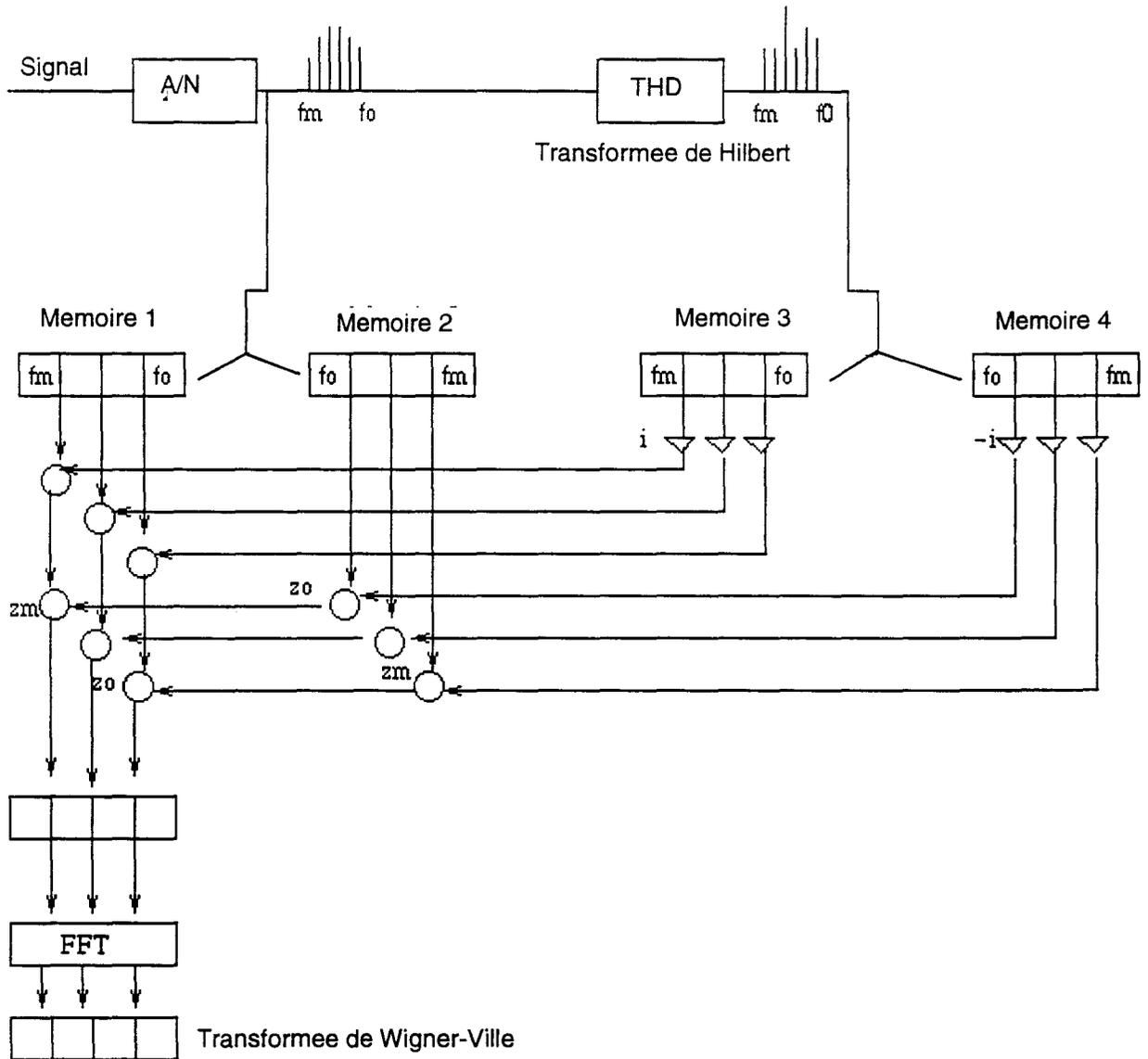


Schéma de calcul de la transformée de Wigner-Ville.

Nous venons de décrire les principales propriétés de la transformée de Wigner-Ville monodimensionnelle.

Nous nous proposons d'utiliser cette transformée sur une séquence d'images. Dans ce cas, nous ne travaillons plus en monodimensionnel mais en tridimensionnel. .

2.CAS TRIDIMENSIONNEL

Dans ce chapitre, nous décrirons le moyen de transposer les calculs de la transformée de Wigner-Ville à une dimension au cas tridimensionnel.

Nous y décrirons ensuite comment il est possible, après calcul de la transformée de Wigner-Ville 3D, de calculer les vecteurs déplacement des pixels d'une image.

2.1.DÉFINITION

La transformée de Wigner-Ville d'un signal monodimensionnel, $f(t)$ est fonction de deux variables $DWV(x,v)$.

La transformée de Wigner-Ville d'une fonction de deux variables, $f(x,y)$ est alors fonction de quatre variables notée $DWV(x,y,\omega_x,\omega_y)$ [Peyr 94]. En chaque point de l'image, on obtient donc un "spectre local".

Si on étend la formule de la transformée de Wigner-Ville au cas tridimensionnel, on obtient une fonction de six variables dont la définition est la suivante :

$$W_f (x, y, t, \omega_x, \omega_y, \omega_t) = \iiint_{\alpha, \beta, \delta = -\infty}^{\infty} R_f (x, y, t, \alpha, \beta, \delta) \exp [-j (\alpha \omega_x + \beta \omega_y + \delta \omega_t)] d\alpha d\beta d\delta \quad (8)$$

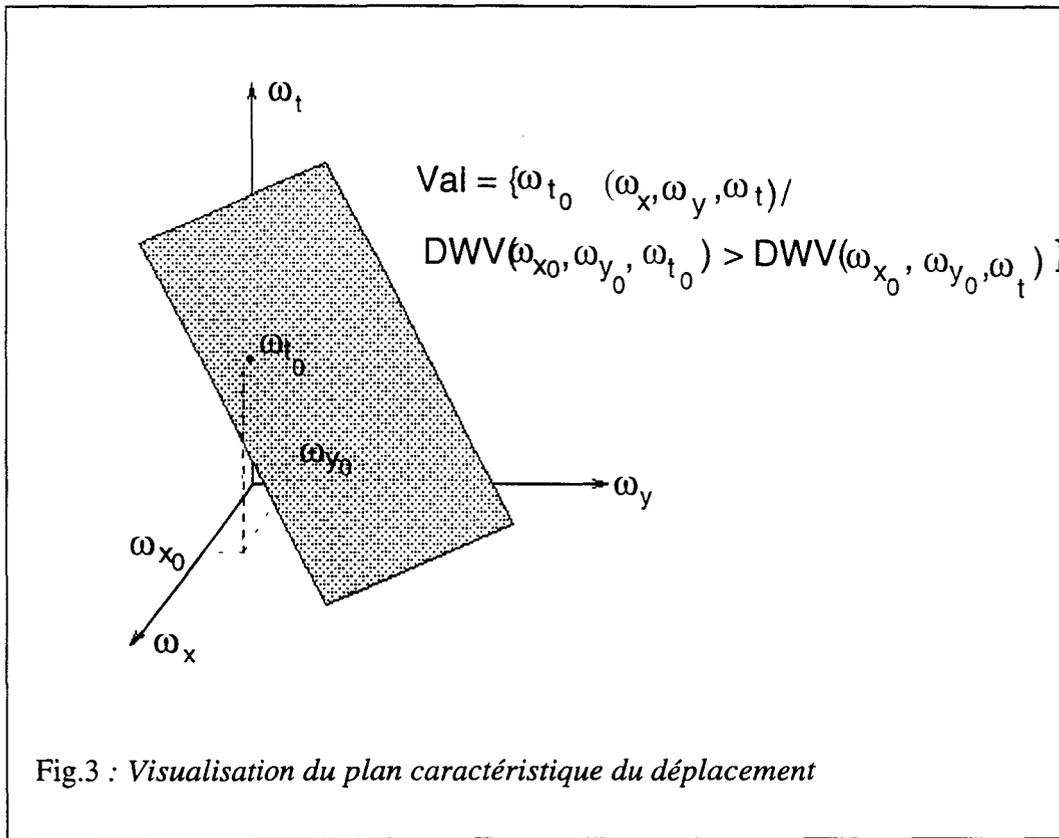
$$R_f (x, y, t, \omega_x, \omega_y, \omega_t) = f (x + \alpha / 2, y + \beta / 2, t + \delta / 2) f^* (x - \alpha / 2, y - \beta / 2, t - \delta / 2)$$

2.2.INTERPRÉTATION

En chaque point (x_0, y_0, t_0) de la séquence, on obtient alors un spectre tridimensionnel $DWV(x_0, y_0, t_0, \omega_x, \omega_y, \omega_t)$ dont l'interprétation permet de calculer le vecteur déplacement du point de coordonnées (x_0, y_0, t_0)

En théorie, la transformée de Wigner-Ville3D d'un pixel en translation constante au cours du temps est un plan que l'on visualisera de la manière suivante :

On suppose que les valeurs maximales de la transformée de Wigner-Ville 3D se situent sur ce plan. Pour chaque fréquence $(\omega_{x0}, \omega_{y0})$ on recherche la valeur maximale de la transformée, qui se situe à une certaine fréquence ω_{t0} que l'on visualise (Fig.3)



Nous allons maintenant passer en revue les différentes phases de calcul de la transformée de Wigner-Ville3D.

Les calculs s'effectuent en construisant un cube centré sur l'image pour laquelle on recherche les vecteurs déplacement. Après avoir effectué le produit du cube par son image miroir, on applique la FFT tridimensionnelle sur le résultat obtenu.

2.3.CALCUL DE LA TRANSFORMÉE DE WIGNER-VILLE 3D

2.3.1.Calcul de la FFT 3D

L'utilisation de la FFT n'est pas un problème car son noyau est séparable. On peut décomposer la FFT3D en une transformée de Fourier 1D suivie d'une transformée 2D ; ou inversement.

2.3.2.Calcul du signal analytique 3D

Pour un signal monodimensionnel, le calcul du signal analytique est trivial ; il suffit de mettre à zéro les fréquences négatives et à doubler les autres. En ce qui concerne le cas bidimensionnel, et à plus forte raison tridimensionnel, cela n'est pas aussi simple.

F.Peyrin dans [Peyr 94] explique comment on peut transposer le calcul du signal analytique associé à un signal 1D au calcul en 2D. Elle propose de calculer le spectre du signal bidimensionnel et de mettre à zéro les fréquences situées sur un plan inférieur au plan déterminé par une droite bien précise.

L'inclinaison de la droite est caractérisé par l'angle θ .

Le calcul du signal analytique, noté $z\theta^*(x,y)$, associé au signal bidimensionnel $f(x,y)$ sera donné par la relation :

$$z_q(x, y) = (f + i \cdot f^* \cdot k_q)(x, y) \quad (9)$$

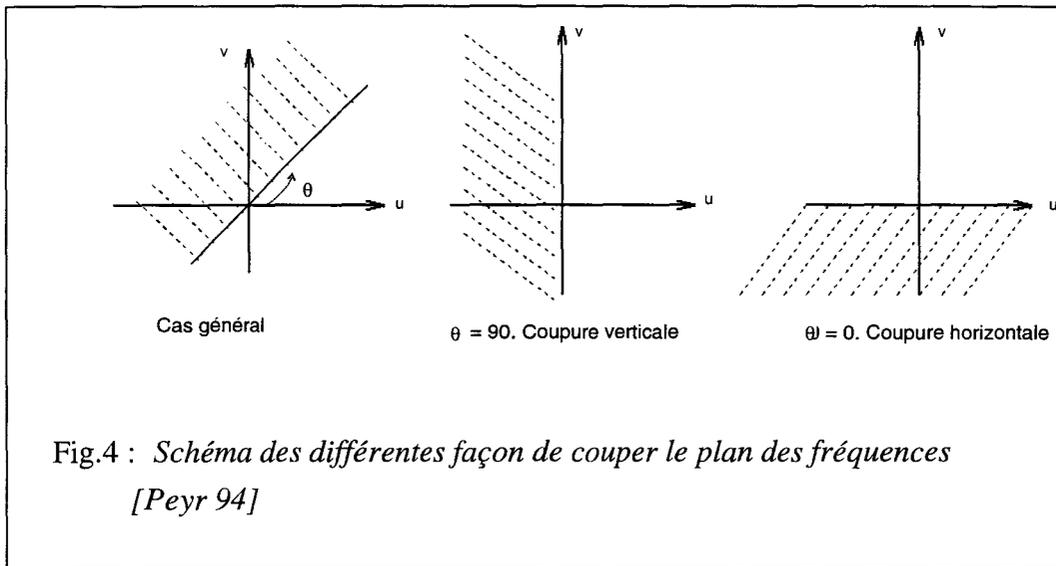
$$\text{avec } k_q(x, y) = \left(\frac{1}{x \cos \theta + y \sin \theta} \right) \cdot d(y \cos \theta - x \sin \theta)$$

Dans le domaine fréquentiel, les relations précédentes s'écritront de la manière suivante:

$$Z_q(x, y) = F(u, v)(1 + \text{sign}(u \cos \theta + v \sin \theta)) \quad (10)$$

$$\text{avec } \text{sign}(t) = 1 \quad \text{ssi } t > 0$$

La figure 4 représente les différentes façons de mettre à zéro les coefficients du spectre du signal.



Suivant la coupure utilisée, on éliminera une certaine partie des interférences. Cette méthode n'est donc vraiment exploitable que si l'on connaît la position des termes à éliminer, ou si l'on arrive à estimer quelles interférences nuisent le plus à l'interprétation du signal.

De plus, cette méthode est difficilement applicable au cas tridimensionnel car elle nécessite l'emploi de deux transformées de Fourier par pixel, ce qui alourdit beaucoup les calculs.

Une solution plus simple est d'adapter le filtre monodimensionnel décrit plus haut au cas bidimensionnel. Dans ce cas, chaque image de la séquence sera filtrée par ce filtre.

3. TRANSFORMÉE DE WIGNER-VILLE 3D ET DEPLACEMENT

Comme nous l'avons dit précédemment, la transformée de Wigner 3D d'une image en translation est un plan. Nous nous proposons d'en faire la démonstration et d'étudier les comportements de la transformée pour différents types de mouvements.

3.1. DWV D'UNE IMAGE EN TRANSLATION

Considérons une séquence d'images en translation constante.

Si on note $f(x,y,t)$ l'intensité d'un pixel P de coordonnées (x,y) à l'instant t , on peut écrire

$$f(x, y, t) = f(x, y) * \delta(x - V_x t, y - V_y t) \quad (11)$$

Développons la transformée de Wigner - Ville de $\delta(x - V_x t, y - V_y t)$

$$\begin{aligned} W_\delta(x, y, t, w_x, w_y, w_t) = & \iiint_{\alpha, \beta, \delta} \delta\{(x + \alpha / 2) - V_x(t + \tau / 2), (y + \beta / 2) - V_y(t + \tau / 2)\} \\ & * \delta\{(x - \alpha / 2) - V_x(t - \tau / 2), (y - \beta / 2) - V_y(t - \tau / 2)\} \\ & * \exp[-j(\alpha \omega_x + \beta \omega_y + \tau \omega_t)] d\alpha d\beta d\tau \end{aligned} \quad (12)$$

$$\begin{aligned} = & \iiint_{\alpha, \beta, \delta} \delta\{(x + \alpha / 2) - V_x(t + \tau / 2)\} \delta\{(x - \alpha / 2) - V_x(t - \tau / 2)\} \\ & \delta\{(y + \beta / 2) - V_y(t + \tau / 2)\} \delta\{(y - \beta / 2) - V_y(t - \tau / 2)\} \\ & \exp[-j(\alpha \omega_x + \beta \omega_y + \tau \omega_t)] d\alpha d\beta d\tau \end{aligned} \quad (13)$$

$$\begin{aligned} = & \iiint_{\alpha, \beta, \tau} \delta\{(x - V_x t) + (\alpha / 2 - V_x \tau / 2)\} \delta\{(x - V_x t) - (\alpha / 2 - V_x \tau / 2)\} \\ & \delta\{(y - V_y t) + (\beta / 2 - V_y \tau / 2)\} \delta\{(y - V_y t) - (\beta / 2 - V_y \tau / 2)\} \\ & \exp[-j(\alpha \omega_x + \beta \omega_y + \tau \omega_t)] d\alpha d\beta d\tau \end{aligned} \quad (14)$$

$$= \iiint_{\alpha, \beta, \tau} \{\delta(\alpha - V_x \tau) \delta(x - V_x t)\} \{\delta(\beta - V_y \tau) \delta(y - V_y t)\} \exp[-j(\alpha \omega_x + \beta \omega_y + \tau \omega_t)] d\alpha d\beta d\tau \quad (15)$$

$$= \delta(x - V_x t, y - V_y t) \int_{\tau} \left\{ \int_{\alpha} \int_{\beta} \delta(\alpha - V_x \tau) \delta(\beta - V_y \tau) \exp[-j(\alpha \omega_x + \beta \omega_y)] d\alpha d\beta \right\} \exp(-j\tau \omega_t) d\tau \quad (16)$$

$$= \delta(x - V_x t, y - V_y t) \int_{\tau} \exp[-j(V_x \omega_x + V_y \omega_y) \tau] \exp(-j\tau \omega_t) d\tau \quad (17)$$

$$= \delta(x - V_x t, y - V_y t) \int_{\tau} \exp[-j(V_x \omega_x + V_y \omega_y + \omega_t) \tau] d\tau \quad (18)$$

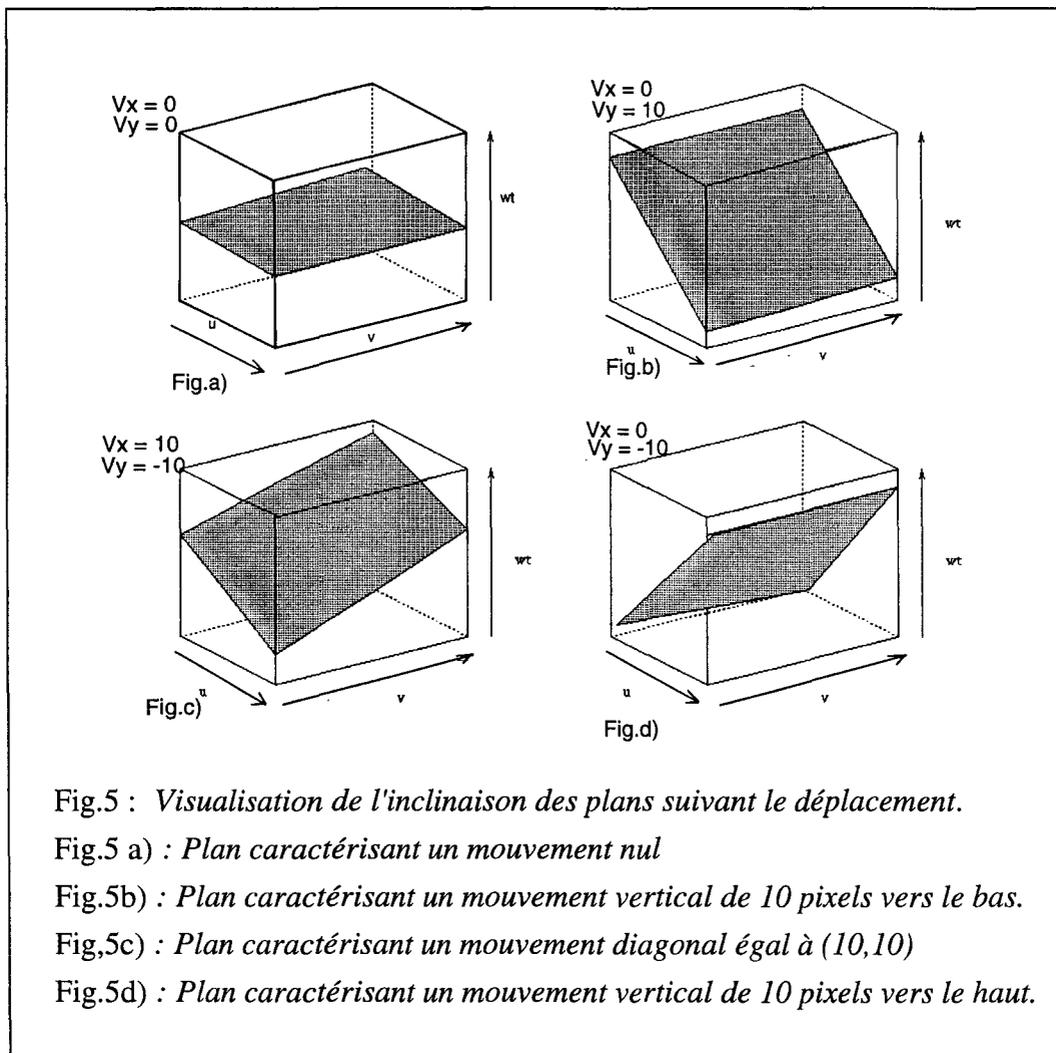
$$= \delta(x - V_x t, y - V_y t) \delta(V_x \omega_x + V_y \omega_y + \omega_t) \quad (19)$$

La transformée de Wigner-Ville d'une séquence d'images en translation constante est un plan d'équation :

$$V_x \omega_x + V_y \omega_y + \omega_t = 0 \quad (20).$$

Le calcul de l'inclinaison de ce plan nous donnera la valeur de V_x et V_y au point P.

La figure 5 représente l'inclinaison des différents plans suivant les vecteurs déplacement.



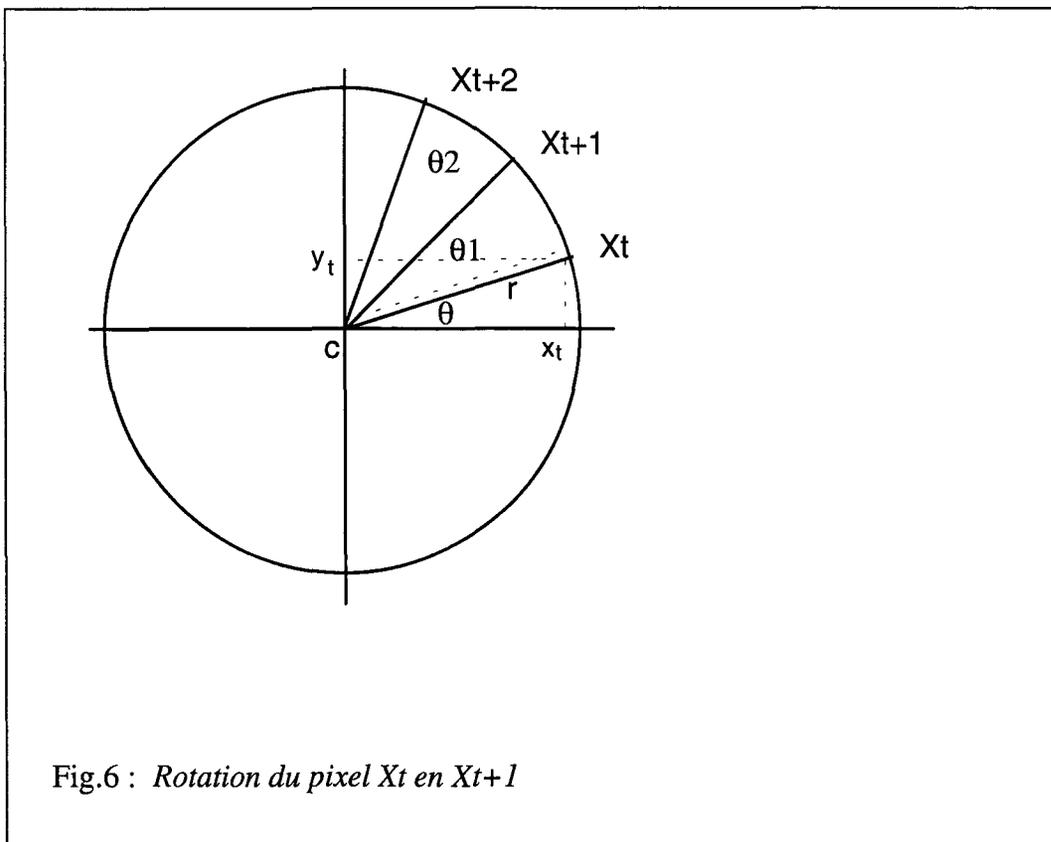
En ce qui concerne la "Fonction test", on obtiendra un pic situé en (V_{x_0}, V_{y_0}) , vecteur déplacement du pixel considéré.

3.2.DWV D'UNE IMAGE EN ROTATION

Soit un pixel ayant un mouvement de rotation uniforme.

De manière simplifiée, on peut représenter son mouvement comme indiqué sur la figure 6.

X_t représente la position du pixel X à l'instant t , X_{t+1} , sa position à l'instant $t+1$ et X_{t+2} à l'instant $t+2$.



Si l'on projette X_t sur les axes horizontaux et verticaux, on a alors :

$$(21) \begin{cases} x_t = c + r \cos \theta \\ y_t = c + r \sin \theta \\ V_x = \partial x / \partial t \\ V_y = \partial y / \partial t \end{cases}$$

L'intensité d'un point P de l'image K (instant t) peut se déduire de celle du même point situé dans la trame précédente par une relation similaire à celle utilisée dans le cas de la translation. on remplacera V_x et V_y par leurs valeurs.

On a donc :

$$f(x, y, t) = f(x, y) * \delta\left[x - \left(-r \frac{\partial \theta}{\partial t} \sin \theta\right)t, x - \left(r \frac{\partial \theta}{\partial t} \cos \theta\right)t\right] \quad (22)$$

Dans le cas d'une rotation uniforme, la dérivée de l'angle θ par rapport au temps est une constante.

En développant de la même façon que précédemment, on arrive au résultat suivant :

La transformée de Wigner-Ville est nulle partout, sauf pour les points d'équation :

$$\omega_x(-r \sin \theta \cdot \frac{\partial \theta}{\partial t}) + \omega_y(r \cos \theta \cdot \frac{\partial \theta}{\partial t}) + \omega_t = 0 \quad (23)$$

La valeur de V_x varie en fonction de $\sin \theta$, celle de V_y en fonction de $\cos \theta$.

Comme on travaille sur plusieurs images consécutives, les valeurs de V_x et de V_y ne seront pas fixes. Pour chaque image utilisées pour les calculs, on aura un plan d'inclinaison différente.

3.3.CALCUL DE L'INCLINAISON DU PLAN

Soit (V_{x_0}, V_{y_0}) le vecteur déplacement d'un point P(x,y) de l'image.

La valeur de (V_{x_0}, V_{y_0}) est liée à l'inclinaison du plan obtenu après le calcul de la transformée de Wigner-Ville. Afin de déterminer le vecteur déplacement du point P, pour chaque couple (V_x, V_y) , allant de $-V_{x_{\max}}$ à $V_{x_{\max}}$ (de même pour V_y), nous avons sommé toutes les valeurs répondant à l'équation du plan :

$$V_x \omega_x + V_y \omega_y + \omega_t = 0. \quad (24)$$

Puis nous avons recherché le maximum qui nous donne directement (V_{x_0}, V_{y_0}) .

Cette méthode est longue et les résultats dépendent de l'amplitude que l'on donne à $V_{x_{\max}}$ ou $V_{y_{\max}}$. Si le mouvement est faible, on aura peu de valeurs différentes à tester pour le couple (V_x, V_y) ; $V_{x_{\max}}$ et $V_{y_{\max}}$ étant faibles.

En cas de mouvements importants, on augmente considérablement les calculs.

CHAPITRE III

CALCULS ET RESULTATS

D'après ce que nous avons vu précédemment, il semble tout à fait possible d'estimer les vecteurs mouvement d'une séquence animée par le biais de la transformée de Wigner-Ville 3D.

Un article datant de 1986 écrit par Jacobson et Weschler [Jacob 86] aborde d'ailleurs ce sujet. Les auteurs proposent d'estimer le flot optique d'une fonction image, $f(x,y,t)$, générée par calculs. Dans ce cas, les images sont bien définies et varient uniformément dans le temps.

Dans le cadre de notre étude, nous nous proposons de calculer les vecteurs déplacement d'une image réelle telle que les images de télévision qui ne sont pas synthétisées de façon mathématique.

Après avoir décrit rapidement notre algorithme de calcul de la transformée de Wigner-Ville tridimensionnelle, nous l'appliquerons sur des séquences d'images.

Les tests préliminaires, décrits dans le second chapitre, ont été effectués sans aucune optimisation, et ceci afin de vérifier rapidement l'existence de plans caractérisant le déplacement ainsi que le moyen de calculer leurs inclinaisons.

Dans la troisième chapitre, nous décrivons la différence entre la calcul des vecteurs déplacement effectués sur des points réels ou sur des points fictifs.

Les trois chapitres suivants sont consacrés à l'optimisation des paramètres de calculs, ainsi que les résultats obtenus.

Dans le dernier chapitre, nous visualisons les plans caractérisant les déplacement, ainsi que la fonction test de divers mouvements que ce soit sur des images de synthèses ou des images réelles.

1. DESCRIPTION DES CALCULS

Soit une image pour laquelle on désire connaître les vecteurs déplacement en chaque point. Comme celle-ci fait partie d'une séquence animée, on dispose donc de N images se succédant à intervalles réguliers.

Nous expliquerons tout d'abord comment calculer la transformée de Wigner-Ville 3D au point P(x,y), pour ensuite déterminer comment à partir de celle-ci on peut estimer les composantes (V_x,V_y) du vecteur déplacement du point P.

1.1. Théorie

Rappelons, pour mémoire, la formule théorique de la transformée de Wigner-Ville 3D d'une fonction f(x,y,t) :

$$Wf(x, y, t, u, v, \omega) = \iiint_{\alpha, \beta, \gamma = -\infty}^{\infty} f(x + \alpha / 2, y + \beta / 2, t + \gamma / 2) \cdot f^*((x - \alpha / 2, y - \beta / 2, t - \gamma / 2) \exp(-2i\pi(\alpha u + \beta v + \gamma \omega)) d\alpha d\beta d\gamma \quad (1)$$

Ce calcul fait intervenir trois paramètres α , β et τ qui déterminent le domaine d'intégration.

En pratique, la transformée de Wigner-Ville s'effectue par l'intermédiaire de la transformée de Fourier 3D et s'écrit de la manière suivante :

$$W(x, y, z, u, v, \omega) = \iiint_{\eta, \xi, \mu = -\infty}^{\infty} F(u + \eta / 2, v + \xi / 2, \omega + \mu / 2) \cdot F^*(u - \eta / 2, v - \xi / 2, \omega - \mu / 2) \exp(2i\pi(\eta x + \xi y + \mu \omega)) d\eta d\xi d\mu \quad (2)$$

Avec :

"*" opérateur conjugué

F transformée de Fourier de f.

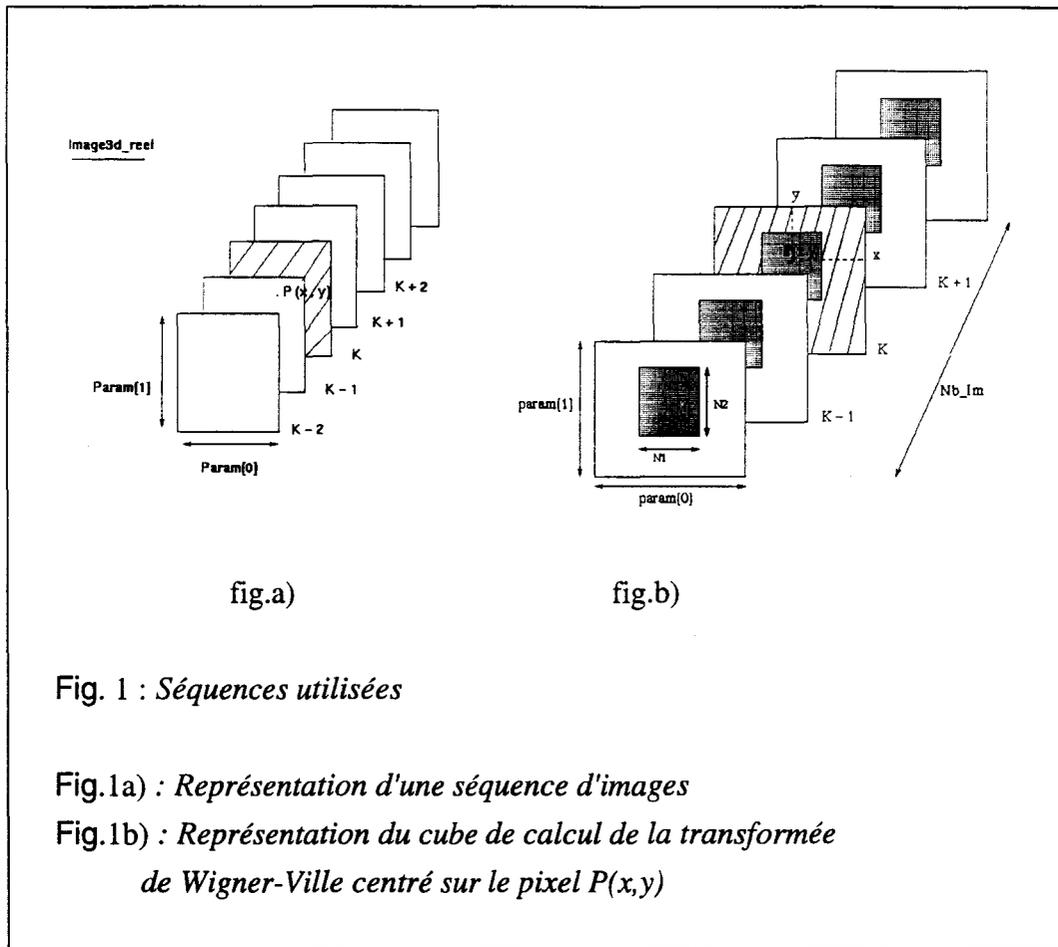
1.2. Méthode de calcul

On construit un cube centré sur le point P, puis on effectue le produit du cube par son image miroir par rapport au point P, et on calcule enfin la transformée de Fourier 3D de ce produit.

La première étape consiste donc à déterminer les dimensions de ce cube.

Par la suite, nous noterons N₂ la dimension verticale du cube, N₁ la dimension horizontale et Nb-Im la profondeur qui représente donc le nombre d'images utilisées pour le calcul de la transformée de Wigner-Ville 3D.

La figure 1 représente une séquence d'images (Fig.a) ainsi que le cube nécessaire aux calculs (Fig.b).



Nous appellerons "Image3D-reel" la séquence d'images que nous avons à notre disposition.

Param[1] est le nombre de lignes de l'image, qui est généralement égal à 576, param[0] le nombre de colonnes, égal à 720.

Les trames 1 et 2 composant une images possèdent donc chacune 288 lignes de 720 pixels.

"sequence3D-reel" est le bloc où s'effectuent les calculs. Celui-ci est centré sur le point $P(x,y)$.

Par la suite, tous les calculs s'effectuent sur "sequence3D-reel".

Comme nous l'avons vu précédemment, le calcul de la transformée de Wigner-Ville (DWV) se fait par l'intermédiaire de la FFT3D. Cette dernière s'applique sur le produit du cube, noté "sequence3D-reel", par son symétrique par rapport au point P.

La figure 2 décrit la manière de calculer le produit du cube par son miroir, en prenant le pixel $P(x,y)$ comme centre de symétrie.

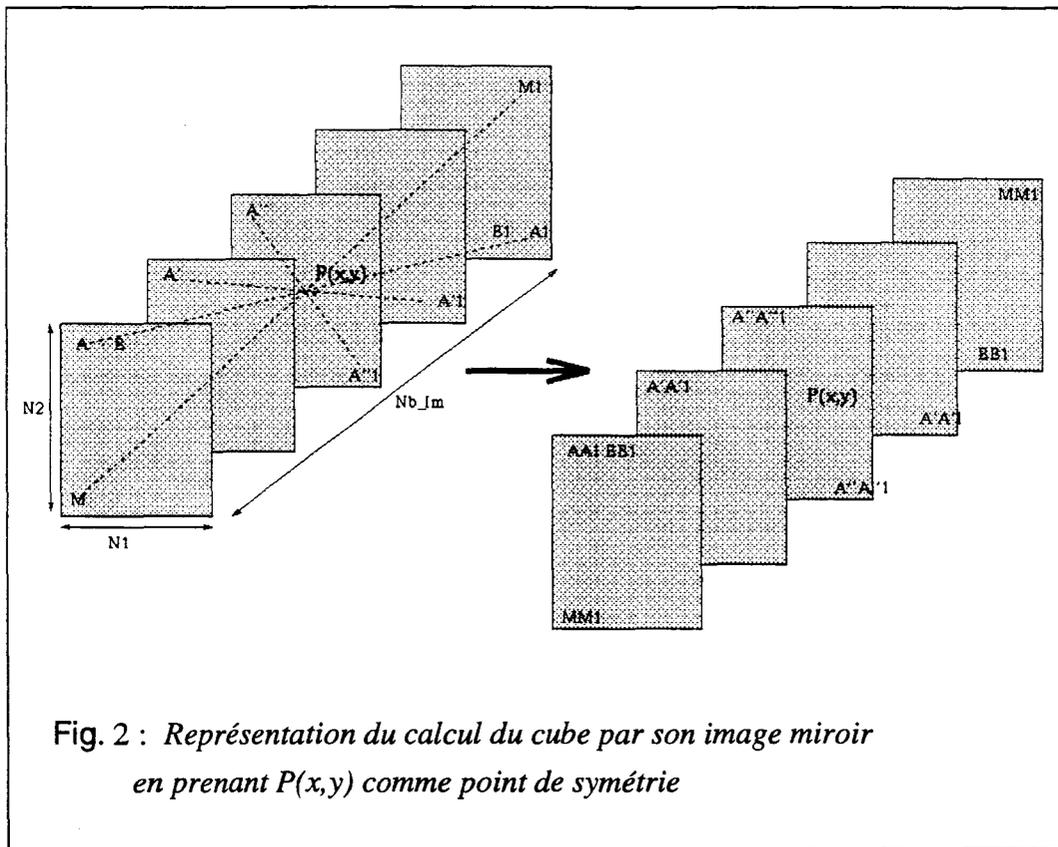


Fig. 2 : Représentation du calcul du cube par son image miroir en prenant $P(x,y)$ comme point de symétrie

Une fois la transformée de Wigner-Ville calculée, l'étape suivante consiste à estimer le vecteur déplacement du point $P(x,y)$

1.3. Calcul théorique de V_x, V_y

La transformée de Wigner-Ville 3D d'une image en translation est nulle partout sauf sur un plan d'équation défini par :

$$uV_x + vV_y + Nb - Im = 0 \quad (3)$$

Pour chaque point de coordonnées (x, y) , on peut alors estimer son vecteur déplacement d'après l'inclinaison du plan obtenu après le calcul de la transformée de Wigner-Ville.

Cependant la DWV est une transformée qui comporte de nombreux termes d'interactions; il est donc fort probable qu'il y ait des fréquences non nulles hors du plan décrit précédemment. Le plan d'inclinaison (V_x, V_y) n'est donc pas parfaitement défini.

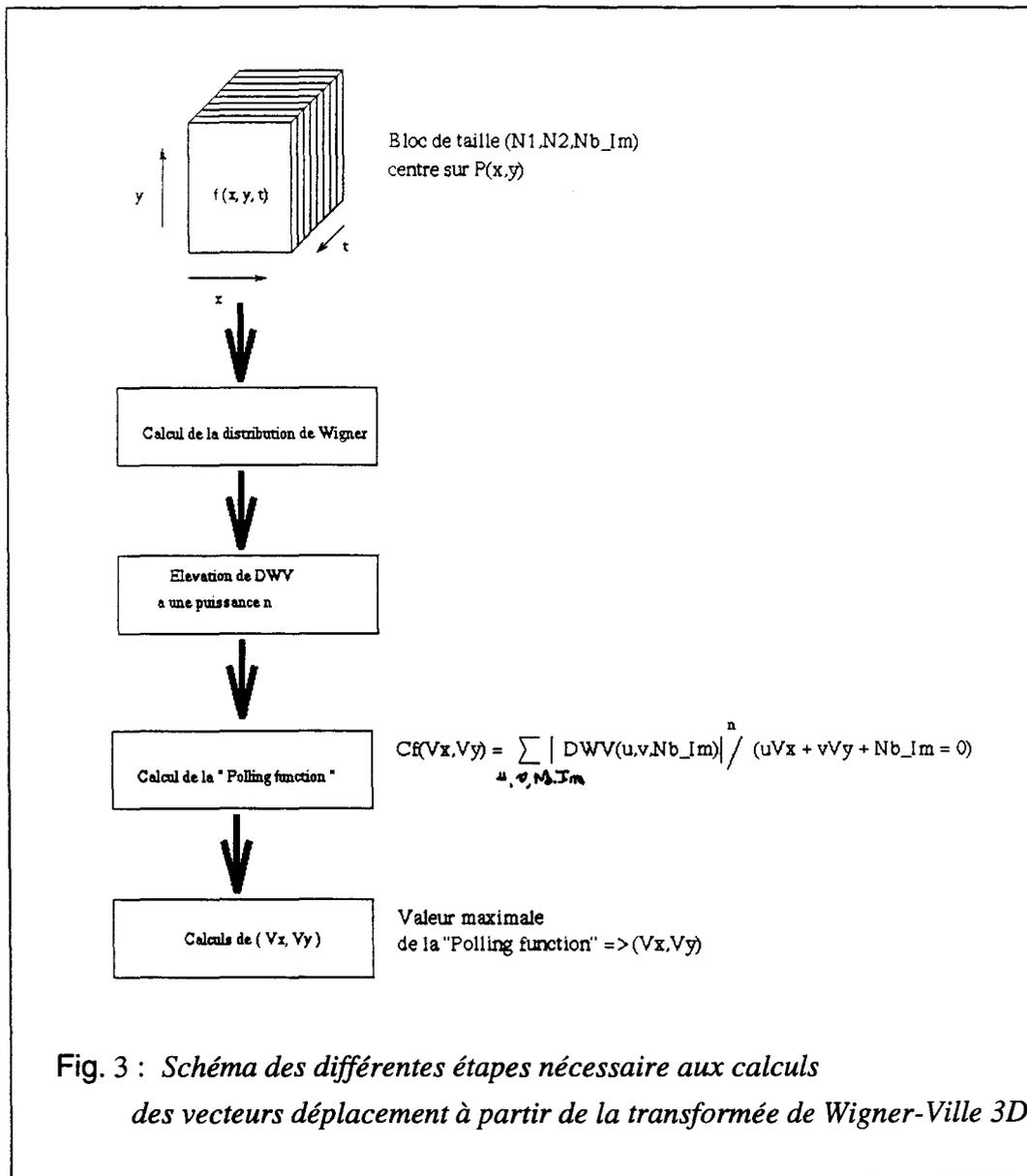
On peut toutefois supposer que le plan caractérisant le vecteur déplacement (V_x, V_y) sera un plan pour lequel les valeurs de la distribution de Wigner-Ville seront les plus importantes. Ce qui revient à supposer que les termes d'interférences sont négligeables par rapport aux coefficients de la transformée de Wigner-Ville.

Comme on ne possède aucune information à priori sur le vecteur déplacement du point $P(x, y)$; pour chaque couple (V_x, V_y) , on sommerá toutes les valeurs de la transformée de Wigner-Ville correspondant à l'équation du plan :

$$uV_x + vV_y + Nb - Im = 0 \quad (4)$$

On obtient donc un tableau, que nous appellerons fonction test dont la valeur maximale nous donne l'amplitude et la direction du vecteur déplacement du point P.

La figure 3 résume les différentes étapes du calcul des vecteurs déplacement à partir de la transformée de Wigner-Ville.



Les calculs étant décrits, nous allons à présent effectuer les premiers tests afin de vérifier si en pratique il est effectivement possible d'obtenir les vecteurs déplacement des pixels d'une image par le biais de la transformée de Wigner-Ville.

2.TESTS PRELIMINAIRES

2.1.Introduction

Dans cette partie, nous allons calculer la transformée de Wigner-Ville tridimensionnelle, et vérifier s'il est bien possible, à partir de cette dernière d'en extraire les vecteurs déplacement.

Aucun paramètre n'est optimisé. Cela fera l'objet du chapitre suivant.

2.2.Séquences utilisées

Les différentes séquences sont construites à partir de l'image "boy" (voir en annexe) qui représente un garçon et ses jouets le tout sur un fond uniforme.

Les séquences utilisées sont formées d'images réelles, de type images de télévision, dont nous connaissons parfaitement les vecteurs déplacements car nous les simulons. Un algorithme permet, à partir d'une image réelle ou de synthèse, de former une séquence d'images dont on aura préalablement déterminé le vecteur déplacement de tous les pixels de l'image.

Comme tous les points de l'image ont le même mouvement uniforme, on peu aisément vérifier l'exactitude des résultats obtenus.

Nous avons fait suivre le nom de l'image servant à créer la séquence animée par les valeurs de V_x , et de V_y .

Par exemple la séquence formée à partir de l'image "boy.lum" et dont tous les points se déplacent horizontalement de 2 pixels vers la droite par image sera appelée "boy20.lum".

2.3.Description des calculs

Le but de ces premiers calculs étant de vérifier s'il est possible de calculer des vecteurs déplacement par le biais de la transformée de Wigner-Ville, nous avons choisi N_1 , N_2 , N_b -Im de façon arbitraire, que nous avons fixé à (11*11*11).

Les calculs sont effectués sur le signal original, c'est à dire sans utiliser le signal analytique associé au signal réel.

Comme nous n'interprétons pas directement la transformée de Wigner-Ville pour analyser le signal, nous avons supposé que les termes d'interférences dus au non surséchantillonnage du signal n'interviendraient pas, ou peu, dans le calcul des vecteurs déplacement.

Le calcul de la fonction test est identique à celui décrit dans le chapitre précédent. Cependant les valeurs de la transformée de Wigner-Ville sont élevées à la puissance quatre de façon à favoriser la prise en compte des valeurs les plus importantes de la transformée.

Afin d'accélérer les calculs, nous avons tout d'abord recherché la valeur des vecteurs déplacement pour un pixel sur quatre de l'image.

Cependant, en procédant de cette manière, on peut effectuer les calculs sur des points sans grande importance dans l'image tel que les fonds uniformes. On risque alors de passer à coté de points intéressants de l'image comme par exemple ceux qui appartiennent à des contours.

Pour cela, nous avons essayé d'extraire de façon simple et rapide les principaux points de l'image.

2.3.1. Filtrage de Sobel

Comme le but principal de ce filtrage n'est pas d'obtenir les points caractéristiques afin de suivre leur évolution au cours du temps, mais de repérer les points les plus représentatifs de l'image nous utiliserons un filtre de Sobel dont nous rappelons ici la définition.

Ce filtre est décomposable en un filtrage horizontal suivi d'un filtrage vertical.

	1 2 1		1 0 -1
filtrage horizontal	0 0 0	filtrage vertical	2 0 -2
	-1 -2 -1		1 0 -1

Après l'application du filtre horizontal et vertical, on obtient deux images, l'une mettant en évidence les contours verticaux de l'image originale, et l'autre les contours horizontaux (voir en annexe boyH et boyV). On ne retient que les valeurs maximales des deux images pour obtenir une image représentative des contours de l'image originale (voir en annexe boy-filt).

Le calcul de la transformée de Wigner-Ville 3D ne s'effectue pas sur les images filtrées, mais sur la séquence d'images originale. L'image filtrée sert à repérer les points caractéristiques de l'image pour lesquels on appliquera la transformée de Wigner-Ville 3D afin d'en calculer leur vecteur déplacement.

Dans ce cas, on n'effectue plus les calculs des vecteurs sur tous les points de l'image, ou sur des points choisis arbitrairement.

Il existe alors des pixels pour lesquels on ne connaît pas les vecteurs déplacement. Nous les appellerons les points non calculés auxquels il faut tout de même affecter une valeur de vecteur, quelle soit nulle ou pas.

2.3.2. Affectation d'une valeur de (V_x, V_y) aux points non calculés

Nous avons tout d'abord fixé à zéro les valeurs de V_x et V_y pour les points non calculés. Les résultats ne sont pas satisfaisants. En effet, un pixel situé à l'intérieur d'un objet, et non sur le contours aura dans ce cas un vecteur déplacement nul, ce qui n'est pas le cas en réalité.

Nous avons ensuite conservé le dernier vecteur calculé sur un contour afin de l'appliquer aux points non calculés. Mais dans ce cas, il y a propagation de l'erreur.

La meilleure solution est de conserver le vecteur trouvé précédemment si la différence entre la valeur du point de l'image K et le même pixel déplacé de (V_x, V_y) de la trame $K+1$ est inférieure à la différence obtenue en prenant comme vecteur le vecteur nul.

Les problèmes énoncés précédemment sont représentés sur la figure 4. On peut soit assimiler le vecteur nul au pixel de coordonnées (x,y) de l'image K , ou supposer que ce dernier se déplace de (V_x, V_y) qui est le vecteur trouvé pour le pixel P_c de coordonnées $(x-1,y)$ et appartenant à un contour.

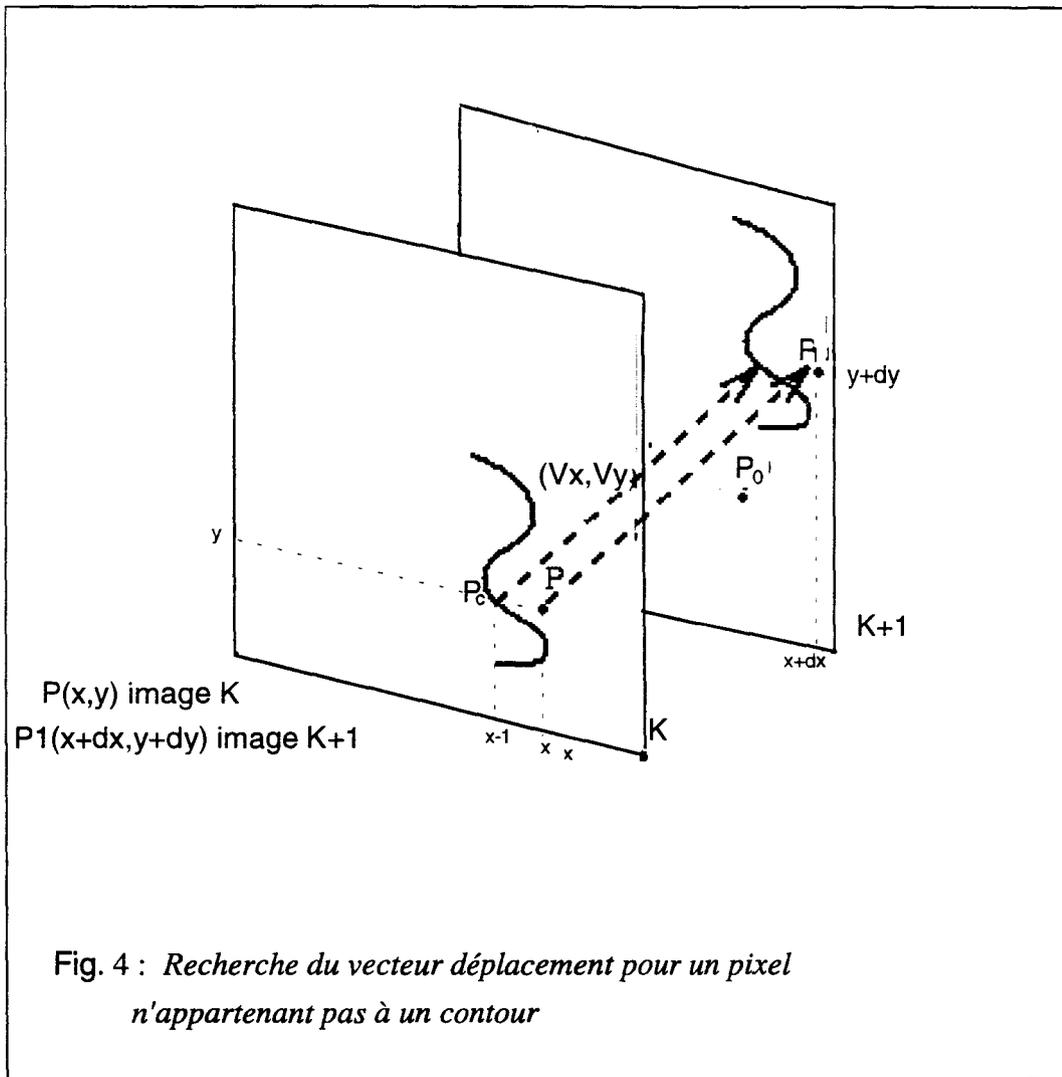
Afin de résoudre ce problème, on calcule alors les différences suivantes :

$|P - P_1|$ et $|P - P_0|$.

P_0 étant le pixel de l'image $K+1$ ayant les mêmes coordonnées que $P(x,y)$; il correspond alors au vecteur nul.

P_1 a pour coordonnées $(x+dx, y+dy)$, dx et dy étant les composantes suivant l'axe horizontal et vertical du vecteur déplacement.

Si $|P-P_1|$ est inférieure à $|P-P_0|$, on assimilera le vecteur (V_x, V_y) au pixel $P(x,y)$. Dans le cas contraire, on lui assimilera le vecteur nul.



2.3.3. Tests sur le calcul de (V_x, V_y)

L'amplitude et la direction du vecteur déplacement sont obtenues en recherchant la valeur maximale de la "fonction test". Cependant, ce pic n'est pas toujours bien déterminé. On sélectionne donc les quatre pics principaux.

Pour chacun d'eux on calcule la différence entre deux images consécutives entre le pixel P et le même pixel ayant subi le déplacement du vecteur trouvé. L'erreur la plus faible correspondra au vecteur déplacement.

La figure 5 représente la façon dont on calcule l'erreur pour les trois ou quatre valeurs principales de la "Fonction test"..

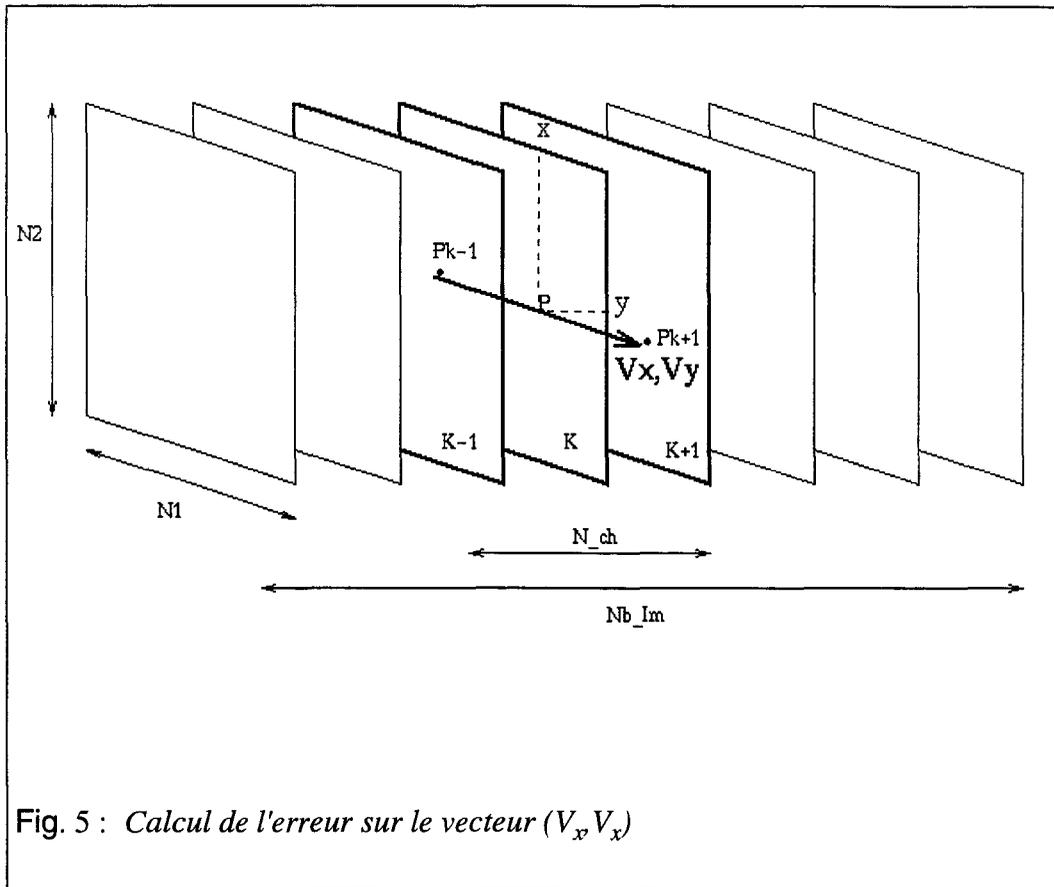


Fig. 5 : Calcul de l'erreur sur le vecteur (V_x, V_y)

L'erreur est égale à :

$$|P_{k-1}(x - V_x/2, y - V_y/2) - P_{k+1}(x + V_x/2, y + V_y/2)| \quad (5)$$

2.4. Tests sur les différents algorithmes

Les tests sont effectués en comptabilisant le pourcentage de vecteurs exacts suivant le mouvement étudié. Ils sont regroupés Tab1.

La première colonne indique la séquence utilisée, indiquant le déplacement de tous les pixels entre deux images consécutives de la séquence.

La seconde colonne regroupe les pourcentages de vecteurs exacts lorsque les calculs sont effectués sur les points caractéristiques de l'image obtenus après un filtrage de Sobel, avec les tests sur les points non calculés et le calcul de trois valeurs de vecteurs déplacement.

Pour la troisième colonne, les calculs sont identiques mis à part que l'on n'a pas effectué de tests sur les points non calculés. Leur vecteur est alors mis à zéro.

Pour la dernière colonne on s'est contenté de ne calculer qu'une seule valeur du vecteur (V_x, V_y) qui correspond au maximum de la "Fonction test".

Les calculs ont été effectués sur des blocs de taille $(11*11*11)$; N_1 et N_2 et Nb-Im varient donc entre -5 et 5.

cube de taille $(11*11*11)$			
Sequence utilisée	Sobel + test sur (V_x, V_y) + test sur points non calculés	Sobel + test sur (V_x, V_y) pas de test sur les points non calculés	Sobel sans aucun test
boy-20	73%	52,5%	16,7%
boy22	77%	58,2%	18,2%
boy30	65%	48%	14%
boy-23	36,8%	15,3%	2,5%
boy32	55,8%	34,1%	6,8%
boy40	51%	28,3%	3%

Tab.1 : Résultats des tests préliminaires obtenus en effectuant les calculs sur un bloc de taille $(11*11*11)$.

La troisième et la dernière colonne sont les plus intéressantes à interpréter.

En effet, pour la troisième colonne, on obtient réellement le pourcentage de vecteurs exacts par rapport au nombre de vecteurs calculés.

En ce qui concerne la seconde colonne, ces chiffres sont plus importants à cause du test sur les pixels pour lesquels aucun calcul n'a été effectué.

On remarque que les résultats sont nettement moins bons pour un déplacement de $(3,2)$ par rapport à un déplacement de $(3,0)$.

On observe une importante différence entre les valeurs obtenues pour un déplacement de (3,2) et un déplacement de (-2,3) alors que l'amplitude est du même ordre de grandeur. Cela s'explique par le fait qu'en ce qui concerne le déplacement (-2,3), le calcul est effectué sur l'image 5, tout comme pour les autres déplacement ; mais dans le cas d'un déplacement de (-2,3), on tombe sur une zone peu intéressante. En fait, les calculs ne sont pas effectués sur la même portion de l'image, il est alors difficile de comparer les lignes d'une même colonne entre-elles.

En ce qui concerne la dernière colonne, on s'aperçoit que les résultats sont beaucoup plus faibles que ceux obtenus pour la deuxième colonne. Il apparaît donc que la recherche du maximum de la "fonction test" ne donne pas automatiquement la valeur exacte du vecteur déplacement.

2.5.Conclusion

Les premiers résultats sont satisfaisants, puisque l'on peut donc effectivement calculer les vecteurs déplacement à partir de la transformée de Wigner-Ville.

En ce qui concerne les faibles mouvements horizontaux ou verticaux ((1,0) ou (0,2)) les résultats sont très satisfaisants (plus de 50% de vecteurs exacts) par contre, dès que l'amplitude du mouvement est plus importante, les résultats sont nettement moins bons surtout si le mouvement est diagonal.

Il apparaît aussi nécessaire de ne pas se contenter uniquement de la valeur maximale de la "Fonction test" pour déterminer les valeurs des vecteurs déplacement.

Il faut cependant souligner que ces premiers tests ont été effectués sans aucune optimisation concernant la façon de calculer la transformée de Wigner-Ville 3D, aussi bien en ce qui concerne le pourcentage de vecteurs exacts trouvés, que la diminution des temps de calculs.

Pour l'instant, tous les calculs ont été effectués directement sur le point $P(x,y)$ qui est un point de l'image existant. Cela peut-être utilisé dans le cas de la compression par exemple, où l'on a besoin de connaître le vecteur déplacement du pixel à envoyer afin d'augmenter les taux de compression.

Mais il peut arriver que l'on ait besoin d'estimer le vecteur déplacement d'un point n'existant pas, et que nous appellerons points fictifs.

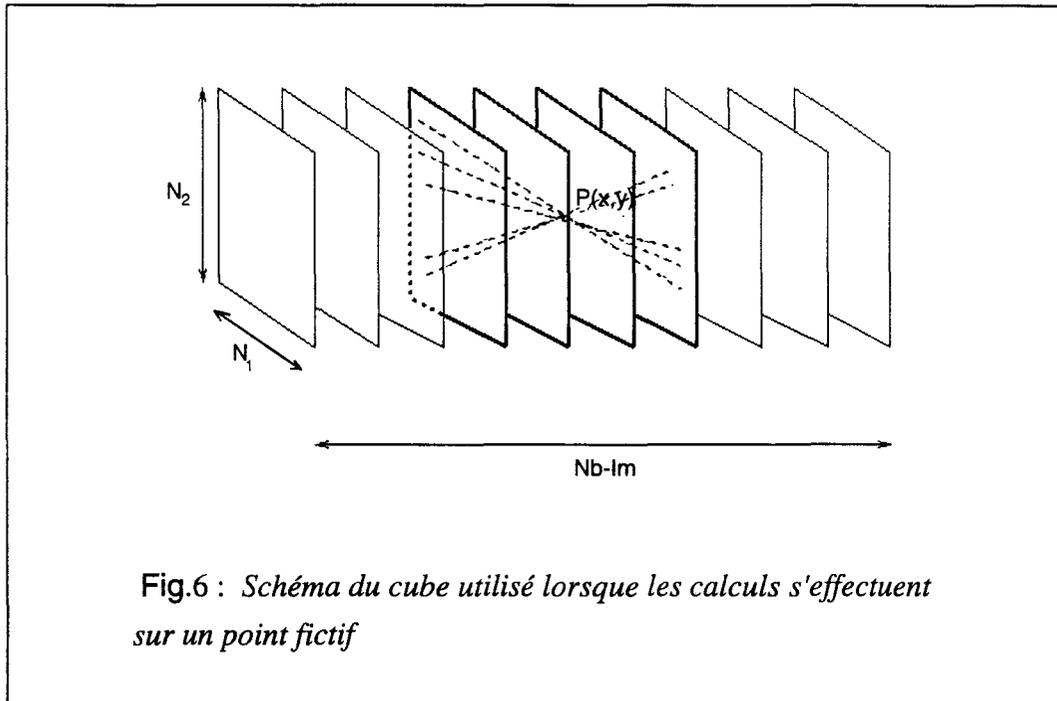
3.CALCUL SUR UN POINT FICTIF

Dans le cas où on désire améliorer la résolution de l'image en créant de nouveaux pixels, l'algorithme précédent n'est pas vraiment exploitable. Pour y remédier, nous avons essayé de calculer directement la transformée de Wigner-Ville pour le point fictif dont on désire connaître le vecteur déplacement.

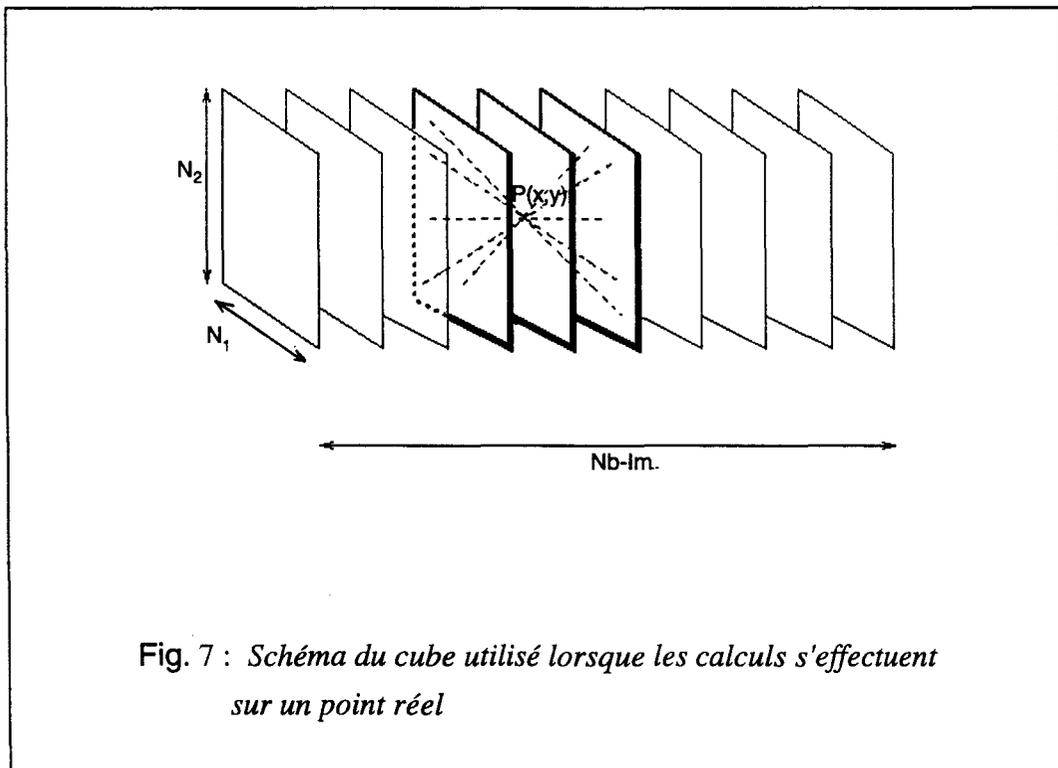
Le point de symétrie des calculs ne sera plus le point $P(x,y)$ d'une image réelle, mais un point fictif n'existant pas.

Les calculs s'effectuent alors sur un nombre d'images pair, alors que précédemment $Nb-Im$ était impair.

Les figures 6 et 7 illustrent les deux manières de former le cube centré sur le point pour lequel on recherche le vecteur déplacement suivant que ce dernier soit un point fictif (Fig.6), ou un point réel (Fig.7).



Dans le cas de la figure 6, le point de symétrie n'appartient à aucune image chargée.

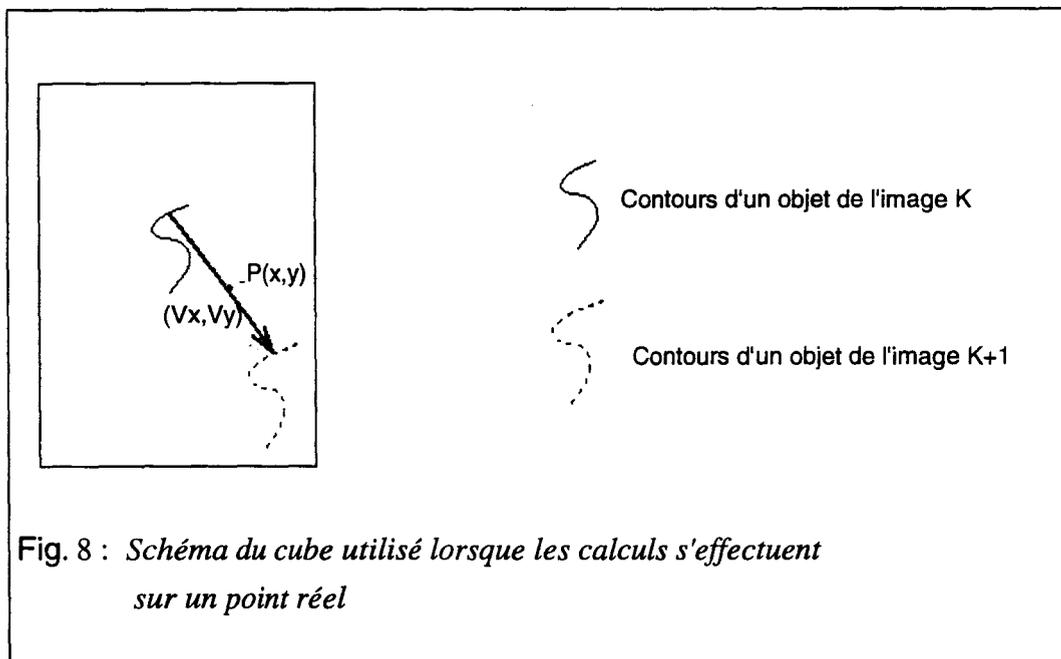


Le nombre d'images non nulles est impair, ici il est égal à trois, et le point de symétrie des calculs appartient à l'image centrale.

Si par exemple on désire créer une image intermédiaire entre deux images consécutives d'une même séquence, il est alors nécessaire d'estimer les vecteurs déplacement des pixels de l'image à créer afin de les interpoler à l'aide des images de la séquence. Dans ce cas, on utilise un point fictif comme point de symétrie des calculs.

Etant donné que l'on recherche le vecteur vitesse d'un point à reconstituer, on ne peut plus utiliser une détection des contours afin de ne travailler que sur les points caractéristiques de l'image.

La figure 8 représente le déplacement d'un contour entre deux images, ainsi que le pixel $P(x,y)$ pour lequel il faut estimer le vecteur déplacement si l'on désire recréer une image intermédiaire par exemple.



On remarque bien que le point P n'appartient à aucun contour (Fig.8). Dans ce cas, on est obligé d'effectuer les calculs sur tous les pixels de l'image.

4.OPTIMISATION DES PARAMETRES DE CALCUL

4.1.Introduction

Les précédents tests ont été effectués sans aucune optimisation concernant la manière de calculer la transformée de Wigner-Ville. Or le calcul de celle-ci fait intervenir plusieurs paramètres à optimiser afin d'améliorer les résultats.

Des tests sur le calcul du signal analytique seront ensuite effectués afin d'évaluer l'influence de ce dernier sur les résultats obtenus.

4.2.Tests sur la taille du cube : $N_1, N_2, Nb-Im$

4.2.1.Importance de la taille du cube

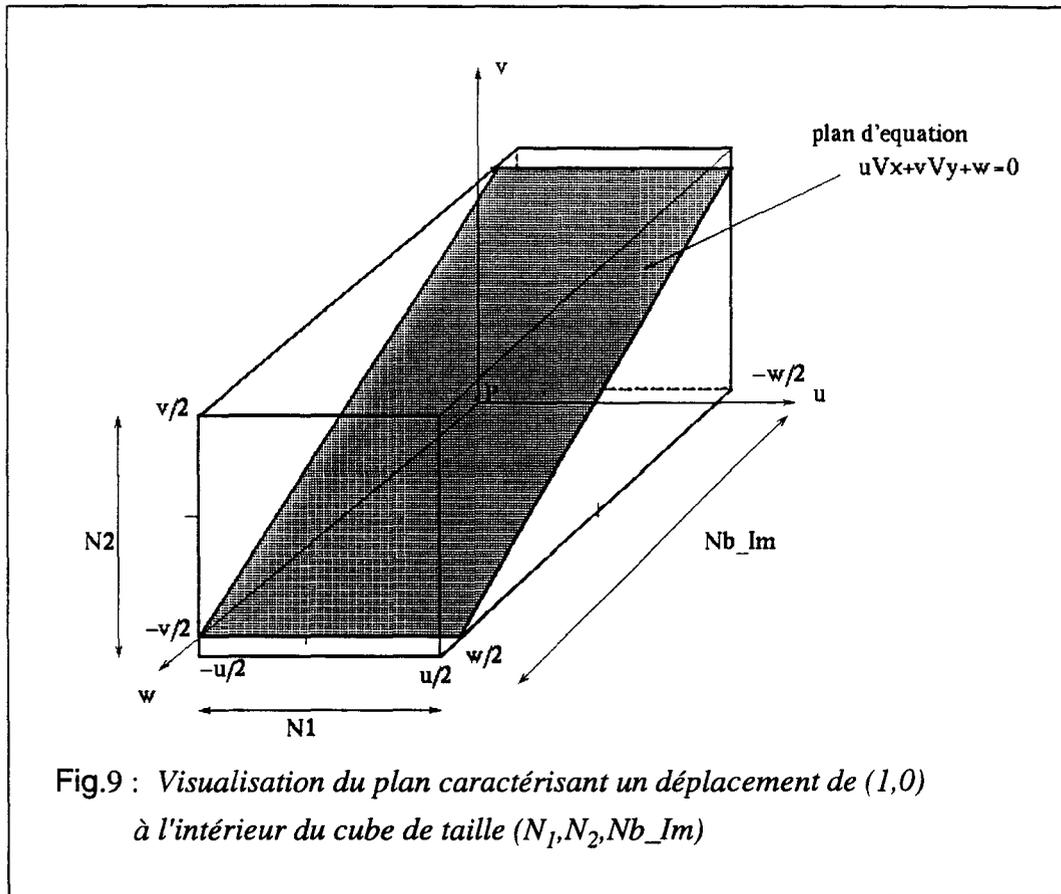
Le choix de la taille du cube jouera un rôle important sur les résultats obtenus.

En effet, le plan caractérisant le vecteur déplacement (V_x, V_y) a pour équation :

$$uV_x + vV_y + w = 0. \quad (6)$$

Les valeurs que peuvent prendre u, v et w sont fonctions de la taille du bloc sur lequel s'effectue les calculs.

Si la taille du bloc est égale à $(N_1, N_2, Nb-Im)$ u variera de $-N_1/2$ à $N_1/2$, v de $-N_2/2$ à $N_2/2$ et w de $-Nb-Im/2$ à $Nb-Im/2$.



A l'aide d'un exemple simple, on peut facilement vérifier l'importance de la taille du bloc dans la recherche du vecteur déplacement.

Par exemple, si l'on cherche à déterminer le vecteur (6,4), on devra calculer l'inclinaison d'un plan décrit par : $6u + 4v + w = 0$.

Si Nb_Im est faible ; égal à 3 par exemple ; pour les différentes valeurs de Nb_Im , on recherche les valeurs de u et v vérifiant une des équations suivantes :

$$6u + 4v = -1$$

$$6u + 4v = 0$$

$$6u + 4v = 1$$

Seul le couple $u = 0, v = 0$ et $w = 0$ vérifiera une des trois équations. Il sera alors quasiment impossible de calculer l'inclinaison du plan car le couple (0,0) appartient à tous les plans possibles. On ne pourra donc pas extraire de valeur maximale dans le tableau permettant le calcul de (V_x, V_y) .

En outre, si la taille du cube est trop faible, le résultat du calcul de la FFT3D ne sera pas suffisamment précis pour donner une représentation correcte du signal.

Par contre, si la taille du bloc est trop importante, on risque d'avoir plusieurs mouvements différents à l'intérieur de ce cube, ce qui faussera les calculs. En effet, dans ce cas, on observera plusieurs plans d'inclinaisons différentes. On tiendra en fait compte de pixels dont le mouvement n'est pas du tout en relation avec celui du point $P(x,y)$.

Avant d'effectuer les tests sur les valeurs de N_1, N_2 et $Nb-Im$, quelques modifications ont été apportées à l'algorithme de calcul de la transformée de Wigner-Ville tridimensionnelle.

4.2.2.Modification de l'algorithme de calcul

Utilisation de la transformée de Fourier rapide :

Afin de pouvoir utiliser l'algorithme rapide de calcul de la FFT, on ajoute une ligne, une colonne et une image, égales à zéro, afin que les dimensions du cube, N_1, N_2 et $Nb-Im$, soient égales à une puissance entière de deux.

La figure 10 illustre la manière dont nous avons modifié l'algorithme.

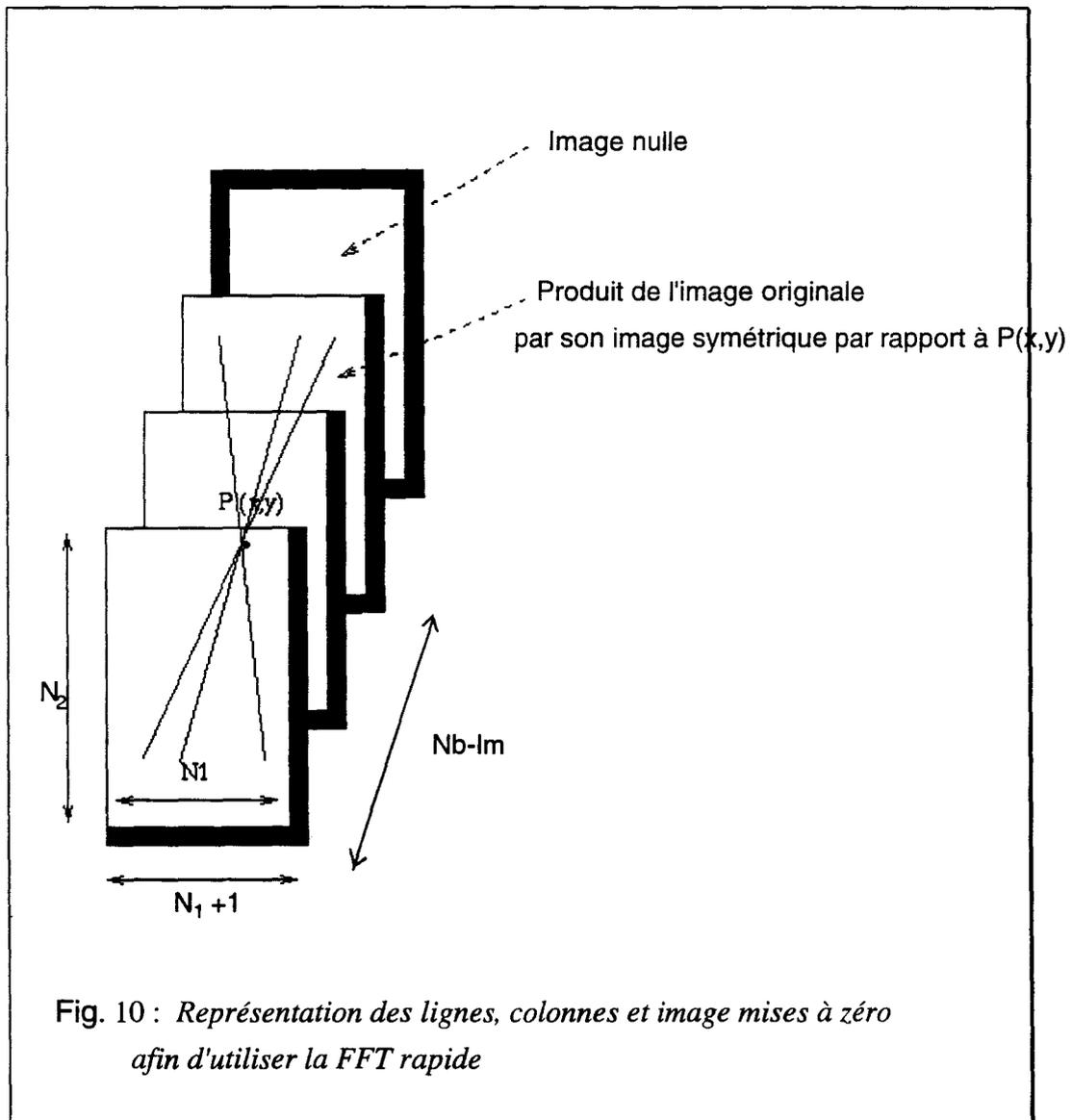


Fig. 10 : Représentation des lignes, colonnes et image mises à zéro afin d'utiliser la FFT rapide

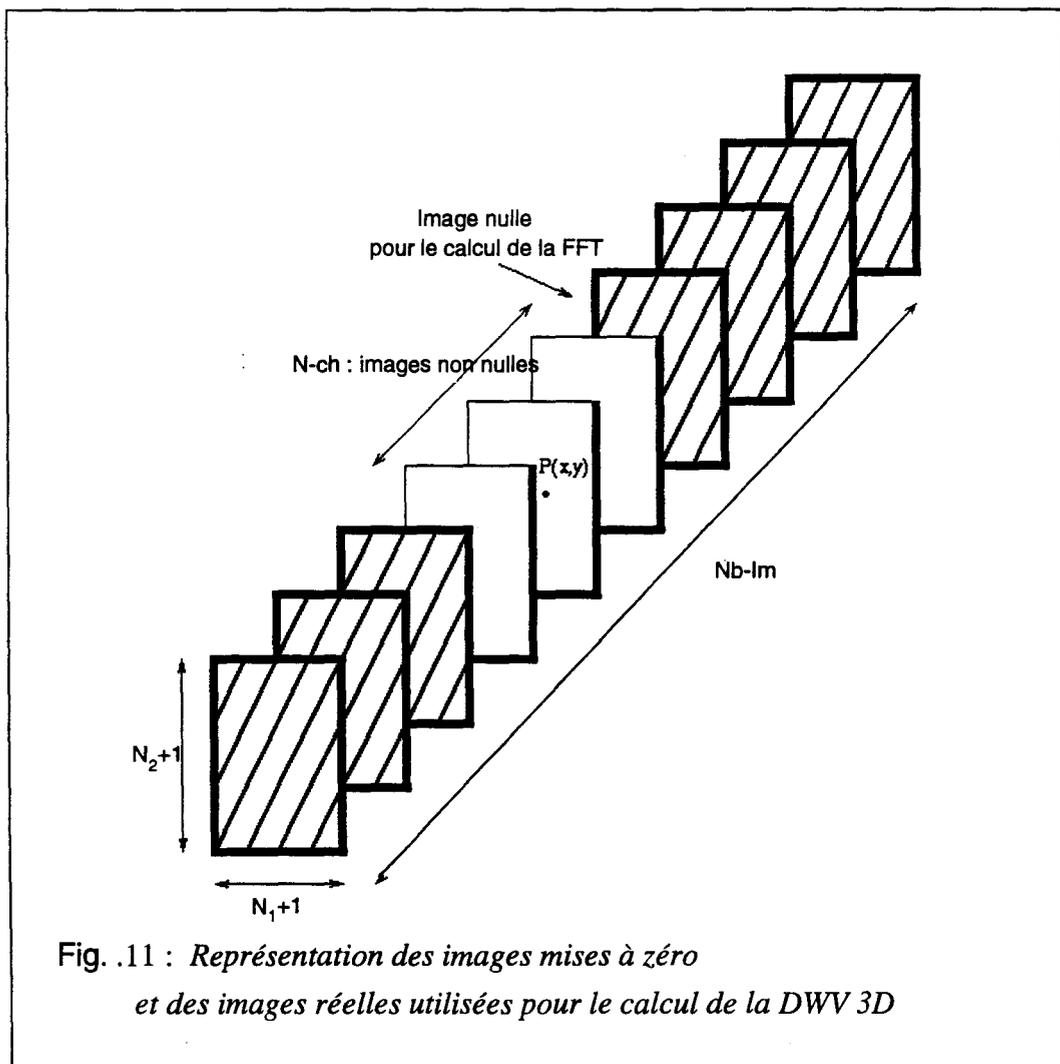
On peut donc désormais utiliser des blocs de taille $(4*4*4)$ ou $(8*8*8)$, $(64*64*64)$ etc..

Modification du nombre d'images chargées :

Les calculs s'effectuent sur Nb-Im images chargées qui peut varier de 4 à 32, avec une meilleure résolution temporelle pour un grand nombre d'images chargées. Cependant, il semble difficilement concevable que l'on puisse avoir un mouvement de translation homogène sur un tel nombre d'images dans le cas d'une séquence réelle.

Pour y remédier, plutôt que de charger 16 ou 32 images, nous avons essayé de réduire le nombre d'images utilisées pour le calcul de la transformée de Wigner-Ville tridimensionnelle. On ne charge alors que N-ch images centrales de part et d'autre du pixel P de coordonnées (x,y) les autres étant mise les autre à zéro.

La figure 11 explique la manière dont cela est réalisé.



Dans le cas de la figure 11 le nombre d'images non nulles, N-ch, est égal à 3.

On travaillera donc toujours sur Nb-Im images mais avec seulement N-ch images réellement chargées, et ceci pour conserver une certaine amplitude temporelle.

4.2.3. Description des tests

Comme précédemment, la séquence est formée à partir de l'image "boy", tous les pixels de l'image ont le même déplacement, et le nom de la séquence est suivi du vecteur déplacement fixé. Par exemple, "boy20" est la séquence formée de l'image "boy" dont tous les pixels se déplacent de (2,0) à chaque image.

Les tests ont été effectués tous les 32 pixels en horizontal et en vertical. Comme tous les pixels de l'image ont le même mouvement, la taille du bloc n'a pas d'importance car on n'a pas à se préoccuper de savoir s'il existe des mouvements d'amplitudes différentes dans ce bloc.

Nous avons utilisé l'image filtrée afin de déterminer les points caractéristiques de l'images sur lesquels les calculs seront effectués. Comme on affecte le même vecteur à tout un bloc de taille (32*32), on recherchera le vecteur déplacement du point caractéristique le plus significatif. Pour cela on détectera le pixel ayant la plus forte valeur, après filtrage.

Pour le calcul de la "Fonction test" les coefficients de la transformée de Wigner-Ville sont élevés à la puissance quatre.

Nous avons effectué plusieurs tests pour les différentes tailles du cube : (15*15*15), (31*31*31) ou (63*63*63), pour chacun desquels le nombre d'images non nulles est soit égal à 3, 5 ou 9.(Tab.2)

Sequence	3 images non nulles			5 images non nulles			9 images non nulles			
	Taille du cube	15*15*15	31*31*31	63*63*63	15*15*15	31*31*31	63*63*63	15*15*15	31*31*31	63*63*63
boy-20		5%	27%	56%	11%	37%	61%	8%	33%	53%
boy22		22%	31%	55%	31%	42%	69%	27%	42%	58%
boy30		6%	70%	78%	55%	72%	86%	58%	53%	73%
boy-23		6%	25%	41%	11%	23%	53%	3%	17%	47%
boy40		39%	55%	66%	30%	55%	72%	41%	42%	56%

Tab.2 : Pourcentages de vecteurs exacts suivant la taille du bloc et le nombre d'images non nulles.

Les résultats sont meilleurs si on charge 5 images non nulles plutôt que 9, pour une taille de cube de $(31*31*31)$ ou $(63*63*63)$. Par contre, cela est l'inverse pour un cube de calcul de taille $(15*15*15)$.

Si l'on utilise 9 images pour estimer les vecteurs déplacement, on risque de prendre en compte des pixels n'ayant plus aucun rapport avec le mouvement du pixel considéré.

En comparant les résultats obtenus pour la séquence boy32 et boy50, on remarque que les résultats sont meilleurs pour un déplacement de $(5,0)$ bien que celui-ci soit plus important que $(3,2)$. Cela est dû au fait que dans un cas le mouvement est parfaitement horizontal, et dans l'autre non. Pour le mouvement égal à $(5,0)$, les plans sont parallèles et bien définis, ce qui n'est pas le cas des plans obtenus avec un déplacement de $(3,2)$.

4.2.4. Conclusion

Le meilleur compromis est de choisir un nombre d'images non nulles chargées égal à 5. En effet, si le nombre d'images véritablement chargé est trop faible (3 par exemple), on ne possède pas assez d'information temporelle pour estimer correctement le vecteur on se retrouve dans le cas où il n'existe pas assez de points appartenant au plan caractéristique du mouvement.

Il est à noter que pour des faibles mouvements, des blocs de taille $(8*8*8)$ ou $(16*16*16)$ suffisent aux calculs, mais deviennent trop petits en cas de mouvements de fortes amplitudes.

Ainsi que nous l'avons remarqué lors des tests préliminaires, les meilleurs résultats sont obtenus pour des déplacements de faibles amplitudes ou des mouvements simplement horizontaux ou verticaux. Le calcul des vecteurs déplacement est plus hasardeux lorsque les mouvements sont diagonaux.

4.3. Tests sur la manière de calculer (V_x, V_y)

Jusqu'à présent, nous avons calculé l'amplitude du vecteur déplacement en élevant les coefficients de la transformée de Wigner-Ville 3D à la puissance quatre ; et ceci afin d'éliminer au maximum les faibles valeurs de cette transformée.

Dans les tableaux ci-dessous, nous avons regroupé les différents résultats suivant la puissance utilisée. Pour chaque mouvement bien défini, nous avons comptabilisé le pourcentage de vecteurs exacts.

La séquence utilisée est formée des images boy.

4.3.1. Tests sur les points réels

Les calculs ont été effectués après filtrage de Sobel sur les points caractéristiques les plus significatifs. Les calculs ont été effectués sur les points réels avec un pas de 32 en horizontal et en vertical. Des tests ont été effectués sur d'images progressives (Tab.3a et Tab.3b) et sur des images entrelacées (Tab.4a et Tab.4b) avec un nombre d'images non nulles chargées (N-ch) est égal à 5.

Dans le premier tableau sont regroupés les résultats obtenus en élevant les valeurs de la transformée de Wigner-Ville à la puissance quatre, alors que dans le second tableau, les coefficients de la transformée sont élevés à la puissance deux.

Sequence	cube (15*15*15)	cube(31*31*31)
boy20	78%	81%
boy22	31%	47%
boy-23	8%	25%
boy40	47%	66%

Tab.3a : Coefficients à la puissance 4
Images progressives

Sequence	cube (15*15*15)	cube(31*31*31)
boy20	91%	86%
boy22	23%	33%
boy-23	5%	16%
boy40	50%	62%

Tab.3b : Coefficients à la puissance 2.
Images progressives

Sequence	cube (15*15*15)	cube (31*31*31)
boy20	85%	95%
boy22	20%	45%
boy-23	56%	25%
boy40	57%	50%

Tab.4a : Coefficients à la puissance 4.
Images entrelacées

Sequence	cube (15*15*15)	cube (31*31*31)
boy20	90%	95%
boy22	10%	35%
boy-23	0%	8%
boy40	55%	75%

Tab.4b : Coefficients à la puissance 2.
Images entrelacées

Les pourcentages obtenus sont plus importants lorsque l'on utilise une puissance de quatre pour les mouvements diagonaux. En ce qui concerne les mouvement horizontaux, les résultats diffèrent peu suivant la puissance utilisée.

On remarque aussi que le fait d'effectuer les calculs sur des images progressives ou entrelacées modifie peu les résultats obtenus.

4.3.2. Tests sur les points fictifs

Nous allons maintenant effectuer des tests similaires mais sur des points fictifs. Comme on ne peut plus utiliser un filtrage de Sobel afin de déterminer les points caractéristiques de l'image, les tests ont été effectués en prenant arbitrairement un pixel sur huit en horizontal et en vertical. Les tests ont été effectués sur des images progressives (Tab.5a et Tab.5b) et sur des images entrelacées (Tab.6a et Tab.6b). Le nombre d'images chargées non nulles est égal à 4.

	POINT FICTIF / IDWDI ⁴	
	cube (16*16*16)	cube (32*32*32)
boy20	75,5%	58,6 %
boy22	54,4%	79,8 %
boy-23	13,8%	33,6%
boy40	61,7%	58,6%

	POINT FICTIF / IDWDI ²	
	cube (16*16*16)	cube (32*32*32)
boy20	87%	79%
boy22	63,5%	81%
boy-23	28,5%	37,8%
boy40	76,4%	89%

Tab.5a : Coefficients à la puissance 4.
Images progressives

Tab.5b : Coefficients à la puissance 2.
Images progressives

Sequence	cube (16*16*16)	cube(32*32*32)
boy20	70%	80%
boy22	58%	82%
boy-23	18%	50%
boy40	30%	60%

Sequence	cube (16*16*16)	cube(32*32*32)
boy20	85%	83%
boy22	70%	83%
boy-23	19%	53%
boy40	53%	89%

Tab.6a : Coefficients à la puissance 4.
Images progressives

Tab.6b : Coefficients à la puissance 2.
Images progressives

On remarque que cette fois, les pourcentages de vecteurs exacts sont plus importants lorsque l'on élève les coefficients de la transformée de Wigner-Ville 3D à la puissance deux plutôt qu'à la puissance quatre. Cependant, l'influence de la puissance est moins importante que pour les calculs sur les points réels.

Jusqu'à présent, nous avons calculé directement la transformée de Wigner-Ville sans nous préoccuper du problème de suréchantillonnage du signal. Nous allons en étudier son influence sur les résultats.

5.SURECHANTILLONNAGE DU SIGNAL

5.1.Introduction

Le calcul de la transformée de Wigner-Ville nécessite de suréchantillonner le signal d'origine. En général on passe par le calcul du signal analytique. Il existe deux manières d'effectuer celui-ci ; soit par le biais de la FFT, soit par un filtrage de Hilbert.

On peut aussi ne pas utiliser le signal analytique et suréchantillonner directement le signal.

5.2.Utilisation de la FFT

Cette méthode consiste à calculer la FFT 3D du signal, à mettre à zéro les fréquences négatives pour ensuite repasser dans le domaine temporel. Nous n'avons pas retenu cette solution car elle suppose le calcul de deux FFT3D par pixel, ce qui augmente considérablement les calculs.

5.3.Calcul de la transformée de Hilbert par l'intermédiaire d'un filtre

Plutôt que d'utiliser la FFT afin de calculer la transformée de Hilbert, on préfère utiliser le filtre décrit dans le chapitre II. Dans ce cas, l'image entière est préfiltrée en début de calcul. Cette méthode présente l'avantage de ne pas être trop lourde en calculs car ceux-ci ne sont effectués qu'une seule fois. On obtient ainsi un signal imaginaire sur lequel on effectue la transformée de Wigner-Ville tridimensionnelle.

5.4. Préfiltrage de l'image originale

Nous avons cette fois employé un préfiltrage sur l'image entière, ce qui permet de n'effectuer qu'une seule fois les calculs de suréchantillonnage.

Les coefficients du filtre sont les suivants : 3, 25, 150, 150, -25, 3.

Le filtre est décomposable suivant X et Y.

On effectue donc le calcul de la transformée de Wigner-Ville 3D sur la séquence d'images préfiltrées.

Nous avons regroupé les résultats obtenus suivant la puissance utilisée, et le moyen de suréchantillonner le signal.

Les séquences sont formées d'images progressives.

Test sur les points réels :

	cube (15*15*15)	cube (31*31*31)
boy20	91,6%	92,6%
boy22	21,8%	47,7%
boy-23	1%	9,7%
boy40	67,5%	87,4%

Tab.7a :Coefficients à la puissance 4.

	cube (15*15*15)	cube (31*31*31)
boy20	91%	93 %
boy22	10,6%	32,5%
boy-23	0%	2%
boy40	68,7%	92%

Tab.7b :Coefficients à la puissance 2.

	cube (15*15*15)	cube (31*31*31)
boy20	89%	91,7%
boy22	69,2%	53,8%
boy-23	1,5%	8,3%
boy40	67,4%	85,8%

Tab.8a :Image préfiltréeCoefficients à la puissance 4.

	cube (15*15*15)	cube (31*31*31)
boy20	92,5%	94%
boy22	14,6%	40,4%
boy-23	0%	1,5%
boy40	69,4%	91,7%

Tab.8b :Image préfiltréeCoefficients à la puissance 2.

	cube (15*15*15)	cube (31*31*31)
boy20	91,7%	92,7%
boy22	21%	44,8%
boy-23	1,2%	9,9%
boy40	67,3%	87,3%

Tab9.a :Signal analytiqueCoefficients à la puissance 4.

	cube (15*15*15)	cube (31*31*31)
boy20	94%	93,4%
boy22	10,6%	34,5%
boy-23	0%	2%
boy40	68,5%	91,9%

Tab.9b :Signal analytiqueCoefficients à la puissance 2.

Que l'on utilise une image préfiltrée ou que l'on passe par le calcul de la transformée de Hilbert, le pourcentage de vecteurs exacts est plus important lorsque l'on élève les coefficients de la transformée de Wigner-Ville à la puissance quatre.

Cependant, le fait d'utiliser le préfiltrage en XY ou le filtre de Hilbert n'améliore pas beaucoup les résultats.

Test sur les points fictifs

	cube (16*16*16)	cube (32*32*32)
boy20	75,5%	58,6 %
boy22	54,4%	79,8 %
boy-23	13,8%	33,6%
boy40	61,7%	58,6%

Tab.10a : Coefficients à la puissance 4.

	cube (16*16*16)	cube (32*32*32)
boy20	87%	79%
boy22	63,5%	81%
boy-23	28,5%	37,8%
boy40	76,4%	89%

Tab.10b : Coefficients à la puissance 2.

	cube (16*16*16)	cube (32*32*32)
boy20	78%	82%
boy22	57%	87,5%
boy-23	12%	41,5%
boy40	62,3%	71,4%

Tab.11a : Image préfiltrée. Coefficients à la puissance 4.

	cube (16*16*16)	cube (32*32*32)
boy20	88%	79,8%
boy22	68%	89,4%
boy-23	15,7%	32,6%
boy40	77,7%	89,9%

Tab.11b : Image préfiltrée. Coefficients à la puissance 2.

	cube (16*16*16)	cube (32*32*32)
boy20	88,5%	82%
boy22	55%	83,8%
boy-23	14,8%	41,4%
boy40	73%	90,6%

Tab.12a : Signal analytique. Coefficients à la puissance 4.

	cube (16*16*16)	cube (32*32*32)
boy20	88,4%	82,8%
boy22	55,5%	83,8%
boy-23	14,8%	26,4%
boy40	73,2%	90,6%

Tab.12b : Signal analytique. Coefficients à la puissance 2.

Les résultats sont meilleurs lorsque l'on utilise une image préfiltrée ou la transformée de Hilbert.

Dans ce cas, le pourcentage de vecteurs exacts est plus important lorsque l'on élève les coefficients de la transformée à la puissance quatre plutôt qu'à la puissance deux.

6. INFLUENCE DU BRUIT

Les séquences utilisées sont formées d'images de télévision, donc naturellement bruitées. Cependant, comme tous les pixels se déplacent d'un même mouvement uniforme à chaque image, le bruit de l'image est lui aussi animé du même mouvement. Il n'intervient donc qu'indirectement dans les calculs. Le fait que l'image soit bruitée introduit tout de même des termes parasites dans la transformée de Wigner-Ville. Nous visualiserons cela dans le chapitre suivant.

Cependant, jusqu'à présent, tous les pixels de l'image avaient le même mouvement uniforme. Or, si la séquence est réellement bruitée, le bruit ne varie pas uniformément mais de façon quelconque. Pour simuler cela, nous avons additionné à la séquence en mouvement une séquence représentant du bruit, donc des pixels variant de façon totalement arbitraire.

Dans ce cas, les résultats obtenus sont beaucoup moins bons que précédemment. Cependant, on s'aperçoit que la baisse du pourcentage est due au fait que dans les zones uniformes les vecteurs sont totalement erronés, ce qui n'était pas forcément le cas sur les séquences non bruitées.

Les séquences utilisées sont formées d'images progressives.

6.1. Calcul sur les points réels de l'image

Image non suréchantillonnée

Sequence	cube (15*15*15)	cube (31*31*31)
boy20	25%	28%
boy22	6%	17%
boy-23	5%	14%
boy40	16%	14%

Tab.13a : Coefficients à la puissance 4.

Sequence	cube (15*15*15)	cube (31*31*31)
boy20	23%	28%
boy22	5%	11%
boy-23	3%	17%
boy40	20%	22%

Tab.13b : Coefficients à la puissance 2.

Image suréchantillonnée

Sequence	cube (15*15*15)	cube (31*31*31)
boy20	31%	27%
boy22	5%	9%
boy-23	2%	8%
boy40	8%	19%

Tab.14a : Coefficients à la puissance 4.

Sequence	cube (15*15*15)	cube (31*31*31)
boy20	33%	28%
boy22	6%	6%
boy-23	6%	12%
boy40	22%	23%

Tab.14b : Coefficients à la puissance 2

6.2. Test sur les points réels

Calculs sur les points fictifs de l'image. Ils ont été effectués en prenant arbitrairement un pixel sur huit.

Sequence	cube (16*16*16)	cube (32*32*32)
boy20	23%	23%
boy22	18%	21%
boy-23	11%	11%
boy40	10%	20%

Tab.15a : Coefficients à la puissance 4.

Sequence	cube (16*16*16)	cube (32*32*32)
boy20	23%	22%
boy22	22%	26%
boy-23	8%	11%
boy40	10%	20%

Tab.15b : Coefficients à la puissance 2

6.3. Test sur les images suréchantillonnées

Sequence	cube (16*16*16)	cube (32*32*32)
boy20	23%	31%
boy22	19%	25%
boy-23	6%	10%
boy40	5%	20%

Tab.16a : Coefficients à la puissance 4.

Sequence	cube (16*16*16)	cube (32*32*32)
boy20	28%	31%
boy22	24%	31%
boy-23	13%	9%
boy40	12%	26%

Tab.16b : Coefficients à la puissance 2

Dans l'ensemble, les résultats sont beaucoup moins bons que ceux obtenus sur une séquence non bruitée. Par contre les différences pour les mouvements diagonaux sont plus réduites entre les deux séries de tests. Cela s'explique par le fait que pour les mouvements diagonaux il existait déjà de nombreux termes d'interférences entre les composantes du signal. Dans ce cas, les interférences introduites par le bruit aléatoire ont une influence moindre, ce qui n'est pas le cas pour les mouvements purement horizontaux ou diagonaux.

7. PRECISION DU DEMI-PIXEL

Auparavant, tous les tests ont été effectués avec une précision du pixel. Mais il est aussi possible de calculer le vecteur déplacement avec une précision de l'ordre du demi-pixel.

Lorsque l'on recherche les valeurs de V_x et de V_y on additionne tous les coefficients de la transformée de Wigner-Ville, élevés à une certaine puissance, vérifiant l'équation :

$$uV_x + vV_y + w = 0. \quad (7)$$

Il est tout à fait possible de choisir V_x et V_y non entiers. Dans ce cas les résultats sont plus précis, mais les calculs sont aussi beaucoup plus longs.

Les résultats sont meilleurs dans le cas des faibles mouvements, ou dans le cas d'objets proches.

8.MÉTHODES DE CALCUL DE L'INCLINAISON DU PLAN

8.1.Méthode directe

Jusqu'à présent, nous avons calculé l'inclinaison du plan représentatif du vecteur déplacement pour un point donné en testant toutes les valeurs possibles de (V_x, V_y) . Cette méthode est assez lourde, cependant elle fonctionne relativement bien. Nous avons simulé artificiellement le plan pour une inclinaison donnée. Pour cela nous n'avons conservé que les valeurs de la transformée de Wigner-Ville vérifiant l'équation du plan donné. Les résultats sont exacts à 100%, ce qui est la moindre des choses car pour de telles conditions on se trouve dans un cas idéal.

Cependant cette méthode est lourde en calcul.

De plus, il faut fixer de manière arbitraire les valeurs maximales de V_x et V_y . Plus le vecteur recherché sera supposé élevé, plus on devra effectuer de tests sur différentes valeurs de (V_x, V_y) .

Si on fixe une valeur trop faible pour $V_{x_{\max}}$ et $V_{y_{\max}}$ qui représente l'amplitude maximale du déplacement, on ne peut pas trouver la valeur exacte du vecteur.

Si on fixe $V_{x_{\max}}$ et $V_{y_{\max}}$ arbitrairement très élevés, on effectuera des calculs inutiles.

Nous avons essayé d'utiliser une méthode de régression, ce qui permet d'obtenir directement la valeur de (V_x, V_y) au point considéré sans avoir besoin de construire la "Fonction test" et d'en extraire sa valeur maximale.

8.2. Calcul de (V_x, V_y) par régression

Le calcul de (V_x, V_y) consiste à rechercher le plan d'inclinaison V_x et V_y passant par un maximum de points de la transformée de Wigner-Ville. En théorie, ce plan passe par tous les points de la transformée car celle-ci est nulle sauf sur le plan à déterminer. Cela peut donc se résoudre par un calcul de régression.

Nous supposons que les valeurs maximales de la transformée sont situées sur le plan que l'on cherche à déterminer. Pour chaque fréquence (u, v) , on recherche la valeur maximale de la transformée, et on obtient ainsi une valeur que nous appellerons $N-Im$ et vérifiant l'équation suivante :

$$uV_x + vV_y + N-Im = 0. \quad (8)$$

On obtient alors le système suivant :

$$(9) \left\{ \begin{array}{l} u_0V_x + v_0V_y + N - Im_0 = 0 \\ u_1V_x + v_1V_y + N - Im_1 = 0 \\ u_2V_x + v_2V_y + N - Im_2 = 0 \\ \cdot \\ \cdot \\ \cdot \end{array} \right.$$

où (V_x, V_y) sont les inconnus.

Nous nous proposons de résoudre ce système, surdimensionné, par une régression.

De manière générale, on cherche à résoudre l'équation suivante :

$$-N-Im = uV_x + vV_y. \quad (10)$$

En multipliant cette équation par V_x et par V_y et en effectuant la somme sur toutes les équations possibles, on obtient :

$$-\sum_{u=u_0}^{u_N} N - \text{Im}.u = V_x \sum_{u=u_0}^{u_N} u^2 + V_y \sum_{u,v} vu \quad (11)$$

$$-\sum_{v=v_0}^{v_N} N - \text{Im}.v = V_x \sum_{u,v} uv + V_y \sum_{v=v_0}^{v_N} v.^2 \quad (12)$$

$$V_x = \frac{-\sum N - \text{Im}.u * \sum v^2 - \sum -N - \text{Im}.v * \sum v.u}{\sum u^2 * \sum v^2 - \sum u.v * \sum u.v} \quad (13)$$

$$V_y = \frac{\sum -Num - \text{Im}.u * \sum u.v - \sum -N - \text{Im}.v * \sum u.v}{\sum u^2 * \sum v^2 - \sum u.v * \sum u.v} \quad (14)$$

La somme s'effectue sur le nombre d'équations obtenu.

Nous avons testé cette méthode sur un plan bien défini, les résultats sont exacts à 100%.

L'avantage de cette méthode est que l'on n'est plus obligé de fixer une valeur maximale pour l'amplitude des vecteurs à estimer. De plus les calculs sont plus rapides que précédemment.

Nous avons ensuite testé cette méthode sur une image réelle. Dans ce cas les résultats sont beaucoup moins bons que le calcul direct car on ne tient pas compte de la valeur de la distribution de Wigner-Ville contrairement à la méthode précédente où l'on somme les valeurs de la transformée.

Pour y remédier, il faudrait ajouter un coefficient de pondération suivant les valeurs de la transformée.

9.VISUALISATION DES PLANS ET DE LA FONCTION TEST

Dans ce chapitre nous allons visualiser les différents plans obtenus suivant les déplacements étudiés ainsi que les fonctions tests qui leurs sont associées.

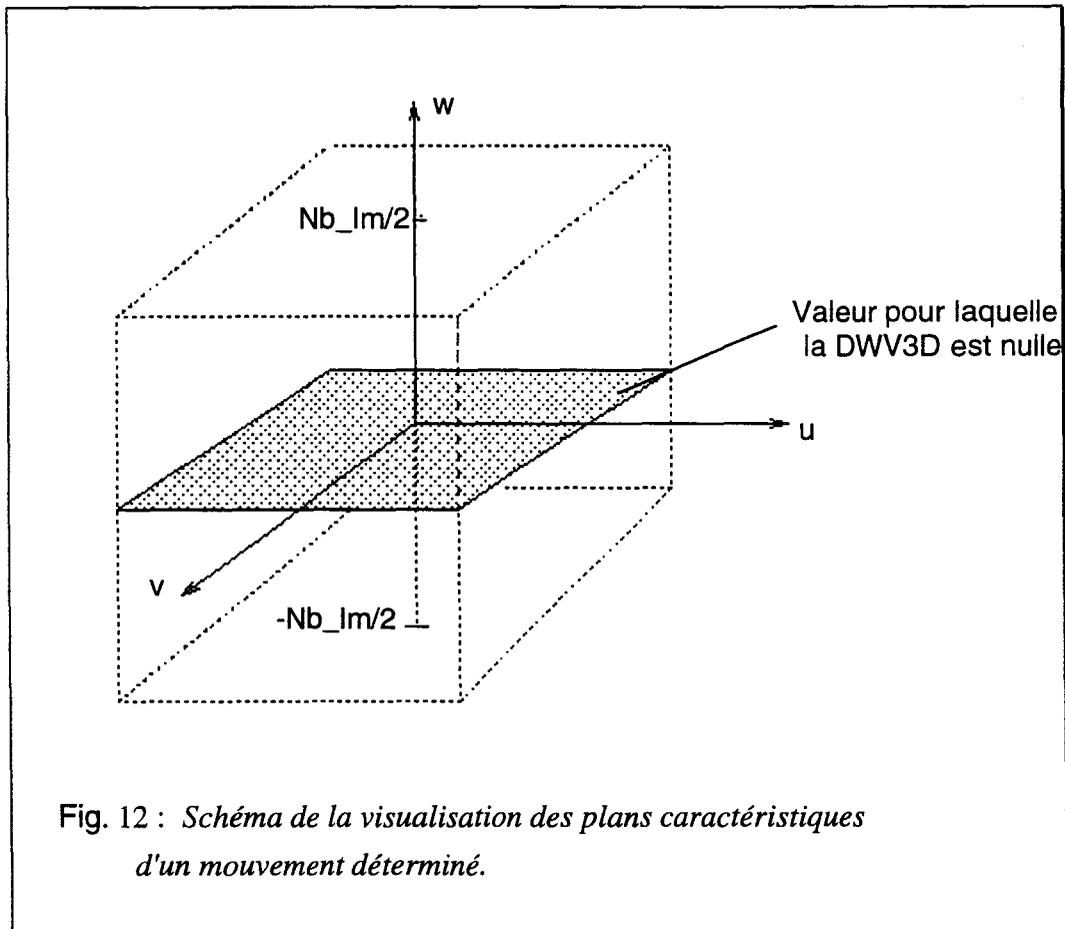
Les séquences utilisées sont soit formées d'images de synthèse, soit d'images réelles.

9.1.Description de la visualisation de plans

Afin de visualiser les plans, nous avons procédé de la manière suivante :

Pour chaque valeur de (u_0, v_0) , on visualise la valeur de ω_0 correspondant à la valeur maximale de la transformée de Wigner-Ville 3D (u, u, ω) .

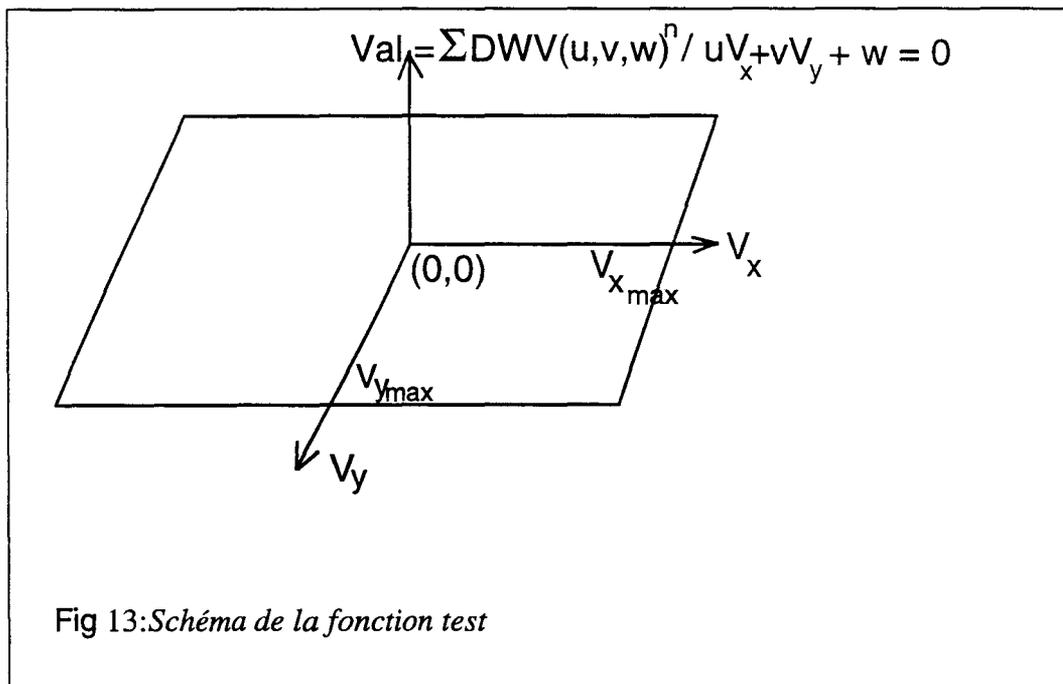
ω variant de $-Nb_{Im}/2$ à $Nb_{Im}/2$, la valeur $\omega = 0,5$ pour un couple (u_0, v_0) fixé signifie que la transformée de Wigner-Ville 3D, $DWV(u_0, v_0, \omega)$, est nulle quel que soit ω .



9.2. Description de la visualisation de la fonction test

La hauteur représente la somme des valeurs de la transformée de Wigner-Ville vérifiant l'équation du plan défini par V_x et V_y .

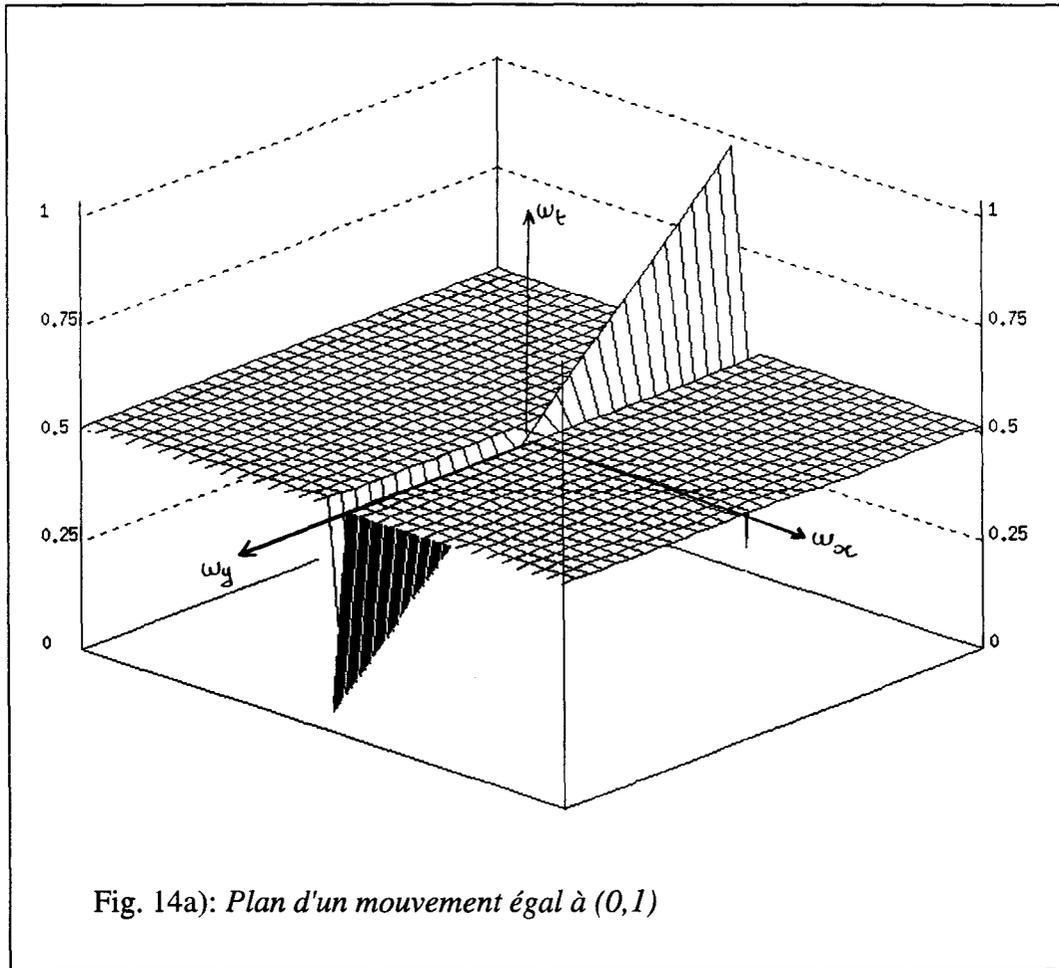
Les amplitudes maximales pour V_x et V_y sont déterminées par $V_{x\max}$ et $V_{y\max}$ qui sont des valeurs fixées par l'utilisateur.

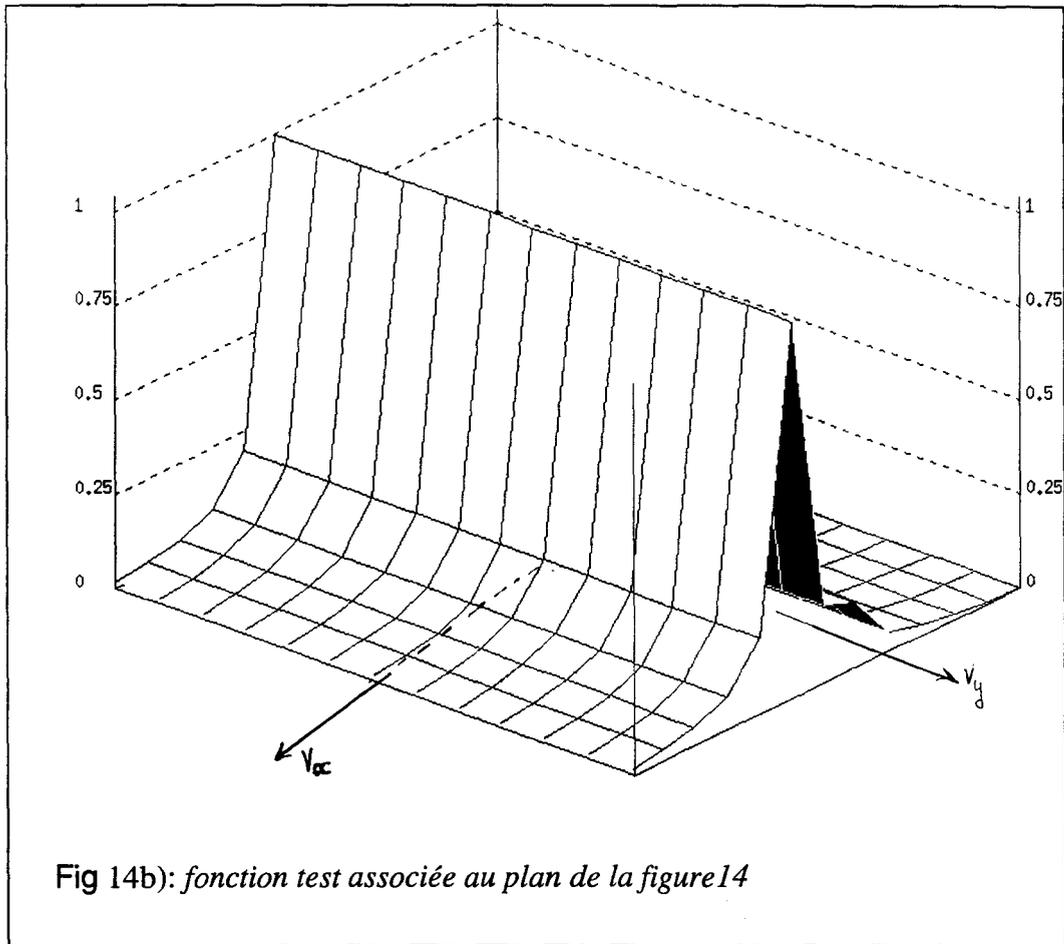


9.3. Visualisation des plans et de la fonction test

La figure 14 représente le plan caractérisant le mouvement d'un pixel appartenant à un trait horizontal se déplaçant verticalement de 1 pixel vers le bas par image.

Les calculs ont été effectués sur les points fictifs. La taille du cube de calcul est égale à $(32*32*32)$ avec 4 images non nulles.





Dans la séquence, seul le trait horizontal est non nul, cela explique pourquoi le plan représenté sur la figure 14a se limite à un seul trait.

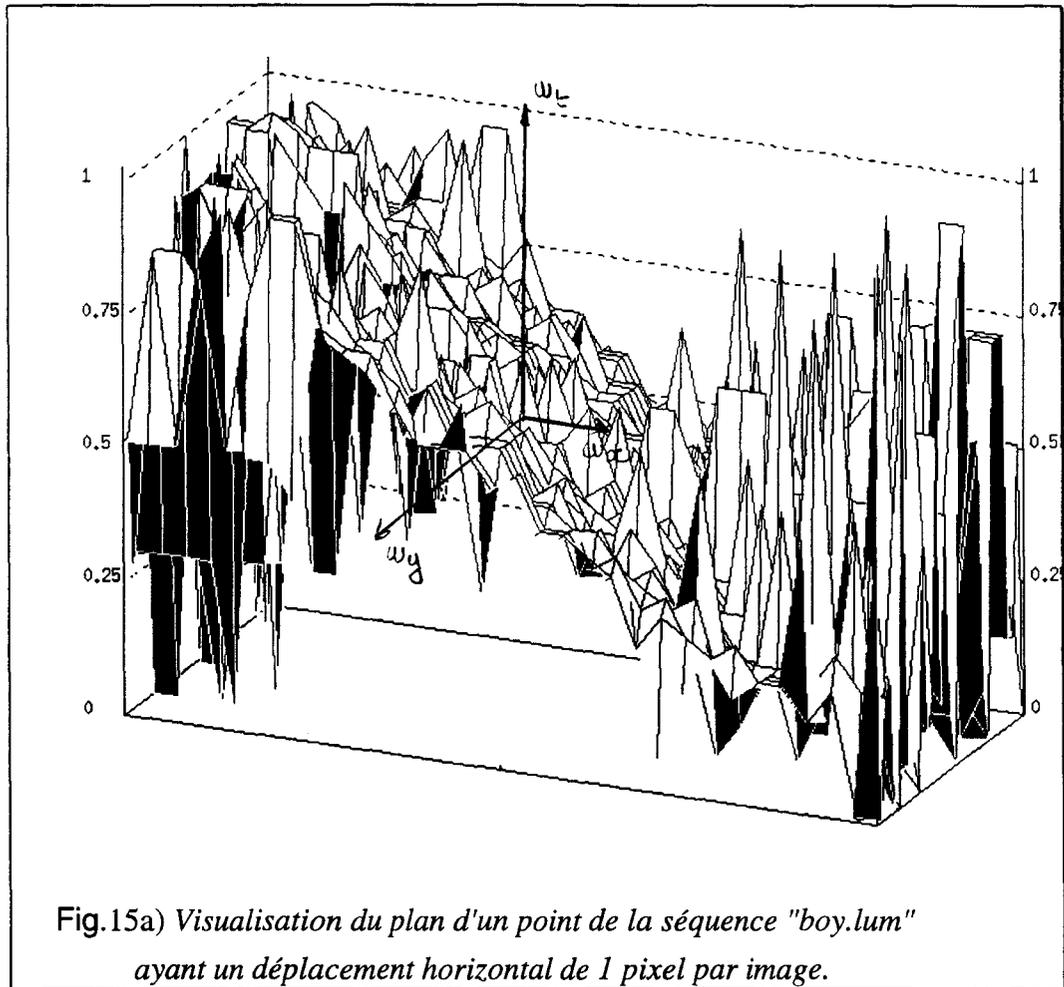
Le vecteur trouvé est exact. En fait, la valeur de V_y est égale à 1, et celle de V_x égale à la valeur maximale fixée pour $V_{y_{\max}}$ (Fig. 14b)). Ce qui s'explique par le fait que la composante verticale du déplacement est infinie le long d'un trait horizontal uniforme.

Les valeurs maximales des amplitudes des vecteurs sont les suivantes : 6 en horizontal et 9 en vertical. Cela signifie que V_x peut varier de -3 à 3 et V_y de -4 à 4.

Le pic de la fonction test est situé à $V_y = 1$, et son amplitude est constante sur toutes les valeurs de V_x possibles.

Nous avons ensuite visualisé les plans obtenus pour la séquence "boy10.lum" (Fig.15a)).

La différence avec la séquence précédente est que celle ci n'est plus une image de synthèse mais une image réelle.



On remarque que pour un même mouvement, le plan est moins bien défini que le plan précédent à cause des termes parasites liés au bruit de l'image. Cependant, comme tous les pixels ont le même mouvement égal à (1,0), les pixels de l'image correspondant à du bruit auront aussi ce même déplacement.

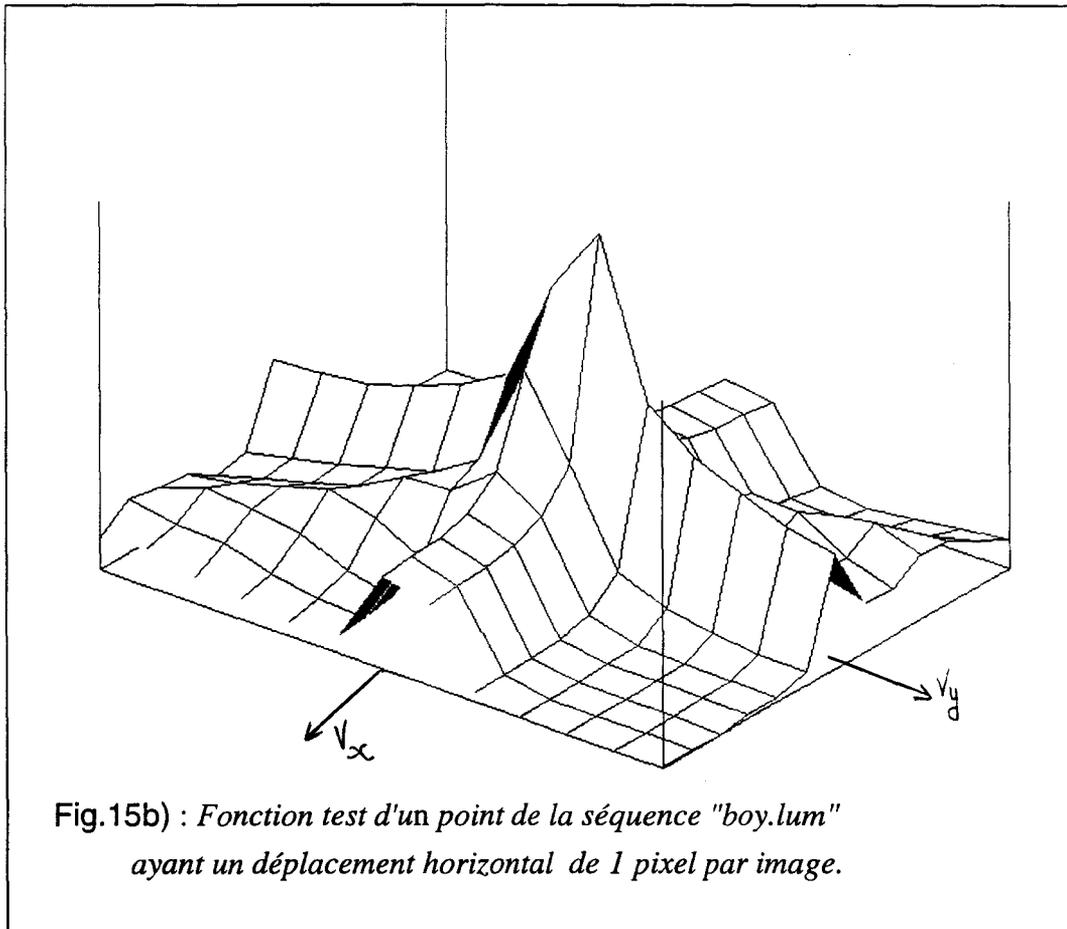
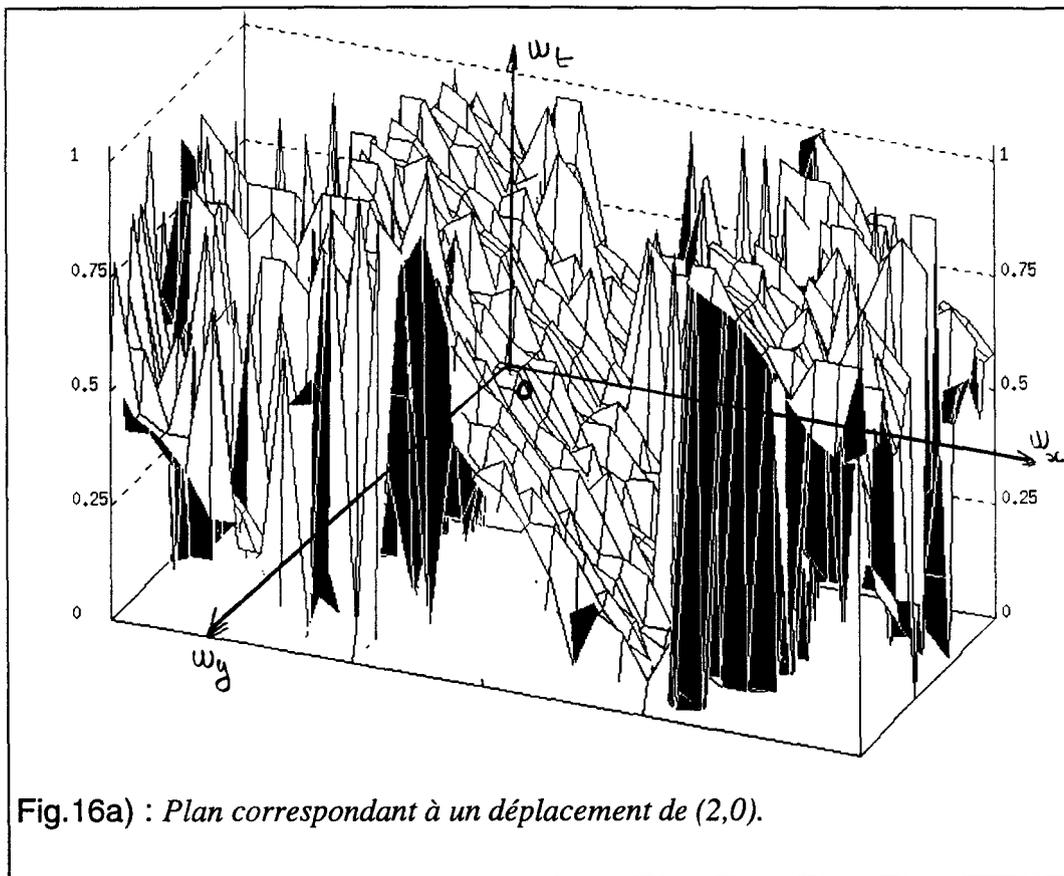


Fig.15b) : *Fonction test d'un point de la séquence "boy.lum" ayant un déplacement horizontal de 1 pixel par image.*

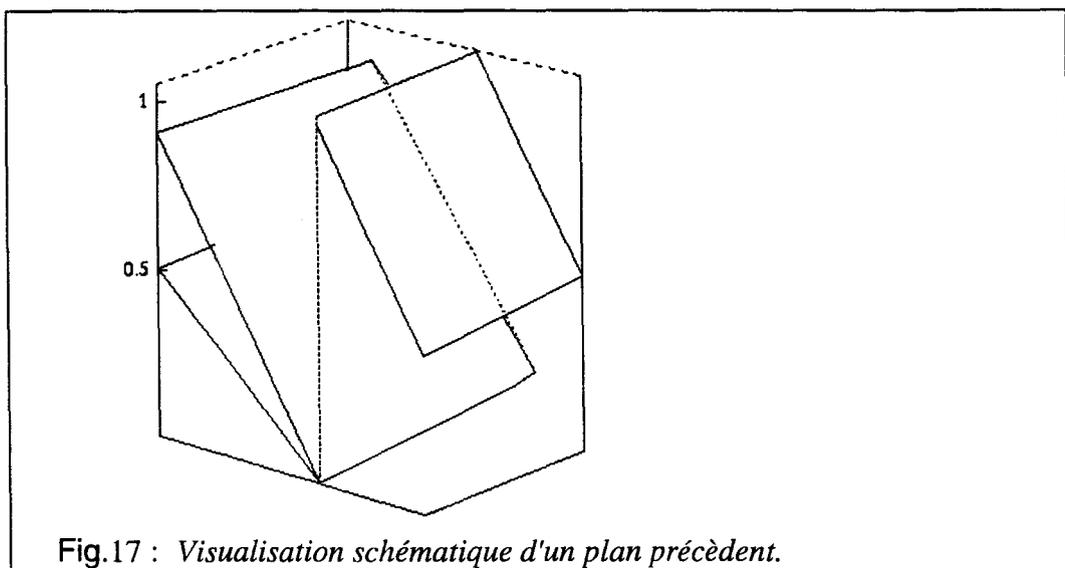
Nous avons visualisé le plan obtenu sur un signal non bruité et celui obtenu sur un signal bruité. On remarque l'apparition de nombreux termes d'interférences dans le second cas.

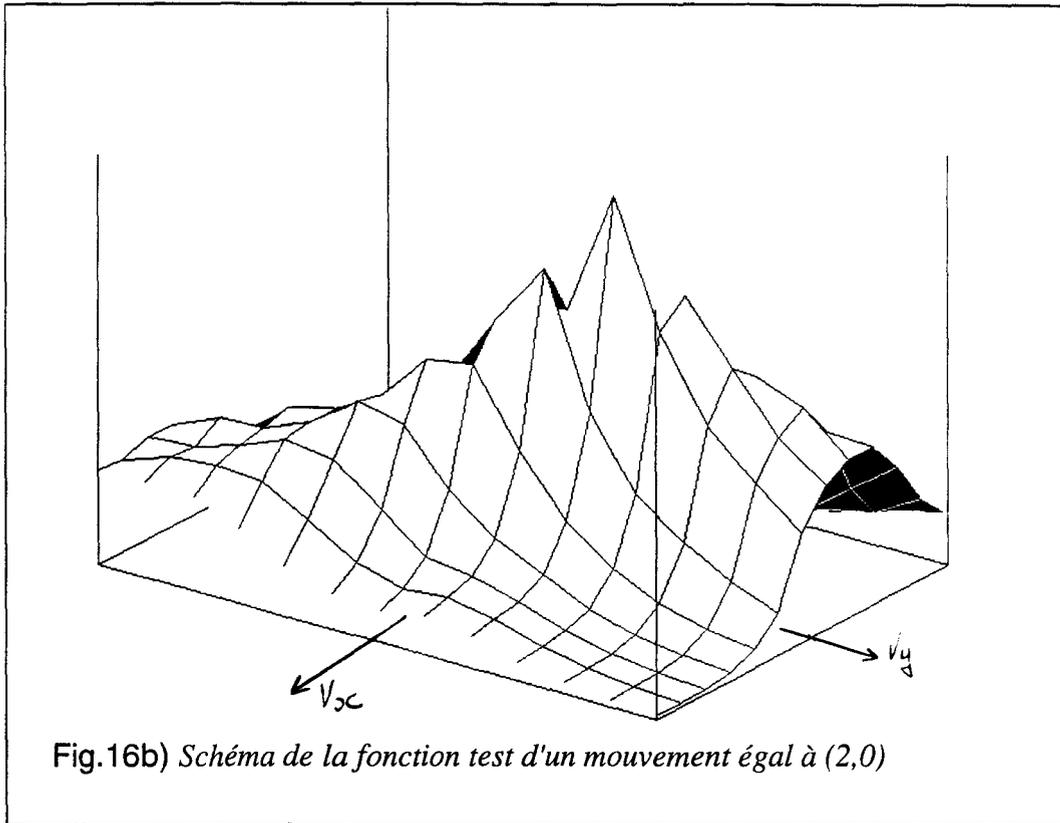
Toutefois, le vecteur trouvé est exact. Le calcul des vecteurs déplacement, bien que long en calculs, est assez robuste au bruit.

Nous avons ensuite visualisé le plan correspondant à un déplacement de 2 pixels vers la droite en horizontal par image, calculé sur la séquence boy (Fig.16a)).

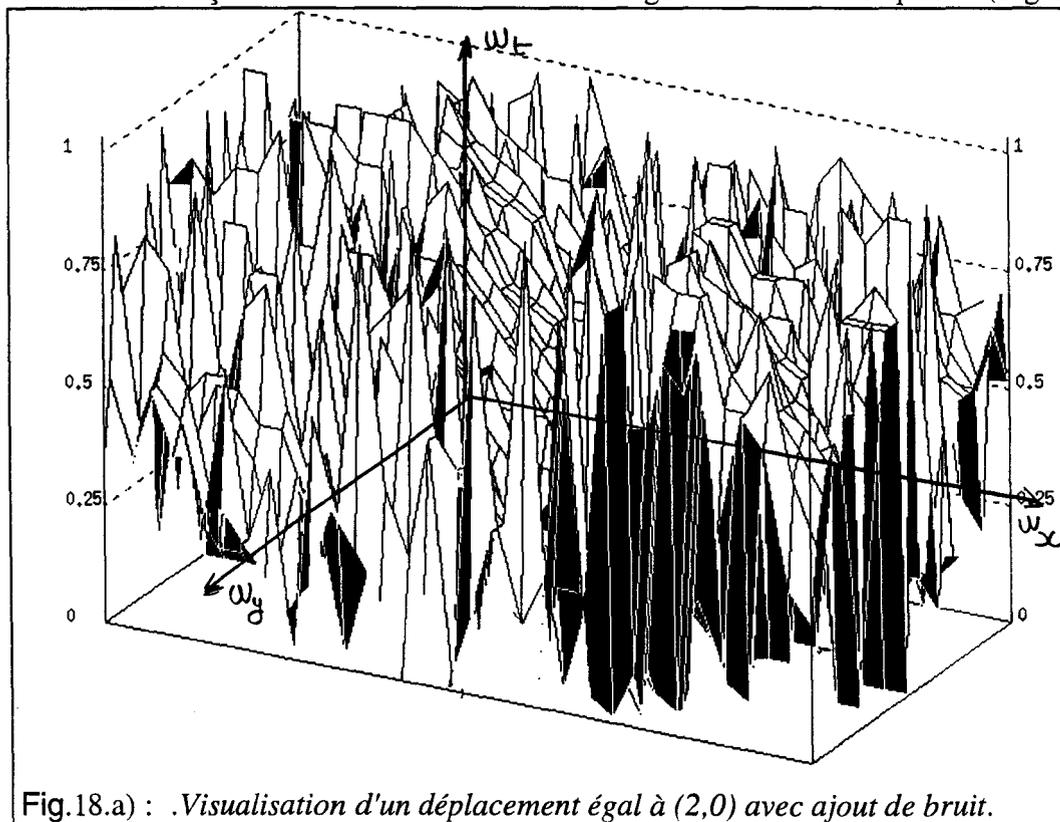


On remarque que l'on n'a plus un seul plan remplissant la totalité du cube. Schématiquement, les plans sont situés de la manière représentée sur la figure 17

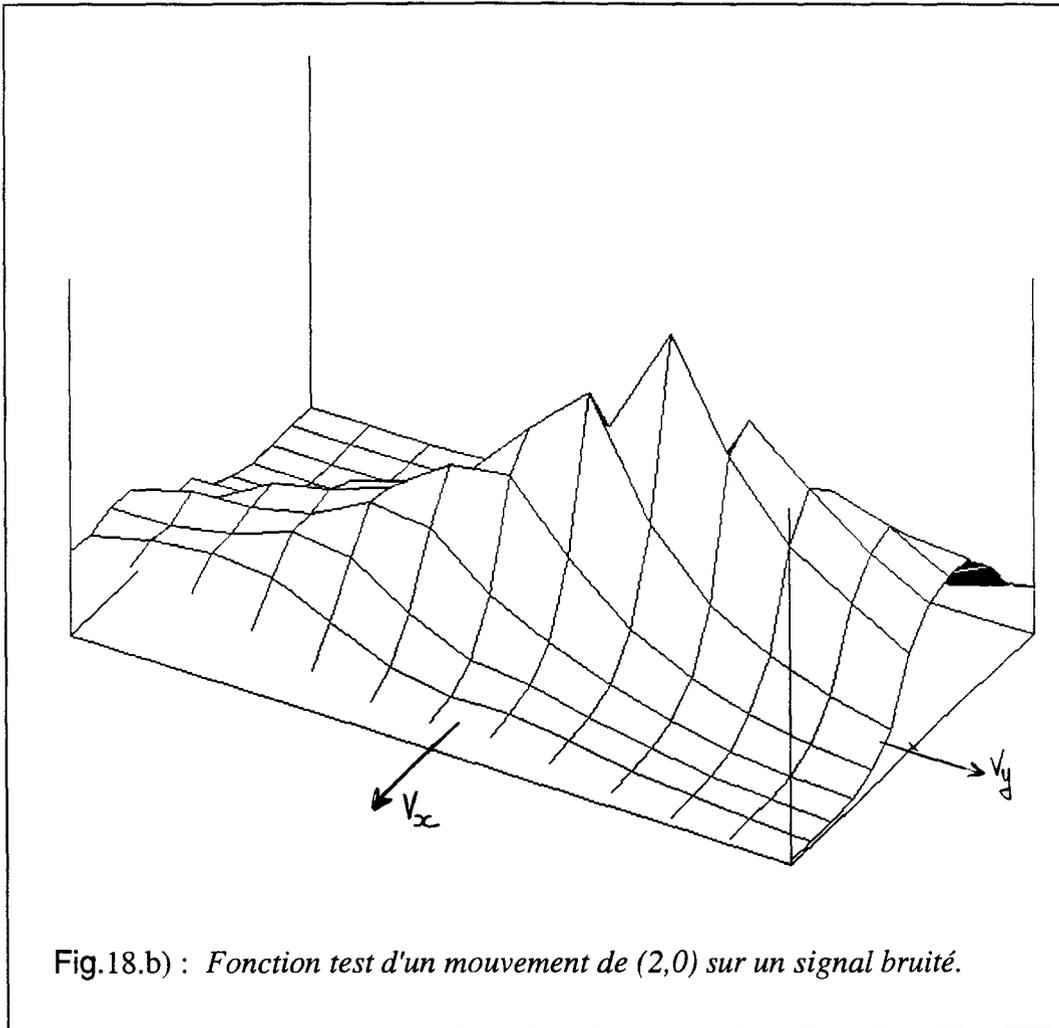




Nous avons aussi effectués des tests sur une séquence pour laquelle nous avons rajouté du bruit variant de façon totalement aléatoire d'une image à l'autre de la séquence (Fig.18.a).



Les plans caractéristiques du déplacement sont encore présent, mais ils sont plus difficiles à distinguer (Fig.18b).



La figure 19a) représente le plan caractérisant un mouvement égal à $(2,0)$ par image lorsque les calculs sont effectués sur le signal analytique.

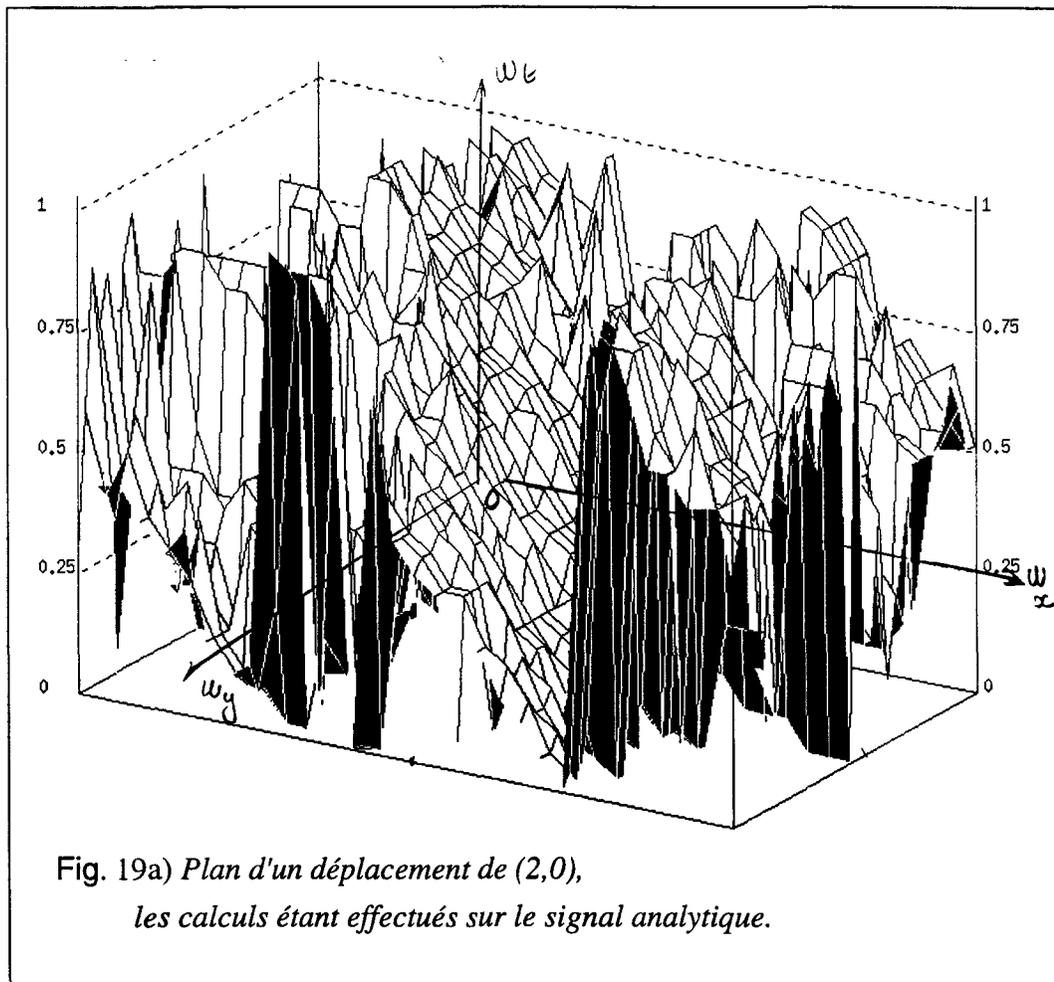


Fig. 19a) Plan d'un déplacement de $(2,0)$,
 les calculs étant effectués sur le signal analytique.

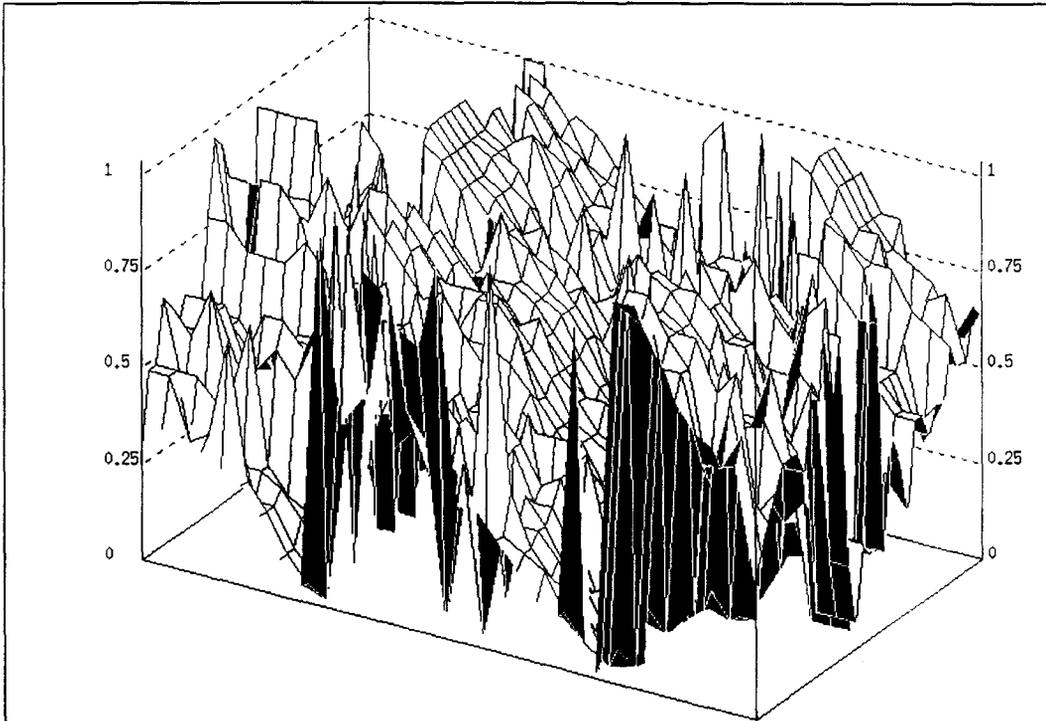


Fig. .20a) : *Calculs effectués sur la séquence préfiltrée.*

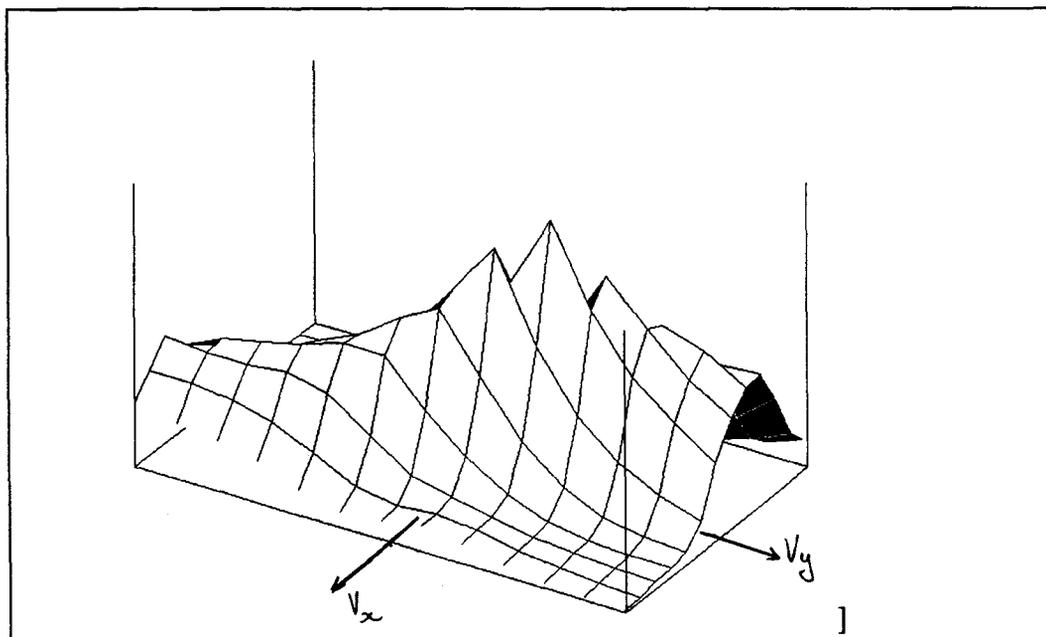


Fig. .20b) : *fonction test de la séquence préfiltrée.*

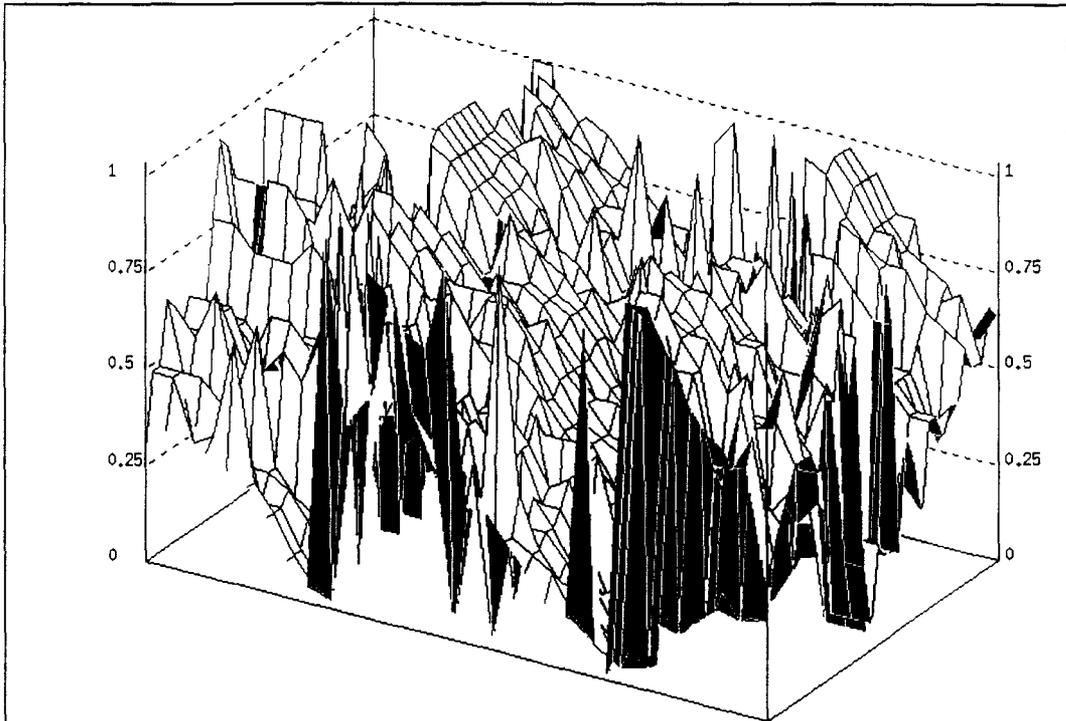


Fig. .20a) : Calculs effectués sur la séquence préfiltrée.

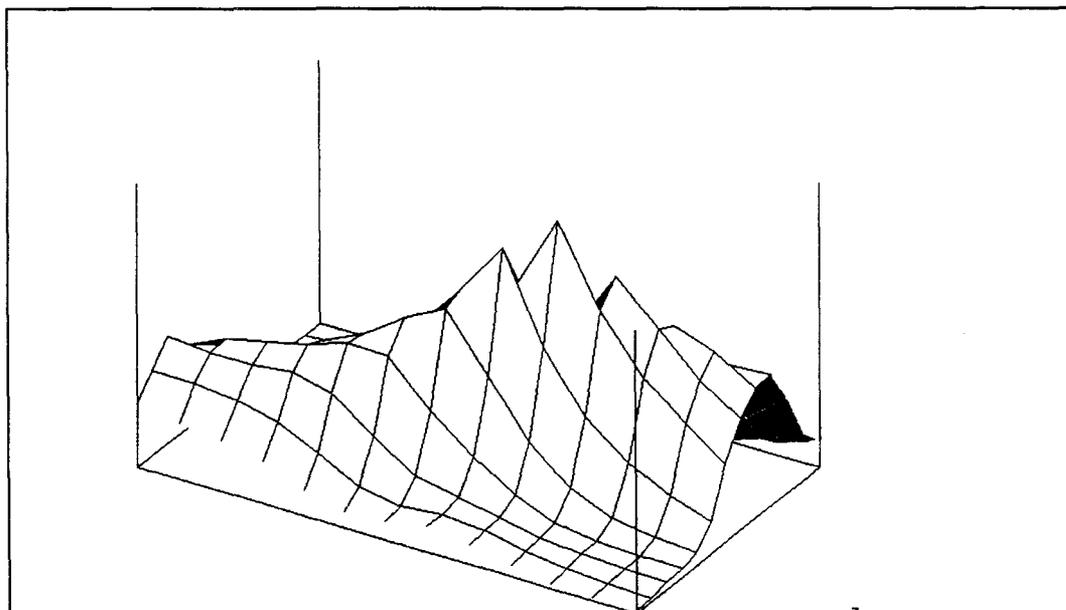


Fig. .20b): fonction test de la séquence préfiltrée.

10.CONCLUSION

Les tests effectués sur les séquences fictives sont assez prometteurs car il est possible de calculer les vecteurs déplacement sur une séquence réelle.

Seulement, jusqu'à présent, les résultats n'ont été effectués que sur des séquences dont le mouvement était uniforme au cours du temps.

L'étape suivante consiste à tester l'algorithme sur des images dont le mouvement n'est pas constant. Pour cela nous nous proposons d'utiliser les images de télévision.

En effet, le système de visualisation actuel présente quelques désagréments.

L'emploi des vecteurs déplacement peut être un plus afin d'améliorer la qualité des images.

CHAPITRE IV

APPLICATION AUX IMAGES DE TELEVISION

Après avoir testé notre algorithme sur des images simples, nous l'avons appliqué sur des images de télévision.

Le mode de visualisation actuel présente certains défauts que nous exposerons dans le premier chapitre.

Pour y remédier, une solution est de modifier la mode de visualisation, ou de modifier la fréquence d'apparition des trames à l'écran.

La solution proposée consiste généralement à doubler la fréquence de trames et donc passer en 100Hz. D'autant plus, que cela peut être effectué simplement en doublant les trames existantes, ainsi que nous l'expliquons dans le deuxième chapitre.

Cependant, bien que très simple, cette méthodes présente encore de nombreux défauts.

La seule solution acceptable est d'utiliser les vecteurs déplacement afin d'interpoler correctement les pixels manquants pour passer au mode 100Hz.

Dans le troisième chapitre, nous exposerons les principaux résultats obtenus grâce aux vecteurs calculés à partir de la transformée de Wigner-Ville tridimensionnelle.

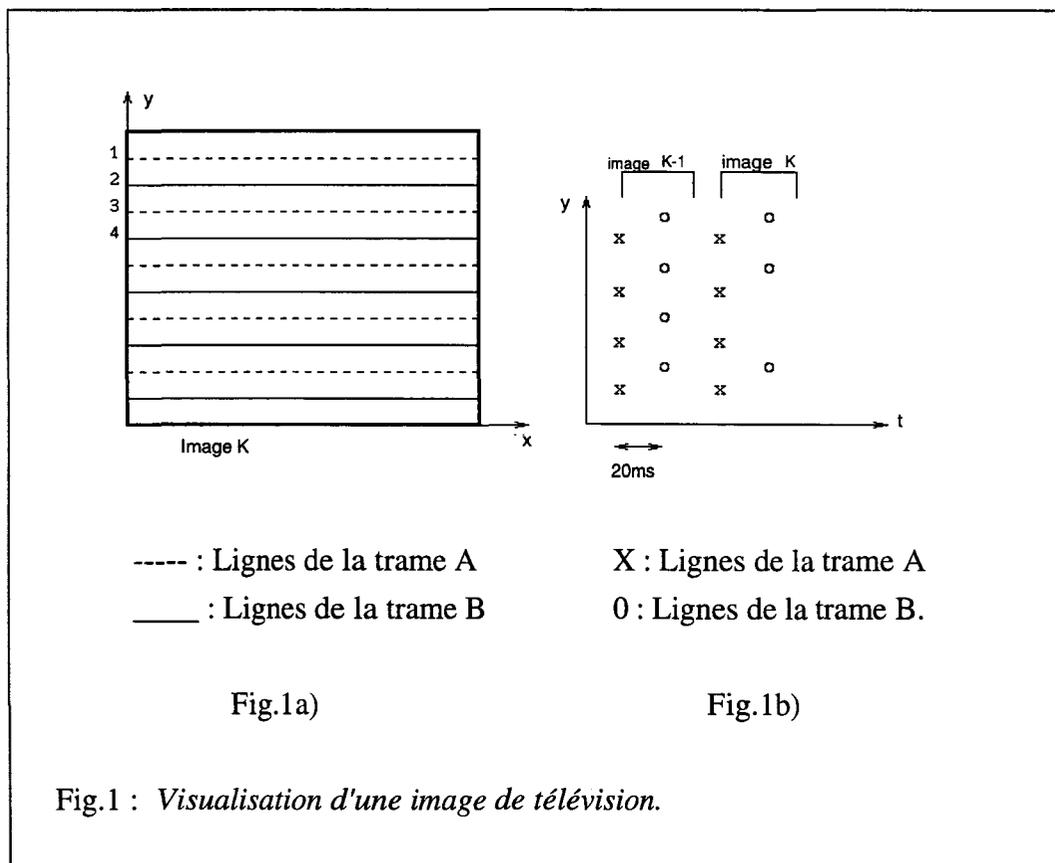
1.LE SYSTÈME DE TÉLÉVISION

1.1.Généralités

Une image de télévision est composée de pixels (picture element). Ce sont de petites surfaces élémentaires porteuses d'une information photométrique (luminance) et colorimétrique (chrominance). Une séquence d'images peut se décomposer en plusieurs images se succédant à une fréquence telle qu'elles sont invisibles pour l'œil humain.

Une image est formée de deux trames entrelacées. La première trame est constituée des lignes impaires de l'image, et la deuxième des lignes paires ; c'est ce que l'on appelle le mode entrelacé.

La figure 1 représente de deux façons différentes la visualisation d'une image entrelacée.



La figure 1.a représente la partie visible du signal vidéo telle qu'on la "voit" à l'écran, avec en traits discontinus la trame impaire et en continus la trame paire.

Pour la figure 1b on a préféré négliger l'échelle horizontale en faveur de celle du temps, et de la dimension verticale.

Pour simplifier, on suppose que le balayage d'une trame s'effectue quasi instantanément. Les croix et les cercles représentent respectivement les lignes des trames impaires et paires.

En mode entrelacé, chaque image est constituée de deux trames successives que nous noterons A et B.

Par exemple, l'image 1 est formée des trames A_1 et B_1 , l'image 2 des trames A_2 et B_2 et ainsi de suite. La première trame étant constituée des lignes impaires, la seconde des lignes paires.

1.2. Les défauts du 50Hz entrelacé

En France, la norme utilisée est du 50Hz entrelacé, c'est à dire que les trames se succèdent à la fréquence de 50Hz.

Communément, on note cela : 50/1/2.

Le 50Hz progressif est noté 50/1/1

Le 100Hz progressif est noté 100/1/1.

L'utilisation du 50Hz entrelacé présente plusieurs inconvénients que nous allons passer en revue.

1.2.1. Le papillotement des grandes plages (Large area flicker)

Ce défaut est très visible dans les zones brillantes de l'image, essentiellement quand l'observateur est assez loin de l'écran (4 à 5 fois sa diagonale).

L'œil ne perçoit pas les lignes mais une région continue car le stimulus de l'écran ne peut pas être intégré parfaitement par les yeux (Fig.2).

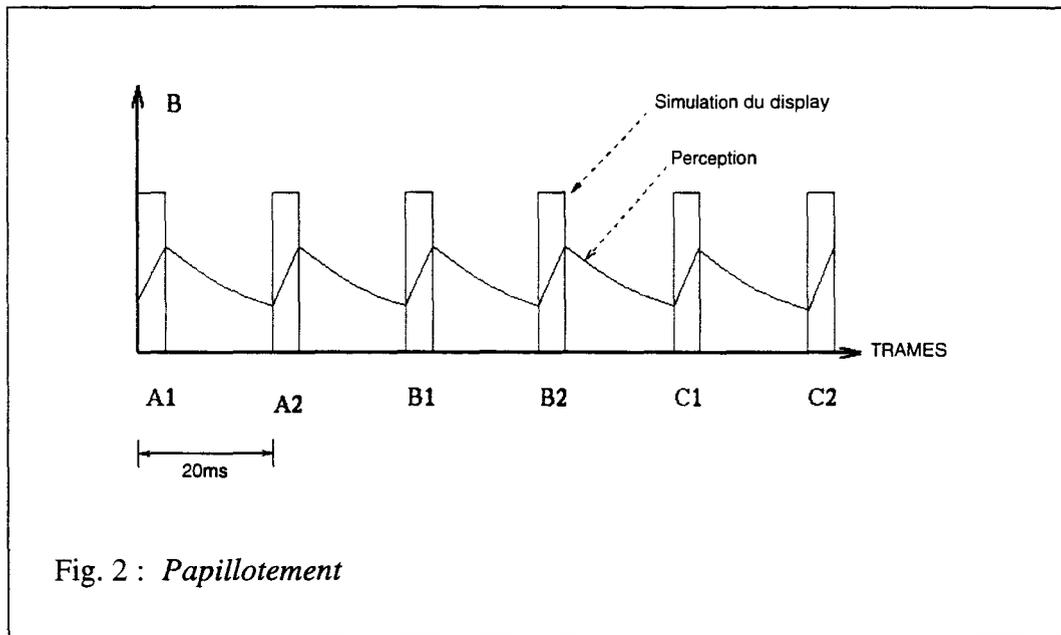


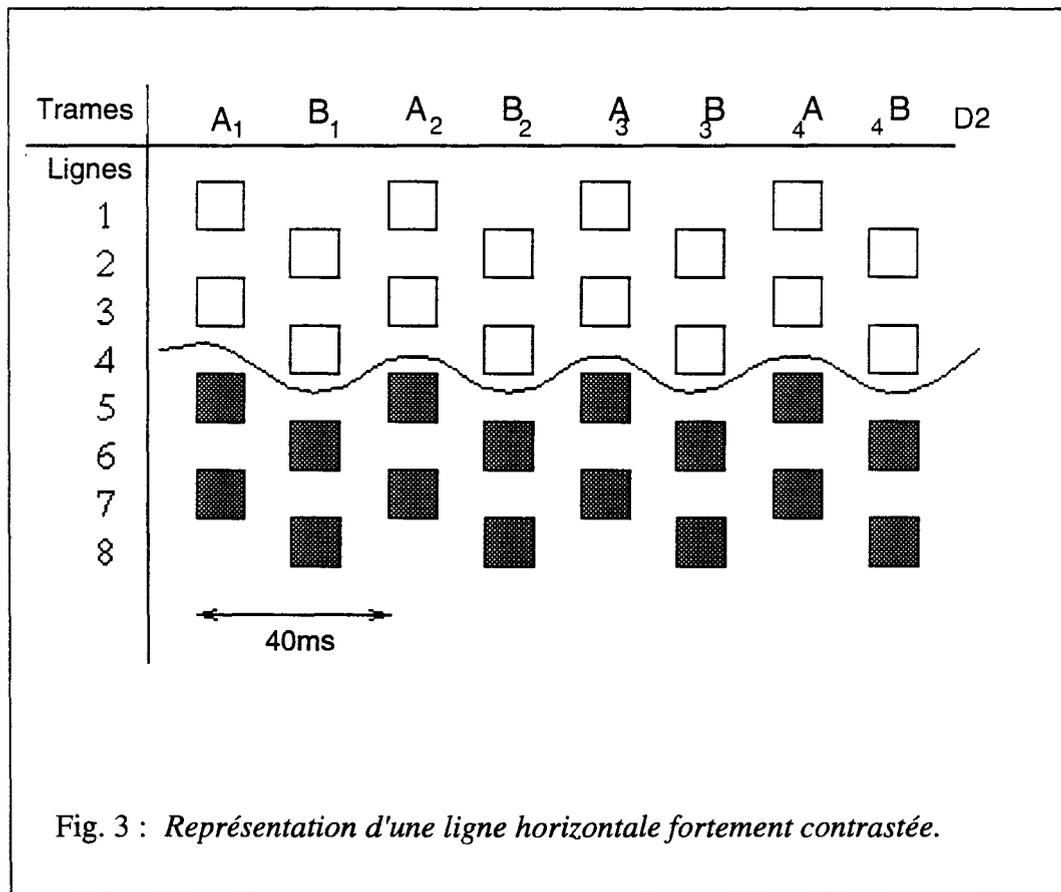
Fig. 2 : Papillotement

Pour éliminer ce papillotement, il est nécessaire d'augmenter la fréquence de trame, et ce suffisamment pour que l'œil perçoive une luminosité constante.

1.2.2. Le scintillement interligne (line flicker)

Le mode de balayage entrelacé fait défaut dans le cas de lignes horizontales avec un fort contraste (Fig.3). En effet, l'œil étant stimulé tous les 30 ms environ (33,3ms), le scintillement interligne à 25 Hz est visible, puisque toutes les fréquences inférieures à 30 Hz sont perçues par l'œil.

Les solutions possibles sont soit d'utiliser un balayage progressif, soit d'utiliser un balayage entrelacé mais avec une fréquence de trame supérieure à 60 Hz.



A cause du mode entrelacé, on a l'impression que la ligne "monte et descend" à chaque visualisation de trame.

Si la fréquence des trames est suffisamment importante, l'œil ne percevra plus cet effet de scintillement. On ne l'élimine pas, mais il n'est plus perçu par le spectateur. Par contre, si on passe en mode progressif le scintillement disparaît complètement.

1.2.3. Le défilement de lignes (line crawling)

Ce phénomène apparaît lorsque l'œil suit un mouvement vertical relativement lent. A cause d'un effet de stroboscope, l'œil suit la structure de ligne.

Ce défaut peut-être réduit en convertissant l'image en format progressif (50 Hz /1/1) ou bien en augmentant la fréquence trame.

Le mouvement perçu par l'œil est représenté par les flèches en trait plein.

Plutôt que de voir un mouvement vertical, le spectateur a l'impression de voir un mouvement diagonal, représenté par les flèches en pointillé. (Fig.4).

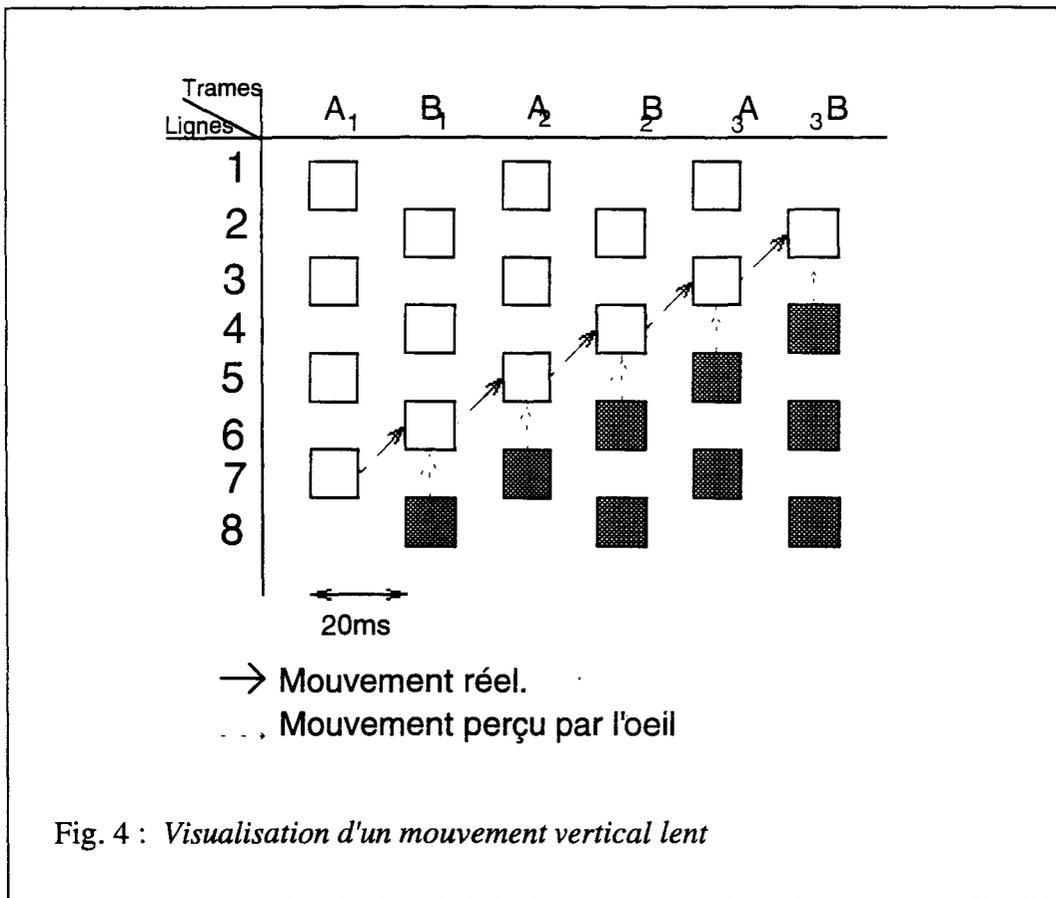


Fig. 4 : Visualisation d'un mouvement vertical lent

Il est donc possible de supprimer, ou tout au moins de réduire les défauts que génère le 50 Hz entrelacé soit en convertissant l'image en format progressif, soit en doublant la fréquence de trame. La première solution (50 Hz progressif) n'élimine pas le scintillement large bande.

La solution retenue est donc de passer au 100 Hz, ce qui a pour effet de translater les défauts à des fréquences moins visibles pour l'œil humain.

2.LA CONVERSION 100HZ

Le passage du 50Hz au 100Hz peut s'effectuer en doublant les trames existantes. Dans ce cas, il n'est pas nécessaire de connaître les valeurs des vecteurs déplacement des pixels de l'image.

Une solution plus élaborée consiste à créer les trames manquantes grâce aux pixels existants et à leur vecteur déplacement.

2.1.Passage au 100Hz sans compensation de mouvement

Soit une image 50 Hz entrelacée, constituée de deux trames A et B.

Pour passer en 100Hz, de manière simple, c'est à dire sans estimation de mouvement, on peut utiliser soit le mode "AABB", soit le mode "ABAB".

2.1.1.Le mode "AABB"

On peut représenter ce mode de visualisation de la manière suivante (Fig.5).

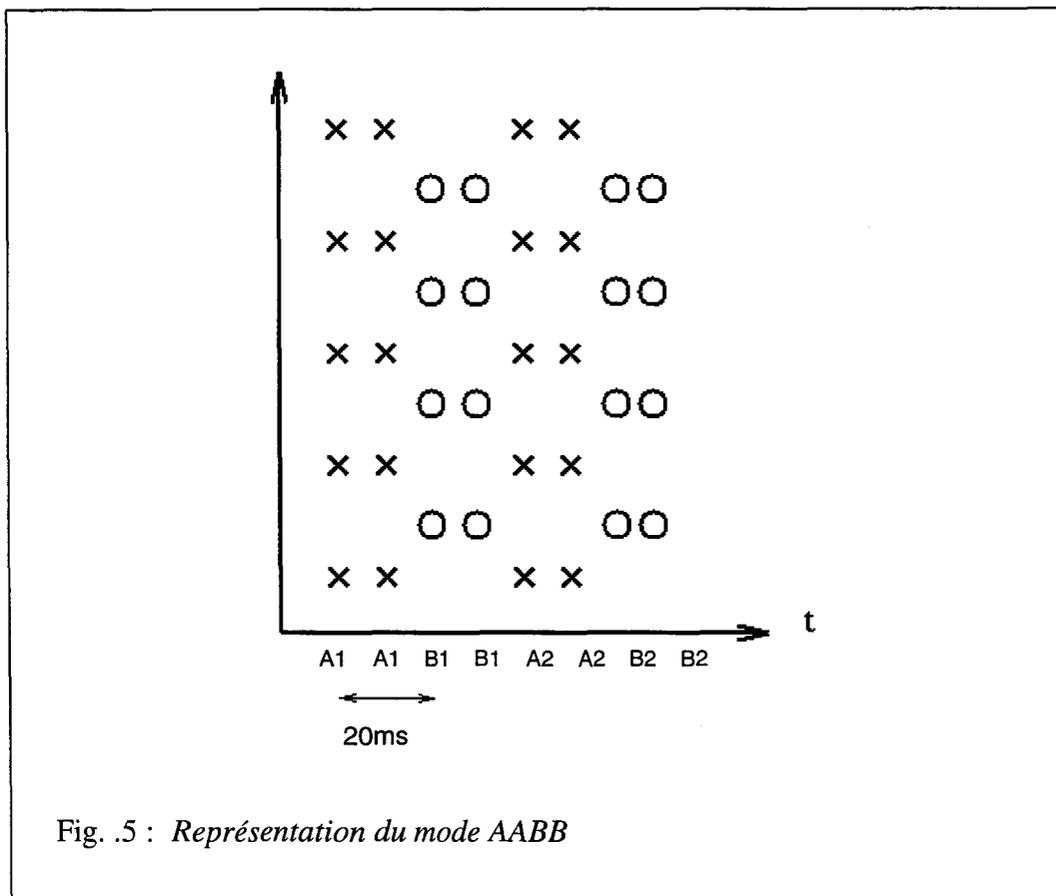


Fig. .5 : Représentation du mode AABB

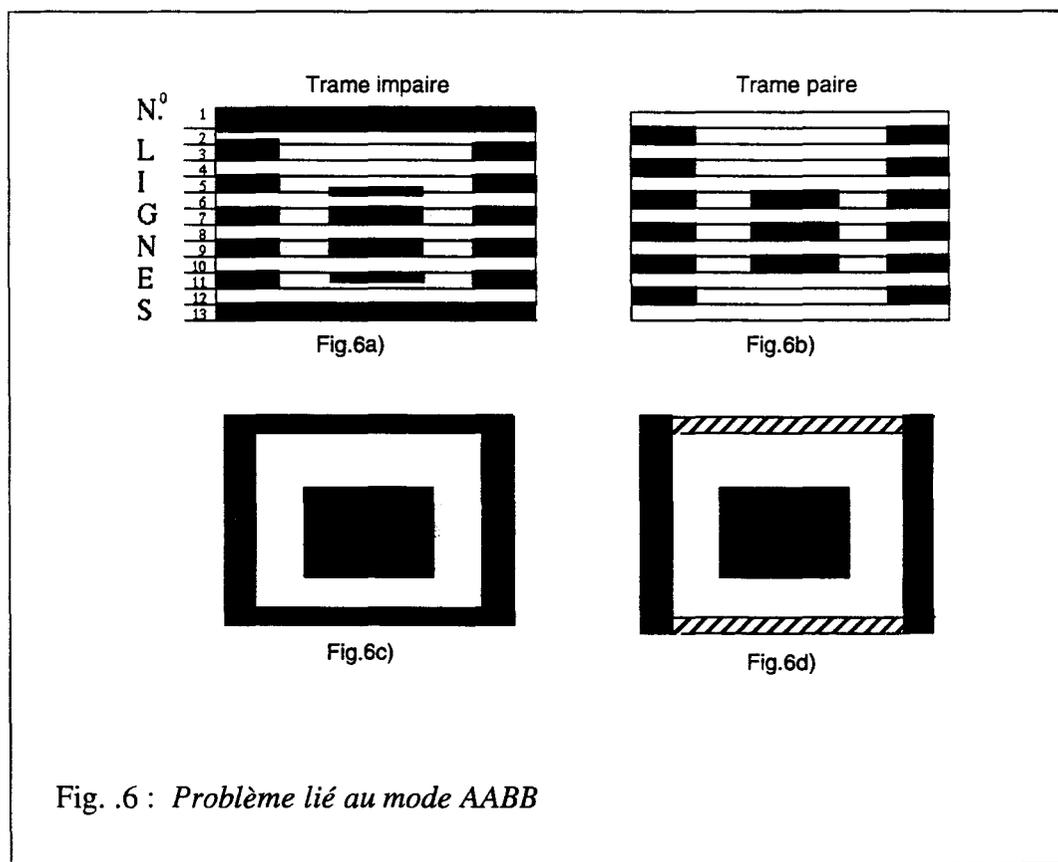
Le principal défaut de ce mode de balayage est dû au fait que les détails sont répétés avec un délai de 10 ms. Lorsque des objets sont en mouvement, il apparaît une deuxième image identique à celle d'origine.

Un autre problème apparaît lorsque une ligne, fine, appartient à la trame impaire et pas à la trame paire.

Reprenons l'exemple décrit dans [Herb] (Fig.6).

Soit une image entrelacée représentant un carré.

Si on visualise séparément les trames A et B constituant l'image, on observe ce qui est représenté sur les figures 6a) et 6b). Comme la première ligne de la trame impaire, représentée en noir sur la figure 6a) ne se retrouve pas dans la trame paire, l'image en mode AABB ne sera pas perçue comme celle de la figure 6c.), mais comme celle de la figure 6d.).



2.1.2. Le mode "ABAB"

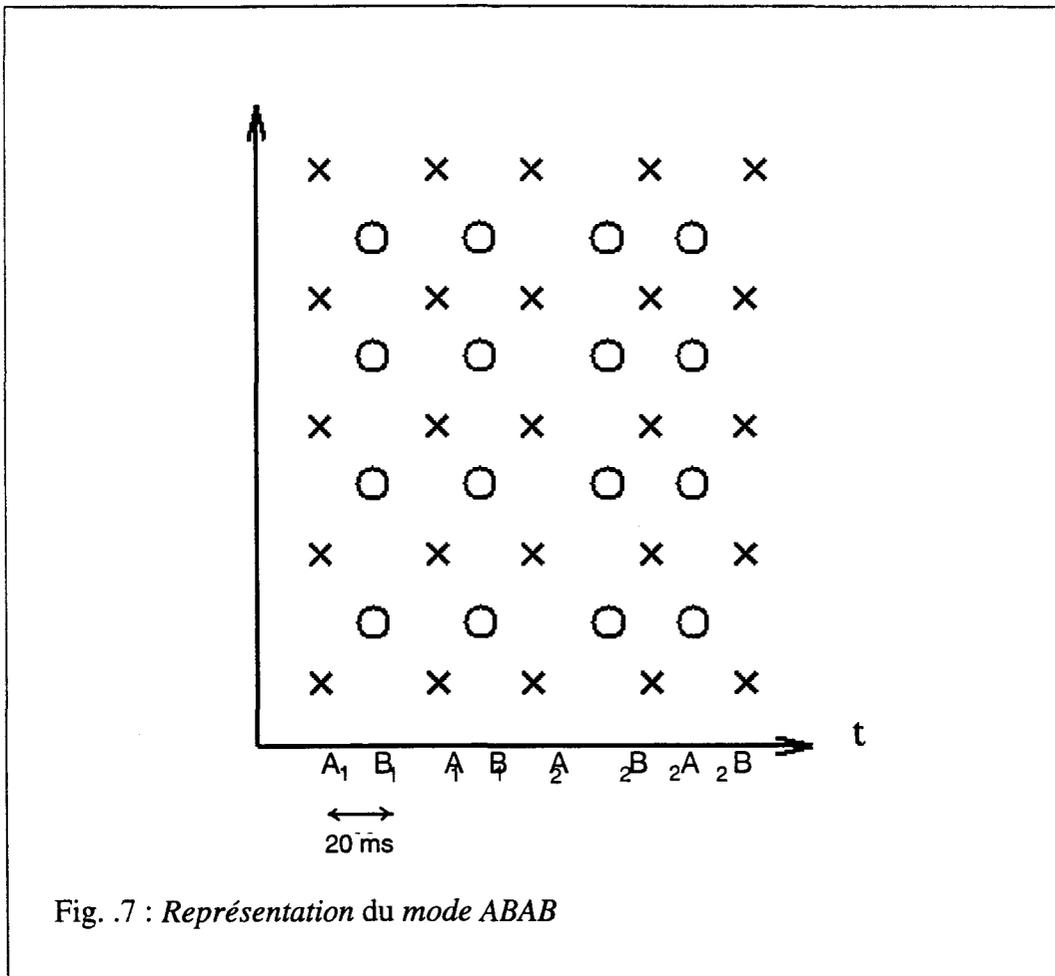


Fig. 7 : Représentation du mode ABAB

Cette méthode permet de réduire le papillotement de grandes plages et du scintillement interligne. Le résultat est idéal dans les zones statiques ce qui n'est absolument pas le cas des zones de mouvement ; en effet, seulement 25% des trames ont une position temporelle correcte. L'image est alors saccadée.

Les seules méthodes pouvant donner des résultats acceptables sont donc celles faisant appel à une estimation de mouvement afin d'interpoler les pixels manquants.

2.2. Passage au 100Hz avec estimation de mouvement

Seule l'estimation de mouvement permet une bonne interpolation des pixels manquants. Nous nous proposons d'utiliser notre algorithme afin de calculer les vecteurs déplacement. Dans un premier temps, nous sommes passé du 50Hz entrelacé au 50Hz progressif pour ensuite passer au 100Hz progressif.

2.2.1. Passage au 50Hz progressif

Dans un premier temps, nous avons calculé le vecteur déplacement des pixels existants. Nous avons travaillé sur les trames de même parité.

Le schéma ci-dessous représente le 50Hz entrelacé et les pixels à interpoler pour passer au 50Hz progressif.

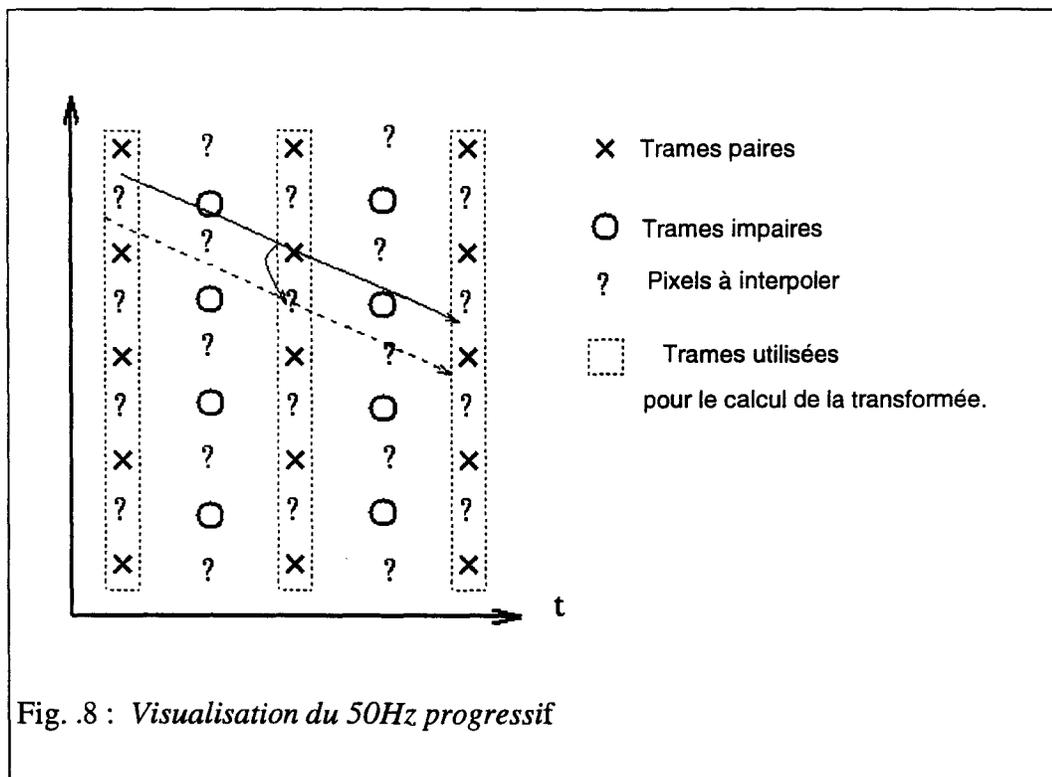


Fig. .8 : Visualisation du 50Hz progressif

Le problème principal est que l'on ne calcule pas directement le vecteur du pixel à interpoler, on est donc obligé de faire des approximations.

On peut supposer que le pixel de la ligne suivante aura le même déplacement que le pixel pour lequel on a calculé le vecteur déplacement (Fig.8).

On peut aussi interpoler directement le pixel de la trame B2 à partir du vecteur (V_x, V_y) (Fig.9).

Le problème est que l'on ne tombe pas forcément sur un pixel à recréer pour passer au 50Hz progressif. On a donc des "trous".

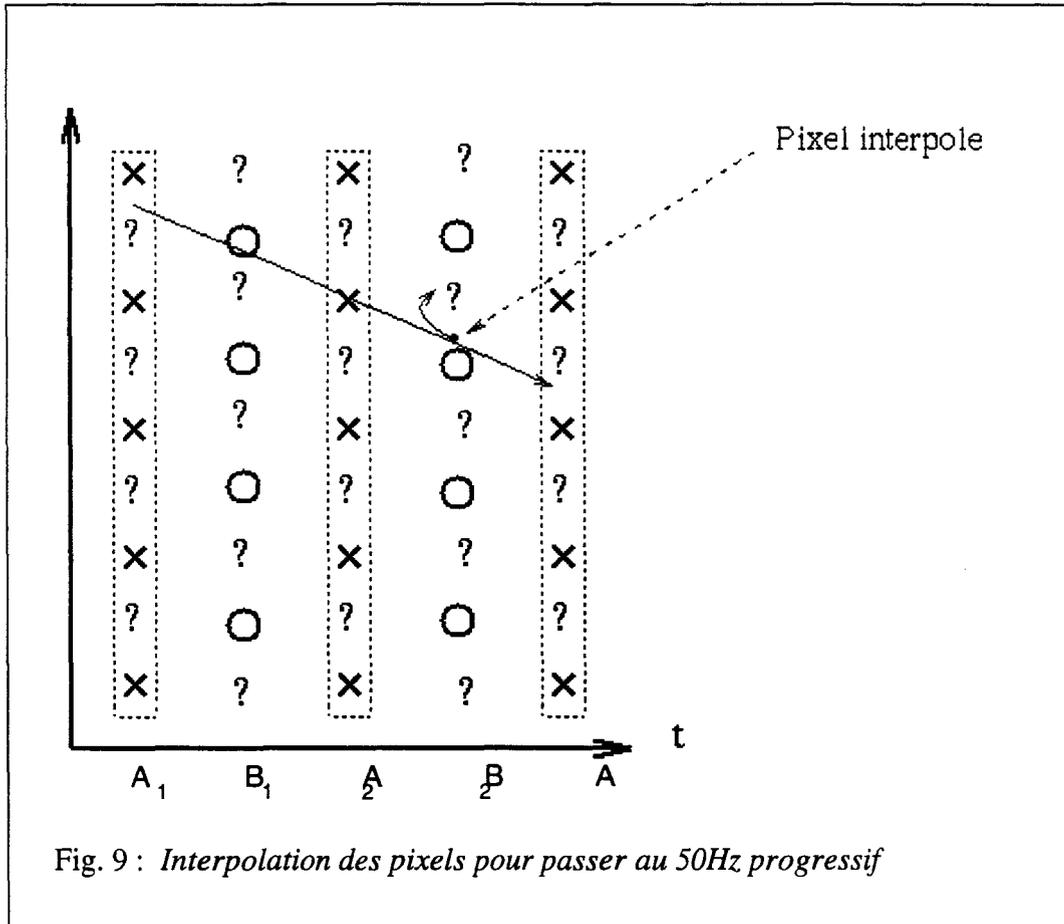


Fig. 9 : Interpolation des pixels pour passer au 50Hz progressif

Afin de "combler" les "trous", il faut ensuite effectuer une interpolation, ce qui alourdit encore les calculs.

Jusqu'à présent, nous avons utilisé des trames de même parités pour le calcul de la transformée de Wigner-Ville.

Nous allons effectuer les calculs mais en se servant cette fois des trames de parités différentes comme nous l'avons représenté sur la Figure 10.

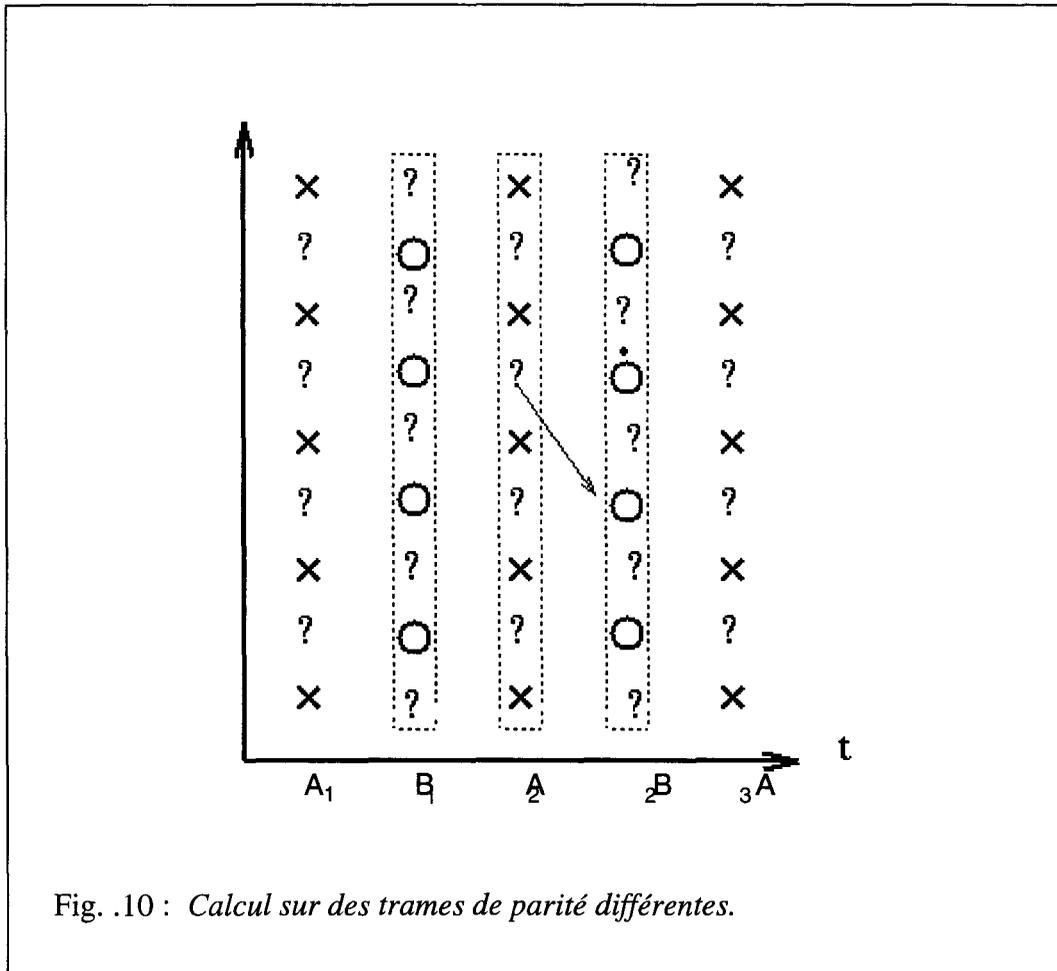


Fig. .10 : Calcul sur des trames de parité différentes.

Sur la Figure 11, nous avons représenté les différentes manières de calculer la transformée de Wigner-Ville tridimensionnelle selon que l'on recherche le vecteur déplacement de points fictifs, ou réels, ou encore si on utilise des trames de parités différentes.

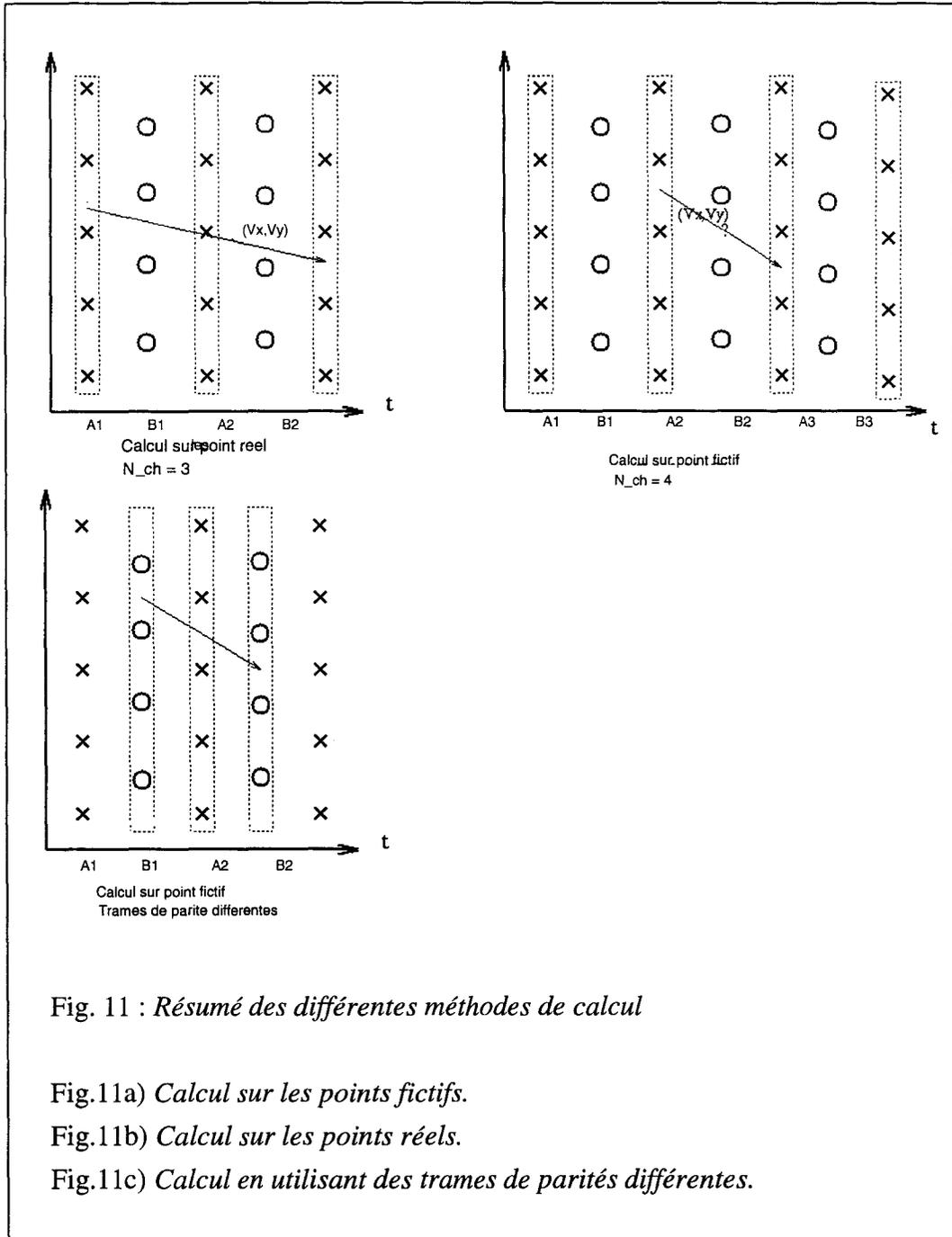


Fig. 11 : Résumé des différentes méthodes de calcul

Fig.11a) Calcul sur les points fictifs.

Fig.11b) Calcul sur les points réels.

Fig.11c) Calcul en utilisant des trames de parités différentes.

2.3. Passage au 100Hz

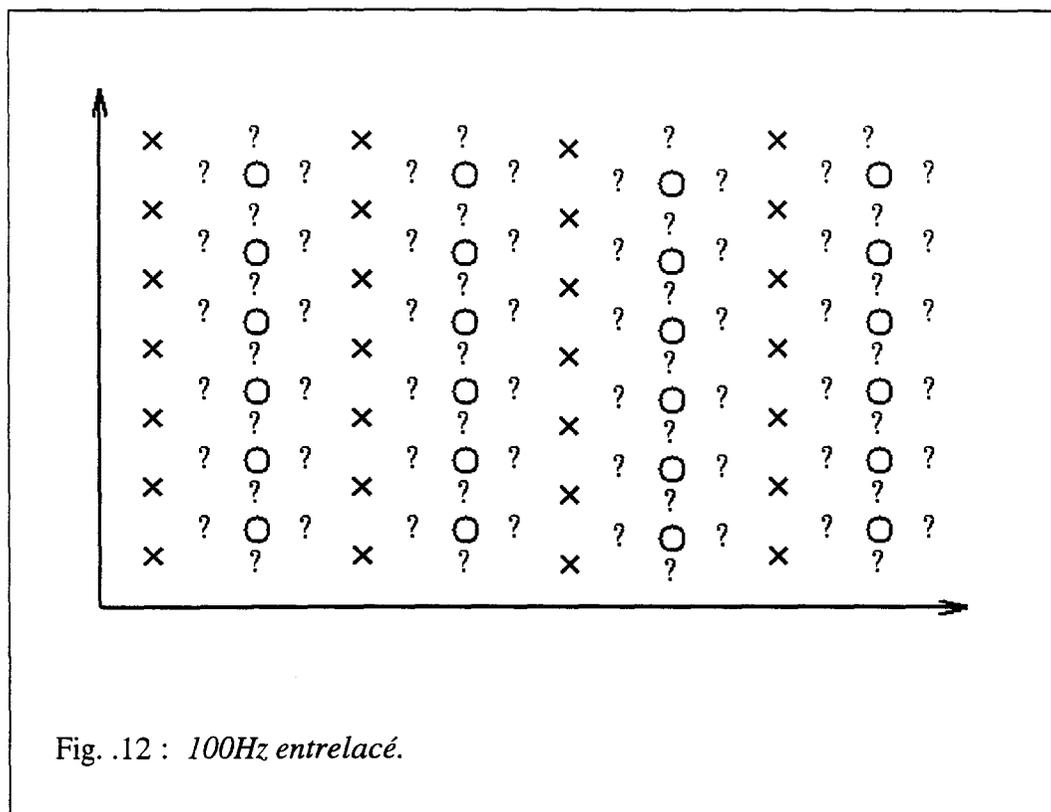


Fig. .12 : 100Hz entrelacé.

Pour passer au mode 100Hz entrelacé, le plus simple est de passer au 50Hz progressif, puis de reconstituer les images manquantes.

L'utilisation de la transformée de Wigner-Ville calculée sur un point réel s'est avérée inutilisable, nous avons donc modifié la transformée afin de pouvoir l'utiliser sur un point fictif.

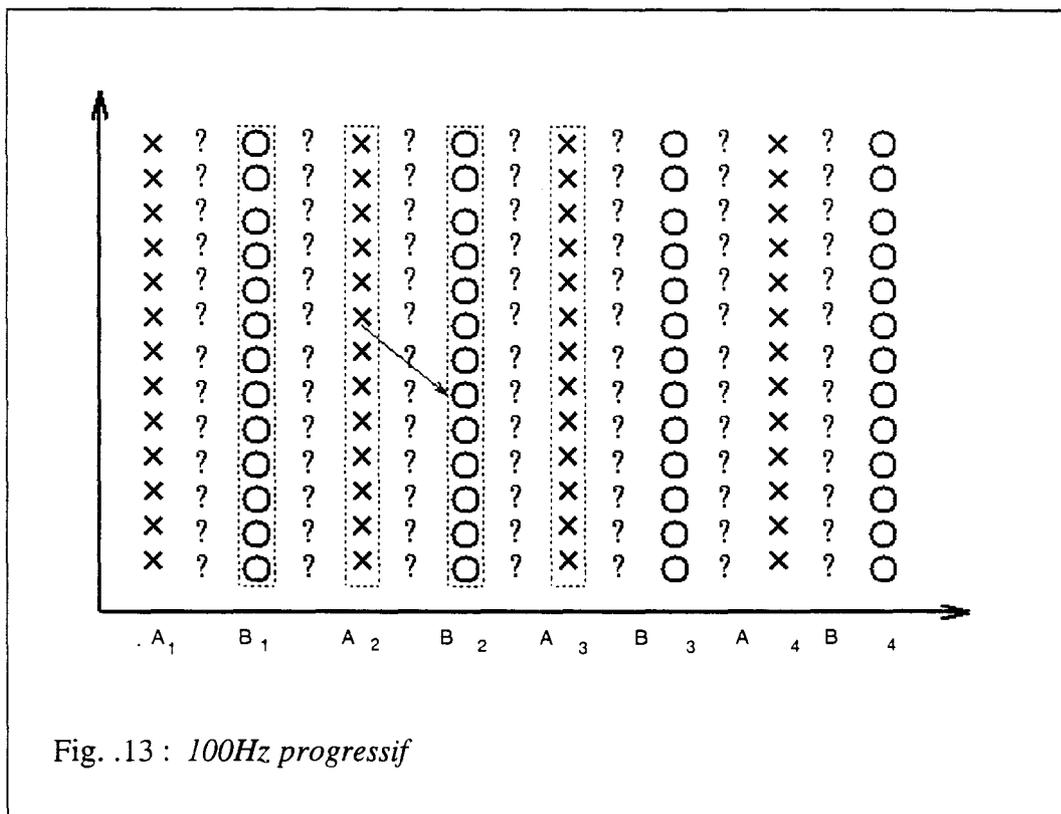


Fig. .13 : 100Hz progressif

Cette fois-ci, on obtient directement le vecteur déplacement du point à construire.
Il est à noter que l'on travaille sur des images et non plus sur des trames.

3. PRINCIPAUX RESULTATS OBTENUS

Afin de mieux visualiser les résultats obtenus nous avons effectués les calculs en vue du passage du 50Hz entrelacé au 50Hz progressif.

La séquence que nous utiliserons est la séquence "four.lum". Elle est divisée en quatre parties.

Le cadran en haut à gauche représente une roue ayant un mouvement de rotation par rapport à son axe central.

Sur le deuxième cadran, la femme a un mouvement quasiment horizontal, assez rapide : de l'ordre de 5 à 6 pixels par trame.

Le troisième cadran représente un mouvement quasiment vertical. Seule les lettres sont en mouvement.

Le mouvement du dernier cadran est très faible.

En mode entrelacé, une image est formée de deux trames se succédant à une fréquence telle qu'elles sont invisibles pour l'œil humain.

Cependant, lorsque l'on visualise ces deux trames simultanément, l'intervalle de temps existant entre deux colonnes consécutives apparaît alors, et ce d'autant plus que le pixel est animé d'un mouvement important.

Les deux images ci-dessous représentent les deux trames composant une image de la séquence four.

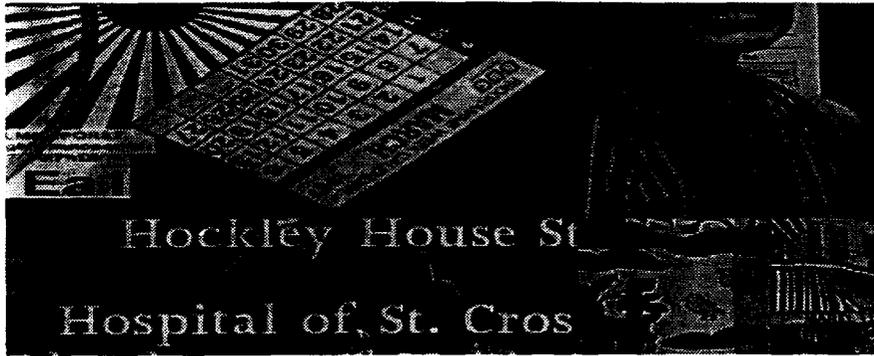


Fig.14.a) : Trame A de l'image "four"

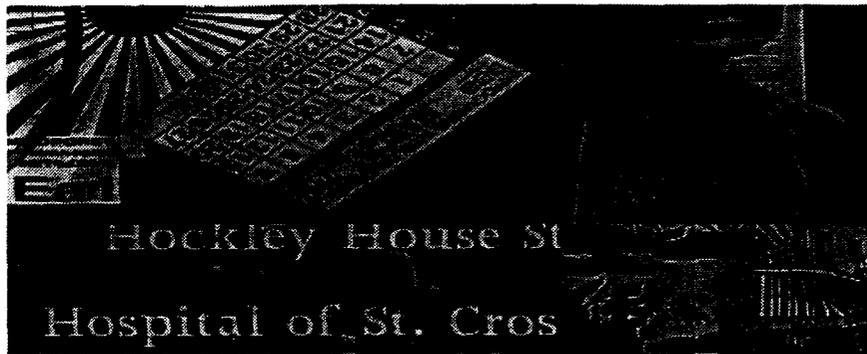


Fig.14.b) : Trame B de l'image "four"

Sur la figure 15, nous avons visualisé les deux trames simultanément.

On remarque alors l'apparition de "dents de souris", caractérisant le mouvement existant entre deux trames successives.



Fig.15) :*Visualisation simultanée des deux trames de la séquence "four".*

Les deux trames formant une image entrelacée se succèdent à un intervalle de temps égal à 20ms. Chaque ligne se succède à un intervalle de 64ms.

Donc, si on visualise les deux trames simultanément, on verra apparaître dans le même temps deux lignes ayant un décalage temporel de 20ms. D'où l'apparition de "dents de souris", visible sur la figure 15.

Il est évident que l'on ne visualise pas les images telles quelles, mais on envoie les deux trames à un intervalle de temps telle que l'oeil ne perçoive pas le décalage entre les deux trames.

Le format progressif consiste à visualiser l'image en une seule fois. Il est alors nécessaire de "recaler" les deux trames.

Dans le cas de la séquence utilisée, cela est surtout visible dans les chiffres de calendrier, au début de la figure de la femme.

3.1. Test sur la manière de calculer le vecteur déplacement

Lorsque l'on estime les vecteurs déplacement à partir de la fonction test, on peut soit ne sélectionner que le principal pic de la fonction test, soit conserver les quatre premiers pics afin de déterminer, par un calcul d'erreur celui que l'on conservera comme vecteur déplacement.

Les deux figures ci-dessous représentent les résultats obtenus suivant que l'on ne sélectionne qu'un seul pic ou quatre pics de la fonction test.

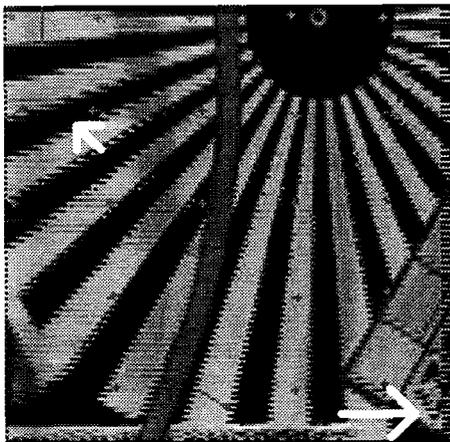


Fig.16a) : *Bloc de taille (8*8*8)*
1 vecteur sélectionné



Fig.16.b) : *Bloc de taille (8*8*8)*
4 vecteurs sélectionnés

Lorsque les calculs sont effectués en retenant les quatre principaux pics de la fonction test. Les résultats sont meilleurs Les résultats sont visibles sur les branches de la roue, et sur le chiffre 7 du calendrier.

Nous allons à présent visualiser les résultats obtenus lorsque l'on effectue les calculs sur les points fictifs, en conservant quatre images non nulles chargées

4. Calculs sur les points fictifs.

Les vecteurs sont estimés en élevant les valeurs de la transformée à une certaine puissance.

les figures 17.a et 17.b représentent les résultats obtenus suivant la puissance utilisée.

Les calculs sont effectués sur des blocs de taille (32*32*32).



Fig.17a) : *Coefficients élevés à la puissance 4.*

Fig.17b) : *Coefficients élevés à la puissance 2.*

On remarque que les résultats sont meilleurs lorsque les coefficients sont élevés à la puissance quatre. Cela est surtout visible dans le chiffre 1, indiqué par la flèche, du calendrier lorsque l'on élève les coefficients de la transformée de Wigner-Ville à la puissance quatre.

Les figures 18a) et 18b) représentent le champ de vecteur obtenu.

Les vecteurs sont visualisés à l'aide d'un code de couleur.

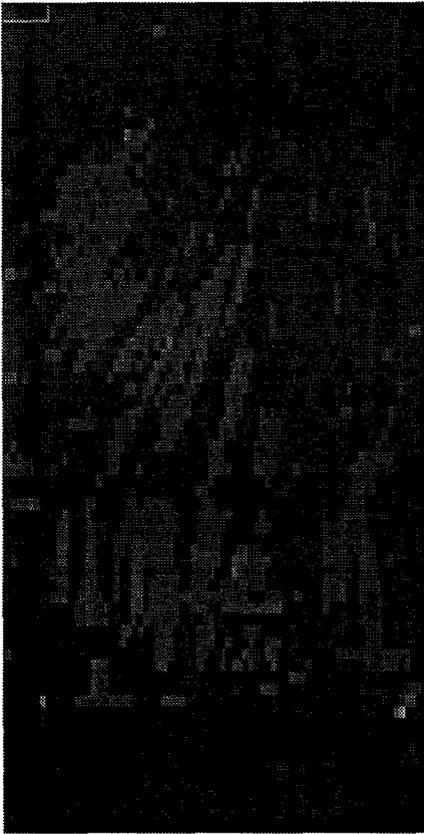


Fig.18a) : *Coefficients à la puissance 4.*

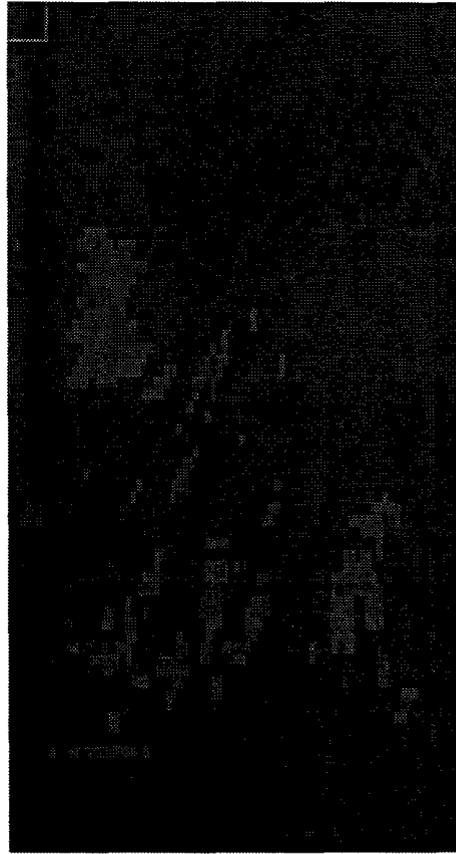


Fig.18b) : *Coefficients à la puissance 2.*

La figure 18a) représente le champ de vecteurs associé à la figure 17.a).

La figure 18b) représente le champ de vecteurs associé à la figure 17.b).

Dans les deux cas, les vecteurs trouvés sont assez homogènes.

Lorsque l'on élève les coefficients à la puissance quatre, les vecteurs suivent plus la structure de l'image.

Dans le chapitre précédent, nous avons observé que les résultats diffèrent suivant la taille du bloc utilisé.

Nous allons visualiser l'influence de la taille des blocs de calculs sur les résultats obtenus et ce sur des séquences d'images réelles.

4.1. Tests sur la taille des blocs de calcul

Afin de vérifier l'importance de la taille du bloc sur les résultats obtenus, nous avons effectués les calculs des vecteurs sur des blocs de taille $(8*8*8)$, $(32*16*16)$ et $(32*32*32)$.

Si les calculs s'effectuent sur des blocs de taille $(8*8*8)$, la dimension temporelle est égale à 8, pour des blocs de taille $(32*16*16)$ et $(32*32*32)$; elle est égale à 32.

Le nombre d'images non nulles est toujours égal à quatre.

Les figures 19 et 20 représentent le plan associé au pixel de coordonnées $(30,39)$ (Fig.19), et la fonction test associée (Fig.20).

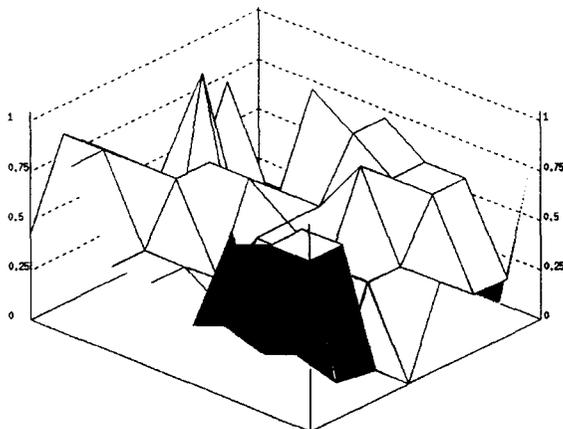


Fig.19) Plan associé au pixel $(30,39)$

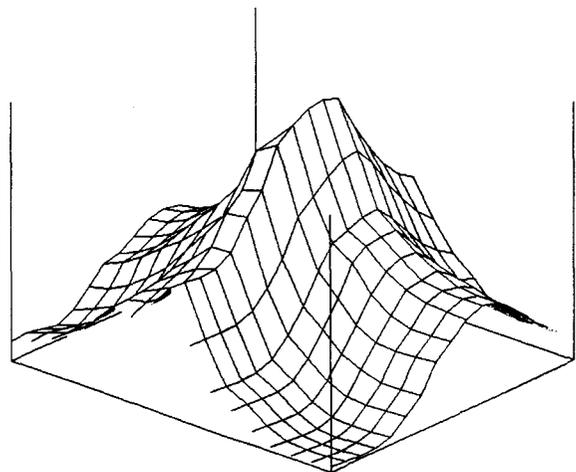


Fig.20) : Fonction test

Le vecteur trouvé est égal à $(1,1)$.

Le plan, caractérisé par très peu de points n'est pas bien défini. Il est alors difficile d'estimer correctement son inclinaison.

Les blocs de taille $(8*8*8)$ ne permettent pas d'estimer correctement les vecteurs déplacement car on ne possède pas assez d'information sur la transformée de Wigner-Ville.

Afin d'y remédier, nous avons modifié la taille du bloc.

* Blocs de taille $(32*16*16)$

Les deux figures suivantes représentent le plan associé au pixel de coordonnées $(30,39)$ et la fonction test correspondante.

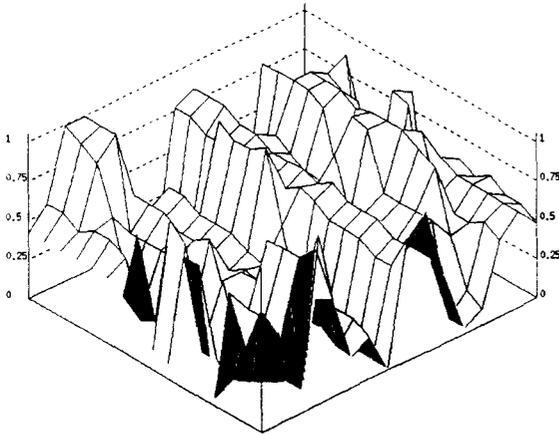


Fig.21) : *Plan associé au pixel $(30,39)$
blocs de tailles $(32*16*16)$.*

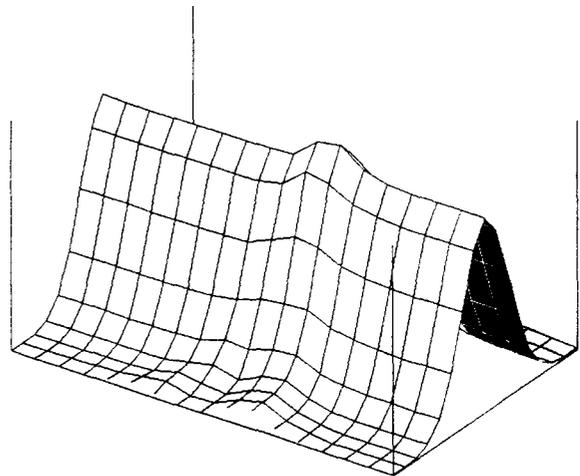


Fig.22) : *Fonction test
Vecteur trouve : $(0,2)$*

On remarque que dans ce cas, le plan est beaucoup mieux défini que précédemment.

La figure ci-dessous représente les résultats obtenus sur des blocs de taille (32*16*16). Cela signifie que l'on a considérablement augmenté la résolution temporelle, en conservant tout de même 4 images non nulles.

Les vecteurs déplacement sont obtenus en conservant les quatre pics principaux de la fonction test.

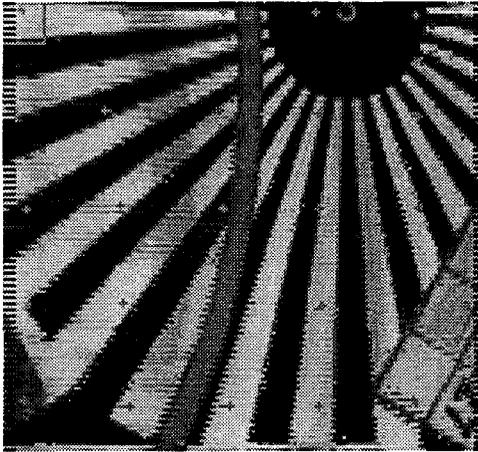


Fig.23) : *Bloc de taille (32*16*16)*

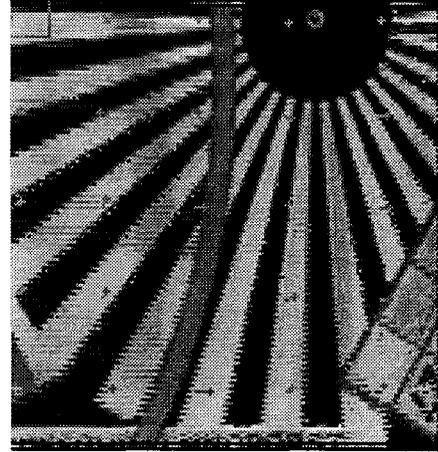


Fig.24) : *Bloc de taille (8*8*8)*

On remarque que les résultats sont meilleurs, surtout dans le chiffre 7 du calendrier, et dans la roue.

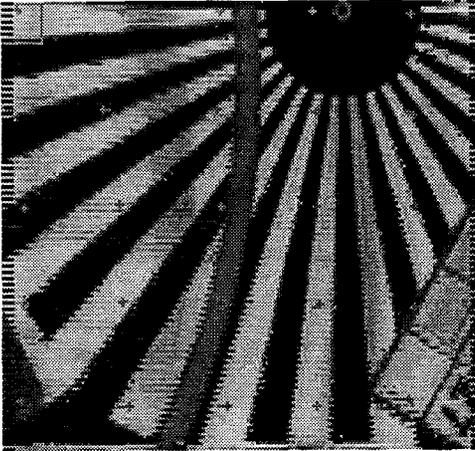
Le fait d'augmenter la dimension temporelle permet d'avoir une meilleure précision sur les valeurs des vecteurs calculés.

De plus, le fait de prendre en considération une zone plus importante centrée sur le pixel pour lequel on recherche le vecteur mouvement n'est pas pénalisante. La taille passe de 8 à 16.

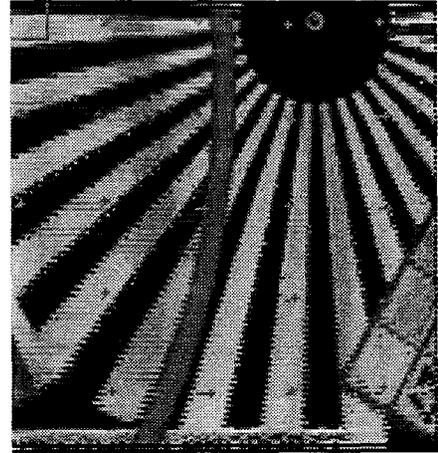
Nous allons vérifier si le fait de prendre en compte un entourage important autour du pixel pour lequel on recherche le vecteur déplacement est pénalisant sur les résultats obtenus.

On peut considérer que si on augmente encore les dimensions de N_1 et de N_2 , on risque de tenir compte de pixels dont les mouvements sont totalement différents.

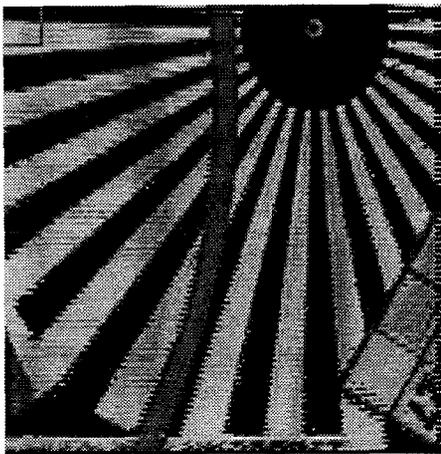
Pour cela, nous avons utilisé un bloc de taille $(32*32*32)$.



Bloc de taille $(32*16*16)$



Bloc de taille $(8*8*8)$



Bloc de taille $(32*32*32)$

L'utilisation de blocs de taille $(*32*32*32)$ permet d'obtenir de meilleurs résultats. Cependant les temps de calcul augmentent considérablement..

Diminution du nombre de points calculés :

Les figures ci-dessous mettent en valeur le fait que notre algorithme est capable d'estimer le vecteur déplacement de pixel dont la luminance est peu contrasté avec un fond fixe.

La figure 22a) représente les résultats obtenus lorsque que l'on effectue les calculs sur tous les pixels de l'image.

Pour la figure 22b), les calculs n'ont été effectués que sur les pixels de fort contraste. Sur les zones d'ombres, précises par les fleches,, les calculs n'ont donc pas été effectués.

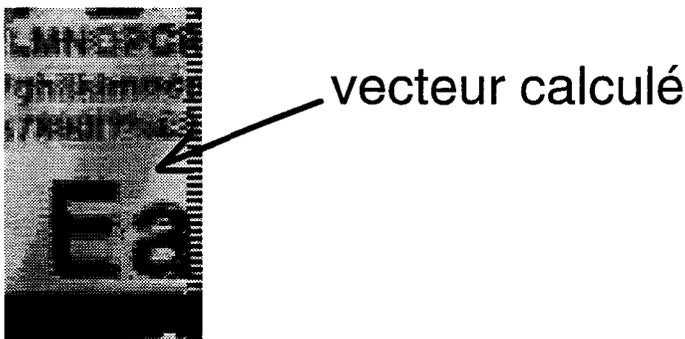


Fig.22a) : Calculs effectués sur tous les pixels de l'image.

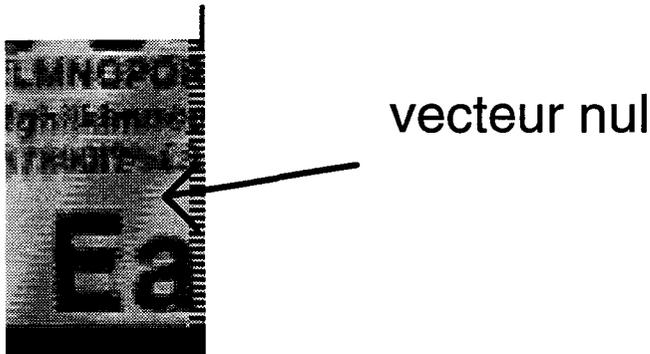


Fig.22bb) Calculs effectués sur les points de fort contraste entre deux images consécutives.

CONCLUSION

L'étude de la transformée de Wigner-Ville en vue d'estimer les vecteurs déplacement d'une séquence d'images animées s'est avérée très intéressante.

L'utilisation de la transformée de Wigner-Ville tridimensionnelle présente l'avantage que son calcul sur un point considéré tient compte de l'influence de l'entourage proche de ce point ; ce qui procure à cette technique une robustesse accrue. L'influence du bruit n'est pas déterminante pour le calcul des vecteurs.

En ce qui concerne l'amélioration de la visualisation des images de télévision, le calcul des vecteurs par le biais de la transformée de Wigner-Ville offre l'avantage considérable de pouvoir calculer directement le vecteur du pixel à interpoler.

On peut donc désormais passer relativement facilement du format entrelacé au format progressif, et même au 100Hz. Cependant, cette transformée nécessite d'utiliser une taille de cube relativement importante afin d'obtenir des plans caractérisant correctement le déplacement du pixel considéré. Cela engendre alors des durées de calculs importantes.

Malgré les résultats encourageants, il reste encore de nombreux cas pour lesquels l'algorithme fait défaut. Dans le cas de mouvement diagonaux de fortes amplitudes par exemple.

De plus, lors des mouvements de grandes amplitudes, l'estimation des vecteurs ne sera correcte que si l'on dimensionne correctement l'amplitude des vecteurs à tester lors du calcul de la "fonction test".

Son principal désavantage reste tout de même son temps de calcul important.

Une étude comparative, dans des conditions similaires avec les méthodes existantes s'avérera intéressante.

BIBLIOGRAPHIE

[Aderk 90]

Aderkaoui

Théorie de l'analyse en multirésolution et applications.

D.E.A automatisme et traitement du signal, option image et science de l'univers. Université de Nice sophia-antipolis. pp : 1.63. juillet 1990.

[BaJ 93]

R.G.Baraniuk and D.L.Jones.

A signal-dependant time frequency representations : optimal kerne design.

IEEE transactions on signal processing. Vol.41, N.4 April 1993.

[BalB 89b]

M.Balanza , A.Biasiolo

On the effects of acceleration in the frequency domain.

[Bergeron 90]

C.Bergeron. E.Dubois.

Parametric block estimation of motion and application to temporal interpolation of video sequences.

IEEE. 1990

[Bhanu 88]

B.Bhanu W.Burger

Approximation of displacement fields using wavefront region growing

Computer vision,graphics, and image processing 41, 306 322. 1988

[Boilot 90]

F.Boilot. P.Vannoorenberghe.

Représentation des signaux non stationnaires en temps-fréquence. Transformation de Wigner_Ville.

Rapport de projet.Université de Lille I. Avril 1990.

[Bouthemy 87]

P.Bouthemy.

Estimation et structuration d'indices spatio-temporels pour l'analyse du mouvement dans une séquence d'images.

Traitement du signal, Vision par ordinateur. Vol 4 N 3 1987.

[CaR 82]

Cafferio.C Rocca.F

The differential method for image motion estimation.

NATO.Advanced study institute on image processing and dynamic scene analysis. June 1982.

[Castro 87]

E.De Castro , C.Morandi

Registration of translated and rotates images using finite fourier transforms.

IEE Transactions on pattern analysis and machine intelligence. Vol PAMI-9 N.5 September 1987

[Chup 91]

Chuveau.

Multiscale motion estimation.

Document Thomson CSF/ Laboratoire électronique de Rennes_Cessam cévigné.

[Daub]

Daubechies.

Orthogonal bases of compactly supported wavelets.

AT&T Bell. Communication on pure and applied mathématique. Vol XLV. pp : 909.996.

[Dreier]

Dreier.

Line flicker reduction by adaptative signal processing.

[Enkel 86]

Enkelmann.

Investigations of multigrid algorithms for the estimation of optical flow fields in image sequences.

I.E.E.E pp : 81.87. 1986.

[FeP 87]

Fernando. Parker.

Motion compensated display conversion

Philips research laboratories.

[Fla 93]

Flandrin. P.

Temps-fréquence.

Traité des nouvelles technologies. Série traitement du signal. Edition Hermes.

[FlaE 85]

P.Flandrin. B.Escudie.

Principe et mise en oeuvre de l'analyse temps-fréquence par transformation de Wigner-Ville.

Traitement du signal. Volume 2 N 2. 1985.

[Flandrin 69]

P.Flandrin.

Représentation temps-fréquence de Wigner-Ville.

Greco 69.

[Grotze]

Grotze.

Generation of motion vectors fields for motion compensated interpolation of HDTV signals

[Haan]

G.De Haan. H.Huijgen.

" New algorithm for motion estimation. "

Philips Research Laboratories.

[HeB 90]

Heitz. Bouthemy.

Estimation et segmentation du mouvement : approche bayésienne et modélisation markovienne des occlusions.

Rapport de recherche INRIA 1990.

[Hos 81]

B.K.P.Horn B.G.Scuink.

Determining optical flow.

Artificial intelligence. Vol.17, pp185-203. 1981.

[Hsieh 90]

Hsieh. Lu. Shyn.

Motion estimation algorithm using interblock correlation.

Electronic letters Vol 26. N 5 1990.

[Huang 91]

Blostein. Huang.

Detecting small, moving objects in image sequences using sequential hypothesis testing.

IEEE Transaction on signal processing. Vol 39 N 7 July 1991.

[Huang 83]

T.S Huang.

Image sequence processing and dynamic scene analysis.

Nato ASI. Springer verlag. Berlin. 1983

[Huang 88]

Liu. Huang.

Estimation of rigid body motion using straight line correspondences.

Computer vision, graphics, and image processing 43, 37-52 . 1988

[Huang 88]

Liu. Huang.

A linear algorithm for motion estimation using straight line correspondences.

Computer vision, graphics, and image processing 44, 35-57 . 1988

[Hwang 90]

T.K.Hwang J.J.Clark

On local detection of moving edge

IEEE 1990

[Hvisto]

H.Hvisto. Juhola. Neuvo.

" Scan rate up-conversion using adaptative weighted median filtering.

[Jain 81]

Jain.J.R Jain. A.K

Displacement measurement and its application in interframe image coding.
IEEE. Transactions on communications. Vol.29. December 19981.

[JaW 86]

Jacobson. Wechsler.

Derivation of optical flow using a spatiotemporal-frequency approach.
Computer vision, graphics and image processing. pp : 29.65. 1986

[JaW 88]

L.D.Jacobson. H.Wechsler

Joint spatial/spatial-frequency representation.
Signal processing 14 37-68. 1988

[Jones 92]

Douglas.L.Jones. Thomas W. Parks

A Resolution comparaison of several time-frequency representations.
IEEE Transactions on signal processing. Vol 40 N2. February 1992

[JeW 92]

Jechang Jeong. William J.Williams.

Kernel design for reduced interferences distributions.
IEEE Transactions on signal processing. Vol 40 N2. February 1992

[Ka 90]

D.Ka.I

A 2-D motion estimation algorithm.
IEEE. 1990.

[Kahn 85]

P.Kahn

Local determination of a moving contrast edge
IEEE transactions on pattern analysis and machine intelligence, vol.PAMI-7,N4 July 1985

[Knee 91]

M.Knee.

Proscan conversion studies
status report. sept.1991.

[Knee 91]

M.Knee.

Motion compensated 100Hz conversion : a discussion paper.

[Konrad 88]

J.Konrad.

Motion compensated interpolation for TV frame-rate conversion.
Les assises des jeunes chercheurs. Oct. 9-13 1988. Cambridge.

[Komarek 90]

T.Komarek. P.Pirsch.

VLSI architecture for hierarchical block matching algorithms.

IEEE. 1990.

[KraH 93]

W.Krattenthaler and F.Hlawatsch

Time-frequency design and processing of signals via smoothed Wigner distribution.

IEEE Transactions on signal processing, Vol.41. N1. January 1993.

[KrC87]

Krishnapuram. Casasent.

Hough space transformation for discrimination and distortion estimation.

Computer vision, graphics and image processing 38 pp : 299.316 1987

[Lavin 86]

Hungwen li. Lavin. Le Master.

Fast Hough transform : a hierarchical approach.

Computer vision, graphics and image processing 36 pp : 139.161. 1986.

[Lopez 88]

Lopez. Krahe. Pousset.

Transformée de Hough discrète et bornée. Application à la détection de droites parallèles et du réseau routier.

Traitement du signal. Vol 5 N4 1988.

[Illin]

Princen. Illingworth. Kittler.

A hierarchical approach to line extraction based on the Hough transform.

[Mallat 88]

Mallat.S

Review of multifrequency channel decompositions of images and wavelet models.

Technical report N.412.

Robotics report N.178

November 1988.

[Mallat 89]

A theory for multiresolution signal decomposition : the wavelet representation.

IEEE Transactions on pattern analysis and machine intelligence. Vol.11, N.7. July 1989.

[Meyer]

Meyer

Les ondelettes.

[Meyer 87]

Meyer. S.Jaffard. O.Rioul.

L'analyse par ondelettes.

Pour la Science. pp : 28.37. Sept 1987

[Millour 89]

Millour.

Une approche multi-résolution pour le calcul de champs denses de vélocité optique.
connaissance des formes et intelligence artificielle. & congrès AFCET INRIA. Tome 3.

[Mon 90]

Monga.O

Segmentation d'images : où en sommes nous ?

Rapport de recherche INRIA N.1216 Avril 1990.

[Mus 85]

Musmann. Pirsch

Advanced in picture coding.

Proceeding of the IEEE. Vol 73. N.4. April 1985.

[Nag 87] H.H. Nagel

On the estimation of optical flow : relations between different approaches and some new results.

Articasiw result

Artificial intelligence N.33 1987 pp 299-324.

[Net 79]

Netravali.Robbins

Motion compensation T.V coding. PartI.

B.S.T.J Vol 53. N3. March 1979.

[Niky 93]

P.Nikyema.

Contribution à l'étude de l'utilisation des techniques multirésolution dans la compensation de mouvement de séquences d'images.

Thèse de doctorat en électronique de l'université de Lille I. Janvier 1993.

[Peyr 93]

F.Peyrin

Transformation de Wigner-Ville des images 2D.

Journées thématiques de GDR TdSI et colloques. INSA Lyon 9-11 mars 1994.

[QiM 92]

S.Qian. J M.Morris

Wigner Distribution decomposition and cross-terms deleted representation.

Signal processing 27 125-144. 1992

thèse Michel kerdranvat.

[Rioul 88]

O.Rioul.

Un algorithme rapide de décomposition discrète de signaux échantillonnés en ondelettes orthogonales.1988

[Riveros 89]

J.F.Vege-Riveros. K.Jabbour.
Review of motion analysis techniques
IEEE proceedings. Vol 136,pt I N 6, dec. 1989.

[Sabata 91]

B.Sabata, J.K.Aggarwal.
Estimation of motion from a pair of range images : a review.
CVGIP : Image understanding. Vol 54. N3 pp: 309.324. Nov 1991.

[Siohan]

Pele. Siohan. Choquet.
Field rate conversion by motion estimation/compensation.

[Thom 87]

Thomas.G.A.
Television measurement for DATV and other applications.
BBC research report. 1987.

[Thomas 89]

R.Thomas. M.Bierling.
Motion compensating interpolation considering covered and uncovered background.
Signal processing : image communication 1989.

[Thomas 90]

Graham. Thomas.
Motion estimation and its application to HDTV transmission and up-conversion using DATV
S.M.P.T.E Journal. pp : 987.992. 1990

[Tsai 90]

Jeng. Tsai
Fast generalized Hough transform.
Pattern recognition letters 11 pp : 725.733. 1990.

[Tubbay 89]

Tugbay. B.Sankur.
Classification and analysis of digital modulation schemes using Wigner-Ville distribution.
douzième colloque Gretsi-Juan-Les-Pins juin 1989.

[Tziritas 88]

G.Tziritas.
Estimation réursive-itérative du mouvement aparent dans une séquence d'images.
Colloque TIPI 1988.

[VaM 91]

Estimation of motion and position of a rigid object using a sequence of images.
(Tridimensional kalman filter approach).
A.C.M Association for Computing Machinery.May 1991.

[Vil 48]

J.Ville.

Théorie et application de la notion de signal analytique.

Câbles et transmission. 2ème A. N.1.

[Zhan 90]

Zhanyi.Destiné.

About Hough transform.

Revue AIM. Liège N3. 1990.

[ZPG 87]

Yue Min Zhu. F.Peyrin. R.Goutte.

Transformation de Wigner_ville:description d'un nouvel outil de traitement du signal et des images.

Annales des télécommunication, 42, N3-4. 1987.

M.Benjelloun. B.Dubuisson.

Analyse de scènes dynamiques. 1 rapport de recherche. Estimation de mouvement

Rapport de recherche INT. UTC (université de technologie de Compiègne).

M.Benjelloun. B.Dubuisson.

Analyse de scènes dynamiques. 2 rapport de recherche. Estimation de mouvement.

Rapport de recherche INT. UTC (université de technologie de Compiègne).

[Wu]

S.F.Wu. J.Kittler.

2-D motion parameters estimation ; a survey.

[Peleg 90]

S.Peleg. H.Rom.

Motion based segmentation.

IEEE. 1990.

[Weys 92]

Weyselomer.

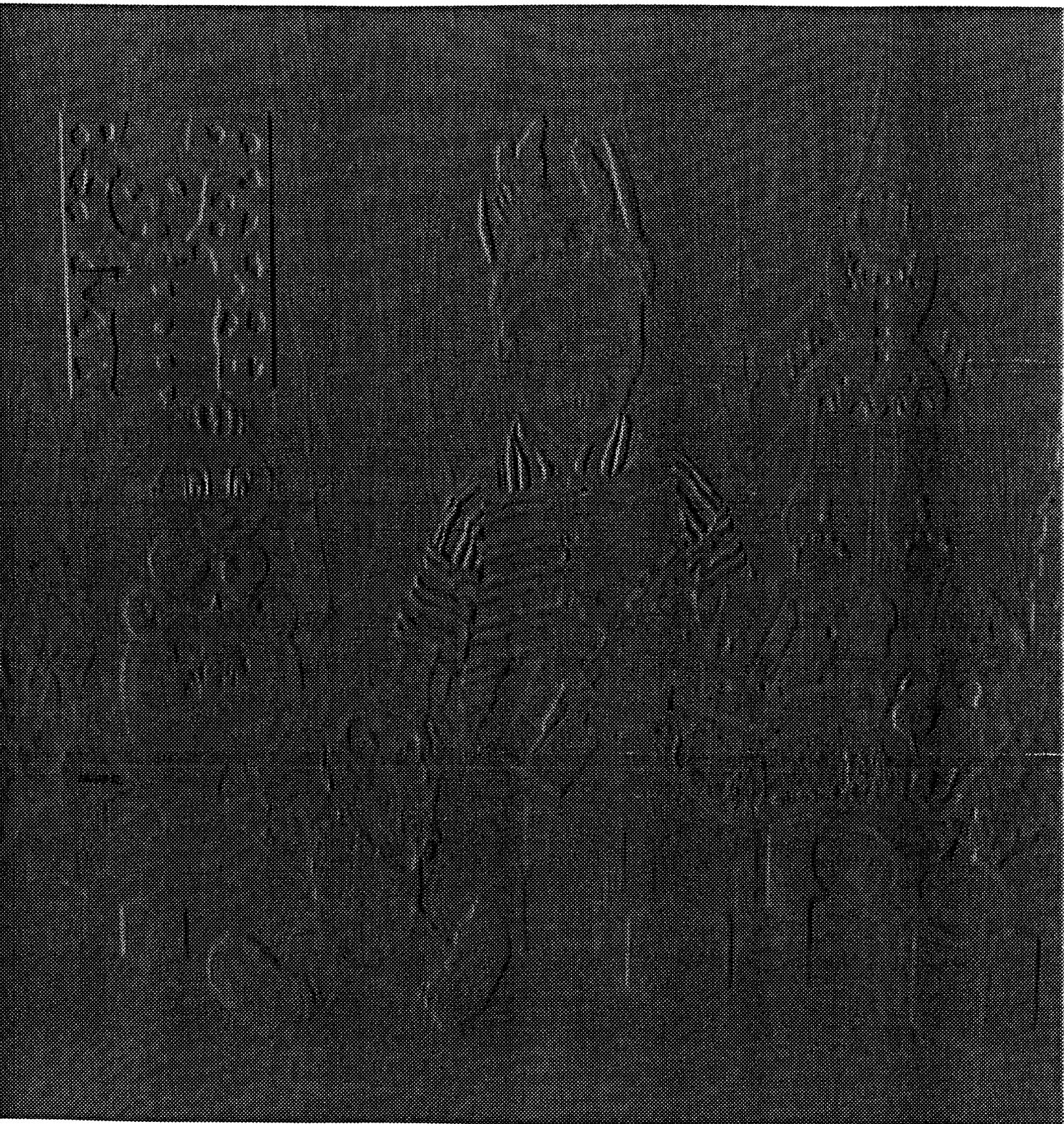
Two-dimensional Fourier transforms.

Wireless world. Vol 92. Dec 1986.

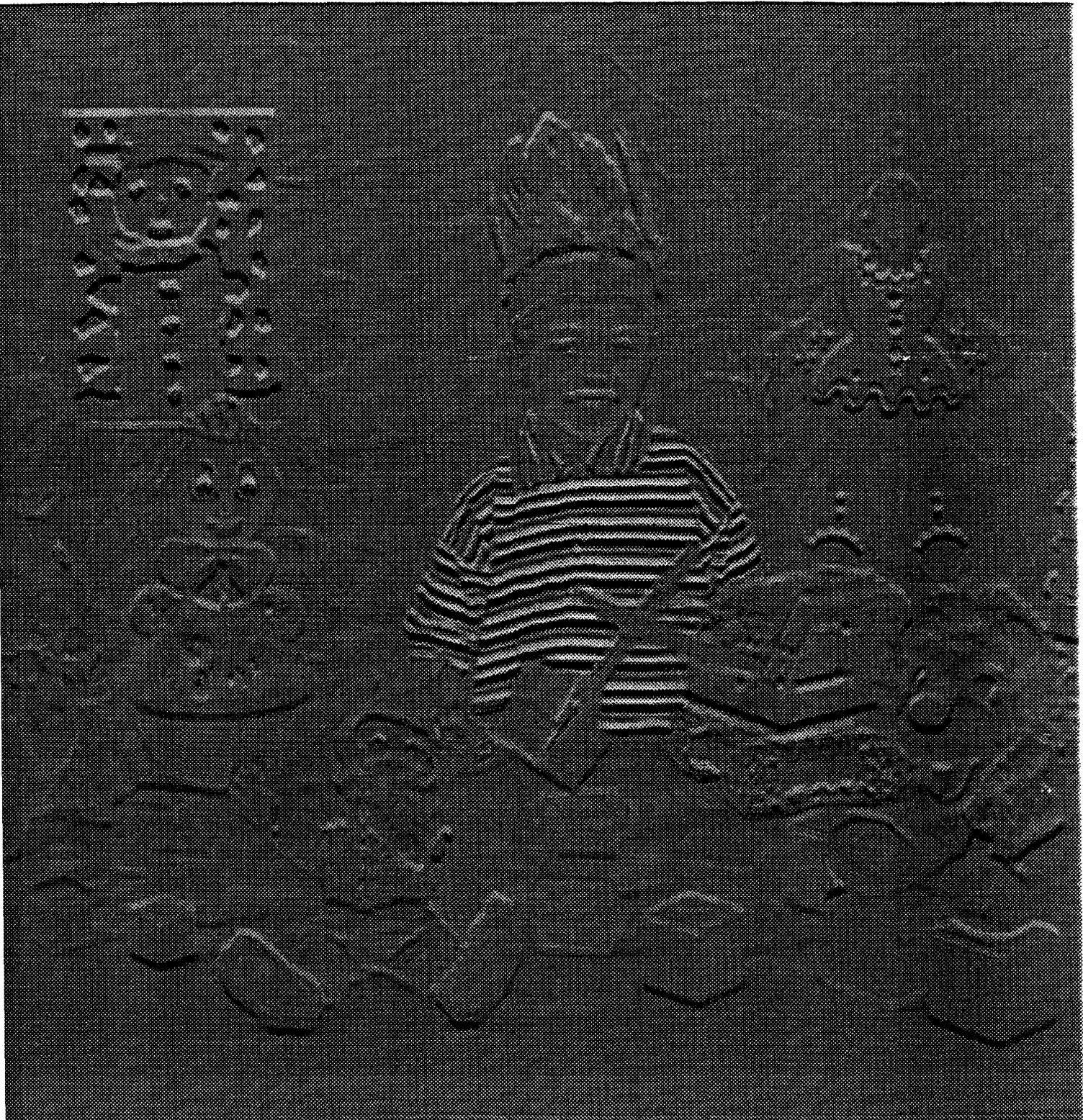
ANNEXE



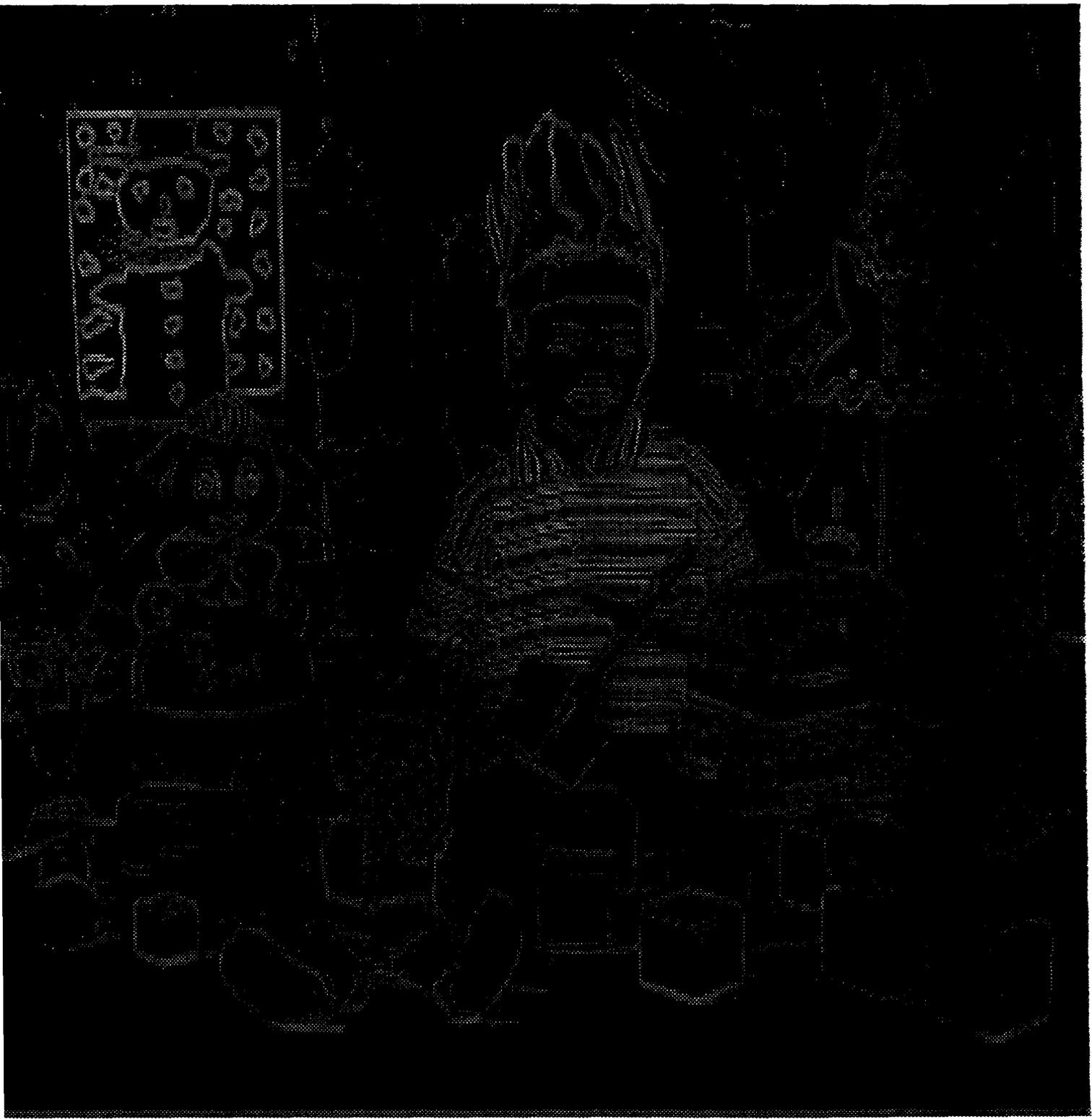
Sequencia de figuras e letras



sequencia de apes jiluf de Soze: touzonte.



reg. pelt.
... ..
... ..
... ..



RESUME

Dans le cadre de cette thèse, nous avons utilisé la transformée de Wigner-Ville tridimensionnelle (DWV 3D) afin d'estimer les vecteurs de mouvement d'une séquence d'images.

Pour cela, nous avons adapté l'algorithme de calcul de la transformée de Wigner-Ville au cas des séquences d'images. Après le calcul de celle-ci, on obtient, pour chaque pixel de l'image, un spectre tridimensionnel dont l'interprétation permet d'estimer le mouvement de ce même pixel. Le calcul de ces vecteurs est basé sur la propriété qu'a le spectre de la transformée de Wigner-Ville d'un pixel en translation constante d'être nul partout sauf sur un plan dont l'inclinaison détermine le mouvement de ce pixel.

Le principal avantage de l'utilisation de la transformée de Wigner-Ville par rapport aux méthodes exploitant les propriétés de la transformée de Fourier, telle que la corrélation de phase par exemple, est qu'elle conserve l'information sur la position du pixel. En effet, lorsque l'on utilise la transformée de Fourier, on obtient un spectre en fréquence, mais pour l'image entière sans connaître le point d'application des vecteurs obtenus.

Bien que les temps de calculs soient assez longs, les résultats sont satisfaisants. Cette méthode est assez robuste au bruit, et les vecteurs trouvés sont homogènes sans que l'on ait besoin d'ajouter des contraintes de lissage.

Nous avons appliqué notre algorithme sur des images de télévision dans le but de passer du format entrelacé au format progressif, ou au 100Hz par compensation de mouvement.

