Chapitre 1

Estimation de mouvement

Sommaire

1.1 Introduction

L'estimation de mouvement (motion estimation en anglais) consiste à mesurer le déplacement des entités composantes d'une scène réelle. Autrement dit, on cherche à quantifier le déplacement de chaque pixel dans la séquence. Le résultat obtenu est quantitatif, cela revient à calculer un vecteur vitesse pour chaque pixel (Figure 1.1).

L'estimation de mouvement est indispensable dans plusieurs domaines puisqu'elle fournit des informations essentielles pour ces derniers. Par exemple dans le domaine d'analyse du flot fluide qui utilise les informations de mouvement en météorologie, aérodynamique et encore en mécanique des fluides. Dans un contexte médical, le mouvement est utilisé pour l'estimation du flot sanguin, le recalage d'images, la détermination du mouvement des cellules, ou encore aider le suivi des cellules individuelles. L'estimation de mouvement est utilisée en vidéo-surveillance (l'expression faciale, la reconnaissance de gestes et aussi l'analyse du comportement), en robotique pour éviter les collisions par la découverte et la détection des obstacles, en techniques de segmentation et de filtrages pour définir les contours des objets, et dans l'analyse des séquences vidéo surtout dans les techniques de compression et d'indexation.

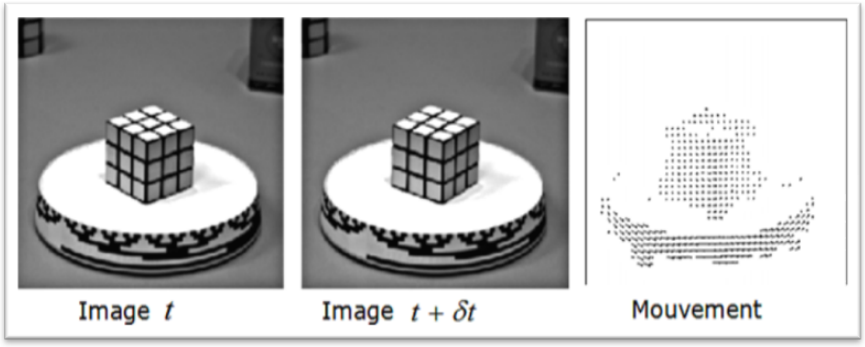

FIGURE 1.1 – Exemple de l'estimation de mouvement avec la séquence «Cube de Rubik»

1.2 Principe de base

Considérons une séquence d'images représentée par la fonction de luminance $I(x, y, t)$, où (x, y) et t représentent respectivement, les coordonnées spatiales et le temps respectivement. Le champ de vitesse produit dans le plan 2D de l'image par des objets en mouvement dans le plan 3D du monde réel ou ce qu'on appel «Flot optique» à un point au temps t est définit comme la vitesse de déplacement de ce point suivant l'équation 1.1 :

$$\vec{v}(v_x, v_y) = (\frac{dx}{dt}, \frac{dy}{dt}) \tag{1.1}$$

Le flot optique ne peut être déterminé que par l'étude des variations de la fonction d'intensité I au cours du temps t, cela peut s'exprimer par la discrétisation ou la dérivée temporelle de l'équation 1.1 comme suit :

$$\frac{d}{dt}I(x, y, t) = \frac{\partial I}{\partial t} + v(v_x, v_y).\nabla I \tag{1.2}$$

Où :

- $I(x, y, t)$: l'intensité lumineuse d'un point $p(x, y)$ à l'instant t
- ∂I : dérivée partielle d'ordre un
- ∇I : le gradient de l'mage

La détermination du flot optique est assez difficile et cela est dû aux variations de l'éclairage de la scène au cours du temps. Donc la prise en considération de la contrainte qui suppose que l'intensité ne varie pas au cours du temps est indispensable, cela est discuté dans la section suivante.

1.3 Contraintes

Le changement de la luminance dans la scène engendre un mouvement qui ne correspond pas au mouvement réel dans les images, d'où les contraintes suivantes doivent prises en considération :

1.3.1 Contrainte de conservation de l'intensité

La contrainte de conservation de l'intensité lumineuse suppose que pendant le déplacement les intensités lumineuses des pixels ne varient pas au cours du temps (restent constantes) (voir Figure 1.2). Cette contrainte peut être exprimée par l'équation suivante :

$$\frac{d}{dt}I(x,y,t) = 0. \tag{1.3}$$

L'approximation discrète de cette dernière équation au temps t est définie comme suit :

$$I(x + v_x, y + v_y, t + \delta t) = I(x,y,t) \tag{1.4}$$

où $v = (v_x, v_y)$ est le vecteur de flot optique d'un pixel $p = (x,y)$ du temps t au temps $t + \delta t$.

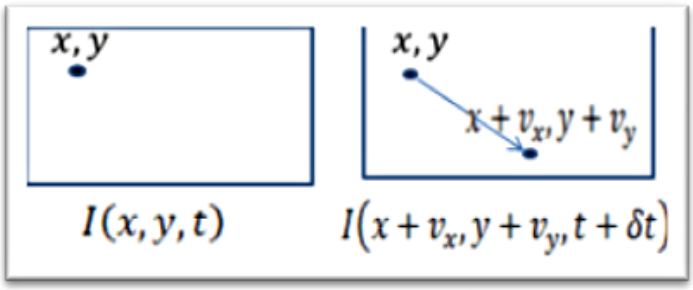

FIGURE 1.2 – L'hypothèse de conservation d'intensités

Par l'utilisation le développement de Taylor du premier ordre au point $p = (x,y)$ on obtient :

$$I(x + v_x, y + v_y, t + \delta t) = I(x,y,t) + v_x\frac{\partial I}{\partial x} + v_y\frac{\partial I}{\partial y} + \frac{\partial I}{\partial t} + \varepsilon \tag{1.5}$$

En négligeant le terme d'ordre supérieur ε on obtient :

$$I(x + v_x, y + v_y, t + \delta t) \approx I(x,y,t) + v_x\frac{\partial I}{\partial x} + v_y\frac{\partial I}{\partial y} + \frac{\partial I}{\partial t} \tag{1.6}$$

Et donc :

$$\frac{I(x + v_x, y + v_y, t + \delta t) - I(x,y,t)}{\partial t} = \frac{\partial I}{\partial x}\frac{v_x}{\partial t} + \frac{\partial I}{\partial y}\frac{v_y}{\partial t} + \frac{\partial I}{\partial t} \tag{1.7}$$

avec $\frac{\partial I}{\partial x} = I_x$, $\frac{\partial I}{\partial y} = I_y$, et $\frac{\partial I}{\partial t} = I_t$ les dérivées partielles d'ordre un du pixel $p = (x,y)$ au temps t.

1.3.2 Le déplacement et l'intervalle le temps sont petits

On peut estimer soit le vecteur vitesse soit le vecteur déplacement noté $d = (d_x, d_y)$ puisqu'ils sont équivalents dans un petit intervalle de temps δt.
Passage à la limite $\delta t \to 0$, on a donc :

$$I_x\frac{\partial x}{\partial t} + I_y\frac{\partial y}{\partial t} + I_t = 0 \tag{1.8}$$

Suivant l'équation 1.1 on a : $v_x = \frac{\partial x}{\partial t}$ et $v_y = \frac{\partial y}{\partial t}$, alors on peut écrire :

$$I_x v_x + I_y v_y = -I_t \tag{1.9}$$

Cette dernière équation(1.9) est connue sous le nom « équation de flot optique ou Equation de Contrainte de Mouvement Apparent (ECMA)», l'ECMA peut s'écrire vectoriellement comme suit :

$$\vec{V}.\vec{\nabla} I = -I_t \tag{1.10}$$

avec $\nabla I = [I_x, I_y]^T$ les gradient spatiaux de l'image au pixel $p(x, y)$ à l'instant t, I_t le gradient temporel (la dérivée de l'intensité par rapport au temps et $V = [v_x, v_y]^T$ le vecteur vitesse transposé.

Comme on peut le constater, l'ECMA définie une seule contrainte pour deux variable (v_x et v_y) sur le mouvement dans la séquence, cependant, cette contrainte ne permet pas à elle seule de mesurer d'une façon unique le champ de vitesse (les deux composantes de V). On a un problème mal posé puisque on ne dispose que d'une seule équation (1.10) pour résoudre un système à deux inconnues c'est-à-dire que seule la composante normale de la vitesse peut être mesurée comme suit :

$$V_n(x, y, t) = -\frac{\partial I}{\partial x}(x, y, t).\frac{\vec{\nabla} I(x, y, t)}{\|\vec{\nabla} I(x, y, t)\|^2} \tag{1.11}$$

Ce phénomène est connu sous le nom de problème d'ouverture qui stipule que pour une position donnée w l'équation du flot n'admet pas de solution unique, donc seul le flot optique normal au contour de l'image est estimé, c'est-à-dire le flot orthogonal au contour local de l'image et qui est orientée dans la direction du gradient(Figure 1.3)

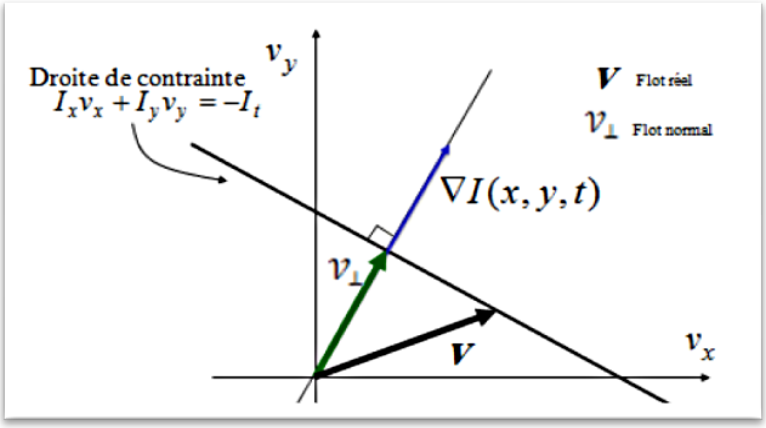

FIGURE 1.3 – Problème d'ouverture

Il faut noter que l'hypothèse de conservation de l'intensité a des limites dans les cas des variations brusques de l'illumination de la scène réelle. L'estimation de mouvement est un problème qui n'a pas toujours de solution dans le cas d'une occultation [1] .

1. Le recouvrement ou non d'une zone pendant le déplacement causé par un autre objet en mouvement

1.3.3 Cohérence spatiale

Les deux contraintes d'écrites ci-dessus ne permettent pas d'estimer le mouvement puisqu'on a un système sur-contrainte donc une troisième contrainte est indispensable, cette dernière est dite "la cohérence spatiale" et suppose que les pixels voisins/adjacents au celui en cours d'étude ont un déplacement proche (direction, la vitesse).

Chapitre 2

Méthodes d'Estimation de mouvement

Sommaire

2.1 Introduction

Nous pouvons classer les méthodes d'estimation de mouvement en cinq classes sont ainsi constitués : les méthodes différentielles, les méthodes de mise en correspondance de blocs, les méthodes fréquentielles, les méthodes neuronales et les méthodes multi-échelles. Dans ce qui suit nous détaillons toutes ces méthodes.

2.2 Méthodes différentielles

Les méthodes différentielles font parties des techniques les plus utilisées pour l'estimation de mouvement, elles sont basées sur la mesure des dérivées spatio-temporelles des images. Ces

méthodes peuvent être classées en deux catégories, les méthodes locales qui peuvent optimiser une fonction d'énergie locale et les méthodes globales qui tentent de minimiser une fonction d'énergie globale.

2.2.1 Horn et Schunck

La méthode de Horn et Schunck 1981 propose de régler le problème d'ouverture par l'ajout d'une contrainte de lissage globale sur le flot optique. Le principe de cette méthode est de combiner l'optimisation de la contrainte du flot optique (ECMA) avec un terme de régularisation global dont le but est de minimiser une expression de la forme de l'équation 2.1 sur toute l'image.

$$\iint [(\nabla I(x,y,t).V(x,y,t) + I_t(x,y,t))^2 + \alpha^2(\nabla v_x(x,y,t))^2 + (\nabla v_y(x,y,t))^2]dxdy \quad (2.1)$$

Où le facteur α correspond à l'influence du terme de régularisation.

Le premier terme dans l'expression ci-dessus est une erreur quadratique moyenne sur la contrainte de mouvement, tandis que le second est un terme de régularisation : il permet de s'assurer que le gradient du champ de vecteur de mouvement prend de petites valeurs ("lissage" de la solution). Une version itérative de la minimisation de l'équation 2.1 est obtenue par l'utilisation de l'approximation de Laplacien et la méthode de Gauss-Seidal :

$$\begin{cases} v_x^{l+1} = v_x^{-l} - \frac{I_x[I_x v_x^{-l} + I_y v_y^{-l} + I_t]}{\alpha^2 + I_x^2 + I_y^2} \\ \\ v_y^{l+1} = v_y^{-l} - \frac{I_y[I_x v_x^{-l} + I_y v_y^{-l} + I_t]}{\alpha^2 + I_x^2 + I_y^2} \end{cases} \quad (2.2)$$

Avec : l représente le nombre d'itérations, v_x^0 et v_y^0 correspond aux vitesses initiales, et v_x^{-l}, v_y^{-l} sont les vitesses moyennes.

Les dérivées statio-temporelles sont calculées par la méthode proposée par Horn et Schunck, qui considère un pixel centré dans l'espace et le temps (voir ka figure 2.1), comme suit :

$$I_x = \frac{1}{4}[I(i+1,j,k) + I(i+1,j,k+1) + I(i+1,j+1,K) + I(i+1,j+1,k+1)] - \frac{1}{4}[I(i,j,k) + I(i,j,k+1) + I(i,j+1,k) + I(i,j+1,k+1)]$$

$$I_y = \frac{1}{4}[I(i+1,j,k+1) + I(i+1,j+1,k) + I(i+1,j,K+1) + I(i+1,j+1,k+1)] - \frac{1}{4}[I(i,j,k) + I(i,j,k+1) + I(i,j+1,k) + I(i,j+1,k+1)]$$

$$I_t = \frac{1}{4}[I(i,j,k+1) + I(i+1,j,k+1) + I(i,j+1,K+1) + I(i+1,j+1,k+1)] - \frac{1}{4}[I(i,j,k) + I(i+1,j,k) + I(i,j+1,k) + I(i,j+1,k)]$$

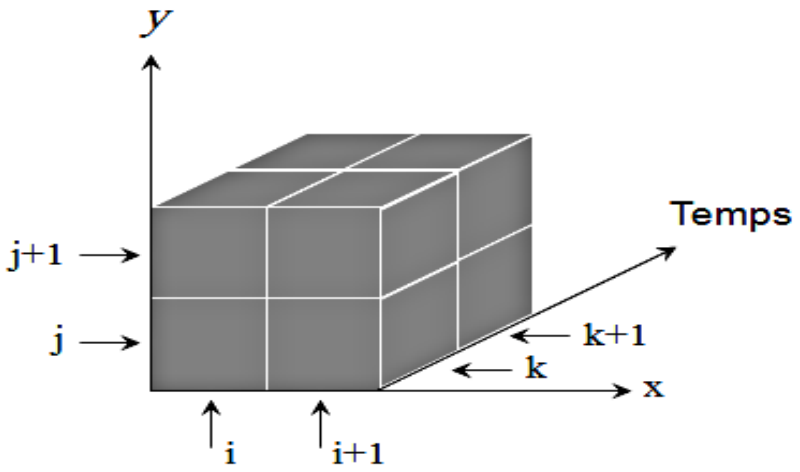

FIGURE 2.1 – Calcul des Dérivées selon Horn et Schunck

Exemple :

Considérant une sphère en rotation sur elle même (Figure 2.2), leur flot optique obtenue par la méthode de Horn et Schunck après une itération, 16 et 100 itérations est schématisé sur la Figure 2.3

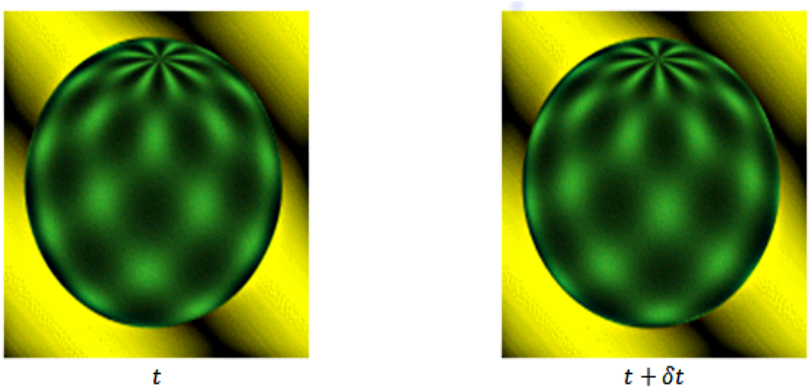

FIGURE 2.2 – sphère en rotation

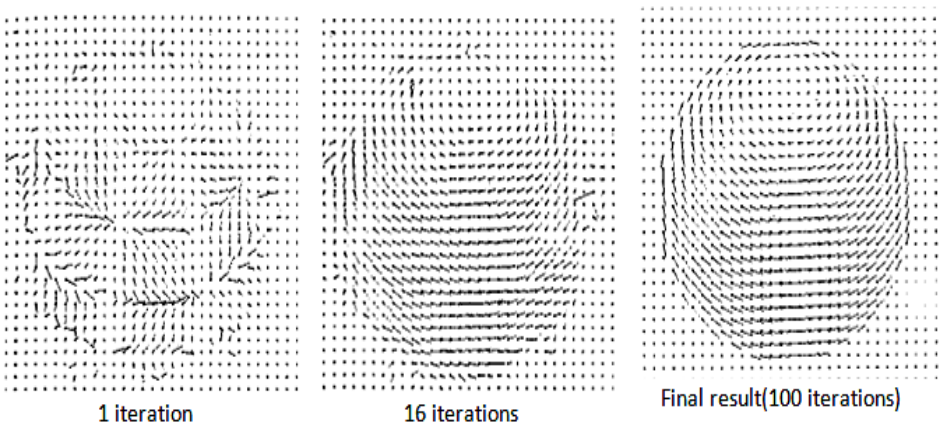

FIGURE 2.3 – Flot optique estimé par la méthode de Horn et Schunck

2.2.2 Méthode de Lucas et Kanade

La contrainte supplémentaire proposée par Lucas et Kanade est que le flot optique soit localement constant, c'est-à-dire que le flot optique $v = (v_x, v_y)$ est constant dans un petit voisinage ou fenêtre qui est centrée au pixel en cours de traitement d'indice (x_i, y_i), alors les pixels qui l'entourent sont indexés de $1..n$.

Cela nous a conduit à un ensemble de n équations, alors l'équation du flot optique 1.9 peut s'écrire comme suit :

$$\begin{bmatrix} I_{x1} & I_{y1} \\ I_{x2} & I_{y2} \\ \vdots & \vdots \\ I_{xn} & I_{yn} \end{bmatrix} \begin{bmatrix} v_x \\ v_y \end{bmatrix} = \begin{bmatrix} -I_{t1} \\ -I_{t2} \\ \vdots \\ -I_{tn} \end{bmatrix}. \tag{2.3}$$

On peur écrire l'équation précédente sous la forme matricielle comme suit :

$$A\vec{V} = -b \tag{2.4}$$

Les auteur utilisent la méthode des moindres carrés pour minimiser $\|A\vec{V} + b\|^2$ comme suit :

$$\vec{V} = (-A^T A)^{-1} A^T b, \tag{2.5}$$

plus formellement on peut écrire :

$$\begin{bmatrix} v_x \\ v_y \end{bmatrix} = - \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} (A^T I_t), \tag{2.6}$$

Exemple : Soit l'image réelle de transverse du thorax à rayons X présentée dans la figure 2.4 nommée Thorax (source : University Medical Center). Cette image présente deux organes du corps humains (cœur et poumons). A partir de cette image, on a appliqué une translation avec un pixel pour générer notre séquence d'images nommée «Paire-Trans-1p. La figure 2.4 présente le flot optique réel et celui estimé par la méthode de Lucas et Kanade avec un fenêtre de taille 11×11.

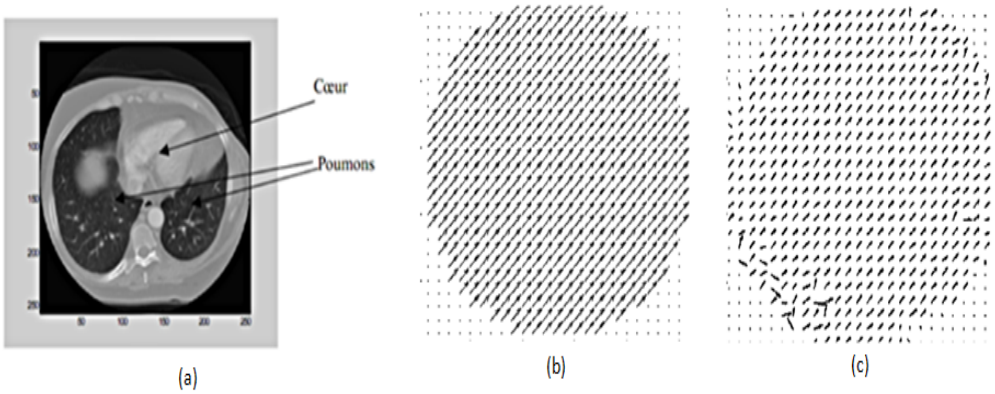

FIGURE 2.4 – (a)L'image réelle Thorax, (b) flot optique réel, (c) flot optique estimé avec la méthode Lucas et Kanade

2.3 Méthodes de mise en correspondance de blocs

L'objectif des méthodes de mise en correspondance de blocs est de minimiser une mesure de similarité entre les images de la séquence. Contrairement aux méthodes différentielles, le même vecteur de mouvement est estimé sur un bloc complet de l'image, cela peut être vu comme une contrainte supplémentaire de lissage pour pouvoir estimer le flot optique. Cette contrainte rend ces méthodes interactives et nécessite un temps de calcul excessivement long, ce qui le rend impropre à une implémentation pratique et en temps réel. Le principe de ces méthodes est décrit ci-dessous.

2.3.1 Principe général

Les algorithmes de mise en correspondance de blocs (MCB) estiment le mouvement entre deux images et le même vecteur de mouvement est obtenu bloc par bloc, c'est-à-dire que tout le bloc prend le même déplacement. Alors, une image est divisée en blocs non superposés de pixels de taille $(N \times N)$. Le bloc actuel noté B_c dans l'image courante I_c est comparé aux blocs correspondants (appelés blocs candidats) dans une zone/fenêtre de recherche de taille $(N + 2W)(N + 2W)$ pixels dans l'image précédente (ou image de référence I_r), où W est le déplacement maximal autorisé et N^2 est le nombre de pixels dans le bloc. Les vecteurs de mouvement peuvent être estimés par la minimisation d'une mesure d'erreur de similarité ou de correspondance entre un bloc dans l'image courante et les blocs candidats. Le vecteur candidat qui minimise une mesure de similarité entre les blocs dans les deux images est défini comme suit :

$$v^* = \arg \min_{x,y \in W} [B_c(x, y), B_r(x + v_x, y + v_y)] \tag{2.7}$$

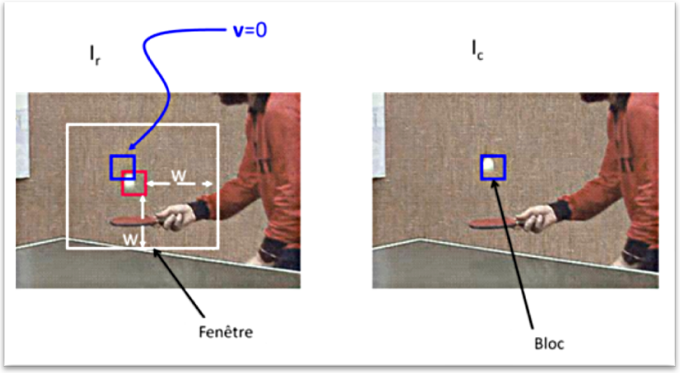

FIGURE 2.5 – Principe des MCB

En général, la fenêtre de recherche correspond à une zone carrée centrée sur le bloc B_r de l'image de référence. La structure de la fenêtre de recherche à un impact important à la fois sur la complexité de l'algorithme d'estimation de mouvement et sur la précision des résultats obtenus. Les différents types de méthodes de MCB diffèrent entre eux selon : le critère de ressemblance et la stratégie de recherche, c'est-à-dire la manière avec laquelle la fenêtre de recherche est parcourue afin de trouver le vecteur de mouvement, qui minimise la similarité entre les blocs. Par la suite, nous discuterons sur ces deux critères.

2.3.2 Critère de similarité

Comme montré dans l'équation précédente 2.7, les blocs dans l'image courante et l'image de référence sont comparés selon une métrique mesurant leur ressemblance, les critères les plus utilisés pour mesurer la ressemblance sont :

- Somme des différences carrées (Square Sum Difference en anglais) : l'approche la plus intuitive consiste à utiliser la distance Euclidienne entre les vecteurs B_c et B_r, c'est-à-dire calculer la norme de leur différence comme suit :

$$SSD(B_c, B_r) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} [B_c(x+i, y+j) - B_r(i,j)]^2 \tag{2.8}$$

  où les valeurs $B_c(i,j)$ et $B_r(i,j)$ représentent les intensités du pixel d'indice (i,j) dans le bloc courant et le bloc de référence respectivement et (x,y) est le vecteur candidat.
- Somme de la différence absolue (Sum Absolute Value en anglais) : Un autre critère de comparaison qui peut réduire les limites du SSD -en se qui concerne la complexité et que la puissance carrée a tendance à augmenter l'erreur en cas de bruit- qui est la somme des différences absolues (SAD :), défini par comme suit :

$$SAD(B_c, B_r) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |B_c(x+i, y+j) - B_r(i,j)| \tag{2.9}$$

2.3.3 Les algorithmes de MCB

Selon la stratégie de recherche on peut distingués plusieurs algorithmes de MCB, dans ce qui suit nous décrivons les algorithmes les plus importants.

- **Recherche exhaustive (Full Search(FS))**

  Afin d'obtenir la meilleure correspondance, il est nécessaire de comparer le bloc courant avec tous les blocs candidats dans la fenêtre de recherche. Cet algorithme examine de manière exhaustive toutes les positions dans la fenêtre de recherche donc le bloc B_c est comparé à tous les blocs candidats dans l'image de référence à la recherche lu critère de similarité le plus petit possible (voir la Figure 2.6). L'algorithme FS est extrêmement complexe, puisque le nombre de blocs candidats est de $(P-N) \times (Q-N) \approx PQ$ et pour chaque bloc candidat, nous devons calculer la mesure de similarité (avec P, Q sont la taille des images), alors une implémentation en temps réel, des stratégies de recherche rapides et efficaces ont été explorées.

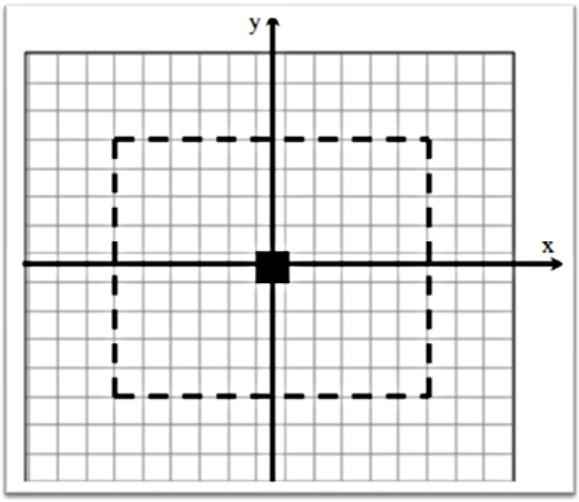

FIGURE 2.6 – La recherche exhaustive

- **Recherche en trois étapes (Three Step Search)**

  Le processus d'estimation de mouvement est résumé en trois étapes et la taille de la fenêtre de recherche est réduite de moitié à chaque étape indépendamment du vecteur vitesse estimé. Donc le pixel du centre d'indice (x, y) et les 8 voisins de pas d-qui représente le déplacement maximal- sont examinés, à l'étape suivante la taille de la fenêtre de recherche est réduite de moitié et le pixel qui minimise le critère de ressemblance devient le nouveau centre de recherche à cette étape. La même procédure que la précédente se répète jusqu'à ce que la taille du pas d devienne un $(= 1)$. L'algorithme de recherche en trois étapes permet de trouver des déplacements de $\pm$ sept pixels dans les deux directions avec seulement 25 calculs de similarités c'est-à-dire neuf pour la première étape et huit pour la deuxième et la troisième étape. Le schéma de recherche correspondant à cet algorithme est représenté sur la Figure 2.7.

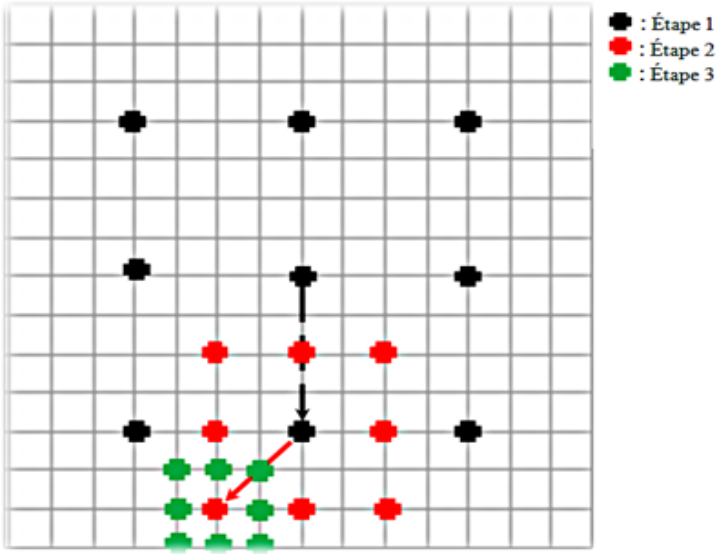

FIGURE 2.7 – Recherche en trois étapes

- **Recherche sur une grille en diamant (Diamond Search(DS))**

  Le voisinage dans l'algorithme de DS prend le forme d'un diamant et non pas un carré comme les algorithmes décrits précédemment et que le nombre des étapes est

illimite. L'algorithme DS utilise deux modèles de recherche (Figure 2.8) :
— LDSP : Large Diamant Search Pattern (voisinage d'ordre 1) avec neuf points de recherche.
— SDSP : Small Diamant Search Pattern (voisinage d'ordre 2) avec cinq points de recherche.

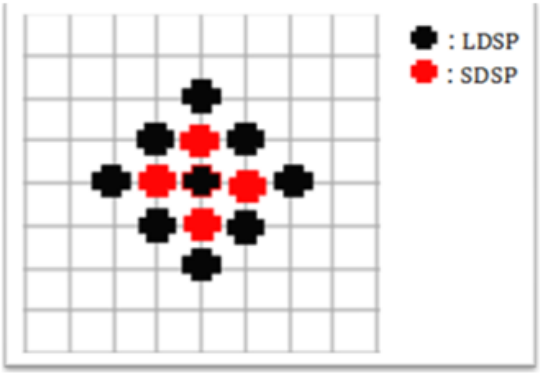

FIGURE 2.8 – Les deux modèles de l'algorithme DS

À la première étape de cette stratégie de recherche, le modèle LDSP est sélectionné, si la meilleure ressemblance n'est pas située au centre du diamant, le point central du diamant est alors remplacé par le nouvel emplacement trouvé. La recherche se poursuit en utilisant le modèle de recherche LDSP jusqu'à ce que l'emplacement qui minimise le critère de ressemblance soit au centre du diamant. Ensuite, en passe au modèle SDSP pour effectuer un processus de raffinement. Si le point qui minimise le critère de ressemblance est situé aux quatre points de recherche à l'entourage du point central du petit diamant, alors le point minimum sera le nouveau point central. Un sous-ensemble doit être vérifié à toute nouvelle étape, car le nouveau modèle est toujours partiellement superposé à l'ancien et l'algorithme s'arrête lorsque le meilleur point situé au centre du SDSP. La Figure 2.9 résume le processus de cet algorithme :

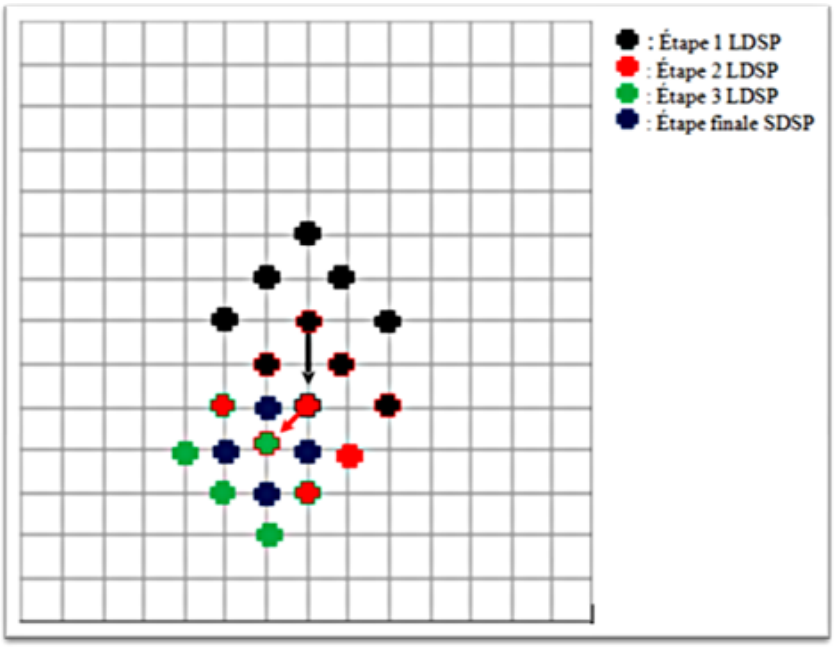

FIGURE 2.9 – L'algorithme DS

Exemple :

Avec l'image réelle présentée dans la figure 2.10 et la paire obtenue (Paire-Trans-1p) par la translation selon la diagonale d'un pixel suivant les axes X et Y de taille 256×256, les performances des algorithmes de mise en correspondance de blocs décrient précédemment (FS, TSS,DS) sont présentées dans le tableau 2.1 suivant et leurs flot optique est schématisé sur la figure 2.10.

	FS	**TSS**	**DS**
Moy_point	199.5156	23.1484	16.3398
Temps(s)	11	2.01	1.3

TABLE 2.1 – performances des méthodes de MCB

avec : **Moy_point** : nombre moyen de points recherchés pour un bloc (de taille 16×16). **Temps** : temps d'exécution en seconde.

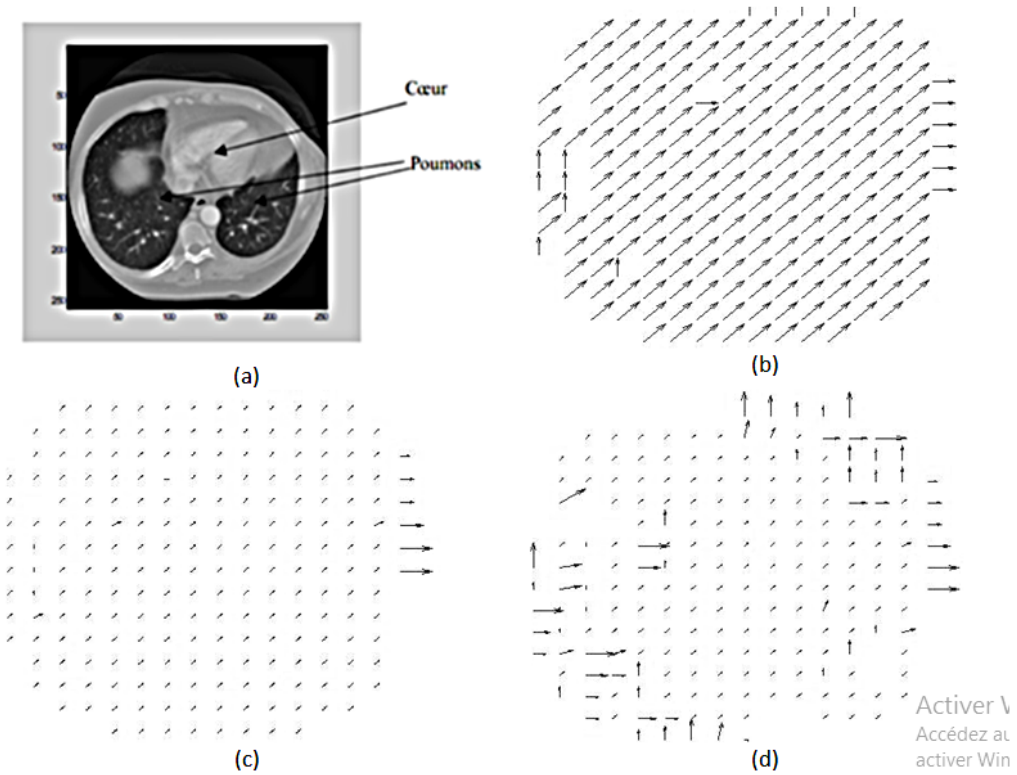

FIGURE 2.10 – Exemple des MCB, (a) l'image Thorax, (b) flot optique de FS, (c) flot optique de DS, (d) flot optique de TSS

2.4 Méthodes fréquentielles

2.4.1 Principe général

Les méthodes fréquentielles sont basées sur la relation entre les coefficients transformés (par exemple, transformée de Fourier ou filtre de Gabor) d'images décalées. Leurs principe est qu'un mouvement de translation pour un objet dans la scène dans le domaine spatial se traduit

dans le domaine fréquentiel par un plan passant par l'origine. L'objectif de telles méthodes est d'estimer les paramètres de ce plan (v_x, v_y). La transformée de Fourier de la séquence d'images $I(x, y, t)$ notée $F(f_x, f_y, f_t)$ en mouvement est donnée par l'équation suivante :

$$F(f_x, f_y, f_t) = F_0(f_x, f_y)\delta(v_x f_x + v_y f_y + f_t) \tag{2.10}$$

où : $F_0(f_x, f_y)$ est la transformée de Fourier à $t = 0$

δ : La distribution de Dirac

f_x, f_y : Fréquences spatiales et f_t : fréquence temporel.

Les fréquences spatiales sont inchangées, mais toutes les fréquences temporelles sont translatées par le produit de la vitesse et des fréquences spatiales. Cela donne l'équation de contrainte de flot optique dans l'espace des fréquences :

$$v_x f_x + v_y f_y + f_t = 0 \tag{2.11}$$

On distingue deux grandes familles des méthodes fréquentielles, les méthodes de la première famille utilise d'information de la phase du signal pour estimer le mouvement dites les méthodes basées sur la phase. Bien que les méthodes de la deuxième famille exploitent l'information de l'énergie de la transformé de Fourier par l'utilisation d'un banc de filtre de Gabor, par la suite nous décrivons un exemple de la famille basée sur la phase qui est "la corrélation de phase" et un autre exemple pour la deuxième famille qui est la méthode basée sur l'énergie de Heeger.

2.4.2 Méthodes fréquentielles basées sur la phase

Ces méthodes exploitent le principe que le décalage dans le domaine spatial est équivalent en un déphasage de phase dans le domaine fréquentiel, la méthode la plus connue est celle de la corrélation de phase qui sera expliquée par la suite.

- **Méthode de la corrélation de phase**

  Le principe de cette méthode est la recherche de la différence de phase à chaque fréquence entre les deux images et le calcul à nouveau de la transformée de Fourier inverse de ces images. Alors que la méthode n'utilise que l'information de phase pour la corrélation est relativement insensible aux changements d'illumination et au bruit.
  Soit $I_1(x, y)$la première image de la séquence et $I_2(x, y)$ une copie de $I_1(x, y)$ translatée à $\vec{d} = (d_x, d_y)$, selon le théorème de Fourier leurs transformées de Fourier sont liées par :

$$G_2(\vec{f}) = G_1(\vec{f}).e^{-2\pi\vec{f}\vec{d}} \tag{2.12}$$

  L'algorithme de corrélation de phase utilise le spectre de puissance croisé normalisé pour obtenir la translation entre ces deux images, en calculant le spectre de puissance croisé normalisé et en extrayant sa phase, on a donc l'équation 2.13 :

$$e^{j\phi(\vec{f})} = \frac{G_1(\vec{f})G_2^*(\vec{f}).e^{-2j\pi\vec{f}\vec{d}}}{|G_1(\vec{f})G_2^*(\vec{f}).e^{-2j\pi\vec{f}\vec{d}}|} \tag{2.13}$$

  Où $G_2^*(\vec{f})$ représente le conjugué complexe de $G_2(\vec{f})$ et ϕ indique la différence de phase.

  La transformée de Fourier inverse du spectre de puissance croisé normalisé donne une image complexe dont le module définit une surface bidimensionnelle avec des fonctions delta, c'est-à-dire des pics notés $p(x, y)$ aux positions correspondant aux décalages

spatiaux entre les deux images. Mathématiquement, la corrélation de phase est définie suivant l'équation 2.14 :

$$p(x,y) = F^{-1}\left(e^{j\phi(\vec{f})}\right) \tag{2.14}$$

Avec F^{-1}désignant la transformée de Fourier inverse.

Les coordonnées (d_x, d_y) correspondent au maximum de l'équation 2.14. Elles peuvent être utilisées comme estimation des composantes horizontales et verticales de la translation entre $I_1(x,y)$ et $I_2(x,y)$ comme suit :

$$(d_x, d_y) = \arg\max(p(x,y)) \tag{2.15}$$

Un exemple expliquant le processus de cette méthode est présenté sur la Figure 2.11.

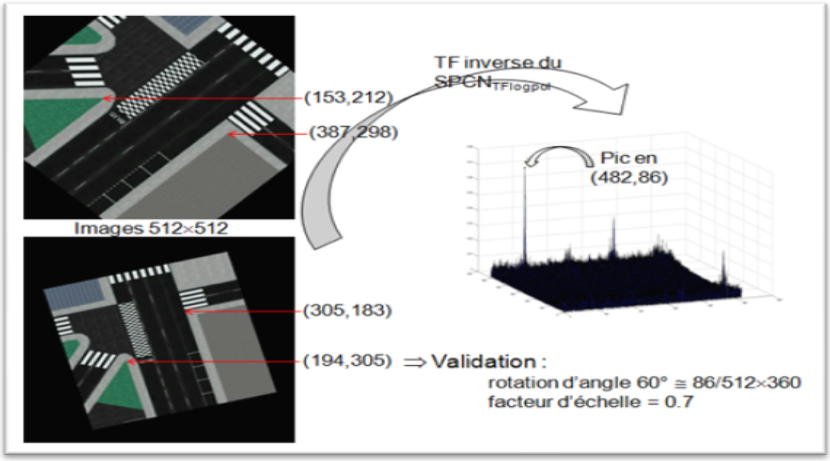

FIGURE 2.11 – Exemple expliquant la méthode de corrélation de phase, cas d'une translation

2.4.3 Méthodes fréquentielles basées sur l'énergie

Le mouvement par les approches basées sur l'énergie est obtenu par l'utilisation des filtres spatio-temporels appliqués à l'ensemble des images de la séquence c'est-à-dire se placer selon une orientation spatiotemporelle spécifique. Cela peut être vu dans l'exemple d'une séquence d'images d'une barre verticale en mouvement de translation vers la droite au cours du temps. Le mouvement apparait comme une orientation en espace-temps.

- **Méthode de Heeger**

  Cette méthode est formulée comme un ajustement des moindres carrés de l'énergie spatio-temporelle à un espace de fréquence. L'énergie locale est extraite à l'aide de filtres de Gabor-3d, avec 12 filtres à chacune des différentes fréquences spatiales, accordés à différentes orientations spatiales et différentes fréquences temporelles(Figure 2.12 [1]).

1. Heeger D.J.(1988), "Optical fow using spatiotemporal filters". Int.J. Com. Vision 1. pp.279-302

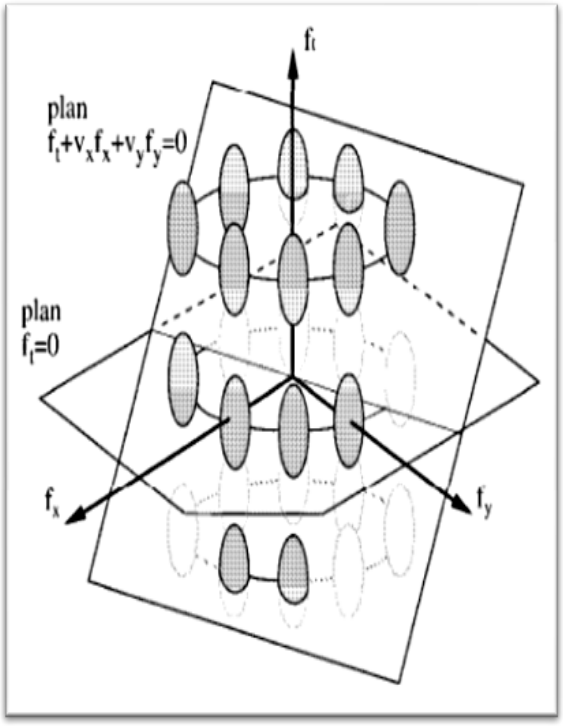

FIGURE 2.12 – Spectres de 12 filtres spatio-temporels orientés de Gabor

Un filtre de Gabor 3D est définit comme suit :

$$g(x,y,t) = \frac{1}{(2\pi)^{\frac{3}{2}}\sigma_x\sigma_y\sigma_t} \exp\left[-(\frac{x^2}{2\sigma_x^2} + \frac{y^2}{2\sigma_y^2} + \frac{t^2}{2\sigma_t^2})\right] \exp[2\pi j(f_{x_0}x + f_{y_0}y + f_{t_0}t)] \tag{2.16}$$

Où σ_x , σ_y et σ_t sont les écarts-types de la composante Gaussienne du filtre de Gabor.

Idéalement, pour un seul mouvement de translation, les réponses $R(v_x, v_y)$ de ces filtres sont concentrées autour d'un plan dans l'espace de fréquence. Heeger mesure la réponse fréquentielle d'un filtre d'énergie de Gabor accordée à la fréquence (f_x, f_y, f_t) de la façon suivante :

$$R(v_x, v_y) = exp\left[\frac{-4\pi^2\sigma_x^2\sigma_y^2\sigma_t^2(v_x f_x + v_u f_y + f_t)^2}{(v_x\sigma_x\sigma_t)^2 + (v_y\sigma_y\sigma_t)^2 + (\sigma_x\sigma_y)^2}\right]. \tag{2.17}$$

La solution proposée par Heeger (1988) est donc une minimisation de l'erreur entre l'énergie de mouvement des filtres pour chacune des images et l'énergie de mouvements devinée sur une image en translation. La vitesse estimée en pixel par image est donnée par le minimum de la fonction $f(v_x, v_y)$, où m_i et R_i représentent respectivement, la somme des énergies et la somme des réponses des filtres qui ont la même orientation spatiale que le filtre i définies comme suit :

$$f(v_x, v_y) = \sum_{i=1}^{12}\left(m_i - \overline{m}_i\frac{R_i(v_x, v_y)}{\overline{R}_i(v_x, v_y)}\right)^2 \tag{2.18}$$

Avec,

$$\overline{m}_i = \sum_{j\in M_i} m_j, \text{ et } \overline{R}_i = \sum_{j\in M_i} R_j(v_x, v_y)$$

Exemple :

La figure 2.13 montre le flot optique estimé par les deux méthodes fréquentielles : la corrélation de phase et la méthode de Hegeer en utilisant la séquence taxi de Hambourg.

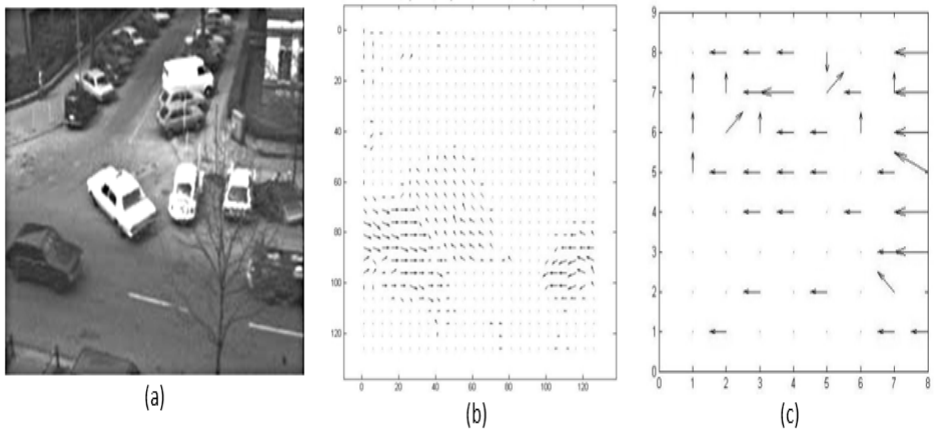

FIGURE 2.13 – (a) image de la séquence Taxi de Hambourg, (b) flot optique estimé par la méthode de Heeger,(c) b) flot optique estimé par la méthode de corrélation de phase

2.5 Méthodes neuronales

Dans la pratique, un autre type de méthodes d'estimation de mouvement est utilisé : les méthodes «neuronales». Ces méthodes utilisent l'apprentissage ainsi qu'un calcul parallèle. Les réseaux de neurones peuvent apprendre ; cela veut dire que les sorties d'un réseau de neurones ne sont pas limitées aux entrées fournies par l'expert. Les réseaux de neurones sont capables de généraliser leurs entrées. Ainsi, pratiquement tous les algorithmes de réseaux de neurones présentent un parallélisme inspiré des réseaux biologiques. Dans ce qui suit, nous décrivons quelques méthodes récentes d'estimation de mouvement basées sur les réseaux de neurones. La première méthode est basée sur les réseaux de neurones à convolution (méthode d'Ahmadi et Patras, 2017), cependant la deuxième méthode est basée sur les cartes auto-organisatrices (méthode de Méthode de Garcia et al. 2016).

2.5.1 Méthodes neuronales basée sur les réseaux à convolution

Les réseaux de neurones à convolution sont considérés comme une nouvelle technologie dans la plupart des domaines de la vision par ordinateur, surtout pour l'estimation de mouvement, la classification et la reconnaissance de formes et de gestes. Dans ce qui suit, nous présentons la méthodes d'Ahmadi et Patras [?], pour l'estimation du flot optique.

- **Méthode d'Ahmadi et Patras**

  Le travail d'Ahmadi et Patras a pour objectif l'estimation du mouvement entre une paire d'images. Pour atteindre cet objectif, les auteurs ont proposé un réseau de neurones à convolutions. L'architecture du réseau est présentée sur la Figure 2.14.

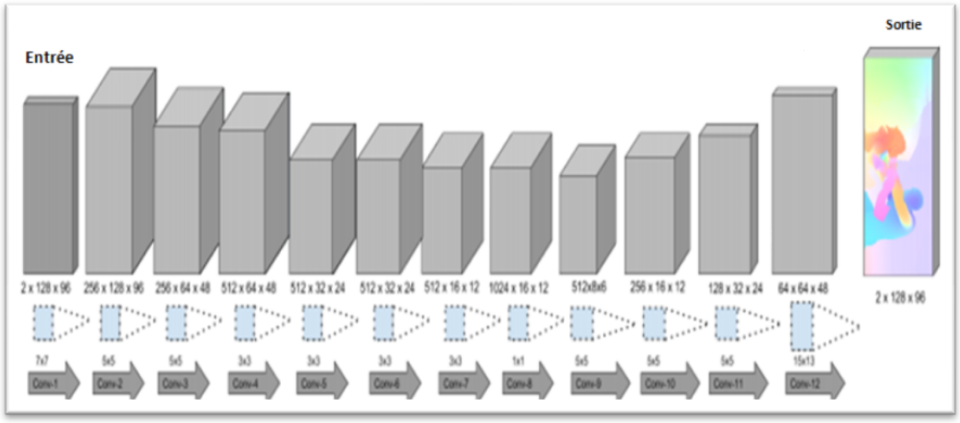

FIGURE 2.14 – L'architecture du CNN proposé par Ahmadi et Patras [?]

Le réseau est constitué d'onze couches plus les couches d'entrée et de sortie. L'architecture du réseau peut être divisée en deux grandes parties. Une première partie pour la représentation des données et des paramètres du réseau qui nécessite quatre opérations de sous-échantillonnages et une deuxième partie pour l'estimation et la reconstruction du champ de mouvement. L'estimation du mouvement nécessite 8 opérations d'échantillonnage, quatre opérations d'échantillonnage pour chaque partie.

Comme dans les approches classiques d'estimation de mouvement dans une séquence d'images, les auteurs ont principalement basé sur l'hypothèse de conservation de l'intensité au cours du temps. La différence avec le travail d'Ahmadi et Patras est que les auteurs utilisent l'hypothèse de conservation de l'intensité pour la création de l'ensemble d'apprentissages du réseau CNN.

Le réseau de neurones fait l'apprentissage par l'optimisation de la fonction de coût suivante :

$$E(F) = \sum_{x,y=1}^{M} \sqrt{(v_x I_x + v_y I_y + I_t)^2 + \epsilon} \tag{2.19}$$

Le calcul des dérivées par rapport aux poids du réseau est effectué selon la formule suivante :

$$\frac{\partial E}{\partial w} = \frac{\partial E}{\partial F}\frac{\partial F}{\partial w} \tag{2.20}$$

avec $\frac{\partial F}{\partial w}$ représentant les dérivées partielles du flot estimé noté F par rapport à ses poids w, qui se calcule de la manière suivante :

$$\frac{\partial E}{\partial F} = \begin{bmatrix} \frac{\partial E}{\partial v_x} \\ \frac{\partial E}{\partial v_y} \end{bmatrix} = \begin{bmatrix} \sum_{x,y=1}^{M} \frac{I_x(v_x I_x + v_y I_y + I_t)}{\sqrt{(v_x I_x + v_y I_y + I_t)^2 + \epsilon}} \\ \sum_{x,y=1}^{M} \frac{I_y(v_x I_x + v_y I_y + I_t)}{\sqrt{(v_x I_x + v_y I_y + I_t)^2 + \epsilon}} \end{bmatrix} \tag{2.21}$$

Les auteurs utilisent le modèle multi-échelles pour surmonter le problème des grands déplacements imposés par l'utilisation de la contrainte du flot optique. La deuxième image est reconstruite à partir de la première et du champ de mouvement estimé, ensuite la nouvelle paire d'images est considérée comme entrée du CNN pour calculer une mise a jour du champ de mouvement. Plusieurs itérations sont effectuées à chaque échelle.

Exemple :

Un exemple de la méthode d'Ahmadi et Patras est illustré sur la Figure 2.15 de la base de données KITTI 2015. Le flot optique estimé est représenté par la palette de couleurs qui associe une couleur à une direction et une saturation à l'amplitude du vecteur vitesse.

FIGURE 2.15 – Exemple de la méthode d'Ahmadi et Patras, (a) image de la séquence, (b) flot optique estimé, (c) codage des couleurs

2.5.2 Méthodes neuronales basée sur les cartes

Les cartes auto-organisatrices présentent l'avantage d'une concurrence entre les neurones d'une couche pour l'optimisation d'une entité ou fonction d'énergie. Nous détaillons ci-dessous la méthode Garcia et al [**?**]. qui utilise le modèle Growing Neural Gas.

- **Méthode de Garcia et al**.

  Cette méthode porte sur une architecture basée sur un réseau de neurones qui estime le mouvement des objets composant une scène vidéo dont le but est de suivre les

objets mobiles. L'utilisation de l'architecture de réseau de neurones permet l'estimation simultanée du mouvement global et local ainsi que la représentation des objets.

Les auteurs utilisent le modèle GNG Growing Neural Gas, sans topologie prédéfinie, pour gérer les séquences d'images. Pour ce modèle et selon les auteurs, il est seulement nécessaire de déplacer la structure du réseau sans nécessité d'ajouter ou de supprimer des neurones. De plus, la structure du réseau de neurones stable permet d'utiliser les vecteurs de référence des neurones comme des caractéristiques dans la séquence vidéo.

Des mesures d'erreur locales sont captées pendant le processus d'adaptation, cela permet d'insérer de nouveaux neurones, donc chaque nouveau neurone est inséré à côté du neurone qui a la plus grande erreur. A chaque étape d'adaptation, une connexion entre le gagnant et le deuxième neurone le plus proche est créée selon le principe de l'algorithme d'apprentissage de Hebb. Ceci est continué jusqu'à ce qu'une condition d'arrêt soit vérifiée.

Le réseau est défini comme suit :

— Un ensemble A de nœuds (neurones). A chaque neurone $c \in A$ est associé un vecteur de référence $w_c \in R^d$.

— Un ensemble d'arêtes (connexions) entre des paires de neurones.

L'algorithme GNG est présenté par la Figure 2.16 ;il est utilisé pour obtenir la représentation de la première image dans la séquence. Cependant, pour les images suivantes, les positions finales (vecteurs de référence) des neurones obtenus à partir de l'image précédente sont utilisées comme points de départ.

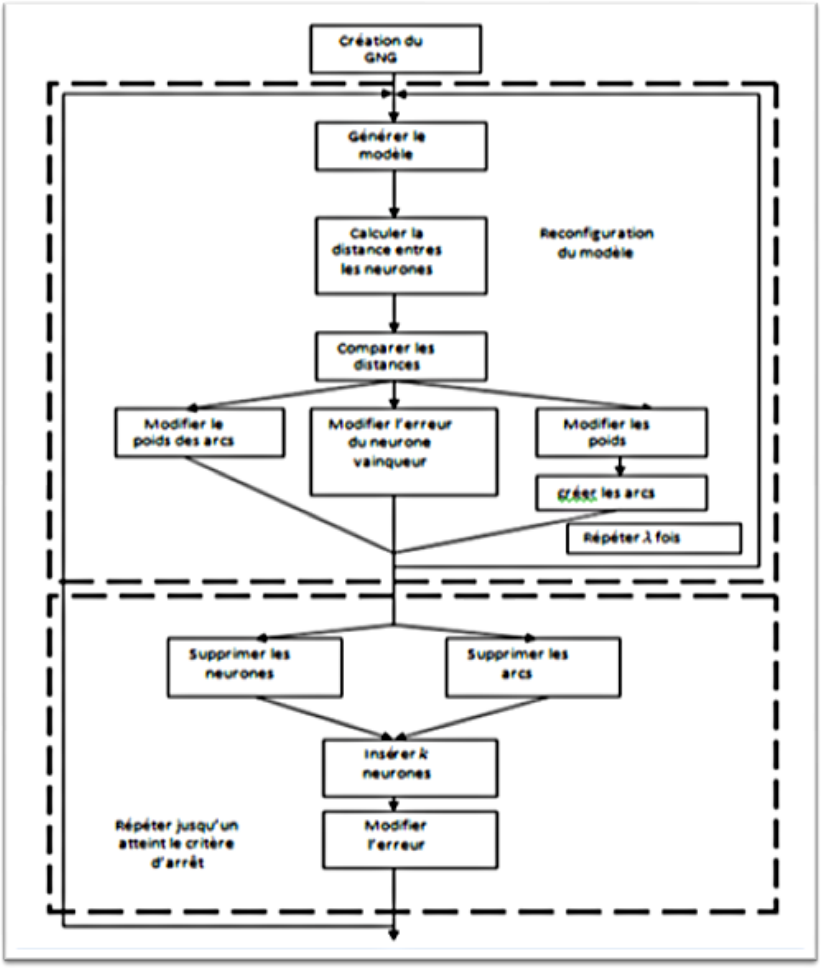

FIGURE 2.16 – L'algorithme d'apprentissage de GNG [?]

Exemple :

Un exemple avec la méthode de Garcia et al., est illustré sur la Figure 2.17 de la base de données CAVIAR [2].

2. https://computervisiononline.com/dataset/1105138692

FIGURE 2.17 – Exemple de la méthode de Garcia et al. tiré de [?]

2.6 Approche multi-échelles

Les méthodes d'estimation de mouvement basée sur l'ECMA sont inadaptées lorsque la quantité de déplacement est importante, pour cela les approches multi-échelles sont apparues. Ces approches permettent d'éviter un certain nombre des problèmes de minima locaux, d'améliorer la précision du calcul du flot optique, de réduire la complexité de calcul et de gérer les grands déplacements, puisque leurs principes est d'utiliser des images sous-échantillonnées et d'estimer le mouvement d'une manière incrémentale c'est-à-dire faire le calcul à une échelle grossière puis améliorer ce résultat en passant à une échelle plus fine.

Leur principe est de réduire progressivement la résolution des images de la séquence pour construire une séquence d'images de plus en plus grossière a partir de la séquence initiale par l'utilisation d'une pyramide [3](voir Figure 2.18). Plus précisément, le mouvement est d'abord estimé au niveau de résolution le plus grossier. Par conséquent, une estimation robuste est obtenue, qui capture les grandes tendances du mouvement. Le champ de mouvement est ensuite projeté au niveau de résolution plus fin suivant et affiné de manière itérative.

3. La pyramide est une méthode pour la représentation multi-résolution d'une image, on trouve essentiellement le pyramide Gaussienne et Laplacienne.

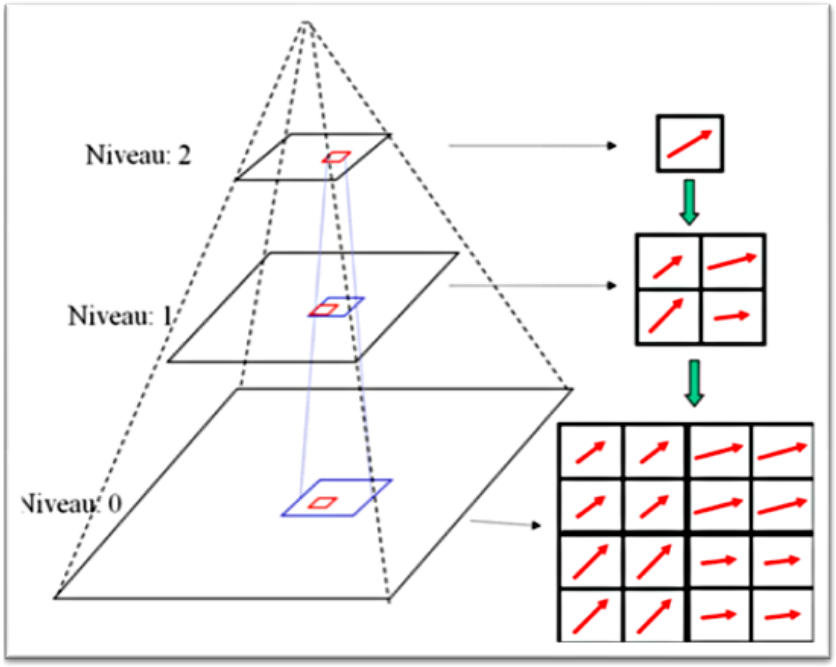

FIGURE 2.18 – Principe des approches multi-échelles

Formellement, à chaque échelle i on calcule le flot optique associé noté v^i avec n'importe quelle méthode d'estimation, ensuite on propage ce champs de mouvement calculé a cet niveau au niveau inférieur $i-1$ par l'utilisation de la technique d'interpolation et ainsi de suite jusqu'au niveau le plus fin (niveau 0) qui correspond à l'image initiale. On récupère alors le flot optique final.

L'estimation résultante est alors :

$$v^{(i,i-1)} = v^i + v^{(i-1)} \tag{2.22}$$

Un exemple d'estimation de mouvement multi-échelle est illustré sur la Figure 2.20 avec la séquence "Bus" (voir la Figure 2.19).

FIGURE 2.19 – Séquence d'images "Bus"

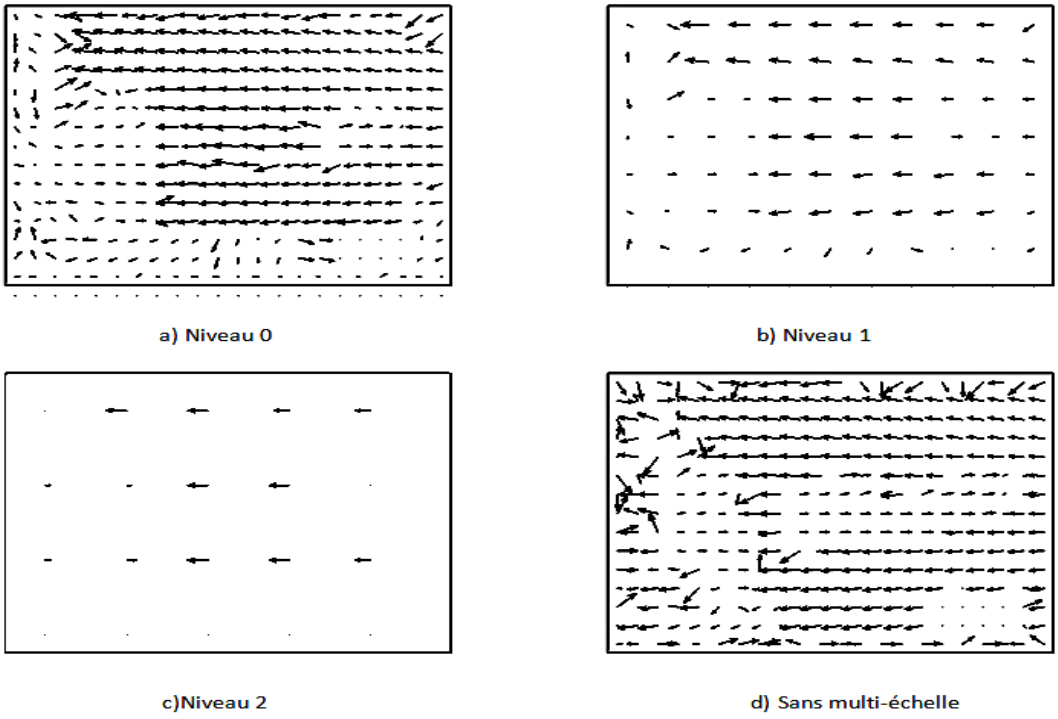

FIGURE 2.20 – Exemple d'estimation multi-échelle