

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE
SCIENTIFIQUE
UNIVERSITÉ LARBI BEN M'HIDI- OUM EL BOUAGHI
FACULTÉ DES SCIENCES EXACTES ET DES SCIENCES DE LA NATURE ET DE LA VIE
Département de Mathématiques et Informatique (MI)

POLYCOPIÉ DE COURS

Détection et Estimation de mouvement

NIVEAU : 2ÈME ANNÉE MASTER
SPÉCIALITÉ : VISION ARTIFICIELLE

PROPOSÉ PAR :
DR. GHOUL KHALID

Édition 1.0

2019/2020

Table des matières

Préambule	iv
Description de la matière	v
1 Introduction	1
1.1 Description du mouvement	1
1.2 Mouvement apparent et mouvement réel	3
1.3 Nature d'objet	5
2 Détection de mouvement	7
2.1 Introduction	7
2.2 Détection basée sur la différence entre images	9
2.3 Test de vraisemblance	15
2.4 Détection basée sur un modèle	18
3 Estimation de mouvement	22
3.1 Introduction	22
3.2 Principe de base	23
3.3 Contraintes	24
3.3.1 Contrainte de conservation de l'intensité	24
3.3.2 Le déplacement et l'intervalle le temps sont petits	25
3.3.3 Cohérence spatiale	26
4 Méthodes d'Estimation de mouvement	28
4.1 Introduction	29
4.2 Méthodes différentielles	29
4.2.1 Horn et Schunck	29
4.2.2 Méthode de Lucas et Kanade	31

4.3	Méthodes de mise en correspondance de blocs	33
4.3.1	Principe général	33
4.3.2	Critère de similarité	34
4.3.3	Les algorithmes de MCB	34
4.4	Méthodes fréquentielles	38
4.4.1	Principe général	38
4.4.2	Méthodes fréquentielles basées sur la phase	39
4.4.3	Méthodes fréquentielles basées sur l'énergie	41
4.5	Méthodes neuronales	43
4.5.1	Méthodes neuronales basée sur les réseaux à convolution	43
4.5.2	Méthodes neuronales basée sur les cartes	45
4.6	Approche multi-échelles	48
	Exercices	52
	Bibliographie	57

Préambule

Le présent document s'agit d'un support de cours concernant la matière intitulée «**Détection et estimation de mouvement**», enseigné au Département de Mathématiques et d'Informatique à l'université d'Oum El Bouaghi et destiné aux étudiants de la deuxième année Master en informatique, option vision artificielle.

Ce support de cours vise à donner aux étudiants une compréhension sur les principes de la détection et l'estimation de mouvement dans les séquences d'images, tout en montrant leurs rôles et leurs fonctionnalités dans les systèmes de vision artificielle. Aussi ce cours vise à former les étudiants dans les traitements des séquences d'images notamment la notion de mouvement :

- Dédire les objets en mouvements,
- Quantifier les déplacements et
- Déterminer l'orientation et le sens du mouvement en chaque pixel de ces objets.

Afin d'atteindre l'objectif tracé, nous avons déployé tous les efforts pour approcher ce travail dans plusieurs aspects et nous avons synthétisé les informations les plus pertinentes en nous appuyant sur des sources variées (ouvrages, notes de cours, articles, sites internet...). Tout en respectant le canevas officiel défini par le ministère de l'enseignement supérieur et de la recherche scientifique. Néanmoins, nous avons conscience que ce document restera partiel, tronqué et non exhaustif. C'est pour cette raison que nous essayons de veiller à une actualisation permanente, dont le but est d'enrichir son contenu. Ainsi, nous serions reconnaissant aux lecteurs de nous signaler toute erreur ou de nous proposer des suggestions dans ce sens.

Description de la matière

Intitulé de la matière : Détection et estimation de mouvement

Volume horaires hebdomadaire : 1h 30 (cours)

Crédits : 6

Coefficients : 3

Contenu

1. Introduction
 1. 1. Description du mouvement
 1. 2. Mouvement apparent et mouvement réel
 1. 3. Nature d'objet
2. Détection de mouvement
 2. 1. Détection basée sur la différence entre images
 2. 2. Test de vraisemblance
 2. 3. Détection basée sur un modèle
3. Estimation de mouvement
 3. 1. Principe de base
 3. 2. Contraintes
4. Méthodes d'estimation de mouvement
 4. 1. Méthodes différentielles
 4. 2. Méthodes de mise en correspondance de blocs
 4. 3. Méthodes fréquentielles
 4. 4. Méthodes neuronales
 4. 5. Approche multi-échelles

Introduction

Sommaire

1.1	Description du mouvement	1
1.2	Mouvement apparent et mouvement réel	3
1.3	Nature d'objet	5

1.1 Description du mouvement

Le mouvement en anglais "motion" est l'un des phénomènes fondamentaux dans la vision naturelle. Les humains et les animaux utilisent le mouvement pour éviter les obstacles, trouver de la nourriture, etc. Le mouvement est aussi important dans de nombreux systèmes de vision artificielle, par exemple : la surveillance, l'interprétation, le suivi, la robotique et la compression vidéo, etc. Le mouvement ou aussi appelé vecteur de mouvement désigne en général, le changement de la position au cours du temps (action physique), c'est-à-dire le déplacement d'un objet par rapport à son ancienne position (voir la Figure 1.1).

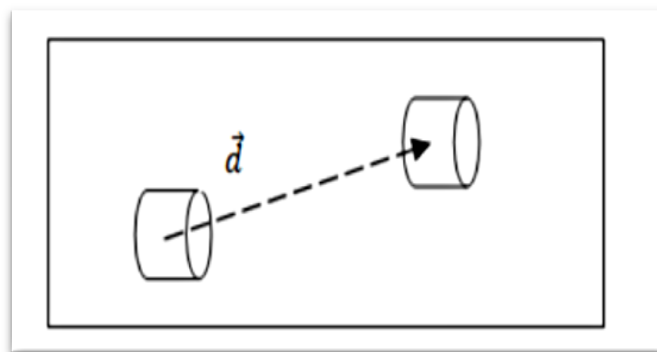


FIGURE 1.1 – Déplacement d'un objet

Le mouvement d'un objet dans la scène est décrit par rapport à une référence (sa position de départ), deux quantités qui caractérisent le mouvement qui sont : la vitesse et la trajectoire.

- La première quantité peut être définie par le quotient de la distance parcourue notée d et qui représente la différence entre les deux positions (départ et finale) par le temps Δt , formellement on peut écrire l'équation 1.1.
- La deuxième quantité est la trajectoire qui correspond aux positions prises à l'objet pendant son déplacement.

$$\vec{v} = \frac{\vec{d}}{\Delta t} \quad (1.1)$$

En réalité, on peut distinguer deux types de mouvements, le mouvement 3D dans l'espace perçu par l'œil humain et le mouvement 2D correspond à la projection du mouvement 3D vers la plan 2D de l'image. On traite le mouvement à partir d'une séquence vidéo ou séquence d'images, dans la séquence vidéo le mouvement est observé par l'affichage d'images successives généralement entre 25 et 30 images par secondes. Une séquence de N images est une série de N images ou aussi appelée 'frame' 2D successives qui représente la scène animée d'objets en mouvement au cours de temps t .

La Figure 1.2 représente la structure d'une séquence d'images, ainsi un exemple d'une séquence nommée 'Taxi de Hambourg'¹ composé de 41 images qui représente la surveillance d'un carrefour par une caméra fixe placée en hauteur.

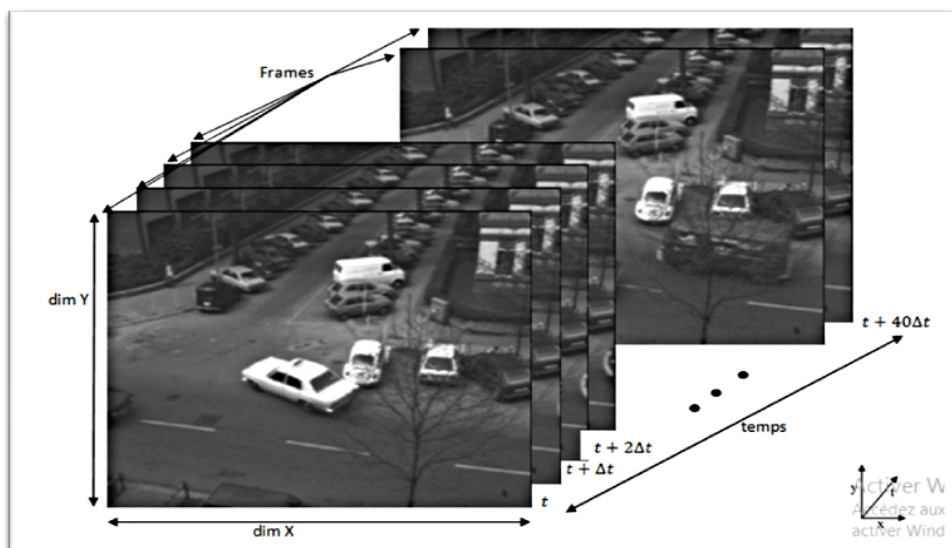


FIGURE 1.2 – La structure d'une séquence d'images

1. Tiré du site : http://i21www.ira.uka.de/image_sequences/#taxi.

En pratique, la définition de mouvement est subjective puisque ce dernier est lié à la variation de brillance dans la séquence, donc le mouvement est observable à cause de changement de la fonction d'intensité, ce qui implique que le mouvement d'un pixel p dans la séquence d'images conduit à un gradient spatio-temporel, comme il est montré dans la Figure 1.3. En général, on a deux mouvements dans la scène, le mouvement des objets et le mouvement de la caméra, dans ce cours on s'intéresse au premier mouvement.

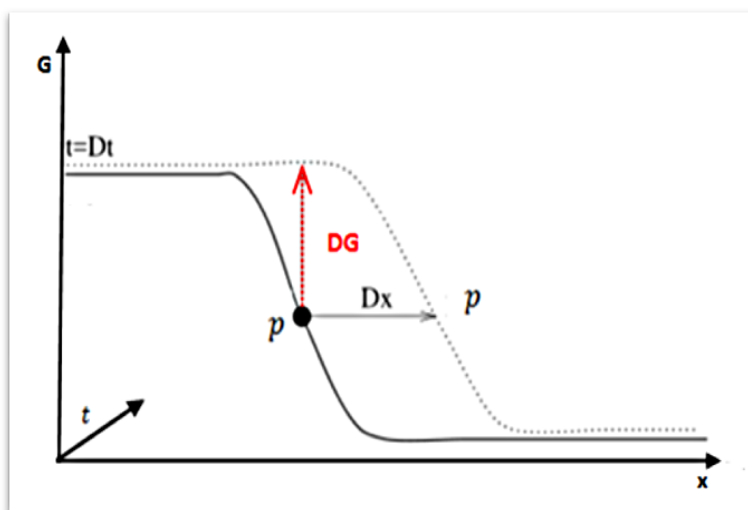


FIGURE 1.3 – Gradient spatio-temporel

1.2 Mouvement apparent et mouvement réel

Les êtres humains ont la capacité de détecter et d'analyser le mouvement dans le monde réel grâce à l'œil, cette dernière permet d'apercevoir et de détecter les objets dans le plan 3D animant une scène réelle. Le mouvement des objets dans le plan 3D n'est observable qu'à travers l'œil ou un système de prise de vue, ce mouvement est dit réel ou encore 3D (tridimensionnel). Donc, pour traiter le problème d'analyse de mouvements dans une séquence d'images, nous devons effectuer la projection du mouvement réel dans un plan 3D de la scène vers un mouvement projeté dans un plan de l'image 2D, le mouvement projeté s'appelle aussi un mouvement apparent (voir Figure 1.4²).

Le mouvement apparent est obtenu par la projection du mouvement des objets dans le plan de la scène 3D sur le plan de l'image 2D, ce mouvement n'est observé que par l'analyse des changements de la fonction de luminances parce que nous avons une seule information qui

2. Tiré de "Video Processing and Communications", Wang et al., Prentice Hall, 2002

est la séquence d'images. Le mouvement apparent est reconnu sous le nom du "flot optique", ce dernier est très utilisé dans le domaine d'estimation de mouvement qui désigne le champ de déplacement des objets dans la séquence d'image.

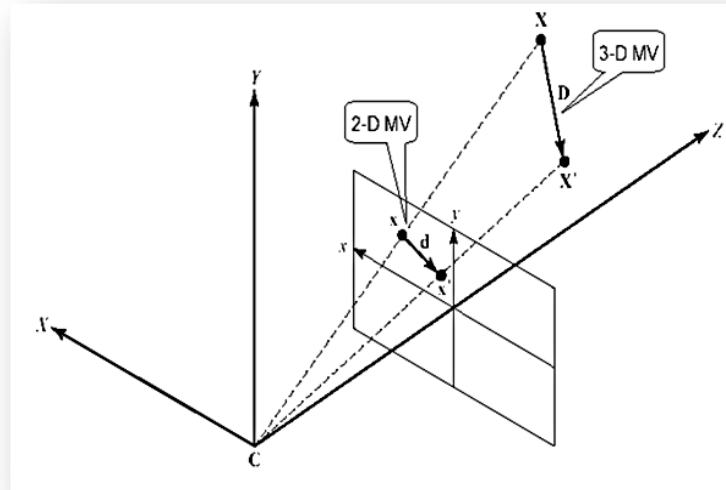


FIGURE 1.4 – Mouvement réel vs mouvement apparent

En général, Le mouvement réel est différent au mouvement apparent parce que :

- Le premier est caractérisé par la troisième dimension (la profondeur), donc nous avons une perte d'une dimension au niveau du mouvement apparent.
- Le mouvement apparent est obtenu via les variations spatiales et temporelles de la fonction de luminance à travers la séquence. Ce mouvement peut être observé dans l'exemple le plus connu, qui représente une sphère homogène en train de tourner sur elle-même. Aucun mouvement ne sera repéré par la projection du mouvement réel dans un plan de l'image 2D, ce qui ne correspond pas à la réalité.
- La séquence des images est affectée par des variations de la luminance et d'éclairage non dues au mouvement des objets dans la scène réelle, comme par exemple le changement de l'orientation du soleil (Voir figure 1.5).

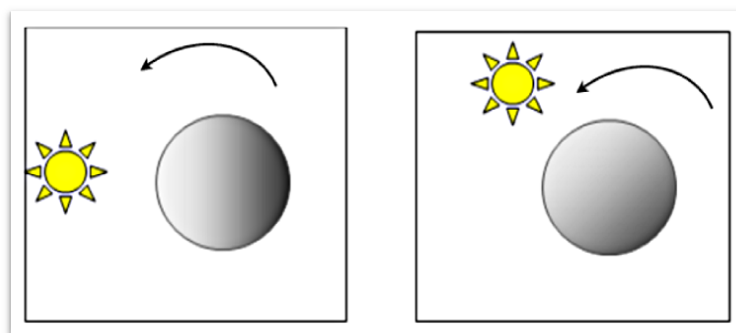


FIGURE 1.5 – Problème de changements d'illumination

1.3 Nature d'objet

Dans le domaine de la vision par ordinateur, la notion d'objet constitue un élément clé, puisque notre étude se concentre toujours sur la notion d'objet. L'objet désigne une zone de l'image qui est caractérisé par sa texture, sa forme, sa couleur, son orientation du gradient ou encore son mouvement et qui représente, en général une entité du monde réel 3D. D'une autre manière on peut dire que les objets dans les séquences d'images représentent les pixels qui appartiennent à l'avant plan ou le fond de la scène.

On distingue les objets réels 3D et les objets apparents qui représentent la projection des objets réels dans le plan 2D de l'image et qui ne sont pas les même dans le cas d'une occultation (voir la Figure 1.6³). On trouve aussi les objets rigides et non rigides dits déformables qui peuvent être différencié par la variation de leurs formes au cours du temps. Les objets rigide gardent leurs forme au cours du temps (par exemple : les véhicules) et les objets non rigide changent de forme d'une image à une autre (par exemple : nuages, cœur, pluie, etc.).

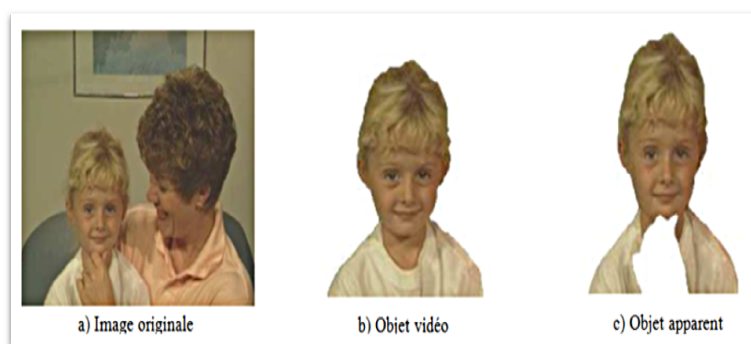


FIGURE 1.6 – Objet Vidéo et objet apparent "séquence de Mother et Daugther"

3. <http://trace.kom.aau.dk/mirrors/h261/1628.html>.

Dans ce support de cours on s'intéresse essentiellement au mouvement des objets dans la scène, pour cela on regroupe les objets selon leurs mouvements en deux grandes catégories, les objets mobiles et immobiles. Les objets mobiles (en mouvements) sont des objets qui se déplacent d'une image à une autre, ils sont caractérisés par leurs mouvements dans la séquence d'images. Selon leurs mouvements nous pouvons distinguer :

- Les objets mobile rapides c'est-à-dire leurs déplacements est important.
- les objets lents sont les objets qui se déplacent lentement dans la scène.
- Les objets qui ont un mouvement simples, sont des objets qui ont le même mouvement (en vitesse et direction) dans la séquence d'images.
- Les objets qui ont un mouvement complexe, sont des objets qui ont un mouvement aléatoire dans la séquence d'images.

Les objets immobiles (stationnaires) qui gardent la même position d'une image à une autre tel que les arrières plans c'est-à-dire sont des objets qui ne se déplacent pas au cours du temps dans la scène.



FIGURE 1.7 – Les objets mobiles et fixes

La Figure 1.7 présente la séquence Taxi de Hambourg (de taille 190×256 pixels) qui représente la surveillance d'un carrefour par une caméra fixe placée en hauteur, cette séquence contient 4 objets mobiles (désignés par cercle rouge) et d'autres objets immobiles (désignés par cercle jaune) : un taxi arrivant de la droite et tournant vers sa droite, un vent arrivant de la droite et se dirige vers la gauche, une voiture arrivant de la gauche et se dirige vers la droite et un piéton se déplace à partir de haut vers la gauche.

Détection de mouvement

Sommaire

2.1 Introduction	7
2.2 Détection basée sur la différence entre images	9
2.3 Test de vraisemblance	15
2.4 Détection basée sur un modèle	18

2.1 Introduction

La détection de mouvement dans les séquences d'images (vidéo) a comme objectif de trouver les points (pixels) qui changent de position au cours du temps dans la séquence, c'est à dire consiste d'associer à chaque pixel une étiquette binaire pour modéliser le changement (valeur 1) ou le non changement de position (valeur 0). Le résultat final de la détection de mouvement est donc une carte binaire qui contient des valeurs 1 et 0, ces dernières peuvent être utilisées pour identifier la présence ou l'absence de mouvement dont le but est de distinguer les objets mobiles et statiques dans la scène (voir la Figure 2.1).

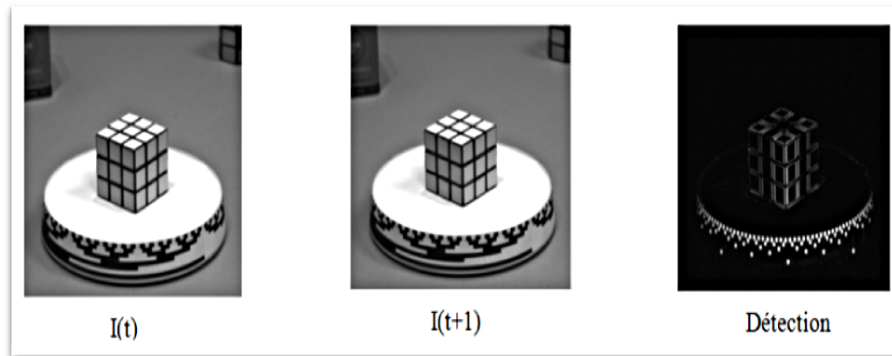


FIGURE 2.1 – Exemple de la détection de mouvement avec la séquence «Cube de Rubik»

La Figure 2.1 représente la séquence cube de Rubik (de taille 240×256 pixels) qui est constituée d'un cube placé au centre d'un plateau circulaire et qui tourne sur lui-même avec une vitesse constante, donc les zones avec couleur blanche représentent le mouvement, ce dernier est dû au déplacement des objets dans la scène. Le processus de détection de mouvement n'est pas toujours possible à cause de plusieurs obstacles, nous pouvons citer :

- La variation de la luminosité et les conditions d'éclairage qui ont des influences sur l'apparence des objets composant la scène.
- Bruit dans la séquence d'images.
- Variabilité de la forme et de la structure des objets de la scène.
- Les zones homogènes où nous ne pouvons pas détecter le mouvement bien que ce dernier existe.

Par conséquent, les méthodes de détection de mouvement sont généralement basées sur la contrainte suivante : l'intensité et l'éclairage de la scène doivent être constants (ou avec de petites variations) au cours du temps. Une autre contrainte que nous pouvons ajouter est que le capteur soit immobile, cette contrainte est utile puisque le processus de détection de mouvement dans les séquences d'images est la première étape dans plusieurs domaines d'application, par exemple :

- La segmentation basée sur l'étude de mouvement
- L'interprétation des séquences d'image pour pouvoir donner des décisions à partir des images.
- Le suivi d'objet.
- L'analyse de comportement.
- La compression et la prédiction vidéo

- La vidéo-surveillance
- La robotique pour éviter les collisions

2.2 Détection basée sur la différence entre images

La méthode intuitive pour la détection de mouvement dans une séquence d'images est de détecter la différence pixel par pixel c'est-à-dire mesurer le changement temporel des pixels dans la séquence, ce qui correspond au calcul du gradient temporel en chaque pixel. Aussi, si dans la séquence on a un mouvement nul, il n'y aura pas de variation entre les images correspondantes ; si par contre dans un mouvement, un changement évident se produira entre les images correspondantes.

Parmi les méthodes de détection de mouvement basées sur la différence entre les images on trouve la méthode de différence ou de soustraction entre deux images successives. Dans une séquence d'images au niveaux de gris, et pour chaque pixel de coordonnées (x, y) dans l'image ou frame I_{t-1} qui correspond à l'image à l'instant $t - 1$, on calcule la différence absolue avec ses coordonnées correspondantes dans l'image suivante (acquise à l'instant t) notée I_t comme suit :

$$\zeta(x, y) = |I_t(x, y) - I_{t-1}(x, y)| \quad (2.1)$$

Pour une image couleur RGB par exemple, on peut calculer cette différence absolue par la distance Euclidienne comme suit :

$$\zeta(x, y) = \sqrt{(I_t^R(x, y) - I_{t-1}^R(x, y))^2 + (I_t^G(x, y) - I_{t-1}^G(x, y))^2 + (I_t^B(x, y) - I_{t-1}^B(x, y))^2} \quad (2.2)$$

Où $I_t^R(x, y)$, $I_t^G(x, y)$ et $I_t^B(x, y)$ représentent l'intensité du pixel de coordonné (x, y) dans les champs R, G et B respectivement du système de couleurs RGB.

En présence de mouvement entre les deux images alors $\zeta \neq 0$ et dans le cas contraire c'est-à-dire $\zeta = 0$ on a un mouvement nul. Puisque les images présentent en générale un bruit lié par exemple au capteur dans les systèmes de surveillance ou d'interprétation (dans la phase d'acquisition), une opération de seuillage est indispensable. Le seuillage permet d'une part d'avoir une information pertinente et d'autre part de produire comme résultat une image binaire qui indiquent les régions de mouvement (seules correspondantes aux variations de l'intensité

lumineuse dans les deux images), formellement on peut écrire :

$$\Psi(\zeta(x, y)) = \begin{cases} 1 & \text{si } \zeta(x, y) > \tau, \\ 0 & \text{sinon} \end{cases} \quad (2.3)$$

où τ est le seuil, qui décide de la sensibilité de la détection de mouvement.

La méthode de détection de mouvement par la différence entre deux images successives est caractérisée par sa simplicité, sa facilité de mettre en œuvre, sa flexibilité, son adaptation au changement de la scène, ainsi cette méthode demande peu de ressources. Comme inconvénient on peut citer la sensibilité au bruit et ne traite pas le problème des zones homogènes.

Exemple : La figure 2.2 montre un exemple de la méthode de différence entre deux images successive :

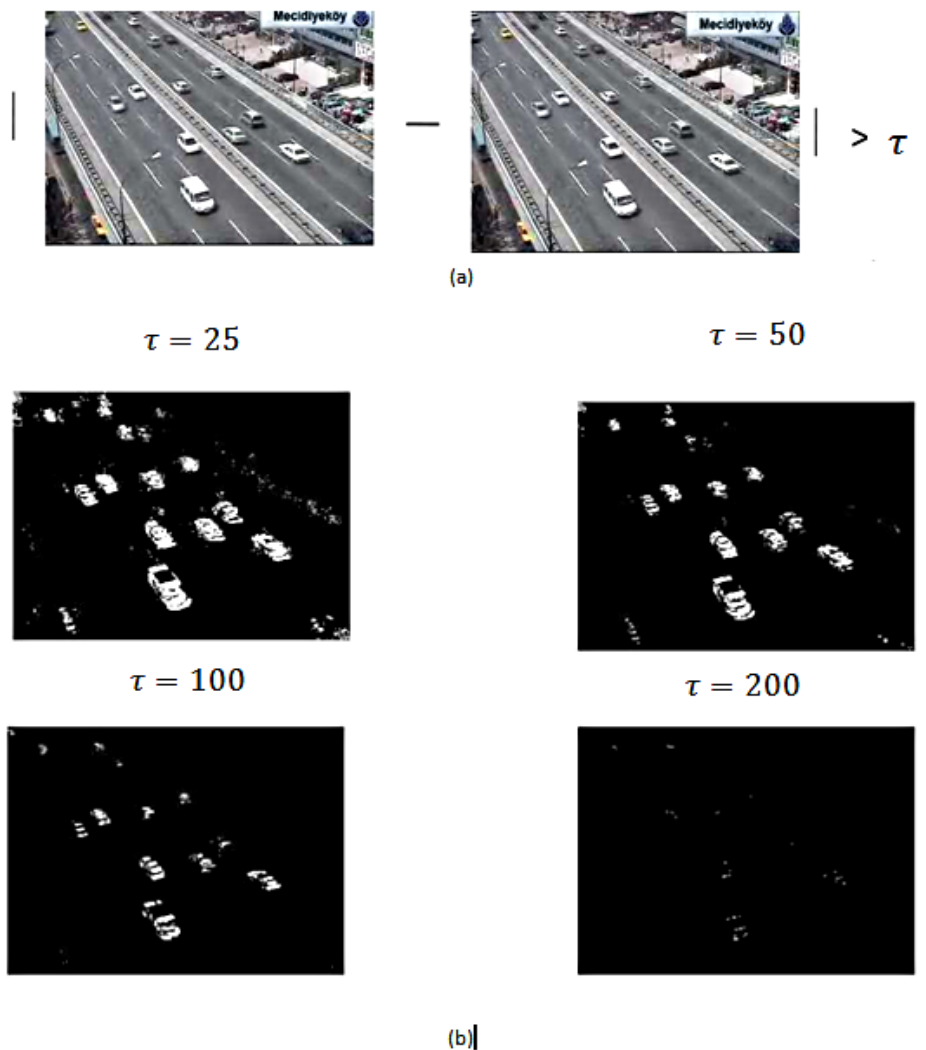


FIGURE 2.2 – Exemple :(a) principe de la détection, (b) résultats obtenues

Pour l'amélioration de la méthode précédente qui utilise une seule opération de différence entre les deux trames, une alternative de cette méthode est proposée, la méthode proposée utilise deux opérations de différences entre trois trames pour traiter le problème de faible mouvement et aussi le problème de bruits. La méthode de différence entre trois trames successives noté $I_{(t-1)}$, I_t et $I_{(t+1)}$ consiste à mesurer la différence entre les trois trames selon les équations 2.4 et 2.5 ensuite une opération de seuillage des résultats obtenus selon les équations 2.6 et 2.7 et enfin combiner les résultats obtenus précédemment selon l'équation 2.8 comme suit :

$$\zeta_1(x, y) = |I_t(x, y) - I_{t-1}(x, y)| \quad (2.4)$$

$$\zeta_2(x, y) = |I_{t+1}(x, y) - I_t(x, y)| \quad (2.5)$$

$$\Psi(\zeta_1(x, y)) = \begin{cases} 1 & \text{si } \zeta(x, y) > \tau_1, \\ 0 & \text{sinon} \end{cases} \quad (2.6)$$

$$\Psi(\zeta_2(x, y)) = \begin{cases} 1 & \text{si } \zeta(x, y) > \tau_2, \\ 0 & \text{sinon} \end{cases} \quad (2.7)$$

$$\zeta(x, y) = \min(\Psi(\zeta_1(x, y)), \Psi(\zeta_2(x, y))) \quad (2.8)$$

Exemple :

Dans la Figure 2.3 illustre un exemple de la méthode de différence avec 3 trames consécutives avec la séquence cube de Rubik.

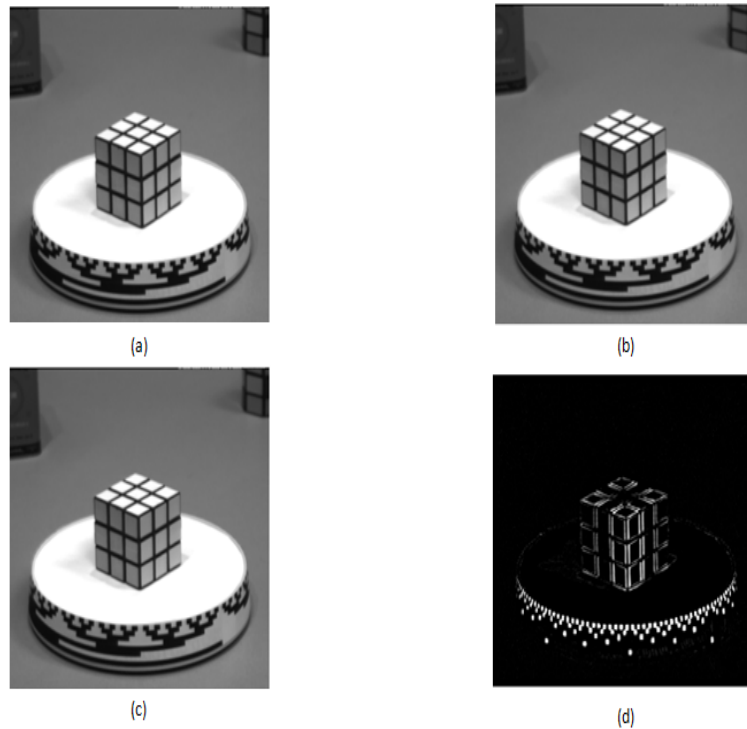


FIGURE 2.3 – Exemple :(a) 1ère image, (b) 2ème image,(c) 3ème image,(d) résultat

Pour surmonter le problème des zones homogènes qui imposent que les images soient bien texturées, une autre méthode qui est basée sur la différence entre les trames appelée la détection de mouvement avec image de référence est proposée. La méthode proposée requière l'utilisation d'une image de référence notée I_R , cette image doit représenter l'arrière plan sans aucun objets mobile c'est-à-dire seuls les objets immobiles sont présents dans cette image, aussi l'image de référence doit prendre en compte le changement de l'arrière plan dans chaque image de la séquence, donc l'image de référence doit être régénérée à chaque Δt (ce qui correspond au temps d'acquisition pour chaque image de la séquence). Donc, la qualité des résultats obtenus est liée avec la qualité de l'image de référence. On distingue deux techniques pour la construction de l'image de référence :

- Soit on sélectionne nous même l'image de référence, on prend par exemple :la première image dans la séquence, ou l'image qui contient peu/aucun mouvement.
- Soit on construit l'image de référence comme suit :

$$I_R = \sum_{i=t-N}^{t-1} w_i I_t \quad (2.9)$$

avec : $w_i = \frac{1}{N}$ où : I_R est l'image de référence

I_t : L'image a l'instant t

N : Le nombre de frame dans ma séquence

w_i Représente le facteur de pondération.

La détection de mouvement avec image de référence consiste à mesurer la différence entre l'image de référence et l'image courante selon l'équation 2.10 et enfin une opération de seuillage des résultats obtenus précédemment pour réduire le bruit :

$$\zeta(x, y) = |I_t(x, y) - I_R(x, y)| \quad (2.10)$$

Cette méthode est moins rapide par rapport aux méthodes vu précédemment puisqu'elle nécessite la reconstruction de l'image de référence à chaque Δt , aussi elle est sensible aux variations considérables de luminosité au cours du temps surtout pour les séquences qui sont composée de nombre important des images, c'est-à-dire que l'image de référence doit avoir les même variations de luminance qui existe dans la séquence d'images.

Exemple :

Un exemple de la méthode de détection avec image de référence est illustré sur la Figure 2.4 :



FIGURE 2.4 – Exemple :(a) image de la séquence, (b) image de référence,(c) résultat(Seuil absolu= 25)

Une autre technique a été proposée par A. Bello ¹, qui utilise à la fois les avantages des deux méthodes décrites précédemment (la soustraction des trois images successives et la méthode de détection avec image de référence) est définie comme suit :

1. Aldo. Bellon. Détection et suivi de véhicules en mouvement dans une séquence d'images. PhD thesis, Université Blaise Pascal, Clermond-Ferrand, Octobre 1996.

- Calculer le détecteur à court terme entre les images $I_{(t-1)}$, I_t et $I_{(t+1)}$ qui permet d'avoir la différence entre les trois trames comme suit :

$$\zeta^1(x, y) = \min(\zeta_1(x, y), \zeta_2(x, y)) \quad (2.11)$$

- Calculer le détecteur à long terme entre l'image de référence I_R et l'image I_t selon l'équation 2.12

$$\zeta^2(x, y) = |I_t(x, y) - I_R(x, y)| \quad (2.12)$$

- Calculer la différence d'image robuste entre les résultats des étapes précédentes comme suit :

$$\zeta^1(x, y) = \max(\Psi(\zeta^1(x, y)), \Psi(\zeta^2(x, y))) \quad (2.13)$$

Exemple :

Un exemple de la méthode de détection de A. Bello est illustré sur la Figure 2.5 avec la même séquence (routière) que l'exemple précédant :



FIGURE 2.5 – Exemple :(a) image de la séquence, (b) résultat obtenu

Les méthodes de détection de mouvement basées sur la différence d'image souffrent du problème de bruit dans la séquence d'images malgré tous les améliorations reconnues, cela provoque des coupures irrégulier dans les zones de mouvement détectées et donc des trous dans des objets mobiles. Ce problème peut être surmonté par une opération de lissage (en utilisant un filtre spatial Gaussien par exemple) avant le seuillage. Aussi, Ce mécanisme de détection de mouvement est limité généralement dans les régions qui ne présentent aucune variation de texture (ou on a un faible gradient d'image).

2.3 Test de vraisemblance

Parmi les méthodes de détection de mouvement on trouve les méthodes qui exploitent le contenu stochastique de la séquence d'image, ces méthodes sont connues sous le nom des méthodes probabiliste. Dans cette section nous présentons la détection de mouvement au sens du maximum de la vraisemblance [Hsu et al. 1984] ou la fonction(test) de vraisemblance c'est-à-dire détection par test des hypothèses statistiques.

Le test de vraisemblance est connu comme un test des hypothèses qui est utilisé pour comparer deux modèles en établissant une hypothèse dite hypothèse nulle, et en utilisant certaines données pour rejeter l'hypothèse dans le cas où la vraisemblance sous l'hypothèse alternative est significativement supérieure à la vraisemblance sous l'hypothèse nulle et réciproquement. Dans notre domaine de la détection de mouvement, on applique le test de vraisemblance comme suit[Hsu et al. 1984] :

Considérons I_t et $I_{(t+dt)}$ les deux trames que l'on cherche à détecter le mouvement entre eux, ces deux trames sont découpée en blocs ou zones notées A_1 , A_2 respectivement centré sur le pixel p de coordonné (x, y) et soit $g_i^1 (i = 1, \dots, m)$, $g_j^2 (j = 1, \dots, n)$ des valeurs observées dans les deux zones A_1 et A_2 qui correspondent à des valeurs de niveaux de gris corrompues par un bruit Gaussien de moyenne 0 et de variance σ^2 noté $N(0, \sigma)$ (voir la Figure 2.6).

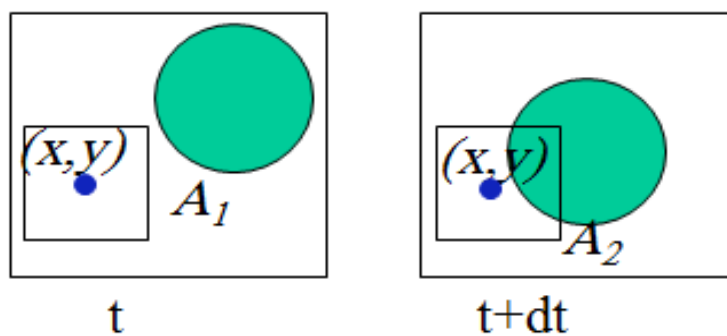


FIGURE 2.6 – Détection par max de vraisemblance

La distribution de probabilité conjointe pour l'observation de ces mesures dans le cas de n valeurs d'échantillon est donnée par la fonction de vraisemblance suivante :

$$L = f(g, \mu, \sigma) = \left(\frac{1}{2\pi\sigma^2} \right)^{\frac{n}{2}} \exp \left(-\frac{1}{2\sigma^2} \sum_{i=1}^n (g_i - \mu)^2 \right) \quad (2.14)$$

où μ représente la moyenne de la distribution des valeurs d'échantillon.

Les hypothèses à tester sont notées comme suit :

- H_0 : A_1 et A_2 proviennent de la même distribution $N(\mu_0, \sigma_0) \implies$ pas de mouvement(zone stationnaire)
- H_1 : A_1 et A_2 proviennent de différentes distributions $N(\mu_1, \sigma)$ et $N(\mu_2, \sigma) \implies$ le mouvement existe(zone mobile)

Si H_1 vraie, la fonction de vraisemblance sera (Loi de Gauss) :

$$L(H_1) = \left(\frac{1}{2\pi\sigma^2} \right)^{\left(\frac{m+n}{2}\right)} \exp \left[-\frac{1}{2\sigma^2} \left(\sum_{i=1}^m (g_i^1 - \mu_1)^2 + \sum_{i=1}^n (g_i^2 - \mu_2)^2 \right) \right] \quad (2.15)$$

Et les dérivées partielles par rapport à σ^2, μ_1, μ_2 seront comme suit :

$$\begin{cases} \hat{\mu}_1 = \frac{1}{m} \sum_{i=1}^m g_i^1, \\ \hat{\mu}_2 = \frac{1}{n} \sum_{i=1}^n g_i^2, \\ \hat{\sigma}^2 = \hat{\sigma}_1^2 = \hat{\sigma}_2^2 = \left(\frac{1}{m+n}\right) \left[\sum_{i=1}^m (g_i^1 - \hat{\mu}_1)^2 + \sum_{i=1}^n (g_i^2 - \hat{\mu}_2)^2 \right]. \end{cases} \quad (2.16)$$

Si H_0 vraie, la fonction de vraisemblance sera $L(H_0)$:

$$\begin{cases} \hat{\mu}_0 = \frac{1}{m+n} \left[\sum_{i=1}^m g_i^1 + \sum_{i=1}^n g_i^2 \right], \\ \hat{\sigma}_0^2 = \left(\frac{1}{m+n}\right) \left[\sum_{i=1}^m (g_i^1 - \hat{\mu}_0)^2 + \sum_{i=1}^n (g_i^2 - \hat{\mu}_0)^2 \right]. \end{cases} \quad (2.17)$$

Alors, le rapport de vraisemblance maximum peut être écrit sous la forme suivant :

$$\lambda = \frac{L(H_1)}{L(H_0)} \quad (2.18)$$

Donc on choisi H_1 si $\lambda > 1$ et H_0 dans le cas contraire, formellement le critère de décision s'écrit comme suit :

$$\begin{cases} H_1 & \text{si } \lambda > 1 \\ H_0 & \text{si } \lambda \leq 1 \end{cases} \quad (2.19)$$

Pratiquement le rapport de vraisemblance maximum peut être écrire sous la forme suivant

[Hsu et al. 1984] :

$$\lambda = \left[1 + \frac{t^2}{m+n-2} \right]^{\left(\frac{m+n}{2}\right)} \quad (2.20)$$

et nous pouvons tester t^2 au lieu de λ puisque cette dernière est une fonction monotone de t^2 , donc on sélectionne un seuil t_α , en fonction d'un niveau de confiance α

Avec :

$$t = \frac{\left(\frac{mn}{m+n}\right)^{\frac{1}{2}} (\hat{\mu}_1 - \hat{\mu}_2)}{\left[\frac{\sum_{i=1}^m (g_i^1 - \hat{\mu}_1)^2}{m+n-2} + \frac{\sum_{j=1}^n (g_j^2 - \hat{\mu}_2)^2}{m+n-2} \right]^{\frac{1}{2}}} \quad (2.21)$$

Cette méthode de Hsu [Hsu et al. 1984] et qui utilise le test de vraisemblance ne permet pas de prendre en compte une variation d'illumination, pour cela des améliorations ont été proposées, alors :

- Pour un changement d'illumination global, une transformation linéaire est utilisée pour que la moyenne (μ) et la variance (σ) de l'image normalisée soient respectivement égales à la moyenne et à la variance de l'image d'origine comme suit :

$$\hat{I}_t(g) = aI_t(g) + b \quad (2.22)$$

tel que :

$$\hat{\mu}_t = \mu_{t+dt}$$

et

$$\hat{\sigma}_t^2 = \sigma_{t+dt}^2$$

alors :

$$\hat{I}_t(g) = \frac{\sigma_{t+dt}}{\sigma_t} (I_t(g) - \mu_t) + \mu_{t+dt} \quad (2.23)$$

- Pour des changements d'illumination locaux, on fait la comparaison des gradients d'intensité, comme suit

$$\frac{1}{N} \sum_{g \in A} \|\nabla I_t(g) - \nabla I_{t+dt}(g)\| \stackrel{\geq}{\leq} \theta \quad (2.24)$$

Où θ représente la seuil.

Exemple² :

Si $m = n = 12$, pour un niveau de confiance $\alpha = 5\%$, puis $t_{0.05} = 1.717$ et $t_{0.05}^2 = 2.948$; pour niveau de confiance $\alpha = 1\%$, on obtient $t_{0.01} = 2.508$ et $t_{0.01}^2 = 6.29$. Lorsque la valeur de t^2 est inférieure à t_α^2 on choisit l'hypothèse H_0 . Sinon, H_1 sera accepté.

2.4 Détection basée sur un modèle

Les méthodes de détection de mouvement basées sur la différence entre les images souffrent de plusieurs problèmes, on trouve trop de pixels en mouvement si le seuil est bas plus la complexité des algorithmes, aussi le bruit de capteur (acquisition+numérisation) dans les applications en temps réel. Alors la solution est d'utiliser les méthodes de relaxation par modèle (ou champs) de Markov (MRF : Markov Random Fields). A cet effet, nous présentons dans cette section la méthode proposée par Bouthemy et Lalande [Bouthemy and Lalande 1993], cet algorithme vise à améliorer la différence d'image par l'utilisation d'un processus markovien.

Pour l'approche Markovienne, la solution du problème de la détection de mouvement est la réalisation la plus probable d'un certain phénomène aléatoire, l'image est considérée comme un ensemble de site noté S où chaque site note s_i représente un pixel de l'image et à chaque site est associé un descripteur qui peut être son niveau de gris ou son étiquette, donc un champ aléatoire X sur S à valeur dans V est défini par :

$$X : \Omega \rightarrow E^S$$

Où :

Ω : Univers probabiliste

E^S : ensemble des valeurs de sites.

Dans notre contexte de la détection de mouvement on a :

$$\left\{ \begin{array}{ll} E = \{0, 1\} & \text{(fixe, mobile)} \\ S = Z^3 & \text{espace-temps discret} \end{array} \right.$$

L'ensemble des sites S est muni d'un système de voisinage noté v qui détermine les relations de dépendances entre les variables aléatoires X_s et dans notre cas de la détection il modélise les relations spatio-temporelles entre les divers sites de la séquence d'images, formellement on a :

2. Hsu Y. Z, NAGEL H.- H., et REKERS G., "New Likelihood Test Methods for Change Detection in Image Sequences" . COMPUTER VISION, GRAPHICS, AND IMAGE PROCESSING 26, 73 – 106(1984).

$$v : S \rightarrow P(S)$$

$$\forall (s, r) \in S \begin{cases} s \notin v(s) \\ s \in v(r) \implies r \in v(s) \end{cases} \quad (2.25)$$

A partir d'un système de voisinage v , un système de cliques est défini. Une clique est soit un singleton de S , soit un ensemble de sites de tous voisins les uns des autres. Une clique est dite d'ordre K , si elle contient K éléments. on peut distinguer trois types de cliques : clique spatiale, clique temporelle et clique spatio-temporelle (voir la Figure 2.7).

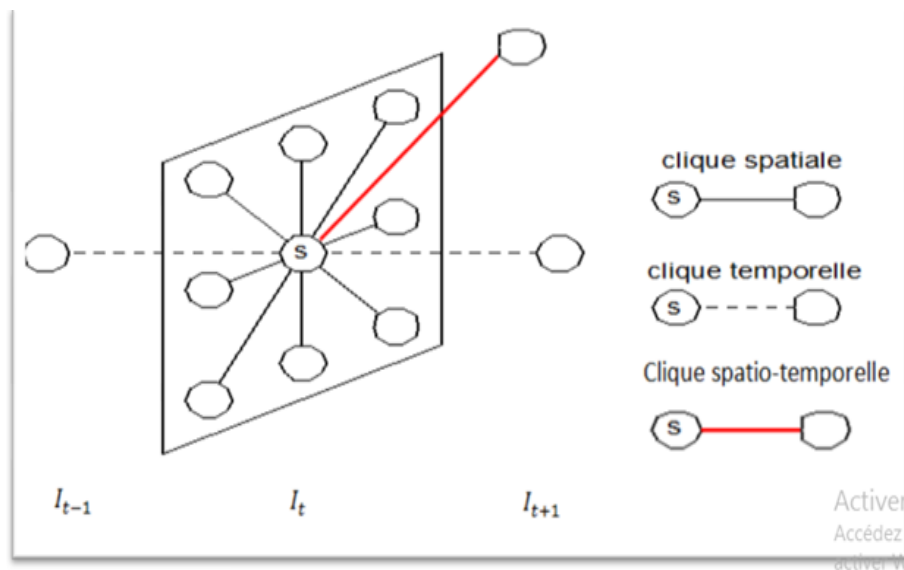


FIGURE 2.7 – système de cliques

L'image est modélisée de façon probabiliste comme une réalisation d'un champ aléatoire, alors selon le modèle markovien « X est un champ de Markov si et seulement si la probabilité conditionnelle locale en un site n'est fonction que de la configuration du voisinage du site considéré ». En d'autre terme, on recherche la réalisation la plus probable d'un champ de Markov dit « caché » X , à valeur binaire sur Z^3 : les étiquètes, à partir d'un champ connu dit « observation » O , qui correspond à la séquence des différences d'images (équation 2.26).

$$O_t(x, y) = | I_t(x, y) - I_{t-1}(x, y) | \quad (2.26)$$

L'estimateur Bayésien du Maximum A Posteriori est utilisé par les auteurs pour estimer les étiquettes appropriées du champ E menu du champ d'observation O qui consiste a la maximisation de la probabilité conditionnelle comme suit :

$$\max \frac{P[O = o | E = e]P[E = e]}{P[O = o]} \quad (2.27)$$

La maximisation du numérateur de l'équation 2.27 est équivalente à la minimisation d'une fonction d'énergie dérivée du théorème de Hammersley-Clifford, qui stipule que les champs aléatoires de Markov suivent la distribution, ce qui nous a permis de décrire un modèle de champ de Markov dans une topologie donnée en spécifiant les potentiels attachés à chaque clique comme suit :

$$P[O = o | E = e] = \frac{e^{-U(o,e)}}{Z} \quad (2.28)$$

où Z est un facteur de normalisation. La fonction énergétique U est donnée par la somme de deux termes comme suit :

$$U(e, o) = U_m(e) + U_a(o, e) \quad (2.29)$$

Avec U_m représente l'énergie qui assure l'homogénéité spatio-temporelle et U_a désigne l'énergie d'adéquation qui assure une bonne cohérence de la solution par rapport aux données observées :

$$U_a(o, e) = \frac{1}{2\sigma^2}[o - \psi(e)]^2 \quad (2.30)$$

avec :

$$\psi(e) = \begin{cases} 0 & \text{si } |I_t(x, y) - I_{t-1}(x, y)| < \tau \\ \alpha & \text{sinon} \end{cases}$$

et

$$U_m = \sum_{c \in C} V_c(e_s, e_r)$$

De plus, V_c est donné selon le modèle de Potts [Potts 1952] qui permet de prendre en compte différentes relations entre différentes valeurs des descripteurs : étiquettes.

$$V_c(e_s, e_r) = V_s(e_s, e_r) + V_p(e_s^t, e_s^{t-1}) + V_p(e_s^t, e_s^{t+1}) \quad (2.31)$$

$$\text{avec : } V_s(e_s, e_r) = \begin{cases} -\beta_s & \text{si } e_s = e_r \\ +\beta_s & \text{sinon} \end{cases}, V_p(e_s^t, e_s^{t-1}) = \begin{cases} -\beta_s & \text{si } e_s^t = e_s^{t-1} \\ +\beta_s & \text{sinon} \end{cases} \text{ et}$$

$$V_p(e_s^t, e_s^{t+1}) = \begin{cases} -\beta_s & \text{si } e_s^t = e_s^{t+1} \\ +\beta_s & \text{sinon} \end{cases}$$

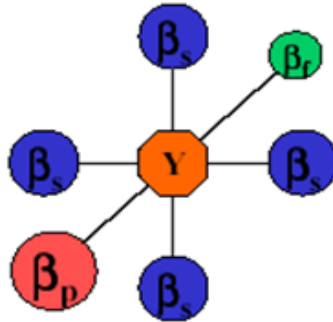


FIGURE 2.8 – Modèle de Potts

Lorsque β est positif, les configurations les plus probables correspondent à des sites voisins de même niveau de gris ou descripteur.

Après avoir défini l'énergie U pour le modèle markovien, les auteurs ont considéré le problème de la minimisation de cette énergie.

Exemple : la Figure 2.9 représente un exemple de la méthode décrite ci-dessus avec deux séquences d'images (Sofa et Canoe) et les paramètres suivants : $\beta_s = 20, \beta_p = 10, \beta_f = 30$ et $\alpha = 10$.

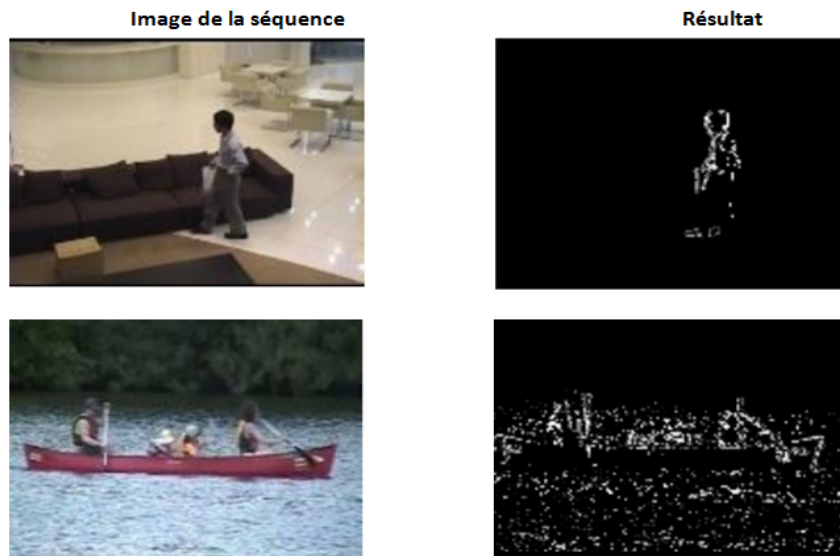


FIGURE 2.9 – Exemple avec de la méthode de Bouthemy et Lalande

Estimation de mouvement

Sommaire

3.1	Introduction	22
3.2	Principe de base	23
3.3	Contraintes	24
3.3.1	Contrainte de conservation de l'intensité	24
3.3.2	Le déplacement et l'intervalle le temps sont petits	25
3.3.3	Cohérence spatiale	26

3.1 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 3.1).

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.

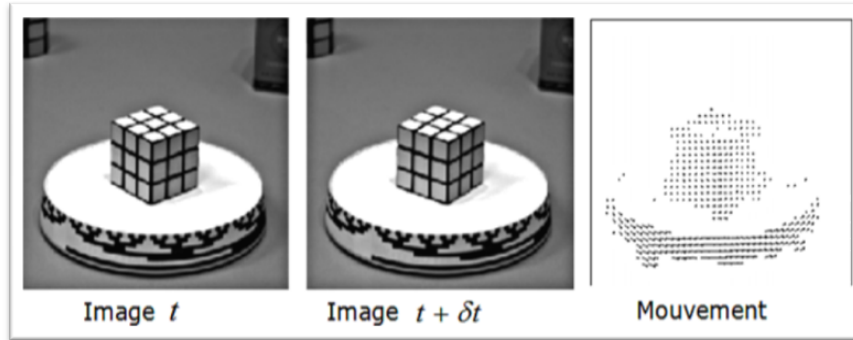


FIGURE 3.1 – Exemple de l'estimation de mouvement avec la séquence «Cube de Rubik»

3.2 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 appelle «Flot optique» à un point au temps t est défini comme la vitesse de déplacement de ce point suivant l'équation 3.1 :

$$\vec{v}(v_x, v_y) = \left(\frac{dx}{dt}, \frac{dy}{dt} \right) \quad (3.1)$$

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 3.1 comme suit :

$$\frac{d}{dt}I(x, y, t) = \frac{\partial I}{\partial t} + v(v_x, v_y) \cdot \nabla I \quad (3.2)$$

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'image

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.

3.3 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 être prises en considération :

3.3.1 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 3.2). Cette contrainte peut être exprimée par l'équation suivante :

$$\frac{d}{dt}I(x, y, t) = 0. \quad (3.3)$$

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

$$I(x + v_x, y + v_y, t + \delta t) = I(x, y, t) \quad (3.4)$$

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

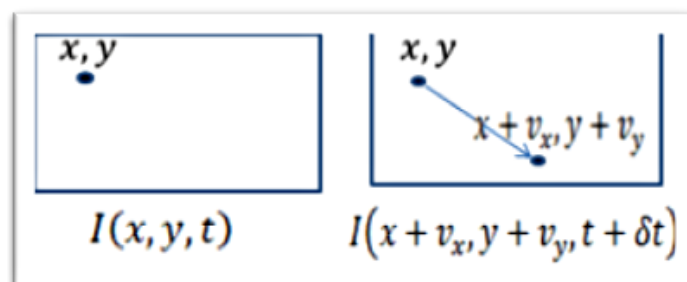


FIGURE 3.2 – L'hypothèse de conservation d'intensités

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

$$I(x + v_x, y + v_y, t + \delta t) = I(x, y, t) + v_x \frac{\partial I}{\partial x} + v_y \frac{\partial I}{\partial y} + \frac{\partial I}{\partial t} + \varepsilon \quad (3.5)$$

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

$$I(x + v_x, y + v_y, t + \delta t) \approx I(x, y, t) + v_x \frac{\partial I}{\partial x} + v_y \frac{\partial I}{\partial y} + \frac{\partial I}{\partial t} \quad (3.6)$$

Et donc :

$$\frac{I(x + v_x, y + v_y, t + \delta t) - I(x, y, t)}{\delta t} = \frac{\partial I}{\partial x} v_x + \frac{\partial I}{\partial y} v_y + \frac{\partial I}{\partial t} \quad (3.7)$$

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 .

3.3.2 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 $\delta t \rightarrow 0$, on a donc :

$$I_x \frac{\partial x}{\partial t} + I_y \frac{\partial y}{\partial t} + I_t = 0 \quad (3.8)$$

Suivant l'équation 3.1 on a : $v_x = \frac{\partial x}{\partial t}$ et $v_y = \frac{\partial y}{\partial t}$, alors on peut écrire :

$$I_x v_x + I_y v_y = -I_t \quad (3.9)$$

Cette dernière équation(3.9) 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 :

$$\vec{V} \cdot \vec{\nabla} I = -I_t \quad (3.10)$$

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 définit 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 (3.10) pour résoudre un système à deux inconnues c'est-à-dire que seule la composante normale de la vitesse peut être mesurée comme suit :

$$V_n(x, y, t) = -\frac{\partial I}{\partial x}(x, y, t) \cdot \frac{\vec{\nabla} I(x, y, t)}{\|\vec{\nabla} I(x, y, t)\|^2} \quad (3.11)$$

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 3.3)

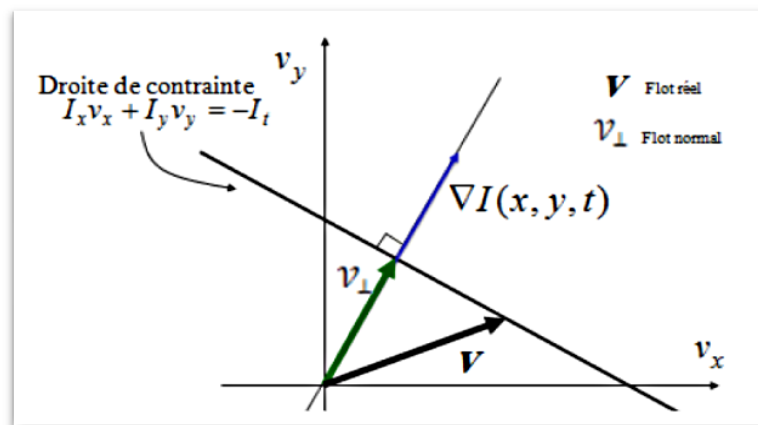


FIGURE 3.3 – Problème d'ouverture

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.

3.3.3 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

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

cours d'étude ont un déplacement proche (direction, la vitesse).

Méthodes d'Estimation de mouvement

Sommaire

4.1	Introduction	29
4.2	Méthodes différentielles	29
4.2.1	Horn et Schunck	29
4.2.2	Méthode de Lucas et Kanade	31
4.3	Méthodes de mise en correspondance de blocs	33
4.3.1	Principe général	33
4.3.2	Critère de similarité	34
4.3.3	Les algorithmes de MCB	34
4.4	Méthodes fréquentielles	38
4.4.1	Principe général	38
4.4.2	Méthodes fréquentielles basées sur la phase	39
4.4.3	Méthodes fréquentielles basées sur l'énergie	41
4.5	Méthodes neuronales	43
4.5.1	Méthodes neuronales basée sur les réseaux à convolution	43
4.5.2	Méthodes neuronales basée sur les cartes	45
4.6	Approche multi-échelles	48

4.1 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.

4.2 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.

4.2.1 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 4.1 sur toute l'image.

$$\iint [(\nabla I(x, y, t) \cdot V(x, y, t) + I_t(x, y, t))^2 + \alpha^2 (\nabla v_x(x, y, t))^2 + (\nabla v_y(x, y, t))^2] dx dy \quad (4.1)$$

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 4.1 est obtenue par

l'utilisation de l'approximation de Laplacien et la méthode de Gauss-Seidal :

$$\begin{cases} v_x^{l+1} = v_x^{-l} - \frac{I_x[I_x v_x^{-l} + I_y v_y^{-l} + I_t]}{\alpha^2 + I_x^2 + I_y^2} \\ v_y^{l+1} = v_y^{-l} - \frac{I_y[I_x v_x^{-l} + I_y v_y^{-l} + I_t]}{\alpha^2 + I_x^2 + I_y^2} \end{cases} \quad (4.2)$$

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 4.1), comme suit :

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

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

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

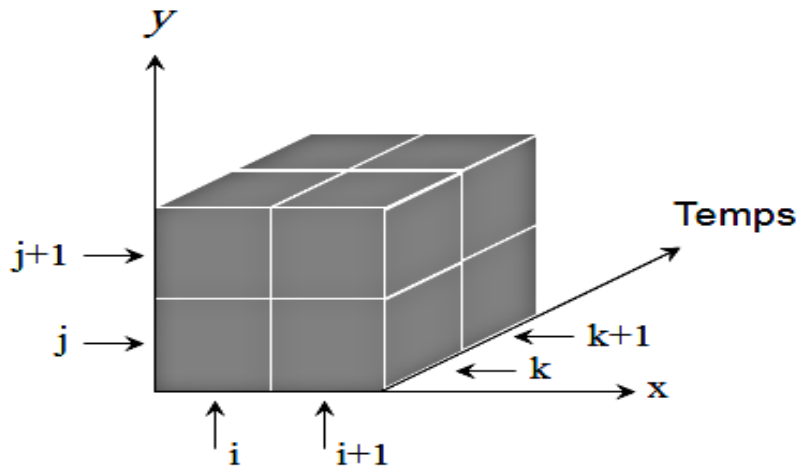


FIGURE 4.1 – Calcul des Dérivées selon Horn et Schunck

Exemple :

Considérant une sphère en rotation sur elle même (Figure 4.2), leur flot optique obtenue par la méthode de Horn et Schunck après une itération, 16 et 100 itérations est schématisé sur la Figure 4.3

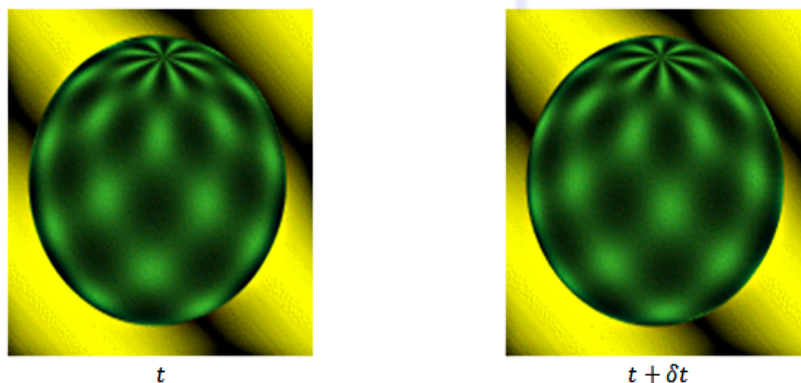


FIGURE 4.2 – sphère en rotation

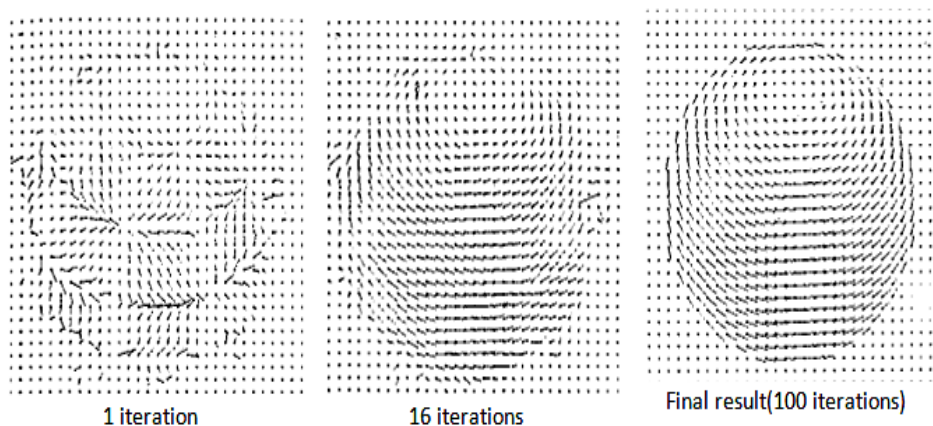


FIGURE 4.3 – Flot optique estimé par la méthode de Horn et Schunck

4.2.2 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 3.9 peut

s'écrire comme suit :

$$\begin{bmatrix} I_{x1} & I_{y1} \\ I_{x2} & I_{y2} \\ \vdots & \vdots \\ I_{xn} & I_{yn} \end{bmatrix} \begin{bmatrix} v_x \\ v_y \end{bmatrix} = \begin{bmatrix} -I_{t1} \\ -I_{t2} \\ \vdots \\ -I_{tn} \end{bmatrix}. \quad (4.3)$$

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

$$A\vec{V} = -b \quad (4.4)$$

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

$$\vec{V} = (-A^T A)^{-1} A^T b, \quad (4.5)$$

plus formellement on peut écrire :

$$\begin{bmatrix} v_x \\ v_y \end{bmatrix} = - \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} (A^T I_t), \quad (4.6)$$

Exemple : Soit l'image réelle de transverse du thorax à rayons X présentée dans la figure 4.4 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 4.4 présente le flot optique réel et celui estimé par la méthode de Lucas et Kanade avec un fenêtrage de taille 11×11 .

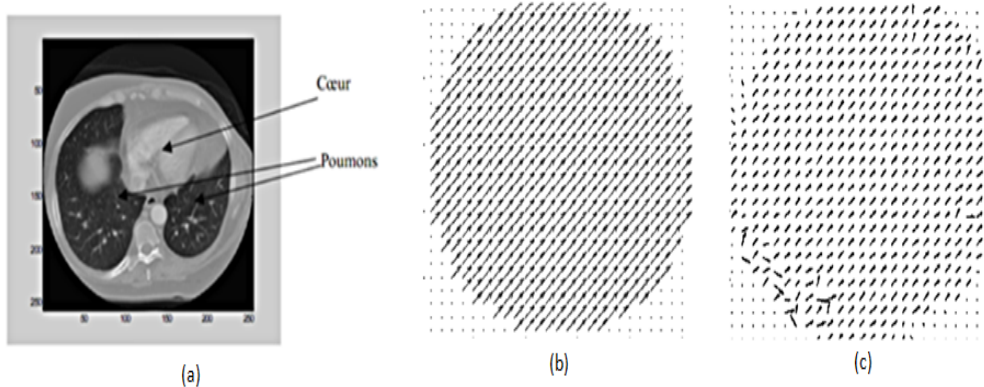


FIGURE 4.4 – (a) L'image réelle Thorax, (b) flot optique réel, (c) flot optique estimé avec la méthode Lucas et Kanade

4.3 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.

4.3.1 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 :

$$v^* = \arg \min_{x,y \in W} [B_c(x, y), B_r(x + v_x, y + v_y)] \quad (4.7)$$

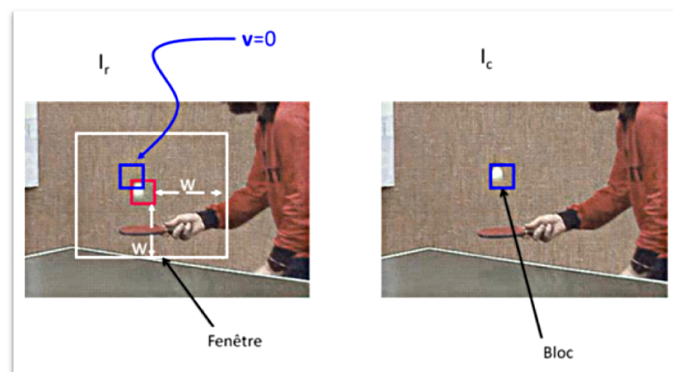


FIGURE 4.5 – Principe des MCB

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 a 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.

4.3.2 Critère de similarité

Comme montré dans l'équation précédente 4.7, 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 :

$$SSD(B_c, B_r) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} [B_c(x+i, y+j) - B_r(i, j)]^2 \quad (4.8)$$

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 :

$$SAD(B_c, B_r) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |B_c(x+i, y+j) - B_r(i, j)| \quad (4.9)$$

4.3.3 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 du critère de similarité le plus petit possible (voir la Figure 4.6). 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.

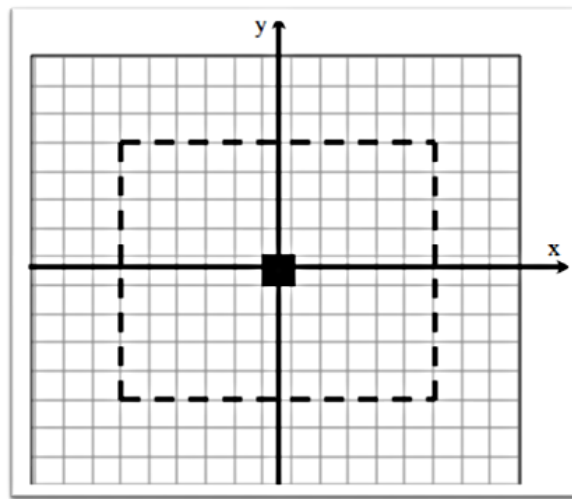


FIGURE 4.6 – La recherche exhaustive

- **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 4.7.

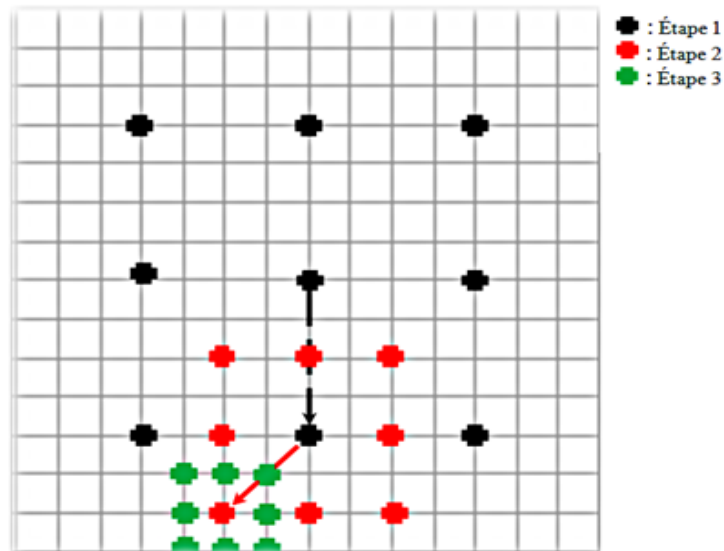


FIGURE 4.7 – Recherche en trois étapes

• 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 illimité. L’algorithme DS utilise deux modèles de recherche (Figure 4.8) :

- 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.

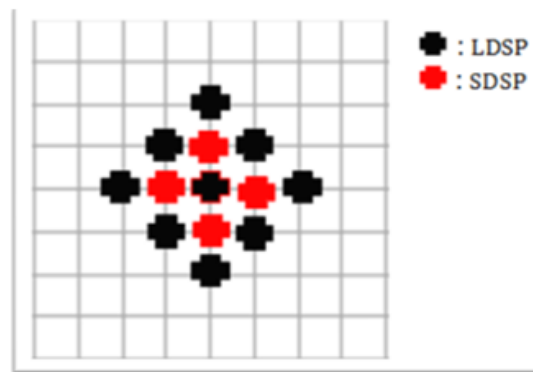


FIGURE 4.8 – Les deux modèles de l’algorithme DS

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 4.9 résume le processus de cet algorithme :

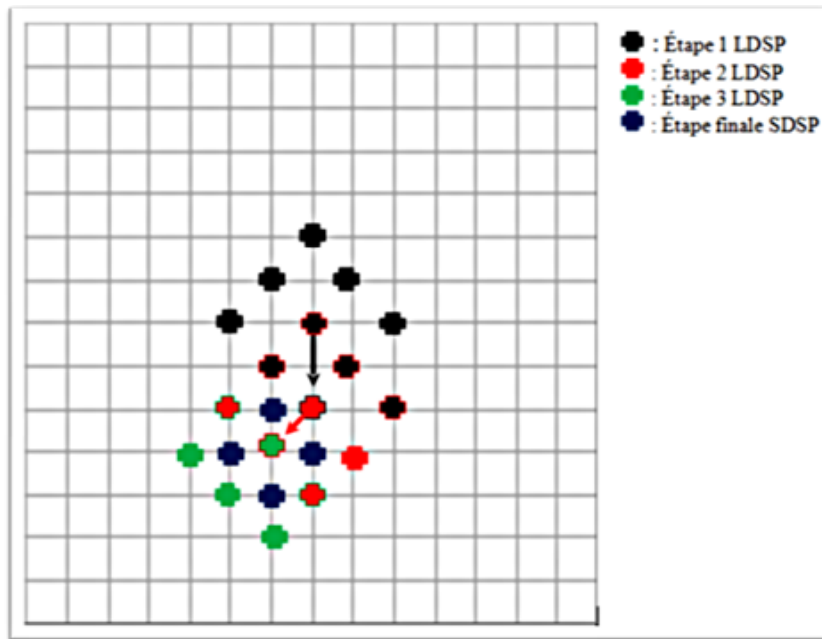


FIGURE 4.9 – L'algorithme DS

Exemple :

Avec l'image réelle présentée dans la figure 4.10 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 décrit précédemment (FS, TSS,DS) sont présentées dans le tableau 4.1 suivant et leurs flot optique est schématisé sur la figure 4.10.

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

TABLE 4.1 – 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.

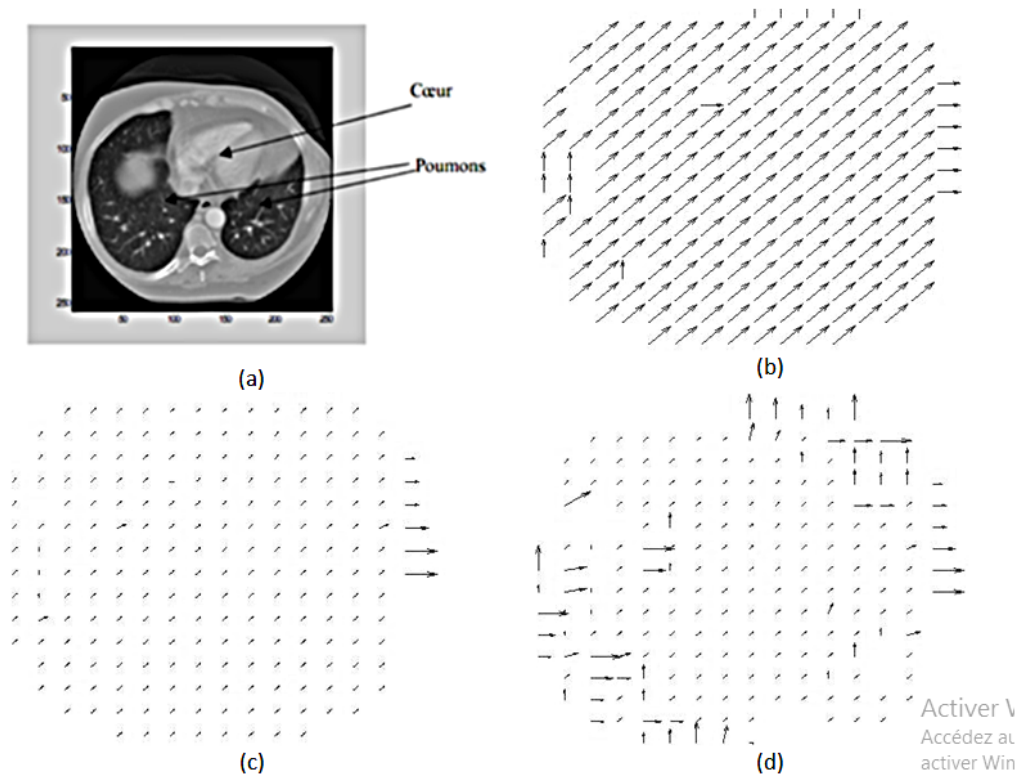


FIGURE 4.10 – Exemple des MCB, (a) l'image Thorax, (b) flot optique de FS, (c) flot optique de DS, (d) flot optique de TSS

4.4 Méthodes fréquentielles

4.4.1 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 :

$$F(f_x, f_y, f_t) = F_0(f_x, f_y)\delta(v_x f_x + v_y f_y + f_t) \quad (4.10)$$

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 temporelle.

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 :

$$v_x f_x + v_y f_y + f_t = 0 \quad (4.11)$$

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.

4.4.2 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 :

$$G_2(\vec{f}) = G_1(\vec{f}) \cdot e^{-2\pi i \vec{f} \cdot \vec{d}} \quad (4.12)$$

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 4.13 :

$$e^{j\phi(\vec{f})} = \frac{G_1(\vec{f})G_2^*(\vec{f}).e^{-2j\pi\vec{f}\vec{d}}}{|G_1(\vec{f})G_2^*(\vec{f}).e^{-2j\pi\vec{f}\vec{d}}|} \quad (4.13)$$

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 4.14 :

$$p(x, y) = F^{-1} \left(e^{j\phi(\vec{f})} \right) \quad (4.14)$$

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

Les coordonnées (d_x, d_y) correspondent au maximum de l'équation 4.14. 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 :

$$(d_x, d_y) = \arg \max(p(x, y)) \quad (4.15)$$

Un exemple expliquant le processus de cette méthode est présenté sur la Figure 4.11.

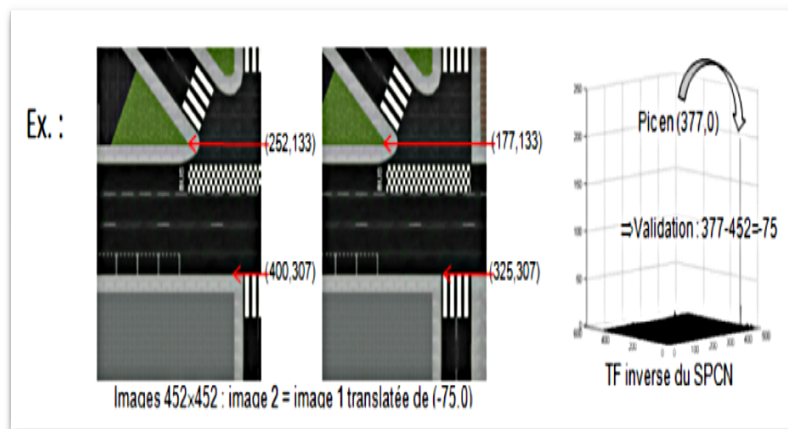


FIGURE 4.11 – Exemple expliquant la méthode de corrélation de phase, cas d'une translation

4.4.3 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 apparaît 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 4.12¹).

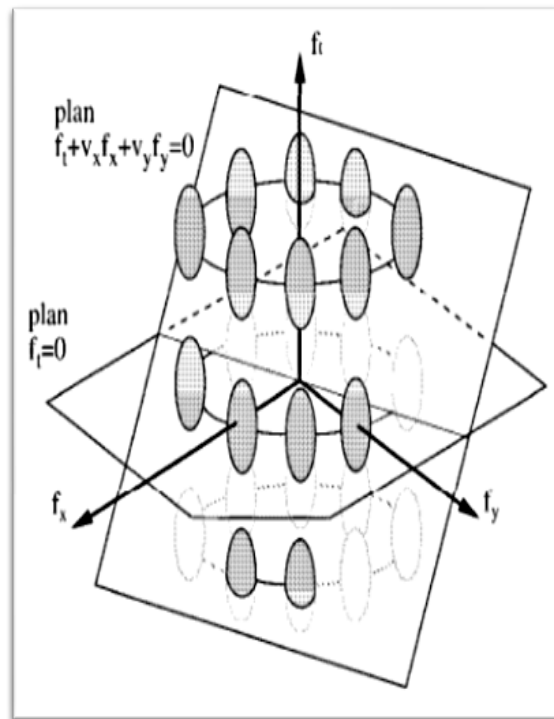


FIGURE 4.12 – Spectres de 12 filtres spatio-temporels orientés de Gabor

Un filtre de Gabor 3D est défini comme suit :

$$g(x, y, t) = \frac{1}{(2\pi)^{\frac{3}{2}} \sigma_x \sigma_y \sigma_t} \exp \left[-\left(\frac{x^2}{2\sigma_x^2} + \frac{y^2}{2\sigma_y^2} + \frac{t^2}{2\sigma_t^2} \right) \right] \exp[2\pi j(f_{x_0}x + f_{y_0}y + f_{t_0}t)] \quad (4.16)$$

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

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 :

$$R(v_x, v_y) = \exp \left[\frac{-4\pi^2 \sigma_x^2 \sigma_y^2 \sigma_t^2 (v_x f_x + v_y f_y + f_t)^2}{(v_x \sigma_x \sigma_t)^2 + (v_y \sigma_y \sigma_t)^2 + (\sigma_x \sigma_y)^2} \right]. \quad (4.17)$$

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 :

$$f(v_x, v_y) = \sum_{i=1}^{12} \left(m_i - \bar{m}_i \frac{R_i(v_x, v_y)}{\bar{R}_i(v_x, v_y)} \right)^2 \quad (4.18)$$

Avec,

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

Exemple :

La figure 4.13 montre le flot optique estimé par les deux méthodes fréquentielles : la corrélation de phase et la méthode de Heeger en utilisant la séquence taxi de Hambourg.

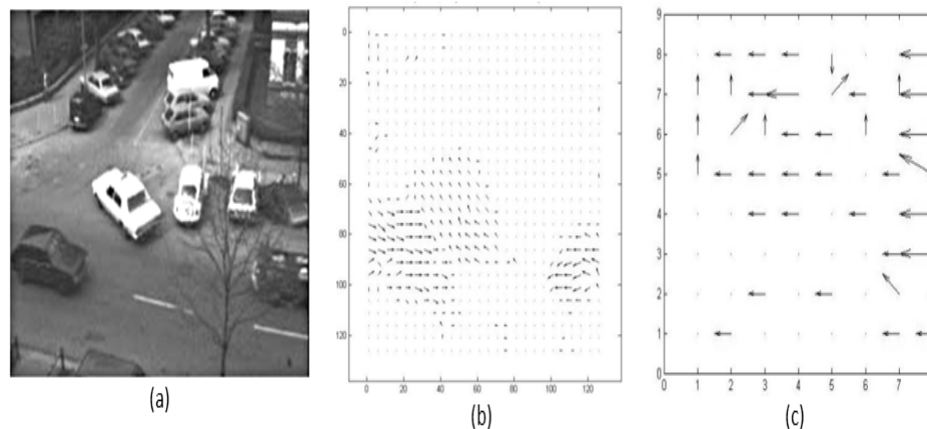


FIGURE 4.13 – (a) image de la séquence Taxi de Hambourg, (b) flot optique estimé par la méthode de Heeger, (c) b) flot optique estimé par la méthode de corrélation de phase

4.5 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).

4.5.1 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 [[Ahmadi and Patras 2017](#)], 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 4.14.

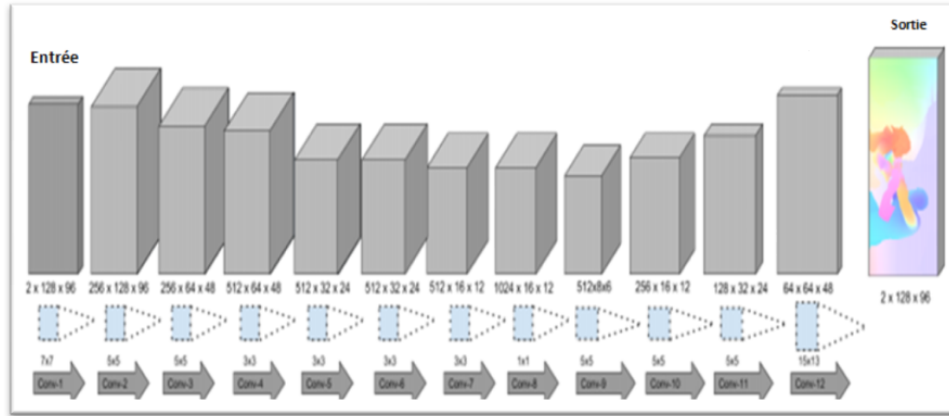


FIGURE 4.14 – L'architecture du CNN proposé par Ahmadi et Patras [Ahmadi and Patras 2017]

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 :

$$E(F) = \sum_{x,y=1}^M \sqrt{(v_x I_x + v_y I_y + I_t)^2 + \epsilon} \quad (4.19)$$

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

$$\frac{\partial E}{\partial w} = \frac{\partial E}{\partial F} \frac{\partial F}{\partial w} \quad (4.20)$$

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 :

$$\frac{\partial E}{\partial F} = \begin{bmatrix} \frac{\partial E}{\partial v_x} \\ \frac{\partial E}{\partial v_y} \end{bmatrix} = \begin{bmatrix} \sum_{x,y=1}^M \frac{I_x(v_x I_x + v_y I_y + I_t)}{\sqrt{(v_x I_x + v_y I_y + I_t)^2 + \epsilon}} \\ \sum_{x,y=1}^M \frac{I_y(v_x I_x + v_y I_y + I_t)}{\sqrt{(v_x I_x + v_y I_y + I_t)^2 + \epsilon}} \end{bmatrix} \quad (4.21)$$

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 à 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 4.15 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 4.15 – Exemple de la méthode d'Ahmadi et Patras, (a) image de la séquence, (b) flot optique estimé, (c) codage des couleurs

4.5.2 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 [Garcia et al 2016], 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 4.16; 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.

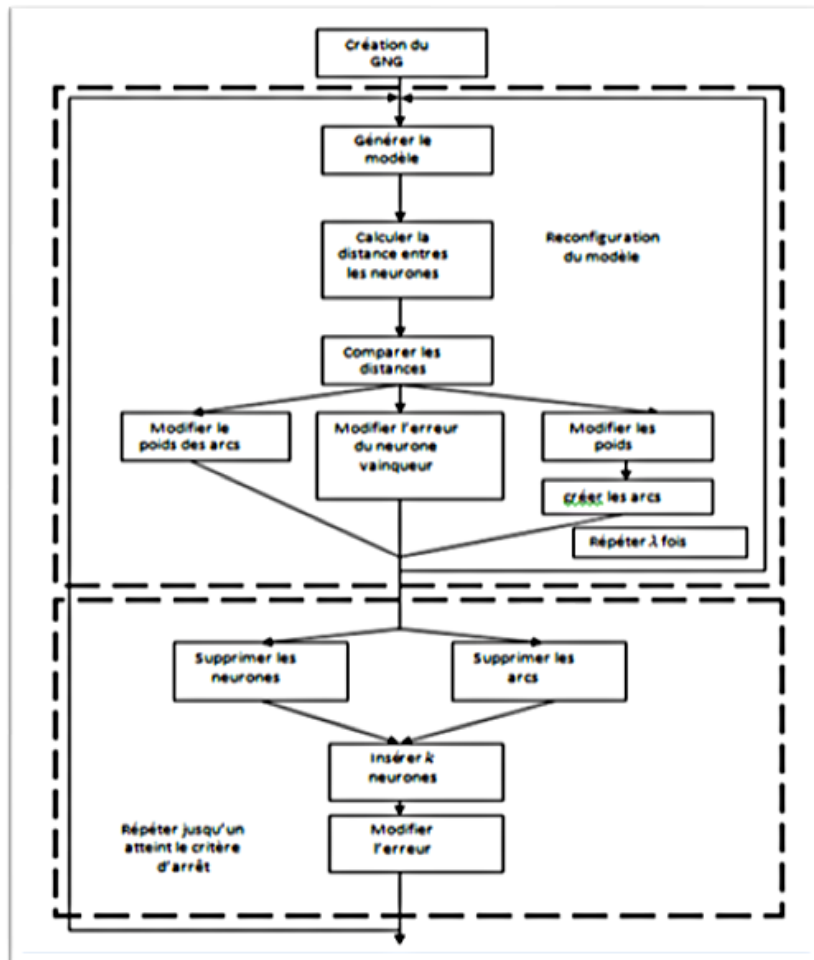


FIGURE 4.16 – L’algorithme d’apprentissage de GNG [Garcia et al 2016]

Exemple :

Un exemple avec la méthode de Garcia et al., est illustré sur la Figure 4.17 de la base de données CAVIAR².

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

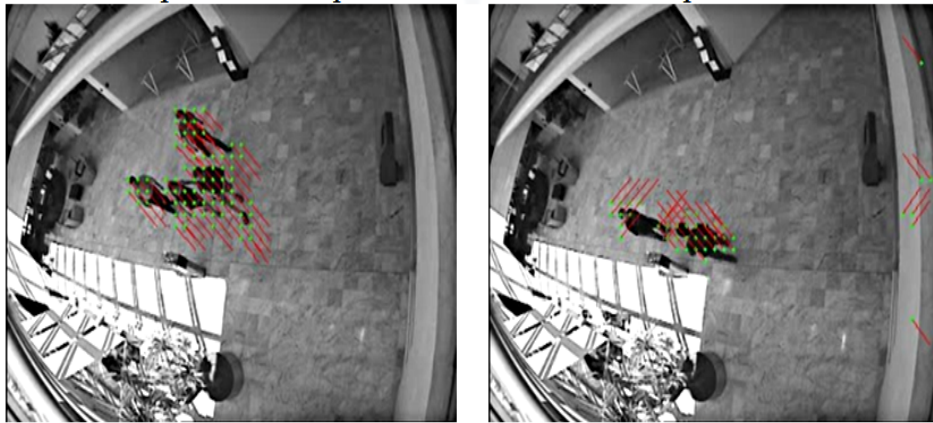


FIGURE 4.17 – Exemple de la méthode de Garcia et al. tiré de [Garcia et al 2016]

4.6 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 pyramide³(voir Figure 4.18). 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.

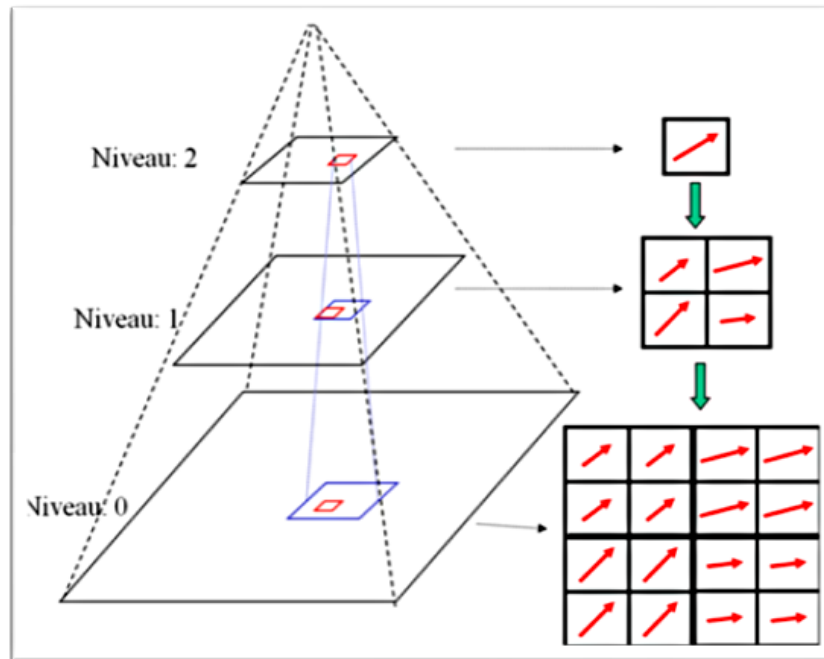


FIGURE 4.18 – Principe des approches multi-échelles

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 :

$$v^{(i,i-1)} = v^i + v^{(i-1)} \quad (4.22)$$

Un exemple d'estimation de mouvement multi-échelle est illustré sur la Figure 4.20 avec la séquence "Bus" (voir la Figure 4.19).



FIGURE 4.19 – Séquence d'images "Bus"

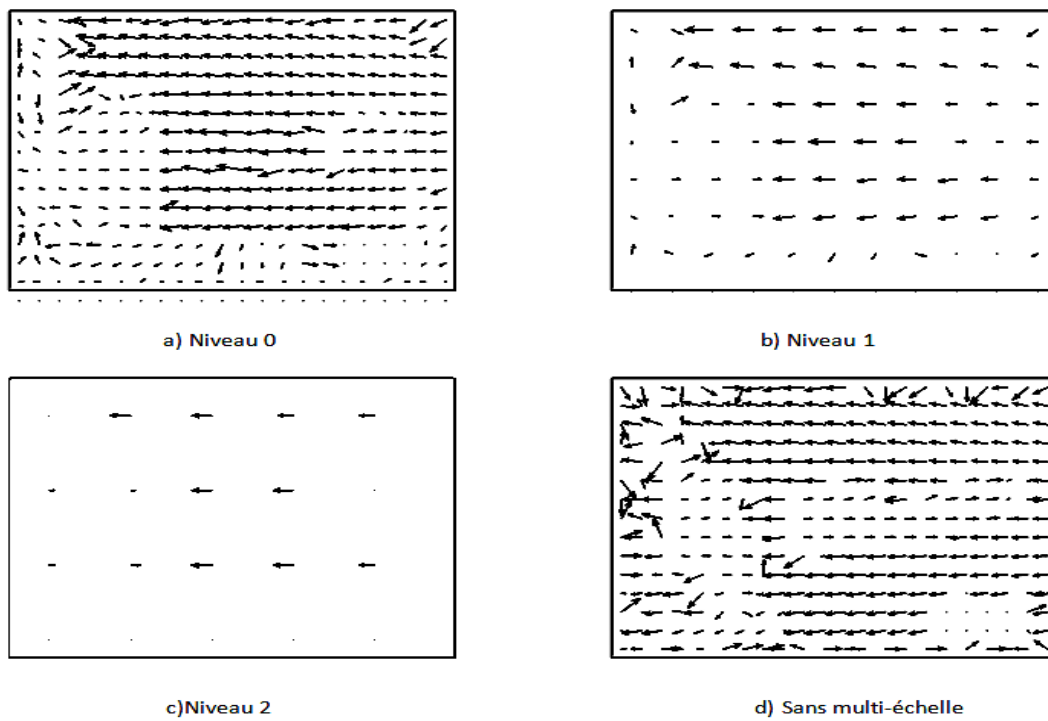


FIGURE 4.20 – Exemple d'estimation multi-échelle

Exercices

Série d'exercices 1

Exercice N°1 :

- Qu'est-ce que le mouvement non rigide et quelle est la différence entre un mouvement rigide et un mouvement non rigide ?
- Quelle est la différence entre la détection de mouvement et la détection d'objets en mouvement ?
- Décrivez la différence entre le mouvement apparent et mouvement réel.
- Quel est l'intérêt de l'opération de seuillage ? Le seuillage d'image est-il considéré comme une technique de détection de mouvement ? Justifier votre réponse.

Exercice N°2 :

- Chargez deux images de la vidéo Taxi de Hambourg
- Affichez les images et comparez-les.
- Quel type de mouvement observez-vous ?

Exercice N°3 :

- Lisez sous Matlab une séquence d'images.
- Faisiez le seuillage des images
- Calculez la valeur absolue de la différence entre la première et la dernière image de la séquence.
- Calculez le «et » logique en suite le « min » entre les deux premières images seuillées, que vous remarquez.
- Calculez le «ou» logique en suite le « max » entre les deux premières images seuillées, que vous remarquez.

Série d'exercices 2

Exercice N°1 :

- Comment les images de différence cumulative peuvent être utilisées dans l'analyse de mouvement ?

Exercice N°2 :

- Comment déterminer l'image de références pour la détection de mouvement avec image de référence ?
- Implémenter et tester sous Matlab l'algorithme de détection de mouvement avec image de référence.

Exercice N°3 :

- Implémenter et tester les algorithmes de la détection de mouvement suivants :
 - * basée sur la différence entre deux les images successives(RGB).
 - * basée sur la différence entre les trois trames.
 - * proposée par A.Bello.
- Comparer entre les trois méthodes.

Exercice N°4 :

- Définissez le critère du test de vraisemblance dans le domaine de la détection de mouvement
- Comment nous pouvons utiliser ce critère pour la détection de mouvement.
- si $m = n = 12$, pour un niveau de confiance 5% alors qu'elle hypothèse on doit choisir (H_0 ou H_1), Justifier votre réponse ?

Exercice N°5 :

- Expliquez l'approche Markovienne pour la détection de mouvement.

Série d'exercices 3

Exercice N°1 :

- Définissez flot optique, à quoi sert le flot optique ? Quel est son intérêt et citez ces deux représentations.
- Qu'est-ce que l'estimation de flux optique ?
- Qu'est-ce qu'un flot optique dense ?
- Quel est le problème d'occultation ? Comment nous pouvons résoudre ce problème.
- Quel est le problème d'ouverture dans le flot optique ? Comment nous pouvons résoudre ce problème.
- Qu'est-ce qu'un champ de flot optique ? Qu'est-ce qu'un champ de mouvement 2D ? Quelle est la différence entre les deux ? Comment sont-ils liés les uns aux autres ?
- Pourquoi considérons-nous l'estimation de mouvement dans une séquence d'images comme un problème mal posé ?
- Qu'implique la contrainte de lissage ? Pourquoi est-ce nécessaire ?
- Quelles sont les hypothèses utilisées pour le calcul de flot optique. Dans le monde réel ces hypothèses sont-elles réalistes ?
- Décrivez le principe de l'estimation de mouvement.

Exercice N°2 :

- Par l'utilisation de développement de Taylor et à partir de l'hypothèse de conservation de l'intensité lumineuse, trouvez l'équation de flot optique.

Série d'exercices 4

Exercice N°1 :

- Comment les dérivées de la fonction d'intensité des composantes de flot optique sont-elles estimées dans la méthode de Horn et Shunck ?
- Qu'elle est la différence entre la méthode de Horn & Shunck et Lucas&Kanade.
- Décrivez les différentes solutions proposées pour la résolution de l'équation du flot optique par les méthodes différentielles.
- Expliquez le principe des méthodes différentielles ?

Exercice N°2 :

- Implémenter la méthode de Horn et Schunck sous Matlab et le tester sur des séquences de synthèse.
- Implémenter la méthode de Lucas&Kanade sous Matlab et le tester sur des séquences de synthèse.
- Comparer les résultats obtenus par les deux méthodes.

Exercice N°3 :

- Chargez une séquence de deux images.
- Affichez les images.
- Quel type de mouvement observez-vous ?
- Évaluez les performances de l'estimation de mouvement par les méthodes de MCB :FS, TSS et DS.
- Modifiez les paramètres : taille bloc et taille de la fenêtre de recherche, que vous remarquez.
- calculer la complexité de chaque algorithme (temps de calcul ou le nombre d'opérations).

Série d'exercices 4

Exercice N°4 :

- Quelles sont les idées sous-jacentes derrière la méthode de la corrélation de phase ainsi que la méthode de Heeger.
- Pourquoi un seul filtre d'énergie de Gabor n'est-il pas suffisant pour l'estimation de mouvement ?
- Implémentez les deux méthodes sous Matlab et les tester sur des séquences de synthèse.
- Comparez les résultats obtenus par les deux méthodes.

Exercice N°5 :

- Quelle sont les avantages des réseaux de neurones.
- Que apporte les CNN.
- Citez les différentes couches d'un CNN.
- Décrivez le processus d'estimation de mouvement dans les CNNs.
- Comparez les CNNs avec les cartes auto-organisatrices.

Exercice N°6 :

- Implémentez et tester l'algorithme de Horn&Shunck-multi-échelle sous Matlab pour qu'il puisse estimer le mouvement de grand déplacement.
- Même question que le précédent pour l'algorithme FS.

Bibliographie

- [Ahmadi and Patras 2017] Ahmadi, A. and Patras, I. (2017); "unsupervised convolutional neural networks for motion estimation", in *Computer Vision and Pattern Recognition (cs.CV)*, Vol abs/1601.06087. 43, 44
- [Ambellouis 2011] Ambellouis, S. (2011), "Analyse du mouvement dans les séquences d'images par une méthode récursive de filtrage spatio-temporel sélectif", *PhD thesis Université Pierre et Marie Curie*.
- [Argyriou 2011] Argyriou, V. (2011), "Sub-Hexagonal Phase Correlation for Motion Estimation", *IEEE Transactions on image processing*, vol. 20, no. 1.
- [Barron et al 1994] Barron, J. L., Fleet, D. J. and Beauchemin, S.(1994), "Performance of optical flow techniques", *Int. J. Comput. Vis.*, vol. 12, no. 1, pp. 43-77.
- [Bouthemy and Lalande 1993] Bouthemy, and P. Lalande, "Recovery of moving object masks in an image sequence using local spatiotemporal contextual information," *Optical Engineering* 32(6), 1205-1212 (1993). 18
- [Cozzella and Spagnolo 2014] Cozzella L, Spagnolo GS (2014), "Phase-only Correlation Function by Means of Hartley Transform". *JSM Math Stat* 1(1) : 1004. 2014.
- [Fleet 1990] Fleet, D.J., Jepson, A.D.(1990), "Computation of component image velocity local phase information". *Int. J. Comp. Vision* 5, pp.77-104.
- [Fleet 1994] Fleet, D. J.(1994), "Disparity from Local Weighted Phase-Correlation", *IEEE Int. Conf. on Systems, Man and Cybernetics*, vol.1, pp. 48-54.

- [Fritzke 1995] Fritzke, B. (1995), "A Growing Neural Gas Network Learns Topologies". *Vol 7. MIT Press (1995) 625-632.*
- [Garcia et al 2016] Garcia-Rodriguez, J. , Orts, S., Angelopoulou, A., Psarrou, A., Azorin-López, J. and Chamizo, J. M. G.(2016), "Real time motion estimation using a neural architecture implemented on GPUs". *Journal of Real-Time Image Processing, Volume 11 Issue 4; Pages 731-749.* [45](#), [47](#), [48](#)
- [Ghoul et al 2018] Ghoul, K., Berkane, M. and Batouche, M., C.(2018) "A neural-based method for optical flow estimation using phase correlation", *In International Journal of Computational Vision and Robotics(IJCVR), 2018, in press.*
- [Heeger 1988] Heeger D.J.(1988), "Optical flow using spatiotemporal filters". *Int. J. Com. Vision 1. pp.279-302.*
- [Horn and Schunck 1981] Horn, B.K.P. and Schunck, B.G. (1981), "Determining optical flow", *Artificial Intelligence, Vol. 17, Nos. 1-3, pp.185-203.*
- [Horn 1986] Horn, B.K.P. (1986), "Robot vision". *MIT Press, ISBN 0-262-08159-8.*
- [Hsu et al. 1984] Hsu Y. Z, NAGEL H.- H., et REKERS G. , "New Likelihood Test Methods for Change Detection in Image Sequences" . *COMPUTER VISION, GRAPHICS, AND IMAGE PROCESSING 26, 73-106 (1984).* [15](#), [16](#), [17](#)
- [Jakubowski and Pastuszak 2013] Jakubowski, M. and Pastuszak, G. (2013), "Block-based motion estimation algorithms a survey". *Opto-Electronics Review, 21(1) :86-102.*
- [Laveau 2005] Laveau, N. (2005), "Estimation, Compression et Filtrage morphologique". *Thèse PhD, École Nationale Supérieure des Mines de Paris.*
- [Lucas and Kanade 1981] Lucas , B. and Kanade,T. (1981), "An iterative image registration technique with an application to stereo video", *Proceedings of DARPA Image Understanding Workshop, Washington, DC.*
- [Liu and Delbruck 2017] Liu , M. and Delbruck, T. (2017), "Block-matching optical flow for dynamic vision sensors : Algorithm and FPGA implementation", *IEEE International Symposium on Circuits and Systems (ISCAS),2017*

- [Metkar and Talbar 2013] Metkar S. and S. Talbar,(2013), "Motion Estimation Techniques for Digital Video Coding", *SpringerBriefs in Computational Intelligence*,DOI : 10.1007/978-81-322-1097-9-2.
- [Pesquet-Popescu et al 2014] Pesquet-Popescu, B., Cagnazzo, M. and Dufaux, F.(2014), "Motion Estimation Techniques", *TELECOM ParisTech*.
- [Pingault 2003] Pingault, M.(2003),"estimations fréquentielle et temporelle du mouvement en transparence additive dans les séquences d'images", *thèse doctorat université Joseph Fourier*.
- [Potts 1952] R. B. Potts, "Some generalized order-disorder transitions", *Proc. Camb. Phil. Soc.*, 48, pp. 106-109, 1952. [20](#)
- [Razali et al 2013] Razali, Y., Alihossein, A., Alfian A. H. and Nasir S. (2013), "A Comparison of Different Block Matching Algorithms for Motion Estimation", *The 4th International Conference on Electrical Engineering and Informatics (ICEEI 2013)*.
- [Xavie 2010] Xavie, M. (2010), "Étude quantitative du mouvement de la paroi du cœur à partir d'images ciné-IRM par des méthodes fréquentielles de flux optique", *thèse de doctorat, université DE Bourgogne*.