

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITÉ IBN-KHALDOUN DE TIARET

FACULTÉ DES SCIENCES APPLIQUEES
DÉPARTEMENT DE GENIE ELECTRIQUE



MEMOIRE DE FIN D'ETUDES

Pour l'obtention du diplôme de Master

Domaine : Sciences et Technologie

Filière : Electronique

Spécialité : Electronique des systèmes embarqués

THÈME

DETECTION ET SUIVI D'OBJET MOBILE DANS UNE SEQUENCE VIDEO

Préparé par :

-Mr. COULIBALY Hassan G-K

-Mr. GUIROU Jean

Devant le Jury :

Nom et prénoms	Grade		Qualité
Y. Belhadji	MCB	Univ. Tiaret	Président
R. Otmani	MAA	Univ. Tiaret	Examineur
A. Ghellab	MCB	Univ. Tiaret	Encadreur

PROMOTION 2018 /2019

REMERCIEMENTS

Nous tenons tout d'abord à remercier Dieu, tout puissant clément et miséricordieux de nous avoir doté d'une patience et d'un courage inouï pour la réalisation de ce modeste travail, qu'il soit béni et glorifié.

Nos profonds remerciements sont adressés à notre respectueux encadreur Mr. A. Ghellab, qui a veillé à la qualité de cette recherche dont nous espérons être à la hauteur de l'ambition souhaitée.

Nos remerciements sont aussi adressés aux honorables membres du jury qui ont accepté la lecture et l'évaluation de ce travail. Sans oublier nos humbles enseignants qui méritent l'honneur et la reconnaissance de notre formation.

Enfin nous remercions nos familles respectives et nos chers amis pour leur soutien moral sans qui tout cela n'aurait pas pu arriver.

Liste des figures

Figure 1.1 : Séquence d'images	4
Figure 1.2 : Projection du mouvement 3D en 2D.....	5
Figure 1.3: Exemple de flot optique.....	6
Figure 1.4: Temps d'acquisition.	9
Figure 1.5: Schéma conceptuel de traitement de mouvement.	10
Figure 2. 1: Block Matching.....	16
Figure 2. 2: Filtre de Kalman.....	25
Figure 2. 3: Représentation d'un coin.....	29
Figure 2. 4: Schéma synoptique différenciant un bord à un coin dans une image. (a) bord et (b) coin	30
Figure 2. 5: Image cameraman.tif soumise au détecteur de Harris	33
Figure 2. 6: Image cameraman.tif avec bruit gaussien soumise au détecteur de Harris.....	34
Figure 2. 7: Image cameraman.tif soumise au détecteur de points clés SIFT	38
Figure 2. 8: La méthode de H&S sur la vidéo visiontraffic	41
Figure 2. 9: La méthode de H&S sur la vidéo visiontraffic face à un bruit gaussien.....	41
Figure 3. 1: Système d'acquisition vidéo.	47
Figure 3. 2: Schéma synoptique montrant le concept temps réel pour notre application.....	49
Figure 3. 3: Organigramme de l'application.	50
Figure 3. 4: Mouvements d'un joueur et de la balle : a) l'image de référence, b) mouvements perçus dans la trame suivante.	51
Figure 3. 5: Mouvements d'étudiants dans la scène : a) l'image de référence, b) mouvements perçus dans la trame suivante.	52
Figure 3. 6: Mouvements de voiture dans la scène : a) l'image de référence, b) mouvements perçus dans la trame suivante.	53

Figure 3. 7: Trames acquises à des instants différents : a) Première trame, b) Deuxième trame.54

Figure 3. 8: Application du détecteur de Harris sur deux trames : a) Première trame, b) Deuxième trame.....55

Figure 3. 9: Image avec une résolution de 640x480.....56

Figure 3. 10: Image avec une résolution de 1334x750.....57

Liste des tableaux

Tableau 2. 1: Etude comparative vis-à-vis du temps de calcul40

SOMMAIRE

REMERCIEMENTS	I
LISTE DES FIGURES	II
LISTE DES TABLEAUX	IV
NOMENCLATURE	VIII
LISTES DES ABREVIATIONS	XI
INTRODUCTION GENERALE	1
CHAPITRE 1 : ETAT DE L'ART	3
1.1 INTRODUCTION	4
1.2 DÉFINITION D'UNE VIDÉO	4
1.3 MOUVEMENT DANS UNE SÉQUENCE VIDÉO	5
1.3.1 Mouvement réel.....	5
1.3.2 Mouvement apparent.....	5
1.3.3 Equation du flop optique	6
1.3.4 Domaines d'utilisation du flot optique.....	7
1.3.5 Contraintes et problèmes du mouvement.....	8
1.4 TEMPS D'ACQUISITION	8
1.5 INFORMATIONS À EXTRAIRE.....	10
1.6 MÉTHODES D'ESTIMATION DU MOUVEMENT	11
1.6.1 Méthodes directes.....	11
1.6.1.1 Méthodes différentielles	11
1.6.1.2 Méthodes de mise en correspondance	11
1.6.1.3 Méthodes fréquentielles	12
1.6.2 Méthodes indirectes.....	12
1.7 CONCLUSION.....	12

CHAPITRE 2 : ETUDE COMPARATIVE ENTRE DIFFERENTES METHODES D'ESTIMATION DE MOUVEMENT	14
2.1 INTRODUCTION	15
2.2 MÉTHODES DE MISE EN CORRESPONDANCE OU BLOCK MATCHING.....	15
2.2.1 Principe de fonctionnement.....	15
2.2.2 Largeur de la zone de recherche.....	16
2.2.3 Taille des blocs.....	16
2.2.4 Types de fonction de coût	17
2.2.5 Algorithmes de Block Matching	17
2.2.5.1 Recherche exhaustive (ES).....	18
2.3 MÉTHODES DIFFÉRENTIELLES	19
2.3.1 Méthodes locales	19
2.3.1.1 Principe de fonctionnement	19
2.3.1.2 Fenêtre pondérée.....	20
2.3.2 Méthodes globales.....	21
2.3.2.1 Principe de fonctionnement	21
2.3.3 Filtre de Kalman méthode différentielle évoluée.....	24
2.3.3.1 Suivi de mouvement par filtre de Kalman	24
2.3.3.2 Principe du filtre de Kalman.....	24
2.4 MÉTHODES FRÉQUENTIELLES.....	25
2.4.1 Méthode de corrélation de phase.....	27
2.5 MÉTHODES INDIRECTES	28
2.5.1 Détecteur de coin Harris.....	29
2.5.1.1 Coin.....	29
2.5.1.2 Différenciation des bords et des coins	30
2.5.1.3 Formulation.....	30
2.5.1.4 Processus de l'algorithme de détection de coins Harris	32
2.5.1.5 Utilisation du detecteur de Harris dans Matlab	33
2.5.2 Scale-Invariant Feature Transform (SIFT).....	34
2.5.2.1 Exemple d'utilisation de la méthode SIFT dans Matlab.....	38
2.6 ETUDE COMPARATIVE.....	39

2.6.1	Temps de calcul.....	39
2.6.2	Sensibilité au bruit.....	40
2.6.2.1	Méthode de H&S	41
2.6.2.2	Méthode de BM	42
2.6.3	Comportement face aux grands déplacements	43
2.6.3.1	Méthode de H&S	43
2.6.3.2	Méthode de BM	44
2.7	CONCLUSION.....	44
 CHAPITRE 3 : MISE EN ŒUVRE D’UN SYSTÈME D’ACQUISITION ET D’ESTIMATION DE MOUVEMENT EN TEMPS RÉEL.....		46
3.1	INTRODUCTION	47
3.2	ACQUISITION D’IMAGES	47
3.3	CARACTÉRISTIQUES DU MATÉRIEL UTILISÉ	48
3.4	ACQUISITION EN TEMPS RÉEL.....	48
3.5	ALGORITHME DE L’APPLICATION	49
3.6	RÉSULTATS DE L’APPLICATION EN TEMPS RÉEL	51
3.6.1	Hypothèse des erreurs d’estimation du flot optique.....	53
3.6.1.1	Surfaces lambertiennes	54
3.6.1.2	Résolution de la camera	55
3.7	CONCLUSION.....	57
 CONCLUSION GENERALE		58
 BIBLIOGRAPHIE		61

NOMENCLATURE

Δt :	Intervalle de temps fixe
$I(x, y, t)$:	Intensité d'un pixel $p = (x, y)$ à l'instant t d'une image numérique
$I(f_x, f_y, f_t)$:	Transformée de Fourier de $I(x, y, t)$
u ou v_x :	Vitesse apparente ou champ de déplacement suivant la direction x
v ou v_y :	Vitesse apparente ou champ de déplacement suivant la direction y
I_x :	Dérivée(variation) de l'intensité d'un pixel par rapport à x (l'axe des abscisses)
I_y :	Dérivée(variation) de l'intensité d'un pixel par rapport à y (l'axe des ordonnées)
I_t :	Dérivée(variation) de l'intensité d'un pixel par rapport au temps
∇I :	Gradient de l'image I au pixel (x, y)
L :	Taille de l'objet d'une scène réelle
$l(t)$:	Taille apparente de l'objet(barre) d'une scène a l'instant t
$l'(t)$:	Dérivée de la taille apparente $l(t)$
D :	Distance entre la barre et le plan optique O
f :	Distance entre le plan optique et le plan de projection
τ :	Temps d'acquisition nécessaire à l'objet pour atteindre la caméra
V :	Vélocité de la barre dans l'espace 3D
I_C :	Image cible (courante)

I_r :	Image de référence.
$V = \begin{bmatrix} u \\ v \end{bmatrix}$:	Matrice de vecteur vitesse constituée de vitesse apparente horizontale et verticale
q_i :	les pixels de la fenêtre centrée et $i = 1$ à n correspondant au nombre de pixels.
A :	Matrice constituée des dérivées spatiales de l'image des différents pixels de la fenêtre centrée
B :	Matrice constituée des dérivées temporelles de l'image des différents pixels de la fenêtre centrée
W :	Matrice diagonale contenant les poids w_i associé à l'équation du pixel q_i .
J :	Fonctionnelle à minimiser par la méthode de Horn & Shunck
e_c :	Equation de contrainte du champ de vitesse
e_s :	Variations spatiales du champ de vitesse
α^2 :	Facteur de pondération qui contrôle la robustesse de la contrainte de lissage
$\Delta^2 u(x, y)$:	Laplacien de la vitesse apparente u
$\bar{u}(x, y), \bar{v}(x, y)$:	Moyennes pondérées de u et v calculé dans un voisinage autour du pixel à l'emplacement (x, y)
$\frac{\partial L}{\partial u}$:	Dérivée partielle de l'intégrale de l'expression de la fonctionnelle J par rapport à u
δ :	Distribution de Dirac
G_t :	Transformée de Fourier de de l'image I_t
χ :	Spectre de puissance croisé normalisé

d :	Transformée de Fourier inverse de χ
$S(x, y)$:	Fonction d'autocorrélation également appelée différence carrée additionnée
M :	Tenseur de structure
λ_{min} :	Plus petite valeur propre du tenseur de structure
λ_1, λ_2 ou m_{11}, m_{22} :	Valeurs propres du tenseur de structure
R :	Réponse de Harris
k :	Constante de la réponse de Harris déterminée empiriquement
$X = (x, y, \sigma)$:	Point clé défini d'une part par ses coordonnées sur l'image (x et y) et d'autre part par son facteur d'échelle caractéristique (σ)
$L(x, y, \sigma)$:	Convolution d'une image I par un filtre gaussien G
$G(x, y, \sigma)$:	Filtre gaussien
$D(x, y, \sigma)$:	Différences de gaussiennes
\hat{x} :	Position précise de l'extremum d'un point clé
H :	Matrice hessienne composée des différentes dérivées de la différence gaussienne de $D(x, y, \sigma)$
D_{xx} :	Dérivée seconde de la différence de gaussienne par rapport à x
D_{yy} :	Dérivée seconde de la différence de gaussienne par rapport à y
D_{xy} :	Dérivée seconde de la différence de gaussienne par rapport à x et y
θ :	Orientation de pixel
r_{th} :	Seuil de la réponse de la matrice hessienne

Liste des abréviations

2D :	Two-Dimensional (deux dimensions)
3D :	Three-Dimensional (trois dimensions)
BM:	Block Matching
BS:	Binary Search
CSA:	Cross Search Algorithm
DoG:	Difference of Gaussians
DS :	Diamond Search
ECMA :	Equation de Contrainte du Mouvement Apparent
ES:	Exhaustive Search
FSS:	Four Step Search
H&S:	Horn & Shunck
HSA:	Hierarchical Search Algorithm
L&K:	Lucas & Kanade
MAD:	Mean Absolute Difference
MDB :	Mesure de Distorsion de Bloc
MSE:	Mean Square Error
OSA:	Orthogonal Search Algorithm
RAM:	Random Access Memory

SIFT:	Scale-Invariant Feature Transform
SPCN :	Spectre de Puissance Croisé Normalisé
SS:	Spiral Search
SSD:	Sum Square Difference
TDL:	Two-Dimensional Logarithmic Search
TF:	Transformée de Fourier
TSS:	Three Step Search

INTRODUCTION

GENERALE

La vision par ordinateur est une branche de l'intelligence artificielle dont le but est de permettre à une machine de comprendre ce qu'elle « voit » lorsqu'on la connecte à une caméra. Cela peut se faire par l'analyse d'une séquence vidéo.

Dans le cadre de cette analyse, les objets peuvent être caractérisés par leurs cohérences spatiale et temporelle. Par conséquent, l'analyse dynamique peut fournir des primitives intéressantes pour la compréhension de la scène. Le mouvement apparent apparaît notamment comme une caractéristique ayant un fort potentiel discriminant. De ce fait, l'estimation de cette primitive est un enjeu crucial pour de nombreuses applications nécessitant la connaissance de l'évolution du contenu de la scène. En outre, cette primitive présente alors un intérêt fondamental pour le développement d'applications fondées sur la poursuite des objets. Le suivi de chaque objet permet d'interagir avec le contenu de la séquence vidéo. D'où l'objet de l'étude de notre mémoire qui concerne la détection et suivi d'objet mobile dans une séquence vidéo. C'est à dire la détection et suivi d'un ensemble de régions d'intérêt en mouvement dans une scène tridimensionnelle.

Pour ce faire nous avons divisé notre mémoire en trois chapitres :

Le premier chapitre établit un état de l'art sur l'estimation de mouvement par les méthodes classiques. Les généralités sont présentées au début du chapitre, ensuite les principales hypothèses et contraintes sont rappelées, puis la présentation des différentes méthodes termine ce chapitre.

Le deuxième chapitre concerne une étude comparative des méthodes d'estimation de mouvement tout en exposant ces dernières. Nous mettrons l'accent sur l'appariement par fenêtre (Block Matching) car elle caractérise l'approche retenue dans ce travail.

Le troisième chapitre présente la mise en œuvre d'un système d'acquisition et d'estimation du mouvement en temps réel. Pour ce faire, nous détaillerons les caractéristiques de notre algorithme et du matériel utilisé puis nous évaluerons les différents résultats obtenus.

La conclusion générale dressera un bilan de ce travail.

CHAPITRE 1 :

ETAT DE L'ART

1.1 Introduction

Avec la généralisation de l'utilisation d'image numérique, l'analyse du mouvement dans les vidéos s'est révélée être un outil indispensable pour des applications diverses.

Ce chapitre introductif est consacré aux descriptions de quelques méthodes usuelles d'estimation du mouvement à partir d'une séquence vidéo. Après une brève discussion sur le mouvement et les problèmes généraux de sa mesure, nous présenterons les principales techniques d'estimation imposées dans le domaine de la vision par ordinateur. La dernière partie du chapitre portera sur une étude comparative des différentes techniques entamées.

1.2 Définition d'une vidéo

La vidéo est une succession d'images animées défilant à une certaine cadence (environ 25 images par seconde) afin de créer une illusion de mouvement pour l'œil humain. Elle peut être analogique (signal continu d'intensité de luminance) ou numérique (suite de trames ou images). Une image numérique désigne toute image acquise, créée, traitée et stockée sous forme numérique (échantillonnée et quantifiée). Elle est composée d'une grille de points élémentaires appelés pixels. Une séquence d'images qui forme une vidéo est une série de N images du temps discrets $t_k = t_0 + k \cdot \Delta t$, où Δt est un intervalle de temps fixe, selon l'échantillonnage de la vidéo, et qui sépare chaque deux trames successives, et $k = 0, 1, \dots, N-1$. [1]

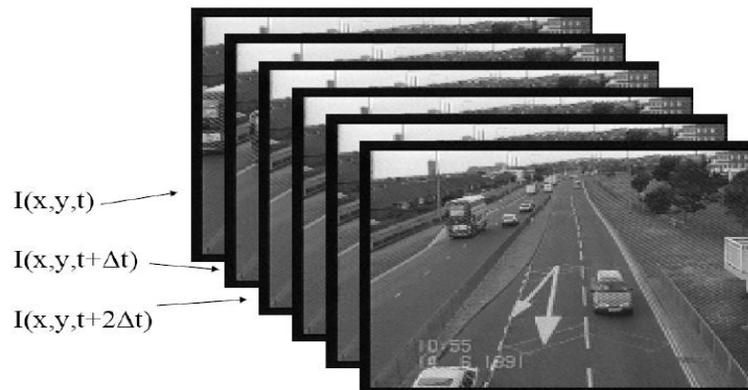


Figure 1.1 : Séquence d'images. [1]

1.3 Mouvement dans une séquence vidéo

Le mouvement est un réel problème en vidéo puisqu'il décrit un contexte en trois dimensions (3D) alors que les images sont une projection de scènes 3D sur un plan bidimensionnel comme montré dans la figure (1.2). [2]

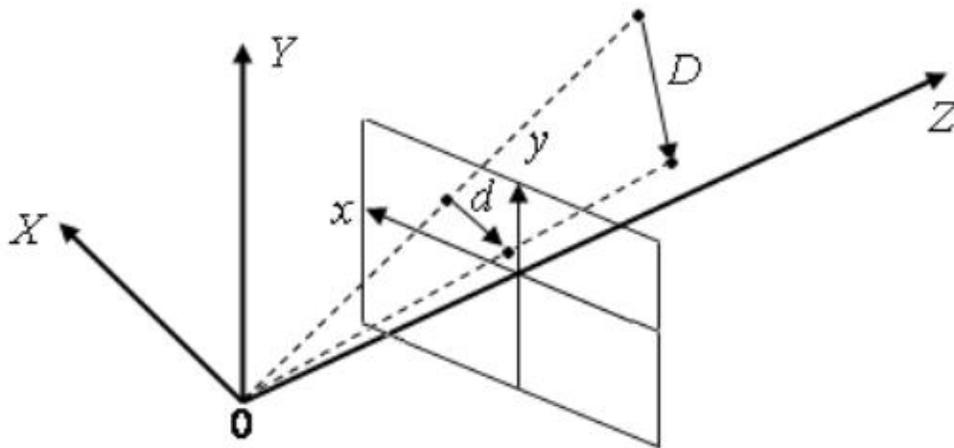


Figure 1.2 : Projection du mouvement 3D en 2D. [2]

Les mouvements sont causés par le déplacement d'un objet dans la scène ou le déplacement de la caméra. Deux types de mouvements sont distingués : on a le mouvement réel et mouvement apparent. [3]

1.3.1 Mouvement réel

C'est le mouvement réel dans la scène 3D, c'est celui observé par l'œil humain ou par la caméra. [2]

1.3.2 Mouvement apparent

Lors de la projection du mouvement réel d'une scène tridimensionnelle sur un plan bidimensionnel, on perd une dimension (perte d'informations). Le mouvement apparent est un mouvement 2D perçu dans une séquence à travers des variations spatio-temporelles de la luminance. Le champ des vitesses apparentes porte le nom de flot optique. Le flot optique est une variable cachée et n'est accessible que par l'analyse des variations temporelles de la séquence d'images. On peut donc définir le flot optique comme le champ de vitesse (ou de

déplacement) produit dans le plan image par des objets en mouvement dans un espace 3D. Il est représenté par des vecteurs comme dans la figure (1.3). [2,3]

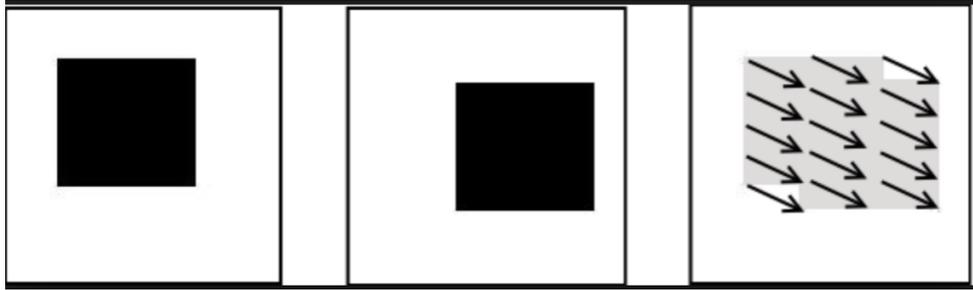


Figure 1.3: Exemple de flot optique. [4]

1.3.3 Equation du flop optique

Une séquence d'images peut être représentée par sa fonction d'intensité $I(x, y, t)$. L'hypothèse de conservation de l'intensité stipule que l'intensité d'un point physique (x, y) de la séquence d'images ne varie pas au cours du temps [2,5], c'est à dire :

$$I(x, y, t) = I(x + dx, y + dy, t + dt) \quad (1.1)$$

Nous pouvons développer le second membre de l'équation (1.1) en séries de Taylor à l'ordre 1 autour de (x, y, t) . [2]

$$I(x + dx, y + dy, t + dt) = I(x, y, t) + \frac{\partial I}{\partial x} dx + \frac{\partial I}{\partial y} dy + \frac{\partial I}{\partial t} dt \quad (1.2)$$

En remplaçant $I(x + dx, y + dy, t + dt)$ par sa valeur dans l'équation (1.1) on obtient l'équation (1.3) :

$$\frac{\partial I}{\partial x} dx + \frac{\partial I}{\partial y} dy + \frac{\partial I}{\partial t} dt = 0 \quad (1.3)$$

En dérivant l'équation (1.3) par rapport à t on obtient l'équation (1.4) :

$$\frac{\partial I}{\partial x} \frac{dx}{dt} + \frac{\partial I}{\partial y} \frac{dy}{dt} + \frac{\partial I}{\partial t} = 0 \quad (1.4)$$

On note :

- $u = \frac{dx}{dt}$: la vitesse apparente suivant la direction x,
- $v = \frac{dy}{dt}$: la vitesse apparente suivant la direction y,
- $\nabla I = \left[\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y} \right] = [I_x, I_y]$ est le gradient de l'image au pixel (x, y) à l'instant t et I_t est la dérivée de l'intensité d'un pixel par rapport au temps. L'équation (1.4) devient :

$$I_x u + I_y v + I_t = 0 \quad (1.5)$$

Cette équation (1.5) est appelée « équation de contrainte du flot optique ou équation de contrainte du mouvement apparent » couramment notée ECMA. [2,5]

1.3.4 Domaines d'utilisation du flot optique

Le flot optique est utilisé dans différents domaines :

- ❖ Compression vidéo
- ❖ Segmentation : météorologie
- ❖ L'imagerie médicale
- ❖ La robotique
- ❖ Le domaine militaire
- ❖ Surveillance du trafic routier.... [2]

Nous nous demanderons donc :

« COMMENT PERMETTRE A LA MACHINE D'ANALYSER, TRAITER ET COMPRENDRE LE MOUVEMENT APPARENT DES OBJETS DANS UNE SCENE TRIDIMENSIONNELLE ? »

1.3.5 Contraintes et problèmes du mouvement

Le changement de luminance est perçu comme un mouvement dans la scène, alors que cela ne correspond pas toujours à un mouvement d'objet, on doit donc supposer que [2] :

- ✚ Les objets sont des corps rigides, c'est-à-dire qu'on doit négliger la déformation de ceux-ci pour des trames voisines,
- ✚ L'intensité de l'objet est constante dans le mouvement,
- ✚ Le mouvement de la caméra influe sur l'appréciation du mouvement de l'objet on va donc supposer que la caméra est fixe,

Aussi les différents problèmes que l'on peut rencontrer sont :

- ✚ Le problème d'occlusion dû à l'apparition ou aux croisements d'objets dans la vidéo (un ou plusieurs objets mobiles)
- ✚ Le problème d'ouverture lorsque l'information apportée par les variations de l'intensité lumineuse contenues dans le voisinage ne permet pas de retrouver l'objet correspondant dans l'image suivante.

Cependant, bien que de nombreuses méthodes d'estimation du mouvement existent, aucune méthode universelle n'a été élaborée pour permettre de déterminer le mouvement dans toutes les situations et pour tous les types d'objets en mouvement. [2, 5, 6]

1.4 Temps d'acquisition

Le temps d'acquisition ne dépend pas de la distance entre l'objet et la caméra et des paramètres de la caméra. Il est donc ainsi possible de calculer le temps, τ , nécessaire à l'objet pour atteindre la caméra seulement à partir d'informations images comme le montre la figure (1.4). C'est-à-dire sans connaître la taille réelle de l'objet ou sa vitesse dans l'espace 3D. Cela signifie que pour estimer un mouvement, on n'a pas besoin de posséder les informations géométriques de l'acquisition (distance de l'objet par rapport à la caméra, taille de l'objet, vitesse d'acquisition...etc.). Cela favorise les algorithmes d'estimation de mouvement à être

génériques ce qui n'est pas le cas pour d'autres domaines de traitement d'image telle que la reconstruction 3D des scènes. [7]

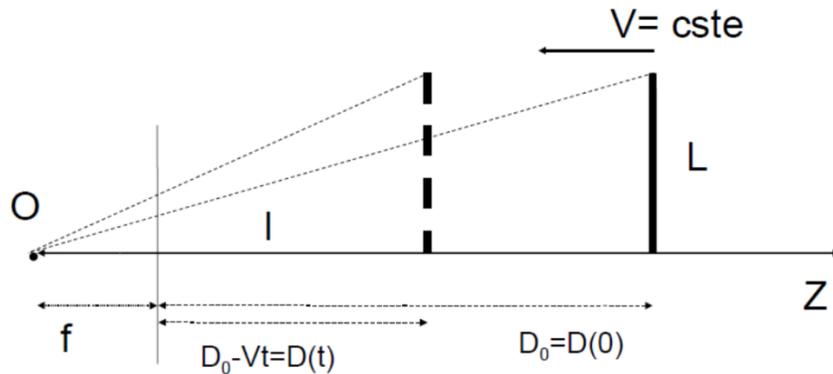


Figure 1.4: Temps d'acquisition. [7]

Soit L la taille de l'objet, l sa taille apparente, D la distance entre l'objet et le plan optique(O), f la distance entre le plan optique et le plan de projection.

A partir du théorème de Thalès, la taille apparente (c'est la distance angulaire entre ses points extrêmes au point d'observation) de la barre $l(t)$ à l'instant t sur le plan image s'écrit comme suit :

$$l(t) = f \frac{L}{D} \quad (1.6)$$

En effectuant la dérivée de l'équation (1.6) par rapport au temps on obtient l'équation (1.7)

$$l'(t) = \frac{dl(t)}{dt} = -f \frac{L}{D^2} \frac{dD}{dt} = f \frac{LV}{D^2} \quad (1.7)$$

On désigne τ comme étant le rapport entre l'équation (1.6) et l'équation (1.7), c'est ainsi qu'on obtient l'équation (1.8) suivante :

$$\tau = \frac{l(t)}{l'(t)} \quad (1.8)$$

On peut donc en déduire comme nous l'avons dit précédemment que le temps d'acquisition peut être calculé à partir d'information images.

1.5 Informations à extraire

Dans le traitement de mouvement, nous avons 3 niveaux conceptuels selon la quantité d'informations à extraire que nous pouvons observer dans la Figure (1.5). [7]

Tout d'abord le modèle non rigide qui concerne les objets déformables. L'inconvénient de ce modèle est que la déformation de l'objet ne peut être considéré comme un mouvement que pour certaine application (battement de cœur...). Ensuite le calcul bas niveau qui a pour but d'extraire l'information utile et pertinente contenue dans l'image en regard de l'application considérée (absence ou présence de mouvement). Le dernier modèle est le modèle rigide c'est celui qui va nous intéresser. Il représente le modèle du solide indéformable, c'est à dire la qualité d'un objet ferme qui ne varie pas d'apparence comme les objets que nous voulons détecter et suivre dans notre mémoire.

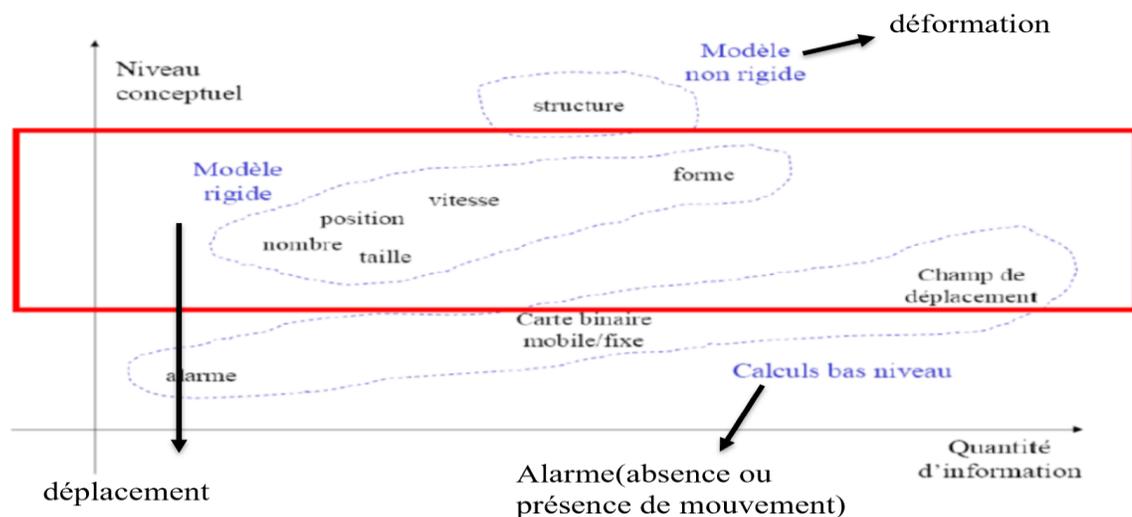


Figure 1.5: Schéma conceptuel de traitement de mouvement. [7]

L'information à extraire est le champ de déplacement c'est-à-dire le flot optique présent dans le modèle rigide. Ce qui se traduit en termes de traitement d'images par l'estimation du mouvement.

1.6 Méthodes d'estimation du mouvement

L'estimation du mouvement consiste à estimer, à partir d'une séquence d'images, le mouvement apparent des objets composant une séquence tridimensionnelle. Les différentes techniques d'estimation du mouvement peuvent être définies en deux catégories : les méthodes basées sur le pixel dites directes et celles basées sur les caractéristiques (points d'intérêt) de l'image dites indirectes [8,9]. Ces différentes méthodes sont brièvement décrites dans ce paragraphe et seront plus détaillées dans le chapitre suivant.

1.6.1 Méthodes directes

Les méthodes directes sont définies comme des méthodes qui récupèrent directement les paramètres inconnus (les champs de déplacement) pour des quantités d'image mesurables à chaque pixel de l'image. Ces méthodes minimisent une mesure d'erreur basée sur la collecte directe d'informations d'image de tous les pixels de l'image. Elles peuvent être divisées en trois parties qui sont : les méthodes de mise en correspondance, les méthodes fréquentielles et les méthodes différentielles. [2, 8,10, 11]

1.6.1.1 Méthodes différentielles

Les méthodes différentielles calculent la vitesse à partir de dérivées spatiales et temporelles d'intensité lumineuse des pixels. Celles-ci sont des méthodes basées sur l'hypothèse de conservation de l'intensité des pixels au cours du mouvement dans une séquence d'images. [2]

Parmi les méthodes différentielles existantes, on peut citer les méthodes de Horn & Shunck (H&S) [12], de Lucas & Kanade (L&K) [13], de Nagel [14]. Le principe de ces méthodes est le même quelle que soit la technique employée. Elles sont basées toujours sur l'équation générale du flot optique pour estimer le mouvement dans la scène. [2]

1.6.1.2 Méthodes de mise en correspondance

Lorsque les déplacements considérés dans les images sont trop grands, typiquement plus que deux pixels par image, ou que le calcul des dérivées spatio-temporelles est sujet à des erreurs numériques, il convient de se tourner vers des méthodes dites de mise en correspondance de blocs ou appariement de blocs ou Block Matching (BM). Ces méthodes consistent à appairer (Matching) chaque région R_i (ou bloc) d'une image au temps t à une autre région R'_i au temps

$t + 1$ et ce en trouvant le déplacement $d = (dx, dy)$ qui minimise une fonction d'erreur et maximise la similarité entre les deux régions.

De nombreux algorithmes d'exploration ont été développés, le plus simple étant l'exploration de tous les blocs de la fenêtre de recherche ("Full Search" ou "Exhaustive Search").[15]

1.6.1.3 Méthodes fréquentielles

Les méthodes fréquentielles d'estimation de vitesse dans les séquences d'images sont fondées sur une caractérisation du mouvement dans le domaine des fréquences. Les fréquences spatio-temporelles sont mises en relation avec la vitesse du stimulus de mouvement, et le flot optique devient l'identification d'un plan d'énergie dans l'espace de la fréquence spatio-temporelle. En général, les mécanismes sensibles au mouvement fondés sur l'énergie orientée spatio-temporelle dans l'espace de Fourier peuvent estimer le mouvement dans les endroits où les autres approches échouent. Ainsi, la détection du mouvement dans l'image revient ici à extraire l'orientation spatio-temporelle. On distingue plusieurs sous-approches telles que les méthodes exploitant la phase du signal. [16]

1.6.2 Méthodes indirectes

Les méthodes indirectes extraient d'abord un ensemble épars de caractéristiques distinctes de chaque image séparément, puis récupèrent et analysent leurs correspondances afin de déterminer le mouvement et la forme. Elles minimisent une mesure d'erreur basée sur les distances entre quelques caractéristiques correspondantes. Nous pouvons citer comme méthodes le détecteur de coin de Harris et la méthode SIFT (Scale Invariant Feature Transform). [8, 9, 17]

1.7 Conclusion

Dans ce chapitre, nous avons présenté un état de l'art de l'estimation de mouvement. Nous avons exposé les principales hypothèses et contraintes dans l'estimation des vecteurs de mouvement. À la fin de ce chapitre, différentes méthodes d'estimation de mouvement ont été présentées de manière non exhaustive à savoir les méthodes différentielles, les méthodes de mise en correspondance de blocs, les méthodes fréquentielles et les méthodes indirectes basées

sur la détection de caractéristiques. On peut observer que certaines techniques opèrent dans le domaine spatio-temporel, d'autres dans le domaine fréquentiel. Ces méthodes feront l'objet de notre étude dans le chapitre 2.

CHAPITRE 2 :
Etude comparative
entre différentes
méthodes
d'estimation de
mouvement

2.1 Introduction

Par définition, le flux optique est le champ de vitesse (ou de déplacement), produit dans le plan image par des objets en mouvement dans un espace 3D. La plupart des méthodes existantes pour l'estimation du flot optique peuvent être classées en quatre catégories : les méthodes différentielles, les méthodes de mise en correspondance, les méthodes indirectes et les méthodes fréquentielles (basées sur l'énergie et sur la phase).

Dans ce chapitre, nous allons décrire ces méthodes citées plus haut. Ainsi, nous mettrons en reliefs les biais et avantages de chacune d'elles. Partant de cela et vu le contexte et le but de ce travail qui doit mener à une application d'estimation du mouvement en temps réel, nous déciderons à la fin du chapitre d'opter pour une de ces méthodes qui sera l'objet du chapitre suivant.

2.2 Méthodes de mise en correspondance ou Block Matching

2.2.1 Principe de fonctionnement

Le principe de fonctionnement pour ces méthodes est basé sur la comparaison entre l'image courante et l'image de référence comme illustré dans la figure (2.1). Chaque image est subdivisée en blocs de taille égale. Nous faisons l'hypothèse que le mouvement des pixels est uniforme à l'intérieur d'un bloc. L'algorithme consiste, pour un bloc de l'image courante, à choisir un bloc dans l'image de référence et à calculer un critère de comparaison entre ces deux blocs, tout en supposant que les pixels du même bloc ayant le même mouvement translationnel auront ainsi le même vecteur de mouvement. [18, 19]

L'opération est répétée en choisissant un autre bloc jusqu'à ce que tous les blocs d'une zone déterminée de l'image de référence (appelée fenêtre de recherche) aient été testés ou jusqu'à ce qu'un critère d'arrêt arbitraire est abouti. Le bloc le plus semblable est ainsi identifié dans l'image de référence pour chaque bloc de l'image courante. Nous obtenons de cette manière pour tout bloc un vecteur de déplacement caractéristique du mouvement de ce dernier. La taille de la fenêtre dépend d'un déplacement maximal autorisé fixé par l'utilisateur.[18, 19]

Les méthodes de Block Matching peuvent être classées en fonction :

- ❖ Du critère de mise en correspondance (type de fonction de cout).
- ❖ De la dimension du bloc et de celle de sa fenêtre de recherche.
- ❖ De la stratégie de recherche (exhaustive ou autre). [18]

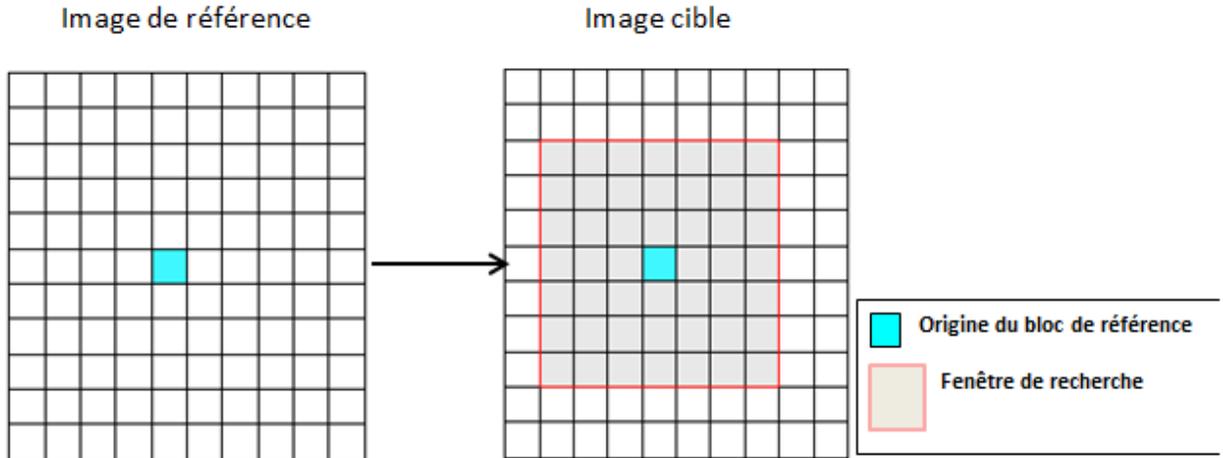


Figure 2.1: Block Matching. [18]

2.2.2 Largeur de la zone de recherche

L'utilisation d'une fenêtre de recherche permet de limiter le nombre de blocs de référence que l'algorithme doit tester. La largeur de la zone de recherche affecte directement la vitesse et la précision de l'encodeur vidéo. Dans la plupart des standards, c'est un paramètre limité à partir des expériences ou imposé par les contraintes des implémentations hard [20]. Cependant, les séquences de mouvement rapide, où les objets changent leurs positions d'une image à l'autre d'une distance considérable, doivent subir une recherche dans une zone couvrant ce déplacement, engendrant bien sûr un temps de calcul plus long. Par contre, les séquences de mouvement lent doivent minimiser la zone de recherche pour profiter du temps de codage. [2, 18]

2.2.3 Taille des blocs

La taille du bloc choisie affecte clairement la recherche et la comparaison des blocs et ainsi la performance de l'estimation de mouvement. En choisissant une taille importante des blocs on réduit la précision des vecteurs du mouvement obtenus du fait que les grands blocs

contiennent des objets de mouvements différents en vitesse et en direction. Mais d'un autre côté, on gagne en matière de taille des informations émises concernant les vecteurs de mouvement puisque le nombre de blocs est réduit. Par contre, le choix d'une taille réduite permet d'explicitier le mouvement des objets et les limites des objets seront bien identifiées. Il permet aussi de produire des vecteurs de mouvement plus précis. D'un autre côté, les petits blocs engendrent un temps de calcul important. [19]

2.2.4 Types de fonction de coût

C'est la fonction principale du BM, qui décide à quel point un bloc est similaire à un autre (elle est appelée aussi Mesure de Distorsion de Bloc (MDB)). Son choix est nécessaire pour obtenir un bloc optimal. Elle a aussi un impact direct sur la complexité de calcul et sur la précision de l'estimation de mouvement. Plusieurs fonctions sont rencontrées dans la littérature dont les plus utilisées sont [18,19] :

✚ Différence moyenne absolue (Mean Absolute Difference (MAD))

$$MAD = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |I_C(x+i, y+j) - I_r(x+i+dx, y+j+dy)| \quad (2.1)$$

Avec I_C l'image cible et I_r l'image de référence.

✚ Erreur quadratique moyenne (Mean Square Error (MSE))

$$MSE = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (I_C(x+i, y+j) - I_r(x+i+dx, y+j+dy))^2 \quad (2.2)$$

Avec I_C l'image cible et I_r l'image de référence.

2.2.5 Algorithmes de Block Matching

Pour trouver un bloc cible dans une image de référence, l'algorithme parcourt l'image bloc par bloc dans un certain ordre. De nombreuses manières existent et chacune possède généralement un ordre de parcours différent. [2,19]

Voici une liste des algorithmes d'estimation de mouvement utilisés dans les normes de codage vidéo.

- Exhaustive Search (ES)
- Cross Search Algorithm (CSA)
- Spiral Search (SS)
- Three Step Search (TSS)
- Two-Dimensional Logarithmic Search (TDL)
- Binary Search (BS)
- Four Step Search (FSS)
- Orthogonal Search Algorithm (OSA)
- Hierarchical Search Algorithm (HSA)
- Diamond Search (DS)

Nous nous contentons de décrire la recherche exhaustive (*Exhaustive Search*).

2.2.5.1 Recherche exhaustive (ES)

Cet algorithme calcule la fonction de cout à chaque emplacement possible dans la fenêtre de recherche. Cela conduit à la meilleure correspondance possible du macrobloc (unité de traitement de formats de compression d'images et de vidéos basée sur des transformations de blocs linéaires) dans le cadre de référence avec un bloc dans un autre cadre. Cependant, il s'agit de l'algorithme d'appariement de blocs le plus long en termes de calcul. A noter qu'une fenêtre de recherche plus grande nécessite un plus grand temps de calcul. [19]

2.3 Méthodes différentielles

L'approche différentielle réalise l'estimation du mouvement en mettant en relation les dérivées spatiales et temporelles des images d'une séquence. On peut diviser les méthodes différentielles en deux groupes : méthodes locales et méthodes globales.[2]

2.3.1 Méthodes locales

Les méthodes à contraintes locales estiment le mouvement en résolvant un groupe d'équations du flot optique, comme étant un système d'équations définies sur un voisinage local (petite région). La contrainte peut agir différemment d'un pixel à un autre d'une manière indépendante. Généralement, ces méthodes font appel à des équations multi contraintes. La méthode la plus connue est celle de Lucas & Kanade (L&K). [13,21]

2.3.1.1 Principe de fonctionnement

La méthode de L&K suppose que le déplacement d'un point de l'image entre deux instants consécutifs est petit et approximativement constant dans un voisinage du point p . L'équation du flot optique peut être supposée vraie pour tous les pixels dans une fenêtre centrée au point p . Le vecteur vitesse local (u, v) doit satisfaire l'équation suivante (2.3) [13,21] :

$$\begin{aligned}
 I_x(q_1)u + I_y(q_1)v &= -I_t(q_1) \\
 I_x(q_2)u + I_y(q_2)v &= -I_t(q_2) \\
 &\cdot \\
 &\cdot \\
 &\cdot \\
 I_x(q_n)u + I_y(q_n)v &= -I_t(q_n)
 \end{aligned}
 \tag{2.3}$$

Avec q_i les pixels de la fenêtre centrée et $i = 1$ à n correspondant au nombre de pixels.

Le système a plus d'équations que d'inconnues et est donc surdéterminé. La méthode de Lucas & Kanade apporte une solution par la méthode des moindres carrés. Elle résout un système d'équations comme indiqué dans l'équation (2.4) [13,21] :

$$A^T A V = A^T B \quad \text{ou} \quad V = (A^T A)^{-1} A^T B \quad (2.4)$$

$$\text{Avec } A = \begin{bmatrix} I_x(q_1) & I_y(q_1) \\ I_x(q_2) & I_y(q_2) \\ \vdots & \vdots \\ I_x(q_n) & I_y(q_n) \end{bmatrix}, V = \begin{bmatrix} u \\ v \end{bmatrix} \text{ et } B = \begin{bmatrix} -I_t(q_1) \\ -I_t(q_2) \\ \vdots \\ -I_t(q_n) \end{bmatrix}$$

Où A^T est la matrice transposée de la matrice A .

Ainsi, on calcule le champ de déplacement $V = \begin{bmatrix} u \\ v \end{bmatrix}$

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \sum_i I_x(q_i)^2 & \sum_i I_x(q_i)I_y(q_i) \\ \sum_i I_x(q_i)I_y(q_i) & \sum_i I_y(q_i)^2 \end{bmatrix}^{-1} \begin{bmatrix} -\sum_i I_x(q_i)I_t(q_i) \\ -\sum_i I_y(q_i)I_t(q_i) \end{bmatrix} \quad (2.5)$$

Avec les sommes allant de $i = 1$ à n .

La matrice $A^T.A$ est appelée tenseur de structure de l'image au point p , également appelé matrice du second moment, c'est une matrice dérivée du gradient de l'image. [13,21]

2.3.1.2 Fenêtre pondérée

La solution donnée ci-avant (équation 2.5) donne la même importance à tous les n pixels q_i de la fenêtre. En pratique, il est préférable de donner un poids plus important aux pixels qui sont proches du pixel p . Pour cela, on utilise la version pondérée de l'équation des moindres carrés.[13]

$$A^T W A V = A^T W B \quad \text{ou} \quad V = (A^T W A)^{-1} A^T W B \quad (2.6)$$

W est une matrice diagonale $n \times n$ contenant les poids $W_{ii} = w_i$ associé à l'équation du pixel q_i .

Alors, le calcul est le suivant :

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \sum_i w_i I_x(q_i)^2 & \sum_i w_i I_x(q_i) I_y(q_i) \\ \sum_i w_i I_x(q_i) I_y(q_i) & \sum_i w_i I_y(q_i)^2 \end{bmatrix}^{-1} \begin{bmatrix} -\sum_i w_i I_x(q_i) I_t(q_i) \\ -\sum_i w_i I_y(q_i) I_t(q_i) \end{bmatrix} \quad (2.7)$$

2.3.2 Méthodes globales

Les méthodes globales consistent à résoudre un problème d'optimisation (local ou global) en minimisant une fonctionnelle, généralement basée sur l'équation du flot optique à laquelle on ajoute une contrainte pour particulariser les solutions. La méthode que nous allons présenter est la méthode de Horn & Shunck (H&S). [6, 12]

2.3.2.1 Principe de fonctionnement

La méthode de H&S minimise simultanément l'équation de contrainte et les variations spatiales du champ de vitesse. La variation pixel à pixel du champ de vitesse peut être quantifiée par la somme des carrés des normes de gradients spatiaux, de chaque composante du vecteur vitesse. Avec cette contrainte, le vecteur vitesse est obtenu par minimisation du critère suivant [6, 12] :

$$J = \iint \left[(I_x u + I_y v + I_t)^2 + \alpha^2 \left(\|\vec{\nabla} u(x, y)\|^2 + \|\vec{\nabla} v(x, y)\|^2 \right) \right] dx dy \quad (2.8)$$

Où l'équation de contrainte du flot optique est donnée par l'équation (2.9) et les variations spatiales du champ de vitesse par l'équation (2.10) :

$$e_c = I_x u + I_y v + I_t \quad (2.9)$$

$$e_s = \|\vec{\nabla} u(x, y)\|^2 + \|\vec{\nabla} v(x, y)\|^2 \quad (2.10)$$

Avec

$$\|\vec{\nabla}u(x, y)\|^2 = \left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2 \text{ et } \|\vec{\nabla}v(x, y)\|^2 = \left(\frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2 \quad (2.11)$$

α^2 , le facteur de pondération qui contrôle la robustesse de la contrainte de lissage. Ce facteur est prépondérant (qui a plus de poids) dans les zones où les gradients de la luminance sont faibles. Pour trouver les valeurs appropriées de u et v , on minimise la fonctionnelle J en résolvant les équations (2.12) multidimensionnelles d'Euler-Lagrange associées. [6, 12]

$$\begin{aligned} \frac{\partial L}{\partial u} - \frac{\partial}{\partial x} \frac{\partial L}{\partial u_x} - \frac{\partial}{\partial y} \frac{\partial L}{\partial u_y} &= 0 \\ \frac{\partial L}{\partial v} - \frac{\partial}{\partial x} \frac{\partial L}{\partial v_x} - \frac{\partial}{\partial y} \frac{\partial L}{\partial v_y} &= 0 \end{aligned} \quad (2.12)$$

où L est l'intégrale de l'expression de la fonctionnelle J , donnant :

$$\begin{aligned} I_x(I_x u + I_y v + I_t) - \alpha^2 \Delta u &= 0 \\ I_y(I_x u + I_y v + I_t) - \alpha^2 \Delta v &= 0 \end{aligned} \quad (2.13)$$

Où les indices représentent à nouveau une différenciation partielle et $\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$ désigne l'opérateur de Laplace. En pratique, le Laplacien est approché numériquement à l'aide de différences finies et peut être écrit :

$$\Delta^2 u(x, y) = \bar{u}(x, y) - u(x, y) \quad (2.14)$$

Où $\bar{u}(x, y)$ est une moyenne pondérée de u calculé dans un voisinage autour du pixel à l'emplacement (x, y) . En utilisant cette notation, le système d'équation (2.13) peut être écrit :

$$\begin{aligned}
(I_x^2 + \alpha^2)u + I_x I_y v &= \alpha^2 \bar{u} - I_x I_t \\
I_x I_y u + (I_y^2 + \alpha^2)v &= \alpha^2 \bar{v} - I_y I_t
\end{aligned}
\tag{2.15}$$

Le système d'équation (2.15) est linéaire dans u et v et peut être résolu pour chaque pixel de l'image. Cependant, comme la solution dépend des valeurs voisines du champ de flux, elle doit être répétée une fois que les voisins ont été mis à jour comme dans l'équation (2.16).

$$\begin{cases}
u^{(k+1)} = \left(\bar{u}^k - I_x \cdot \frac{I_x \bar{u}^{(k)} + I_y \bar{v}^k + I_t}{\alpha^2 + (I_x)^2 + (I_y)^2} \right) \\
v^{(k+1)} = \left(\bar{v}^k - I_y \cdot \frac{I_x \bar{u}^{(k)} + I_y \bar{v}^k + I_t}{\alpha^2 + (I_x)^2 + (I_y)^2} \right)
\end{cases}
\tag{2.16}$$

Où l'indice supérieur $k + 1$ indique la prochaine itération à calculer et k le dernier résultat calculé. [12,21]

Avec :

$$\begin{aligned}
\bar{u}(i, j) &= \frac{1}{6} [u(i-1, j) + u(i, j+1) + u(i+1, j) + u(i, j-1)] + \\
&\frac{1}{12} [u(i-1, j-1) + u(i-1, j+1) + u(i+1, j-1) + u(i+1, j+1)]
\end{aligned}
\tag{2.17}$$

Et

$$\begin{aligned}
\bar{v}(i, j) &= \frac{1}{6} [v(i-1, j) + v(i, j+1) + v(i+1, j) + v(i, j-1)] + \\
&\frac{1}{12} [v(i-1, j-1) + v(i-1, j+1) + v(i+1, j-1) + v(i+1, j+1)]
\end{aligned}
\tag{2.18}$$

2.3.3 Filtre de Kalman méthode différentielle évoluée

Le filtre de Kalman est un filtre à réponse impulsionnelle infinie qui estime les états d'un système dynamique à partir d'une série de mesures incomplètes ou bruitées. Le filtre a été nommé d'après le mathématicien et informaticien américain d'origine hongroise Rudolf Kalman. Il est utilisé dans une large gamme de domaines technologiques (radar, vision électronique, communication ...). Son appellation « filtre » vient du fait qu'il permet, entre autres, d'estimer un signal bruité (bruit blanc), en séparant le signal de son bruit. Ce filtre est utilisé en trajectographie pour estimer une trajectoire comme le cas de Poursuite de cibles (avions, missiles, radars...) en navigation pour localiser un engin, ou encore en balistique pour estimer des conditions initiales inconnues. [22]

2.3.3.1 Suivi de mouvement par filtre de Kalman

Dans le processus du suivi du mouvement, le filtre de Kalman observe un objet quand il se déplace, c'est-à-dire qu'il prend les informations sur l'état de l'objet à ce moment. Puis il utilise ces informations pour prédire où se trouve l'objet dans la prochaine trame. La position de l'objet est corrigée ensuite en considérant la prédiction et aussi l'observation. L'information extraite de la prédiction peut remplacer celle de l'observation lorsque cette dernière n'est pas accessible. [22]

2.3.3.2 Principe du filtre de Kalman

Le filtre de Kalman donne l'estimation d'un processus en utilisant un système à retour sous la forme de mesures (bruités). De cette manière, les équations du filtre de Kalman se regroupent en deux types d'équations :

- Les équations de prédiction (extrapolation).
- Les équations de mise à jour (correction).

Les équations d'extrapolation sont responsables de propager les estimées de l'état présent et les covariances d'erreur pour obtenir les estimées à priori de la prochaine étape. Alors que les équations de correction sont responsables du retour, c'est à dire, introduire une nouvelle mesure avec l'estimée à priori afin d'obtenir une estimée à posteriori améliorée. [22]

Le filtre de Kalman est donc un filtre prédictor-correcteur pour la résolution des problèmes numériques :

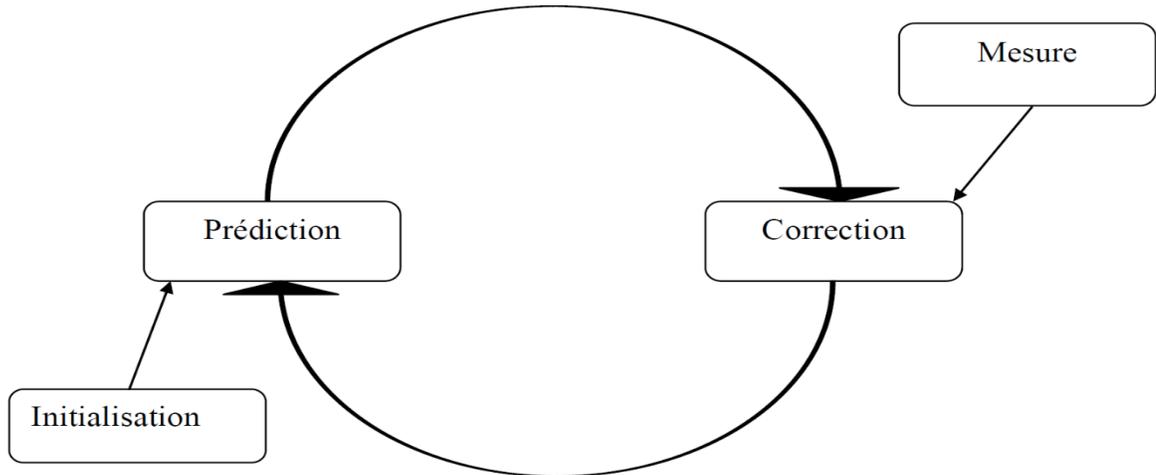


Figure 2. 2: Filtre de Kalman.[22]

2.4 Méthodes fréquentielles

Ces méthodes découlent de l'analyse des séquences d'images par la transformée de Fourier (TF). Ainsi, elles sont basées sur deux hypothèses fondamentales. La première hypothèse est qu'il existe une fonction $I_0(x, y)$, représentant la fonction d'intensité spatiale de référence ($t=0$) telle que [7,18] :

$$I(x, y, t) = I_0(x - v_x(x, y, t), y - v_y(x, y, t)) \quad (2.18)$$

Où $(v_x(x, y, t), v_y(x, y, t))$ est le vecteur vitesse du pixel (x, y) , induit par le déplacement de l'objet en mouvement, au temps t . Cette équation signifie que les objets

conservent la même illumination au cours du temps t , c'est l'hypothèse de conservation de la luminance. [18]

La deuxième hypothèse consiste à considérer que les objets se déplacent suivant des translations uniformes, c'est-à-dire que $v_x(x, y, t) = v_x t$ et $v_y(x, y, t) = v_y t$. La fonction (2.18) devient [18] :

$$I(x, y, t) = I_0(x - v_x t, y - v_y t) \quad (2.19)$$

En calculant la transformée de Fourier spatiale de la séquence dans l'équation (2.19), nous obtenons l'équation :

$$I(f_x, f_y, f_t) = I_0(f_x, f_y) e^{-2i\pi(v_x f_x + v_y f_y)t} \quad (2.20)$$

Où $I_0(f_x, f_y)$ est la transformée de Fourier de $I_0(x, y)$. L'information sur le mouvement se trouve donc dans la phase de la transformée de Fourier spatiale d'une séquence. En effet, lorsqu'un objet se déplace avec une vitesse uniforme, seule la phase de sa transformée de Fourier spatiale évolue au cours du temps. [18]

Le calcul de la transformée de Fourier temporelle de la séquence conduit à :

$$I(f_x, f_y, f_t) = I_0(f_x, f_y) \delta(v_x f_x + v_y f_y + f_t) \quad (2.21)$$

Où δ représente la distribution de Dirac. Ainsi le spectre $I(f_x, f_y, f_t)$ a des valeurs nulles partout, sauf dans un plan passant par l'origine dont l'équation est [18] :

$$v_x f_x + v_y f_y + f_t = 0 \quad (2.22)$$

Cette équation peut aussi être vue comme la transformée de Fourier de l'ECMA. Estimer les paramètres de ce plan, appelé plan de mouvement, est donc équivalent à calculer les vitesses recherchées des objets en mouvement. Cela est équivalent à l'analyse de l'orientation dans l'espace fréquentiel. Les premières méthodes d'estimation du mouvement dans le domaine fréquentiel consistent donc à déterminer l'emplacement des plans de mouvement par des filtres spatio-temporels orientés. Les filtres de Gabor sont un exemple de filtres orientés. Le flux optique d'une séquence d'images peut être estimé grâce à un ensemble de ces filtres [18]. On distingue plusieurs types d'approches parmi lesquelles nous avons celles basées sur la phase (méthode de corrélation de phase...). Ces méthodes exploitent le fait qu'un décalage spatial entraîne un déphasage dans le domaine fréquentiel. [7, 23, 24]

2.4.1 Méthode de corrélation de phase

Cette technique estime la translation 2D entre chaque paire d'images en prenant, pour chacune d'elles, sa transformée de Fourier. Son principe repose sur la recherche de la différence de phases pour chaque fréquence et le calcul de la transformée de Fourier inverse. Si I_1 et I_2 sont les deux images de la scène se recouvrant partiellement et G_1, G_2 leurs transformées de Fourier respectives, alors la différence de phase entre ces deux images dans le domaine de Fourier est égale à leur spectre de puissance croisé normalisé (SPCN) exprimé par [2] :

$$e^{j(\theta_1 - \theta_2)} = \frac{|G_1| \cdot |G_2| e^{j(\theta_1 - \theta_2)}}{|G_1| \cdot |G_2|} = \frac{|G_1| e^{j(\theta_1)} \cdot |G_2| e^{j(-\theta_2)}}{|G_1| \cdot |G_2|} = \frac{G_1 G_2^*}{\|G_1 G_2\|} \quad (2.23)$$

D'autre part, la transformation entre les deux images est une simple translation exprimée par le vecteur $V = (u, v)$, soit :

$$I_2(p) = I_1(p + V) = I_1(p) \delta(p - V) \quad (2.24)$$

Où δ Est la fonction de Dirac.

La TF de cette équation est donnée par :

$$G_2(f) = G_1(f) * e^{j2\pi(fV)} \quad (2.25)$$

Ainsi, il est clair que la différence de $e^{j(\theta_1 - \theta_2)}$ entre les deux images n'est autre que $e^{j2\pi(fv)}$ pour chaque fréquence f .

À partir de l'équation (2.25), on trouve que la transformée de Fourier inverse de cette différence de phase (SPCN) produit la fonction de Dirac du vecteur de translation [2] :

$$d(p) = \delta(p - V) = F^{-1}(e^{j(\theta_1 - \theta_2)}) \quad (2.26)$$

Avec

$$\delta(p - V) = \delta(x - u, y - v)$$

Il est évident d'après l'équation (2.26), que pour calculer le déplacement 2D (translations suivant x et y), il suffit de chercher le pic de cette représentation spatiale (La fonction $d(p)$ présente en général un pic très net en $V = (u, v)$). [2]

L'algorithme de la méthode de corrélation de phase est le suivant :

- Calculer G_1 et G_2 les TF de I_1 et I_2 .
- Calculer χ le SPCN de G_1 et G_2 .
- Calculer d la TF inverse de χ .
- Rechercher V , le maximum de d .

Cette méthode ne renvoie pas un résultat dense (seuls les maximums locaux sont considérés), et elle donne (dans sa version initiale) des déplacements entiers. [2,7]

2.5 Méthodes indirectes

Dans les méthodes indirectes à contrario des méthodes directes, les paramètres du mouvement ne sont pas calculés directement à partir de l'information de la luminance mais plutôt selon l'estimateur du champ dense pour chaque région. Elles ne font qu'extraire un ensemble de caractéristiques distincts de chaque image. [9]

2.5.1 Détecteur de coin Harris

Le détecteur de coin Harris est un opérateur de détection de coin couramment utilisé dans les algorithmes de vision artificielle pour extraire les coins et en déduire les caractéristiques d'une image. La détection d'angle est utilisée pour la détection de mouvement, l'enregistrement d'images, le suivi vidéo, la reconnaissance d'objets, etc. Il a été introduit pour la première fois par Chris Harris et Mike Stephens en 1988. [17]

2.5.1.1 Coin

Un coin est un point dont le voisinage se trouve dans deux directions de bord dominantes et différentes. En d'autres termes, un coin peut être interprété comme la jonction de deux bords, un bord étant un changement soudain de la luminosité de l'image. Les coins sont des caractéristiques importantes de l'image et sont généralement désignés comme des points d'intérêt invariants de la translation, de la rotation et de l'éclairage. Bien que les angles ne représentent qu'un faible pourcentage de l'image, ils contiennent les fonctions les plus importantes pour la restauration des informations sur l'image. La figure suivante montre un schéma synoptique pour un coin. [17]

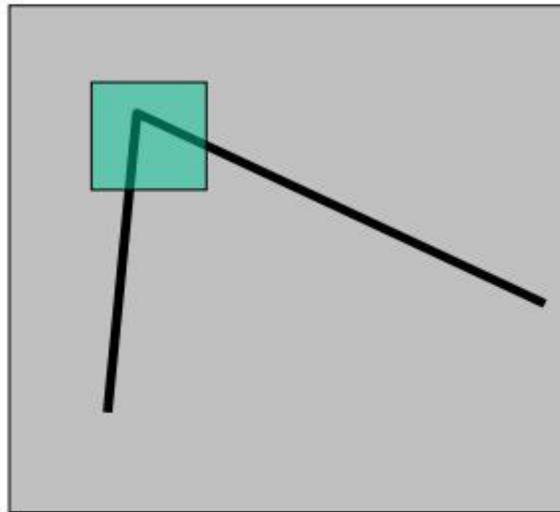


Figure 2. 3: Représentation d'un coin. [17]

2.5.1.2 Différenciation des bords et des coins

Le bord n'est rien d'autre que le côté ou la limite d'un objet. Pour un carré, ses quatre côtés sont représentés comme ses bords. Le point de rencontre de deux bords est un coin indiquant qu'un carré à quatre coins. Le schéma synoptique de la figure (2.4) suivante met en évidence la différence entre un coin et un bord. [17]

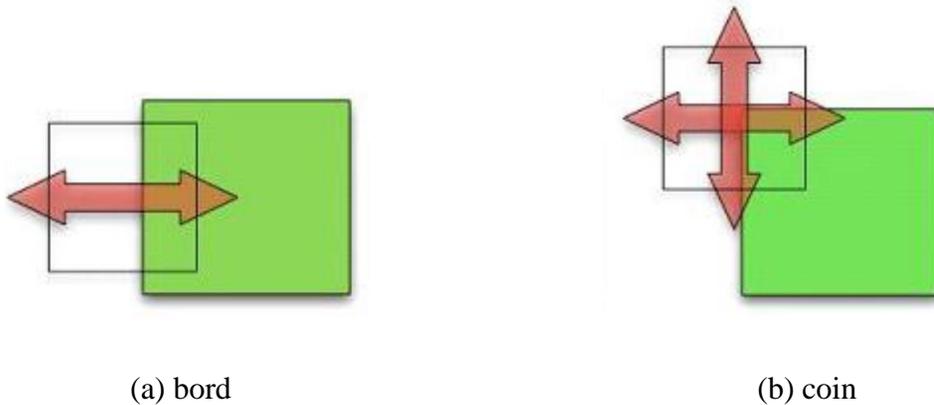


Figure 2. 4: Schéma synoptique différenciant un bord à un coin dans une image. (a) bord et (b) coin. [17]

2.5.1.3 Formulation

Le détecteur d'angle Harris est basé sur la fonction d'autocorrélation locale d'un signal. La fonction d'autocorrélation locale mesure les changements locaux du signal par des plages décalées dans différentes directions. L'algorithme du détecteur d'angle Harris consiste à :

- Trouver des dérivées partielles à partir de l'intensité d'une image
- Calculer la réponse du coin (R)
- Trouver les maxima locaux dans le coin réponse. [17]

Dans Harris, les dérivées x et y de détection des angles sont calculées pour une image, puis le produit de la dérivée est déterminé pour chaque pixel, ensuite La somme suivante du produit est calculée et enfin La matrice de Harris est définie à chaque pixel (x, y) . On pourra

donc calculer la réponse en angle et ainsi trouver les maxima locaux dans la réponse en angle.[17]

La fonction d'autocorrélation est également appelée différence carrée additionnée (SSD). Pour un point (x, y) , sa fonction d'autocorrélation est représentée par :

$$S(x, y) = \sum_{(x_k, y_k) \in W} (I(x_k, y_k) - I(x_k + u, y_k + v))^2 \quad (2.27)$$

$I(x + u, y + v, t)$ Peut-être approché par une expansion de Taylor. On aura I_x et I_y les dérivées partielles de I , tel que :

$$I(x + u, y + v) \approx I(x, y) + I_x(x, y)u + I_y(x, y)v \quad (2.28)$$

Cela produit l'approximation suivante :

$$S(x, y) = \sum_{(x, y) \in W} (I_x(x, y)u + I_y(x, y)v)^2 \quad (2.29)$$

Cette approximation (Equation 2.29) peut être écrite sous forme de matrice :

$$S(x, y) \approx (u \ v)M \begin{pmatrix} u \\ v \end{pmatrix} \quad (2.30)$$

Où M est le tenseur de structure,

$$M = \sum_{(x,y) \in W} \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} = \begin{bmatrix} \sum_{(x,y) \in W} I_x^2 & \sum_{(x,y) \in W} I_x I_y \\ \sum_{(x,y) \in W} I_x I_y & \sum_{(x,y) \in W} I_y^2 \end{bmatrix} \quad (2.31)$$

2.5.1.4 Processus de l'algorithme de détection de coins Harris

Généralement, l'algorithme du détecteur d'angle de Harris peut être divisé en cinq étapes.[17]

✚ COULEUR EN NIVEAU DE GRIS :

Pour accélérer la vitesse de traitement d'image pour une image en couleur il est préférable de la convertir en image de niveaux de gris vu que le détecteur de Harris n'est pas basé sur les couleurs. [17]

✚ CALCUL DES DERIVEES SPATIALES :

Ensuite, nous allons calculer $I_x(x, y)$ et $I_y(x, y)$

✚ CONFIGURATION DU TENSEUR DE STRUCTURE :

Avec $I_x(x, y)$, $I_y(x, y)$ on peut construire le tenseur de structure M .

✚ CALCUL DE LA REPONSE DE HARRIS :

Nous allons calculer la plus petite valeur propre du tenseur de structure avec l'équation d'approximation suivante.

$$\lambda_{min} \approx \frac{\lambda_1 \lambda_2}{(\lambda_1 + \lambda_2)} = \frac{\det(M)}{\text{trace}(M)} \quad (2.32)$$

Où $\text{trace}(M) = m_{11} + m_{22}$

Un autre calcul de réponse de Harris couramment utilisé est présenté ci-dessous,

$$R = \det(M) - k(\text{trace}(M))^2 = \lambda_1 \lambda_2 - k(\lambda_1 + \lambda_2)^2 \quad (2.33)$$

Où k est une constante déterminée empiriquement ; $k \in [0.004, 0.006]$.

2.5.1.5 Utilisation du détecteur de Harris dans Matlab

On a soumis une image <<cameraman.tif>> au détecteur de Harris afin de détecter les différents coins de l'image. La figure (2.5) nous montre le résultat de la détection des différents points caractéristiques de l'image.



Figure 2. 5: Image cameraman.tif soumise au détecteur de Harris

On soumet cette même image à un bruit gaussien afin de juger le comportement du détecteur de Harris face aux bruits.



Figure 2. 6: Image cameraman.tif avec bruit gaussien soumise au détecteur de Harris

On en conclut de la robustesse du détecteur de Harris face au bruits car celui-ci réussit parfaitement à détecter les différents points caractéristiques de l'image avec ou sans bruits. Il offre une bonne répétabilité sous des conditions d'éclairage et de rotation variables. [17]

2.5.2 Scale-Invariant Feature Transform (SIFT)

La Scale-Invariant Feature Transform (SIFT), que l'on peut traduire par « transformation de caractéristiques visuelles invariante à l'échelle », est un algorithme utilisé dans le domaine de la vision artificielle pour détecter et identifier les éléments similaires entre différentes trames. Cette méthode a été développée en 1999 par le chercheur DAVID LOWE. C'est un algorithme de détection de caractéristiques et de calcul de descripteurs. En effet, il s'agit tout d'abord de détecter sur l'image des zones circulaires, centrées autour d'un point-clé et de rayon déterminé appelé facteur d'échelle. Ces descripteurs présentent l'avantage d'être invariants à l'orientation et à la résolution de l'image, peu sensibles à son exposition, à sa netteté ainsi qu'au point de vue 3D. [25,26]

➤ **Détection des points-clés et calcul du descripteur SIFT**

Les points d'intérêts, sont communément appelés points-clés. Un point-clé $p = (x, y, \sigma)$ est défini d'une part par ses coordonnées sur l'image (x et y) et d'autre part par son facteur d'échelle caractéristique (σ). Il s'agit d'une zone d'intérêt circulaire, le rayon de la zone étant proportionnel au facteur d'échelle. Il s'ensuit une étape de re-convergence et de filtrage qui permet d'améliorer la précision sur la localisation des points-clés et d'en éliminer un certain nombre jugé non pertinents. Chaque point-clé restant est ensuite associé à une orientation intrinsèque, c'est-à-dire ne dépendant que du contenu local de l'image autour du point clé, au facteur d'échelle considéré. Elle permet d'assurer l'invariance de la méthode à la rotation et est utilisée comme référence dans le calcul du descripteur, qui constitue la dernière étape de ce processus. [25,26]

➤ **Détection d'extrema dans l'espace des échelles**

La détection s'effectue dans un espace discret que l'on appelle espace des échelles (*scale space*) qui comporte trois dimensions : les coordonnées cartésiennes x et y et le facteur d'échelle σ . On appelle gradient de facteur d'échelle σ (noté L) le résultat de la convolution d'une image I par un filtre gaussien G de paramètre σ , soit [25,26]:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \quad (2.34)$$

Cette convolution a pour effet de lisser l'image originale I de telle sorte que les détails trop petits, c'est-à-dire de rayon inférieur à σ , sont estompés. Par conséquent, la détection des objets de dimension approximativement égale à σ se fait en étudiant l'image appelée différences de gaussiennes (en anglais *difference of gaussians*, DoG) définie comme suit [25,26]

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma), \quad (2.35)$$

Où k est un paramètre fixe de l'algorithme.

➤ **Localisation précise de points clés**

L'étape de détection d'extremums produit en général un grand nombre de points-clés candidats, dont certains sont instables ; de plus, leur localisation, en particulier aux échelles les plus grands, reste approximative. [25,26]

➤ **Amélioration de la précision par interpolation des coordonnées**

Visant à augmenter de façon significative la stabilité et la qualité de la mise en correspondance ultérieure, cette étape, qui est une amélioration de l'algorithme original, s'effectue dans l'espace des échelles à trois dimensions, où $D(x, y, \sigma)$, qui n'est connu que pour des valeurs discrètes de x, y et σ , doit être interpolé.

Cette interpolation s'obtient par un développement de Taylor à l'ordre 2 de la fonction différence de gaussiennes $D(x, y, \sigma)$, en prenant comme origine les coordonnées du point-clé candidat. Ce développement s'écrit comme suit [25,26] :

$$D(X) = D + \frac{\partial D^T}{\partial X} + \frac{1}{2} X^T \frac{\partial^2 D}{\partial X^2} X \quad (2.36)$$

Où D et ses dérivées sont évaluées au point-clé candidat et où $X = (x, y, \sigma)^T$ est un delta par rapport à ce point. Les dérivées sont estimées par différences finies à partir des points voisins connus de façon exacte. La position précise de l'extremum \hat{x} est déterminée en résolvant l'équation (2.37) en annulant la dérivée de cette fonction par rapport à X , on trouve ainsi [25,26]:

$$\hat{x} = - \frac{\partial^2 D^{-1}}{\partial x^2} \cdot \frac{\partial D}{\partial x} \quad (2.37)$$

Un delta \hat{x} supérieur à 0.5 dans l'une des trois dimensions signifie que le point considéré est plus proche d'un des voisins dans l'espace des échelles discret. Dans ce cas, le point-clé candidat est mis à jour et l'interpolation est réalisée à partir des nouvelles coordonnées. Sinon, le delta est ajouté au point candidat initial qui gagne ainsi en précision. [25,26]

➤ **Elimination des points-clés de faible contraste**

La valeur de $D(X)$ aux coordonnées précises \hat{x} du point-clé peut être calculée à partir du développement de Taylor de cette fonction, et constitue donc un extremum local. Un seuillage absolu sur cette valeur permet d'éliminer les points instables, à faible contraste. [25,26]

➤ **Elimination des points situés sur les contours**

Les points situés sur les arêtes (ou contours) doivent être éliminés car la fonction DoG y prend des valeurs élevées, ce qui peut donner naissance à des extremums locaux instables, sensibles au bruit. L'équation (2.38) ci-dessous représente la matrice hessienne qui permettra d'éliminer les points situés sur les contours [25,26] :

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \quad (2.38)$$

Les dérivées doivent être évaluées aux coordonnées du point d'intérêt (x, y, σ) dans l'espace des échelles. Les valeurs propres de H sont proportionnelles aux courbures principales de D , dont seul le rapport r est intéressant. La trace de H représente la somme de ces valeurs, et le déterminant son produit. Par conséquent, en adoptant un seuil r_{th} sur le ratio des courbures ($r_{th}=10$ dans la méthode originale de Lowe), un point-clé candidat va être retenu, selon le critère adopté par Lowe [25,26], si :

$$R = \frac{\text{trace}(H)^2}{\det(H)} = \frac{(r+1)^2}{r} < \frac{(r_{th}+1)^2}{r_{th}} \quad (2.39)$$

➤ **Assignment d'orientation**

L'étape d'assignment d'orientation consiste à attribuer à chaque point-clé une ou plusieurs orientations déterminées localement sur l'image à partir de la direction des gradients dans un voisinage autour du point. Dans la mesure où les descripteurs sont calculés relativement

à ces orientations, cette étape est essentielle pour garantir l'invariance de ceux-ci à la rotation : les mêmes descripteurs doivent pouvoir être obtenus à partir d'une même image, quelle qu'en soit l'orientation. [25,26]

À l'issue de cette étape, un point-clé est donc défini par quatre paramètres (x, y, σ, θ) . Il est à noter qu'il est parfaitement possible qu'il y ait sur une même image plusieurs points-clés qui ne diffèrent que par un seul de ces quatre paramètres (le facteur d'échelle ou l'orientation, par exemple). [25,26]

➤ Descripteur des points clés

Les étapes précédentes ont permis de trouver des emplacements de points-clés à des échelles particulières et de leur attribuer des orientations. Cela garantissait l'invariance de l'emplacement, de l'échelle et de la rotation de l'image. Maintenant, nous voulons calculer un vecteur de descripteur pour chaque point clé de manière à ce qu'il soit hautement distinctif et partiellement invariant des variations restantes telles que l'éclairage, le point de vue 3D, etc. Cette étape est effectuée sur l'image la plus proche de l'échelle du point clé. [25,26]

2.5.2.1 Exemple d'utilisation de la méthode SIFT dans Matlab



Figure 2. 7: Image cameraman.tif soumise au détecteur de points clés SIFT

On remarque que la méthode SIFT reconnaît mieux les objets mais cependant on a trop de points caractéristiques, comme sur les bords de l'image.

2.6 Etude comparative

Pour l'étude comparative, nous avons procédé par élimination c'est-à-dire choisir les méthodes pouvant nous être utiles, c'est ainsi que les méthodes fréquentielles (temps de calculs trop long dû au passage du domaine temporel au domaine fréquentiel et vice-versa) et indirectes (pas adaptées pour des applications génériques) ont été ignorées. Nous allons donc faire une comparaison entre les méthodes différentielles et celles de mise en correspondance. Pour ce faire, nous allons comparer la méthode de H&S et la méthode de BM. La méthode différentielle de L&K ayant été éliminée du fait qu'étant moins adaptée aux grands déplacements que celle de H&S.

Les critères choisis pour comparer les différents types de méthodes restantes sont : la sensibilité au bruit, le temps de calcul et le comportement face au grand déplacement. Le logiciel choisi pour effectuer les différents algorithmes est Matlab, principalement pour la facilité de manipulation des matrices (images 2D) et le toolbox très développé « Image Processing ».

2.6.1 Temps de calcul

Pour évaluer l'efficacité d'une méthode, on peut utiliser comme critère quantitatif, le temps d'exécution, sur une plate-forme de calcul donnée. Cette évaluation dépend des caractéristiques du processeur utilisé. L'évaluation a été faite sur un ordinateur portable ayant les caractéristiques suivantes :

- ❖ Processeur : intel(R) Core (TM) i7-5500U CPU @ 2.40GHZ
- ❖ Mémoire RAM installée : 8.00 G0
- ❖ Type du système : Système d'exploitation 64 bits, processeur x64

Après une évaluation sur 10 mesures on obtient les moyennes suivantes :

Séquence de test	Méthode d'estimation	Temps de calcul (secondes)
Singleball (50 trames)	HORN & SHUNCK	6.5069s
	BLOCK MATCHING (Exhaustive Search)	5.4601s
Visiontraffic (531 trames)	HORN & SHUNCK	74,7844 s
	BLOCK MATCHING (Exhaustive Search)	70,3875 s

Tableau 2. 1: Etude comparative vis-à-vis du temps de calcul

2.6.2 Sensibilité au bruit

Nous allons appliquer aux différentes méthodes un bruit gaussien (de moyenne nulle et de variance égale à 0.01) qui est le modèle de bruit de base utilisé en théorie de l'information pour rendre compte des nombreux processus stochastiques (aléatoires) qui se produisent dans la nature pour voir leurs comportements face à celui-ci.

2.6.2.1 Méthode de H&S



Figure 2. 8: La méthode de H&S sur la vidéo visiontrafic

En ajoutant du bruit gaussien à la figure (2.8), on obtient la figure (2.9) suivante :



Figure 2. 9: La méthode de H&S sur la vidéo visiontrafic face à un bruit gaussien

Au vu de la Figure (2.9) on observe l'estimation du mouvement par la technique de H&S, le bruit gaussien est perçu comme un mouvement dans la scène.

2.6.2.2 Méthode de BM



Figure 2. 10: La méthode de BM sur la vidéo visiontrafic

Après ajout de bruit gaussien a la Figure (2.10) l'on obtient la figure (2.11) suivante :



Figure 2. 11: La méthode de BM sur la vidéo visiontrafic face à un bruit gaussien

La méthode de BM n'est pas trop sensible aux bruits même si on observe quelques flots optiques sur la figure (2.11) qui ne correspondent pourtant pas à des mouvements dans la scène.

On peut en conclure que la méthode de BM est moins sensible au bruit que celle de H&S.

2.6.3 Comportement face aux grands déplacements

Un grand déplacement est défini comme un mouvement rapide de l'objet dans la scène, pour ce faire nous avons provoqué de grand déplacement dans la vidéo *visiontraffic* en passant de 531 trames à 21 trames (un saut répétitif de 25 trames) on a obtenu les résultats suivants :

2.6.3.1 Méthode de H&S

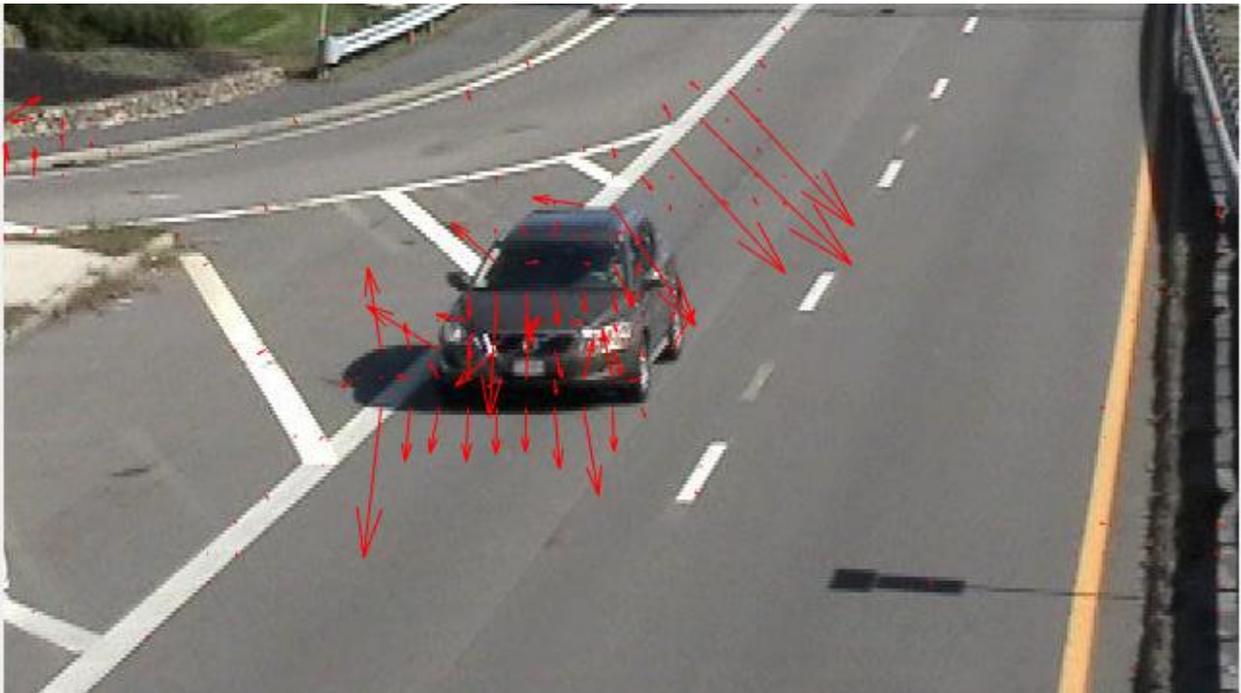


Figure 2. 12: Méthode de H&S face aux grands déplacements

La figure (2.12) nous montre le comportement de cette méthode face aux grands déplacements cependant on découvre quelques erreurs dans l'appréciation du sens de déplacements.

2.6.3.2 Méthode de BM



Figure 2. 13: Méthode de BM face aux grands déplacements

Selon la Figure (2.13) malgré les grands déplacements, on peut en déduire que la méthode de Block Matching n'est pas sensible aux grands déplacements.

Pour finir on peut affirmer que la méthode par appariement est bien meilleure pour les grands déplacements que la méthode différentielle.

2.7 Conclusion

Le but de cette étude comparative est de pouvoir décider pour quelle méthode nous allons opter afin de réaliser une application générique et en temps réel d'estimation de mouvement.

Partant de cette étude, nous déduisons en premier lieu que les méthodes indirectes ne sont pas favorisées pour notre cas car nous désirons réaliser une application générique or celles-ci extraient des caractéristiques à l'objet pour pouvoir le mettre en correspondance ce qui les rend bien adaptées à des applications bien spécifiques et non génériques comme par exemple l'application de surveillance routière.

Ensuite, un système temps réel doit avoir un temps de calcul inférieur à la variation de l'entrée pour ne pas perdre de l'information. Pour cela, les méthodes favorisées sont celles les plus rapides et celle les plus efficaces pour les grands déplacements (frames ratées dues à un temps de calcul excessive=grand déplacement). Un autre critère de choix de méthode est la sensibilité envers le bruit.

Partant de l'analyse énumérée ci-dessus et de l'étude comparative effectuée, pour une application générique et en temps réel, nous optons pour les méthodes les plus génériques, les plus rapides, les plus efficaces en grands déplacement et les moins sensibles aux bruits. Pour cela, nous choisissons celle dite par appariement de blocs ou Block Matching.

Le chapitre suivant sera l'objet d'une application générique et en temps réel pour estimer le mouvement des objets dans une séquence vidéo en se basant sur la méthode de Block Matching.

**CHAPITRE 3 : Mise
en œuvre d'un
système
d'acquisition et
d'estimation de
mouvement en
temps réel**

3.1 Introduction

D'après l'étude critique effectuée dans le chapitre précédent pour différentes méthodes d'estimation de mouvement et vu le but de notre application, nous avons opté pour la méthode de *Block Matching*. Ce choix a été justifié.

Dans ce chapitre, nous mettons en œuvre une application d'acquisition et d'estimation de mouvement en temps réel.

3.2 Acquisition d'images

Le but de l'acquisition d'images est de convertir une vue du monde réel en une image numérique. C'est une mesure spatiale d'une interaction entre une onde et de la matière. L'onde est émise par une source et reçue par un capteur. Dans notre cas c'est notre environnement qui est capturé par une caméra. Après cette acquisition d'images, l'information ainsi obtenue va être numérisée pour être compréhensible par le processeur. Cela se traduit par un échantillonnage et ensuite une quantification. Tout cela peut se faire à travers une webcam. La webcam est un appareil numérique compact que l'on peut connecter à notre PC pour diffuser des images vidéo en temps réel. En effet l'image, au lieu d'être stockée en mémoire, est directement envoyé par câble USB à l'ordinateur qui via un logiciel, l'affiche immédiatement à l'écran. Le capteur étant sans cesse rafraîchi, cela permet de créer des trames d'images, formant ainsi des séquences vidéo. [27, 28]

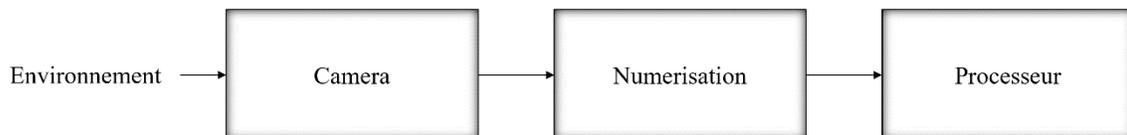


Figure 3. 1: Système d'acquisition vidéo.

Dans notre cas on utilise le logiciel Matlab pour notre acquisition à travers le toolbox de celui-ci appelé Image Acquisition Toolbox.

3.3 Caractéristiques du matériel utilisé

➤ **Rappel des caractéristiques du processeur**

Processeur : intel(R) Core (TM) i7-5500U CPU @ 2.40GHZ

Mémoire RAM installée : 8.00 GO

Type du système : Système d'exploitation 64 bits, processeur x64

➤ **Caractéristiques de la webcam**

Name: 'A4 TECH HD PC Camera'

Resolution: '640x480'

Brightness : 128

3.4 Acquisition en temps réel

Notre application s'effectue en temps réel c'est-à-dire que notre système est soumis à des contraintes de temps, donc l'exactitude des applications ne dépend pas seulement du résultat mais aussi du temps auquel ce résultat est produit. Donc même si le temps de variation est plus rapide que le temps de calcul, l'application reste toujours en temps réel comme le montre la figure (3.2). Notre application ne nécessite pas de capturer toutes les trames mais juste un nombre nécessaire pour pouvoir estimer le mouvement par la méthode de Block Matching. Aussi cette méthode se comporte bien face aux grands déplacements, donc rater quelques trames n'influe pas sur l'estimation de mouvement par cette méthode.

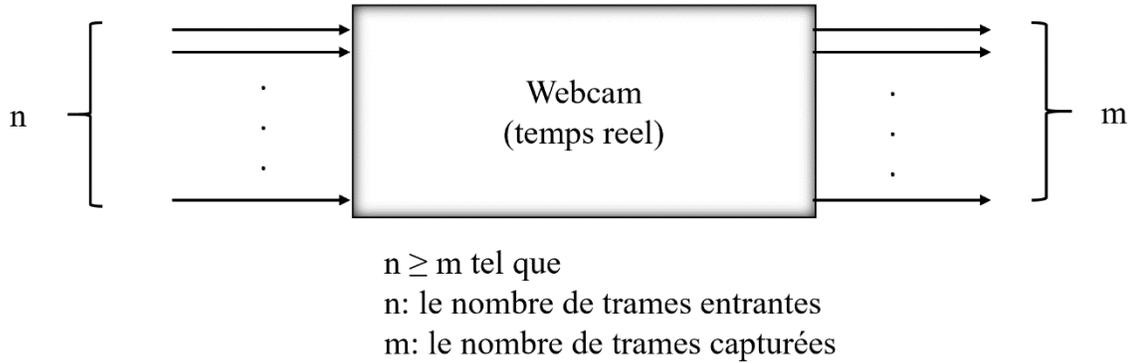


Figure 3. 2: Schéma synoptique montrant le concept temps réel pour notre application.

3.5 Algorithme de l'application

Les méthodes de mise en correspondance sont basées sur l'hypothèse que l'intensité lumineuse des pixels est constante ou faiblement variable le long des trajectoires de mouvement. Ces méthodes cherchent de manière exhaustive des similarités (points, régions) entre deux images successives, de telle sorte qu'un critère de ressemblance soit maximisé. Ces méthodes peuvent être classées en fonction de critère de mise en correspondance, dimension du bloc et celle de la fenêtre de recherche, et de la stratégie de balayage.

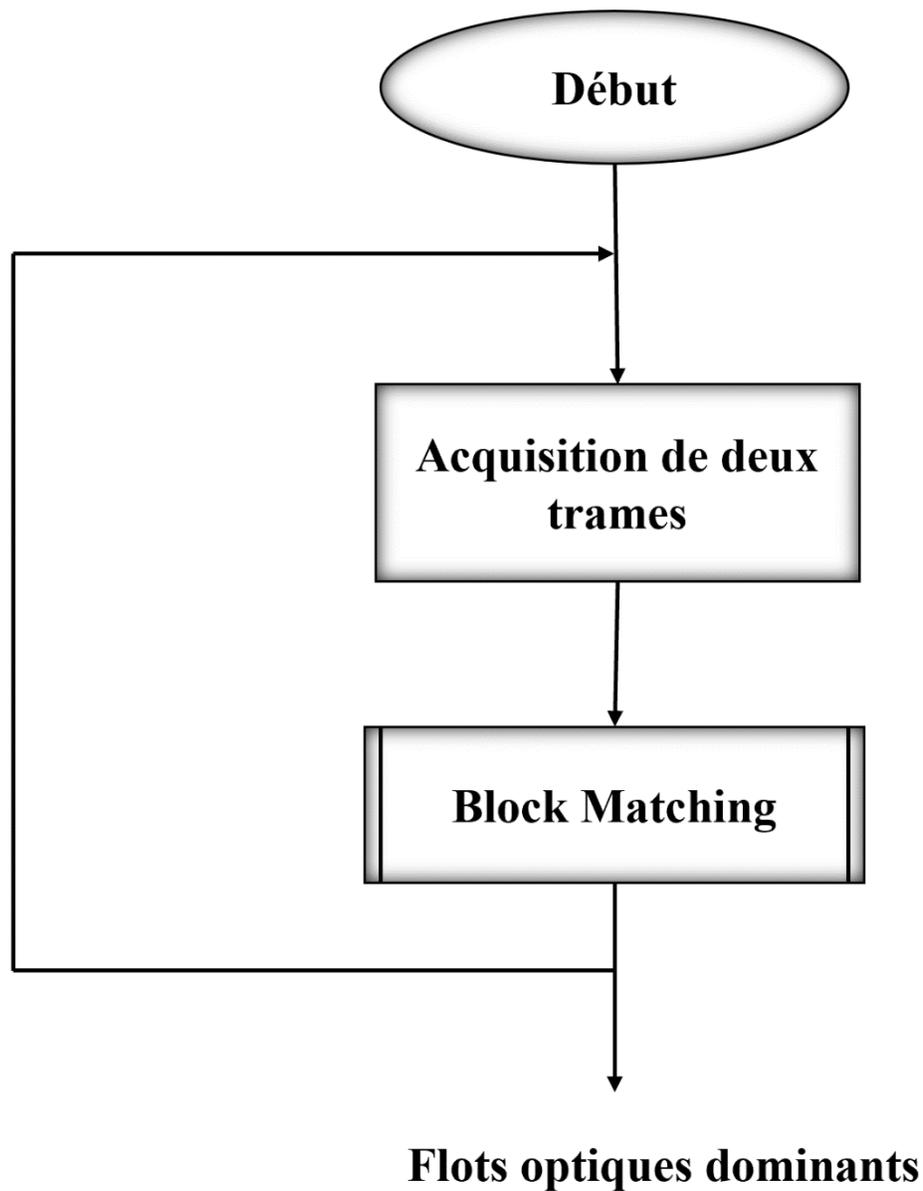


Figure 3. 3: Organigramme de l'application.

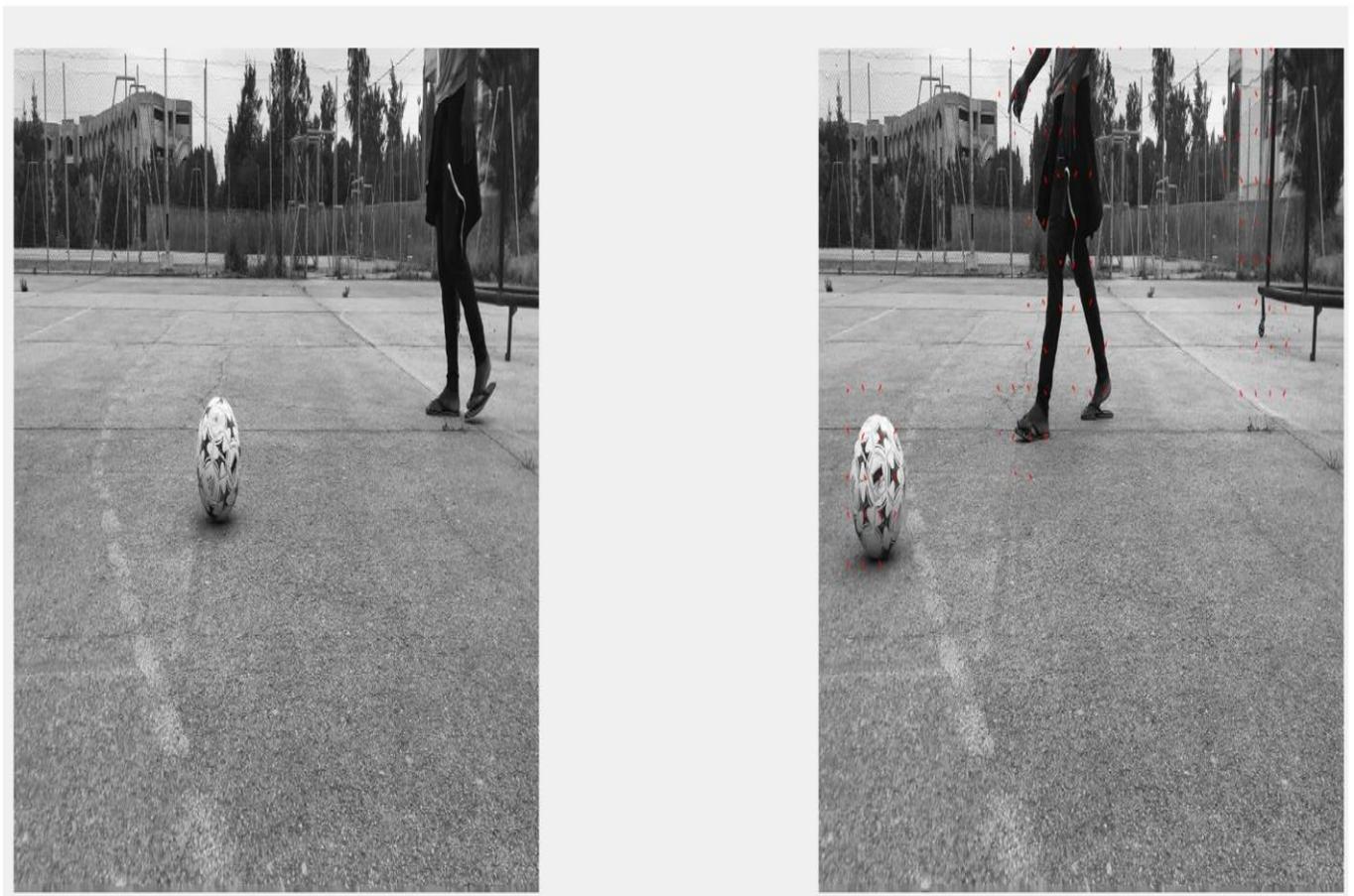
La première fonction de notre algorithme qui est l'acquisition consiste tout d'abord à faire appel à la webcam, puis on fait une acquisition de deux trames.

La seconde fonction consiste à l'application de l'algorithme du BM.

Enfin on affiche les résultats de l'application du BM sur nos trames à travers des flots optiques dominants (éliminations de la plupart des bruits considérés comme des objets mobiles dans la scène). Cet algorithme est répété en boucle jusqu'à un temps voulu.

3.6 Résultats de l'application en temps réel

Nous allons commencer par observer le mouvement à travers des situations communes de la vie c'est à dire des personnes jouant au football (Figure (3.4)), deux étudiants déambulant dans une chambre universitaire (Figure (3.5)) et un trafic routier (Figure (3.6)).



a)

b)

Figure 3. 4: Mouvements d'un joueur et de la balle : a) l'image de référence, b) mouvements perçus dans la trame suivante.



a)

b)

Figure 3. 5: Mouvements d'étudiants dans la scène : a) l'image de référence, b) mouvements perçus dans la trame suivante.



a)

b)

Figure 3. 6: Mouvements de voiture dans la scène : a) l'image de référence, b) mouvements perçus dans la trame suivante.

Les différentes figures ci-dessus représentent des mouvements dans la scène. Chaque figure correspond à deux trames successives, la première trame est celle de référence, elle va être comparée à la seconde pour voir s'il y'a eu un mouvement ou pas. En effet les mouvements effectués dans la scène sont perçus par l'algorithme qui identifie ses mouvements à travers des flots optiques de couleur rouge comme on peut le voir dans ses images. Que ce soit le mouvement du joueur et de la balle, le mouvement des étudiants dans la chambre, ou le mouvement de différentes voitures à un carrefour.

Cependant, on observe quelques erreurs dans l'estimation du flot qui sont sûrement dues aux contraintes de l'estimation du flot optiques que nous avons expliqué dans le premier chapitre (même les intensités lumineuses issues des objets immobiles peuvent varier).

3.6.1 Hypothèse des erreurs d'estimation du flot optique

Nous avons constaté quelques erreurs dans l'estimation du flot optique qui peuvent être dues à divers facteurs : les surfaces non lambertiennes (supposées au départ étant lambertiennes), la résolution de la caméra utilisée pour faire l'acquisition des images.

3.6.1.1 Surfaces lambertiennes

On a supposé que les surfaces sont lambertiennes (l'intensité lumineuse est la même quelque soit l'angle de vue), alors que cela n'est pas le cas de manière pratique. Pour montrer cette contrainte, nous allons utiliser deux méthodes : à partir de l'index de Matlab pour les images et le détecteur de Harris.

✚ Index de Matlab

Pour démontrer cela on effectue l'acquisition d'une même scène à deux instants différents (Figure 3.7). Dans cette figure, on a trois types de valeurs, la première [X, Y] représente les coordonnées du pixel, la seconde qui nous intéresse est l'index, c'est-à-dire l'intensité lumineuse du pixel ou encore son niveau de gris. La troisième valeur ne nous est pas utile vu qu'on utilise une image en niveau de gris. Ainsi pour un pixel de mêmes coordonnées et appartenant à un objet fixe dans la scène, on a un index différent ce qui renforce notre argumentation sur le fait que les surfaces ne sont pas réellement lambertiennes.

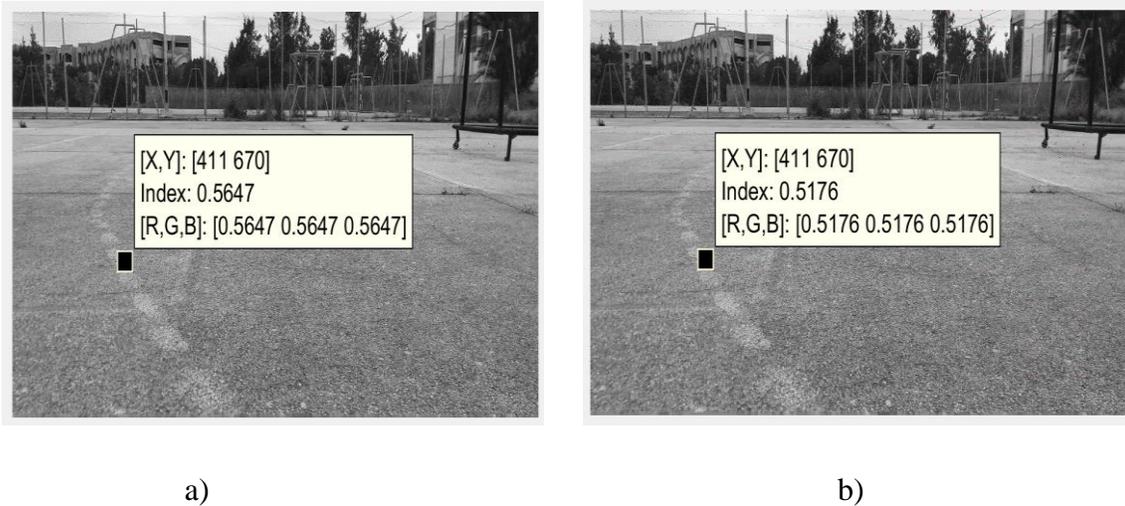


Figure 3. 7: Trames acquises à des instants différents : a) Première trame, b) Deuxième trame.

✚ Détecteur de Harris

A travers le détecteur de coins de Harris (détecteur de points caractéristiques) dont nous avons expliqué le principe dans le chapitre précédent, nous avons pu observer que pour deux trames successives d'un objet immobile on a malgré tout une différence dans la position des points caractéristiques (entouré en rouge) dans la figure (3.8). Ce qui appuie notre argumentation sur le fait que les surfaces ne sont jamais lambertiennes.

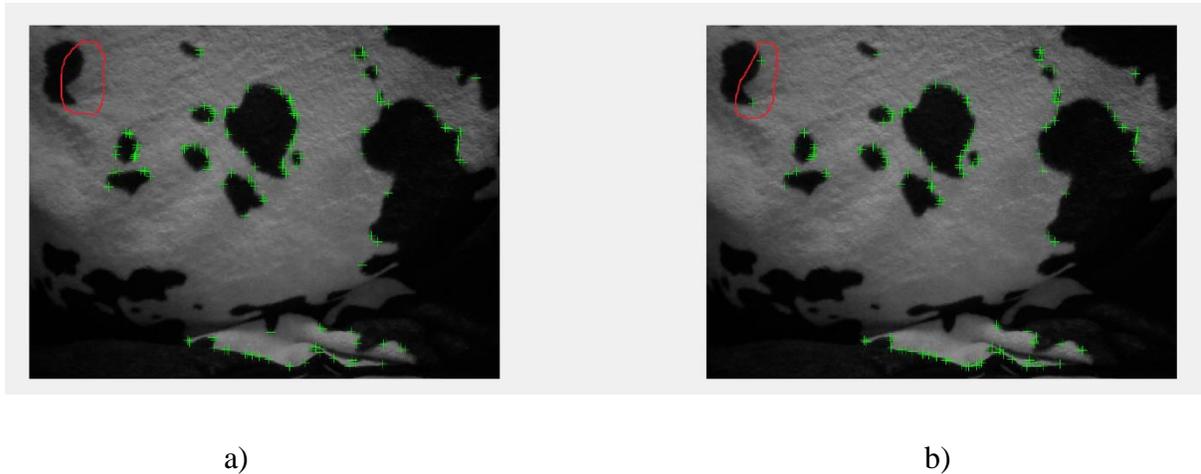


Figure 3. 8: Application du détecteur de Harris sur deux trames : a) Première trame, b) Deuxième trame.

3.6.1.2 Résolution de la camera

La résolution de la caméra influe sur l'appréciation du flot optique pour un mouvement en profondeur. Cela est dû d'une part au nombre de pixel correspondant à la taille d'un objet et qui varie selon la distance entre la caméra et l'objet. D'autre part, l'objet peut bouger dans la profondeur (troisième dimension) en gardant les même coordonnées sur le plan de projection 2D. En effet plus l'objet est proche et plus les petits mouvements sont perçus, alors que pour un objet plus éloigné de la caméra le mouvement doit être plus grand pour être perçu par la caméra. Ce problème est illustré à travers la figure (3.9), l'objet (la voiture) mobile entouré en vert n'est pas perçu comme mobile dans la scène.



Figure 3. 9: Image avec une résolution de 640x480

La solution d'un tel problème peut être dans le choix de la résolution convenable à la profondeur du territoire à surveiller(Figure 3.10) ou par une deuxième caméra placée perpendiculairement.



Figure 3. 10: Image avec une résolution de 1334x750

Cependant l'augmentation de la résolution influe sur la vitesse de calcul, plus la résolution est grande et plus l'algorithme met plus de temps alors qu'avec une résolution réduite on a un meilleur temps de calcul. On doit donc faire le compromis entre résolution et temps de calcul.

3.7 Conclusion

Dans ce chapitre, nous avons mis en œuvre un système d'acquisition et d'estimation de mouvement en temps réel. Ainsi, nous avons décrit notre algorithme d'estimation de mouvement, utilisant la méthode de Block Matching, tout en assurant que celui-ci fonctionne sans perdre le fil de poursuite des objets mobiles. Un ensemble d'exemples a montré l'efficacité de l'algorithme décrit.

Nous avons ainsi mis en reliefs les différentes contraintes liés à l'application et proposé des solutions éventuelles qui pourraient donner suite à ce travail.

CONCLUSION GENERALE

Ce travail de mémoire traite de la détection et du suivi d'objets en mouvement dans une séquence vidéo pour une application en temps réel. L'objectif est de définir un ensemble d'opérations génériques qui peuvent être utilisées quel que soit le domaine d'application.

Dans un premier temps, nous avons présenté un état de l'art sur le domaine. Ainsi, nous avons rappelé les notions relatives à l'estimation de mouvement et ensuite présenté la différence entre le mouvement réel (les scènes réelles 3D), le mouvement apparent (observé dans une séquence temporelle d'images, comme variation spatio-temporelle d'intensité lumineuse) et le mouvement qui peut être estimé à partir du mouvement observé. Cette présentation a été accompagnée des contraintes envers lesquelles des hypothèses doivent être faites pour que le mouvement estimé soit une bonne approximation du champ de mouvement réel. A la fin, les méthodes de base en estimation de mouvement ont été représentées de manière non exhaustive.

Ensuite, nous avons commencé par diviser les techniques d'estimation en deux parties : les méthodes indirectes (basée sur la détection de points caractéristiques) et les méthodes directes (basée sur le pixel). Cependant les méthodes indirectes ne sont pas adaptées pour des applications génériques, nous avons donc travaillé avec les méthodes directes. Elles peuvent être divisées en trois groupes : les méthodes fréquentielles (basées sur la transformée de Fourier), les méthodes différentielles (basées sur les gradients spatiaux et temporels de l'intensité lumineuse des pixels) et les méthodes par appariement (ces méthodes cherchent de manière exhaustive des similarités (points, régions) entre deux images successives, de telle sorte qu'un critère de ressemblance soit maximisé). Pour une utilisation en temps réel, nous devons opter pour des méthodes avec un temps de calcul minimal alors que pour les méthodes fréquentielles on doit faire un passage du domaine temporel au domaine fréquentiel puis retourner au domaine temporel ce qui n'est pas propice pour une application en temps réel. Nous nous sommes donc dirigés vers les deux types de méthodes directes restantes c'est-à-dire les méthodes différentielles et celles de mise en correspondance. Pour les départager afin de trouver celle qui nous scierait le plus, nous avons d'abord comparé les méthodes différentielles entre elles, c'est à dire celle de Lucas & Kanade et de Horn & Shunck ce qui nous a permis d'opter pour cette dernière. Puis, nous avons comparé la méthode différentielle choisie et la méthode d'appariement à partir des critères suivants : la sensibilité aux bruits, le temps de calcul et le

comportement face aux grands déplacements c'est ainsi que nous avons choisi comme méthode pour notre application celle d'appariement : le Block Matching.

Enfin, pour aboutir à notre but planifié, nous avons évalué les points forts et les points faibles de notre algorithme pour une application en temps réel. Pour ce faire, nous avons donné un aperçu sur notre application et les fonctions principales que nous avons implémentées.

- **Perspectives**

Les points suivants pourraient être l'objet d'une suite pour ce travail :

- Concernant le Block Matching, l'utilisation d'autres stratégies de balayage dans la fenêtre de recherche, à part celle utilisée dans ce travail et dite exhaustive, peut encore optimiser le temps de calcul et assurer un fonctionnement en temps réel pour des mouvements à des vitesses éventuellement plus élevées.
- Nos différentes méthodes étudiées sont basées sur la mesure du mouvement. Mais face à certaines conditions météorologiques (brouillard, pluies...) ou même occlusion, notre application s'avère insuffisante. Pour pallier à cela, on pourrait opter par exemple pour un filtre de Kalman, qui lui à partir de la mesure effectuée du mouvement, prédit le mouvement. C'est à dire qu'il peut perdre l'objet de vue et prédire son déplacement à partir d'un système de régulation.
- Un système d'acquisition et d'estimation de mouvement en temps réel doit évidemment avoir un temps de calcul bien inférieur au temps de la variation en entrée (mouvement réel). Pour des vitesses plus élevées des objets mobiles, un système d'estimation de mouvement peut atteindre ses limites et ainsi perdre le fil de poursuite. Dans ce contexte, et par rapport à un processeur d'un ordinateur qui fonctionne en multitâches, on favorise des noyaux de traitement dédiés que pour ce calcul (cas d'un *Digital Signal Processor* DSP).

BIBLIOGRAPHIE

- [1] A. Abidi, M. Slimani, «Suivi d'objets en mouvement dans une séquence d'images,» *memoire de fin d'etude master Informatique, Université Ahmed Draia- Adrar*, p. 112, 2016-2017.
- [2] F. Charif, «Estimation hiérarchique du mouvement par ondelettes géométriques,» *Thèse de doctorat, Université Mohamed Khider-Biskra*, p. 142, 2015.
- [3] S. Ambellouis, «Analyse du mouvement dans les séquences d'images par une méthode récursive de filtrage spatio-temporel sélectif,» *Thèse de doctorat, Université Pierre et Marie Curie*, 2011.
- [4] A. Abonnat et A. Rault, «Projets Génie ElectriquePolytech'Clermond-Ferrand,» 2008. [En ligne]. Available: <http://dirac.epucfe.eu/projets/wakka.php?wiki=P09A05index>.
- [5] E. Bruno, «De l'estimation locale à l'estimation globale de mouvement dans les séquences d'images,» *Thèse de doctorat, Université Joseph Fourier*, Novembre 2001.
- [6] J.L. Barron, D.J. Fleet, S.S.Beauchemin et T.A Burkitt, «Performance of Optical Flow Techniques,» *Department of Computer Science, University of Western Ontario*, July 1993.
- [7] A. Manzanera, «Analyse du mouvement, estimation et poursuite,» sur le site ensta-paristech. Consulté le 7 février 2019.(fichier pdf, p. 56). <http://perso.ensta-paristech.fr/~manzaner/Cours/IMA/tdi6.pdf>
- [8] M. Irani et P. Anandan, «All About Direct Methods», sur wisc.edu. Consulté le 6 Juin 2019. (fichier pdf, p. 12). <http://pages.cs.wisc.edu/~dyer/ai-qual/irani-visalg00.pdf>
- [9] P.H.S. Torr et A. Zisserman, «Feature Based Methods for Structure and Motion Estimation,» sur ox.ac.uk. Consulté le 6 Juin 2019. (fichier pdf, p. 12). <http://www.robots.ox.ac.uk/~vgg/publications/papers/torr00a.pdf>

- [10] D. A. Randriantsoa, «Estimation du mouvement fondée sur un schéma direct et rétrograde application à la segmentation,» *Thèse de doctorat, Université de Bordeaux I*, p. 283, 2002.
- [11] F. Comby, «Estimation du mouvement apparent majoritaire dans une séquence d'images vidéo par accumulation de votes bimodaux sur un histogramme approché,» *Thèse de doctorat, Université Montpellier II*, p. 169, 2001
- [12] B.K.P. Horn et B.G. Shunck, «Determining optical flow,» *Artificial intelligence*, vol. 17, pp. 185-203, 1981.
- [13] B. Lucas et T. Kanade, «An iterative image registration technique with an application to stereo video,» *Proceedings of DARPA Image Understanding Workshop*, pp. 121-130, 1981.
- [14] H.H. Nagel, «Displacement vectors derived from second-order intensity variations in image sequences,» *CVGIP*, pp. 85-117, 1986.
- [15] B. H. Friemel, L. N. Bohs, B. A. McDermott et G. E. Trahey, «initial results with the real-time sad vector velocity estimation system: constant velocity calibration,» *In Ultrasonics Symposium*, pp. 1179-1982, 1992.
- [16] D.J. Heeger, «Optical Flow using Spatiotemporal Filters,» *Int. Journal of Computer Vision*, pp. 279-302, 1987.
- [17] J.J. Anitha et S.M. Deepa, «Tracking and Recognition of Objects using SURF Descriptor and Harris Corner Detection,» *Nehru Institute of Engineering and Technology*, vol. 4, No.2, p. 4.
- [18] D. Amel, «Estimation de mouvement par les méthodes fréquentielles de flux optique, » *memoire de fin d'etude master Informatique*, p. 82, 2012-2013.
- [19] R. Yaakob, A. Aryanfar, A. A. Halin et N. Sulaiman, «A Comparison of Different Block Matching Algorithms for Motion Estimation,» *Procedia Technology*, vol. 11, pp. 199-205, 2013.

- [20] F. Yu et A.N. Wilson Jr., «A flexible hardware-oriented fast algorithm for motion estimation,» *Integrated Circuits and Systems Laboratory*, pp. 2681-2884, 1997.
- [21] D. Patel et S. Upadhyay, «Optical Flow Measurement using Lucas Kanade Method,» *International Journal of Computer Applications*, vol. 61, n° 10, pp. 6-10, janvier 2013.
- [22] H. A. Patel et D. G. Thakore, « Moving Object Tracking Using Kalman Filter », *International Journal of Computer Science and Mobile Computing*, vol. 2, pp. 326-332, 2013.
- [23] D. J. Fleet et A. D. Jepson, «Computation of Component Image Velocity from Local Phase Information,» *IJCV*, vol. 5, n° 1, pp. 77-104, 1990.
- [24] D. J. Fleet, «Disparity from Local Weighted Phase- Correlation,» *IEEE Int. Conf. on Systems, Man and Cybernetics*, vol. 1, pp. 48-54, 1994.
- [25] A. Rajwade, «Scale Invariant Feature Transform (SIFT),» sur [cse.iitb.ac](http://cse.iitb.ac.in/~ajitvr/CS763/SIFT.pdf). Consulté le 7 Mars 2019. (fichier pdf, p. 52). <https://www.cse.iitb.ac.in/~ajitvr/CS763/SIFT.pdf>
- [26] D. G. Lowe, «Distinctive Image Features from Scale-Invariant Keypoints,» *International Journal of Computer Vision*, vol. 60, n° 02, pp. 91-110, Nov. 2004.
- [27] H. Mathieu, « Interaction homme-machine, images, données, connaissances », *Bulletin of Sociological Methodology/Bulletin de Méthodologie Sociologique*, vol. 37, n° 1, p. 82, déc. 1992.
- [28] « How do webcams work? | What parts are inside a webcam? », *Explain that Stuff*. [En ligne]. Disponible sur : <http://www.explainthatstuff.com/webcams.html>.

RESUME

Le travail développé dans ce mémoire porte sur la détection et suivi d'objets mobiles dans une séquence d'images. Le but de notre travail est donc de développer un algorithme permettant d'estimer le mouvement d'objets pour une utilisation en temps réel de celle-ci. Après un état de l'art des différentes méthodes d'estimation, nous avons fait une étude comparative de celles pouvant nous permettre une utilisation en temps réel à savoir : Horn & Shunck, Block Matching. C'est ainsi que nous avons opté pour la méthode par appariement de Bloc ou dite *Block Matching*. A partir de cette méthode nous avons mis en œuvre un algorithme nous permettant d'utiliser le Block Matching en temps réel tout en utilisant un système d'acquisition à base d'une webcam

ملخص

يتناول العمل الذي تم تطويره في هذه الرسالة الكشف عن الأجسام المتحركة وتتبعها في سلسلة من الصور (فيديو). الهدف من عملنا هو تطوير خوارزمية لتقدير حركة الأجسام لاستخدامها في الوقت الفعلي. بعد طرح أحدث أساليب التقدير المختلفة، قمنا بإجراء دراسة مقارنة لتلك التي يمكن أن تسمح لنا باستخدامها في الوقت الفعلي وهي: Block Matching، Shunck & Horn. بهذا الشكل تم اختيارنا لطريقة "مطابقة المناطق" (Block Matching). من خلال هذه الطريقة، قمنا بتطبيق خوارزمية تسمح لنا باستخدام Block Matching في الوقت الفعلي أثناء استخدام نظام تلقي يعتمد على كاميرا ويب.

