

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

UNIVERSITE MOULOU D MAMMERI DE TIZI-OUZOU



FACULTE DU GENIE ELECTRIQUE ET D'INFORMATIQUE
DEPARTEMENT D'ELECTRONIQUE

Mémoire de Fin d'Etudes
De MASTER ACADEMIQUE
Domaine : Sciences et Technologies
Filière : Génie électrique
Spécialité : **Télécommunication et réseaux**

Présenté par
SADOUDI IDIR

Thème
**Utilisation de BM pour l'estimation des
mouvements de translations et d'orientations
des blocs**

Mémoire soutenu publiquement le 06/07/2017 devant le jury composé de :

Président : M. LAZRI
Examineur : F. OUALLOUCHE
Promoteur : DJ. ALOUACHE



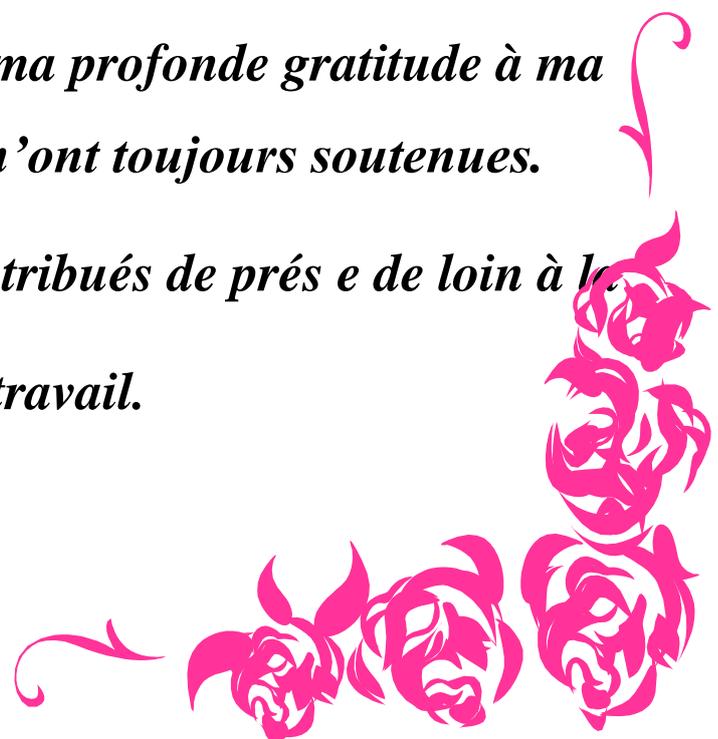
Remerciement

Je tiens a remercier vivement mon dévoué promoteur Mr. ALOUACHE DJAMAL pour son encadrement et pour sa contribution à travers sa disponibilité, pour sa patience et surtout pour sa confiance, ses remarques et ses conseils qui m'ont beaucoup éclairés et à travers lesquels j'ai apprécié l'utilité de ce mémoire.

Je voudrais également remercier les membres de jury pour avoir accepté d'évaluer ce travail et pour toutes leurs remarques et critiques.

Enfin je tiens a exprimer ma profonde gratitude à ma familles et mes amis qui m'ont toujours soutenues.

Et a tous ceux qui ont contribués de prés e de loin à la réalisation de ce modeste travail.

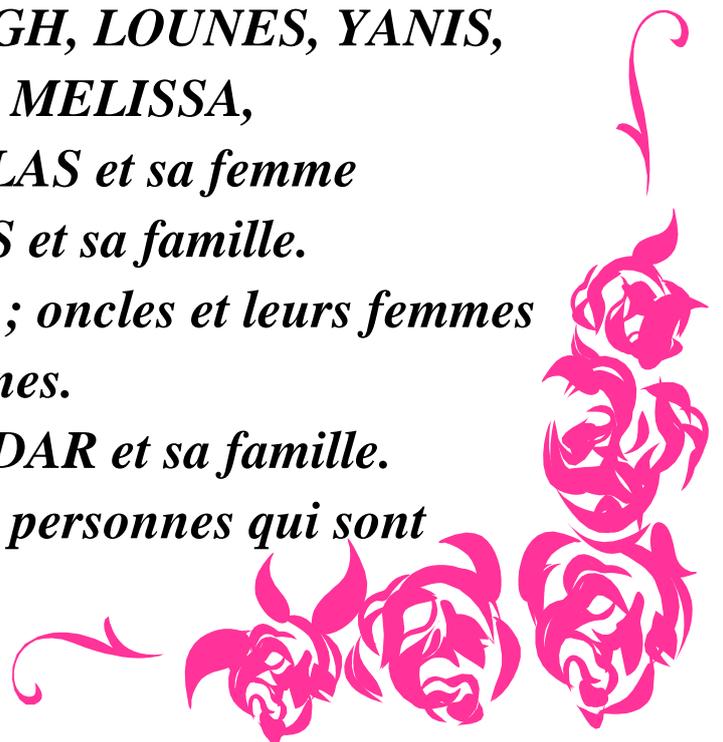




Dédicace

*Je dédie ce modeste mémoire ; tout d'abord à ma
Très chère famille qui ma toujours conseillée et
Orientée :*

- *Ma mère ; qui est la lumière de notre maison.*
- *Mon père ; qui est le guide de la réussite dans ma vie.*
- *Mes chers grands parents bien aimés.*
- *Mon cher frère ainsi que sa femme et ses enfants.*
- *Mes très chers sœurs ; NESRINE et AMEL ainsi que son mari et ses enfants.*
- *Mes chers et fidèles amis(es) qui m'ont soutenue et encourager durant tout le parcours de mes études : MAZIGH, LOUNES, YANIS, MERIEME, SABINE, MELISSA, MASSINISSA, AGHILAS et sa femme DJOUHRA , AMAYES et sa famille.*
- *Ma famille maternelle ; oncles et leurs femmes tante, cousins et cousines.*
 - *mon oncle LEKHDAR et sa famille.*
 - *Je dédis aussi aux personnes qui sont*
Chère à mon cœur .



SOMMAIRE

Introduction générale.....	1
Chapitre I	
I Préambule.....	3
I.1 Définition d'une image.....	3
I.2 les différents types d'image :.....	4
I.2.1 image numérique :.....	4
I.2.2 image binaire	4
I.2.3 image en niveau de gris	5
I.2.4 image couleur.....	5
I.2.5 image multi spectrale.....	5
I.2.6 image négative	6
I.3 compression d'image	7
I.3.1 compression avec perte :	7
• classification en deux classes (seuillage).....	8
• Classification en plusieurs classes	8
I.3.2 compression sans perte	8
• Algorithms RUN LENGTH ENCODING (RLE).....	8
• Algorithme HUFFMAN.....	9
I.4 prétraitement d'image.....	10
I.4.1 extension de dynamique	11
I.4.2 égalisation d'histogramme.....	11
I.4.3 les filtres	12
➤ filtre passe bas (lissage).....	12

SOMMAIRE

➤ Filtre passe haut.....	14
I.5 Conclusion	15
Chapitre II	
Préambule.....	16
II. 1 Le mouvement.....	16
II. 2 Classification du mouvement	17
II.2.1 Le mouvement réel.....	17
➤ Le mouvement d'objet rigide	17
➤ Le mouvement d'objet déformables.....	18
II.2.2 Le mouvement apparent	18
II.2.3 Le mouvement estimé	19
II.3 L'estimation du mouvement.....	20
II.3.1 Estimation des vecteurs de déplacement.....	22
II.3.2 Estimation des vecteurs vitesse	22
II.4 Les problème de l'estimation de mouvement	23
II.4.1 problème d'occlusion	23
II.4.2 Problème d'ouverture	24
II.4.3 Problème de changement d'éclairage de la scène.....	24
II.4.4 problème du modèle de translation	25
II.5 Les méthodes d'estimation de mouvement	25
II.5.1 Méthodes de mise en correspondance	25
➤ Méthode de mise en correspondance dans le plan transformé.....	26

SOMMAIRE

➤ Méthode de mise en correspondance dans le plan image.....	26
II.5.2 Méthodes statistique	26
II.5.3 Méthodes pixels-récurrentes	26
II.5.4 Méthode de LUCAS et KANADE	26
II.5.5 méthode de HORN et SCHUNCK	27
II.5.6 les méthodes basées sur le flot optique	27
II.5.7 les méthodes par appariement de blocs BMA (block matching algorithme)	27
II.6 conclusion	28
Chapitre III	
Préambule.....	29
III.1 Principe général du block matching	29
III.2 Prédiction avant et arrière.....	30
III.3 Critères d'évaluation.....	32
III.4 Stratégies de recherche du meilleur bloc	33
III.4.1 algorithme de recherche exhaustive (Full Search (FS)).....	33
III.4.2 algorithme de recherché en tris pas (Three Step Search (TSS))...34	
III.4.3 Nouvelle recherche en trois pas (New 3SS).....	35
III.4.4 Recherche en quatre pas (four step search(FSS)).....	36
III.4.5 Recherche selon le gradient (Gradient Search GS).....	38
III.4.6 algorithme de recherche en diamant.....	39
III.4.7 recherche logarithmique en deux dimensions (TDL).....	41
III.4.8 recherche binaires (BS).....	42

SOMMAIRE

III.4.9	algorithme de recherche orthogonale (OSA)	44
III.4.10	cross search algorithm	45
III.4.11	Spiral search (SS)	46
III.5	Conclusion	48
Chapitre IV		
	Préambule	49
IV.1	L'algorithme du block matching proposé	49
IV.2	Critères d'évaluation	51
	▪ L'isométrie	51
	▪ Les modifications du contraste	53
IV.3	Application	53
	➤ IV.3.1 Application sur les images de synthèse	54
	IV.3.1.1 Résultats obtenus avec la méthode de block matching	55
	IV.3.1.2 Résultats obtenus avec la méthode proposée	56
	➤ IV.3.2 Application aux images réelles (séquence rubik)	57
	IV.3.2.1 Résultats obtenus par la méthode de block matching	58
	IV.3.2.2 Résultats obtenus avec la méthode proposée	60
IV.4	conclusion	62
	Conclusion générale	63
Annexe		
Bibliographie		
Liste des figures		
Liste des tableaux		

INTRODUCTION GENERALE

Une image dit-on vaut mille mots. En effet, l'image constitue un champ fabuleux d'informations. Sous l'œil du chercheur, une importante quantité d'informations surgit spontanément de l'image, offrant non seulement une description globale et synthétique de ce qu'elle représente plus que tout autre moyen de description existant, mais aussi des instruments révolutionnaires qui permettent non seulement l'observation et la description, mais aussi la manipulation indirecte d'objets qui autrement seraient hors de notre portée et de notre regard.

Durant ces dernières années, nous assistons à la convergence rapide des ordinateurs et de la communication, conduisant à une profonde révolution de l'information, ce qui changera finalement non seulement la façon dont les gens font des affaires, mais aussi l'apprentissage et le divertissement.

Le traitement d'image, inventé au début d'années soixante dix, est de nos jours utilisé dans divers domaines avec des applications de plus en plus nombreuses.

Le concept du traitement d'image est assez simple, il a en effet très souvent pour but la reconnaissance d'objets ou de formes que l'œil humain perçoit rapidement et avec efficacité. En pratique et malgré le foisonnement important de travaux que suscite ce domaine de recherche. Il reste difficile d'extraire du flux de données que contient une image, des informations claires, précises et utiles pour l'application souhaitée.

Les problèmes rencontrés sont d'autant plus frustrants qu'ils peuvent souvent être résolus par un observateur humaine qui reconnaitra aisément une personne ou un objet sur une image un peu floue. Néanmoins, l'analyse d'image est aujourd'hui dotée d'outils mathématiques et informatiques assez puissants qui permettent de repousser ces limites et d'étendre les domaines d'applications notamment à l'analyse des séquences d'images.

L'estimation du mouvement est l'un des domaines du traitement d'images qui joue un rôle considérable dans l'analyse des séquences d'images de part le nombre considérable de ses applications. La mesure de champ de mouvement est une étape de traitement de l'image dite de bas niveau. On lui trouve de nombreuses applications. Comme l'analyse du mouvement de fluides en physique expérimentale, la compensation du mouvement, ou encore son utilisation pour des phases haut niveau, comme la reconstruction de scènes tridimensionnelles. [5]

INTRODUCTION GENERALE

En 1980 HORN et SCHUNCK ont proposé une méthode d'estimation du mouvement basée sur la régularisation. Ce premier travail a été suivi d'un grand nombre de contributions qui ont proposé différentes méthodes alternatives.

La technique la plus populaire adaptée pour l'estimation du mouvement entre deux trames est la technique de mise en correspondance par bloc 'BLOCK MATCHING ALGORITHMS' dans laquelle un bloc de taille $N \times M$ du trame actuelle est comparé avec un bloc correspondant dans une zone dite zone de recherche dans le frame précédente.

Trois éléments principaux doivent être pris en considération dans la technique BMA :

- Critère de mise en correspondance
- Zone de recherche
- Technique de recherche

Cette dernière est la plus importante qui joue un rôle critique dans les performances de l'algorithme de BMA.

En 1981 KOGA et AL ont introduit Three step search qui est un algorithme de recherche en trois pas, puis il est devenu très populaire en raison de simplicité et aussi des performances robustes presque optimales.

Pour élaborer ce travail, nous avons organisé notre thème comme suit :

-Dans le chapitre I on va donner les notions sur les différentes images et les différentes techniques de prétraitement d'image.

-Dans le chapitre II nous donnerons une notion sur les différentes méthodes d'estimation de mouvement.

-Dans le chapitre III on donnera une notion sur les différentes méthodes de mise en correspondance des blocs.

-Dans le chapitre IV, chapitre final qui inclus les résultats pratiques de notre travail. Et on en finira par une conclusion générale de notre travail.

RESUME :

Dans ce travail, nous proposons une amélioration de l'algorithme du Block Matching (BM) qui prend en considération le mouvement de rotation des blocs. L'objectif est de développer une méthode de suivi de mouvement de translation et de l'orientation de chaque bloc dans les séquences vidéo. Il s'agit donc de coder aussi bien les orientations des blocs que les translations locales des blocs en mouvement présents dans les séquences vidéo.

L'estimation et le suivi de mouvements de rotation des objets dans les images sont des problèmes complexes et encore ouverts. Il intéresse de nombreux domaines, comme la robotique, la vidéo surveillance, mais aussi la compression vidéo, Le travail présenté dans cet article concerne le suivi et l'estimation des mouvements de rotation présente dans les séquences vidéo.

Mots clés :

Estimation de Mouvement, image vidéo, mise en correspondance par blocs, Orientation des blocs, *compression d'images MSG, algorithme de Jacquin,*

Chapitre I

GENERALITE ET PRETRAITEMENT D'IMAGE

I.1 Préambule

Ce chapitre introductif est consacré pour les généralités et les prétraitements d'image. L'analyse d'une image ou d'une séquence vidéo par le cerveau humain se fait d'une manière naturelle. Le progrès remarquable qu'a connu l'outil informatique en matière de stockage et la vitesse de traitement et la compression sans sacrifice dans la qualité de l'image. Chacun peut analyser l'image à sa manière, pour en dégager une impression et d'en extraire des informations précises.

Nous avons dans le premier temps donné les différentes définitions et les types d'images. Par la suite les prétraitements d'image et les différentes techniques utilisées pour améliorer l'image.

I.2 Définition d'une image

L'image est une représentation d'un objet par (un dessin, la photographie, peinture.....etc.) aussi c'est un ensemble d'éléments structuré d'information qui ont une signification pour l'œil humain.

L'image est donc un instrument révolutionnaire qui permet non seulement l'observation et la description, mais aussi la manipulation indirecte d'objet, qui autrement serait hors de notre portée et de notre regard.

Née du désir de manipuler et toucher, de modifier et éventuellement d'améliorer la qualité de l'image par laquelle l'homme perçoit le monde extérieur.[1]



Figure I.1 : représentation d'image

Chapitre I

GENERALITE ET PRETRAITEMENT D'IMAGE

I.2 les différents types d'image :

Il y a six formes physiques d'existence d'une image : forme numérique, binaire, en niveau de gris, couleur et multi spectrale.

I.2.1 image numérique :

Est un ensemble fini de pixel coude sur des bits. L'image numérique est divisée en deux sorte d'image numérique : les images matricielles et les images vectorielles.ces image sont essentiellement utilisées pour la réalisation des schémas ou des plans, elles présentent deux avantages :

- Occupent peu d'espace en mémoire.
- Peuvent être redimensionnées sans perte d'information.

I.2.2 image binaire

Appelée aussi image noir et blanc dont les pixels ne peuvent avoir que deux valeurs (0) et (1).

'0' Correspond a un pixel noir et '1' correspond a un pixel blanc.

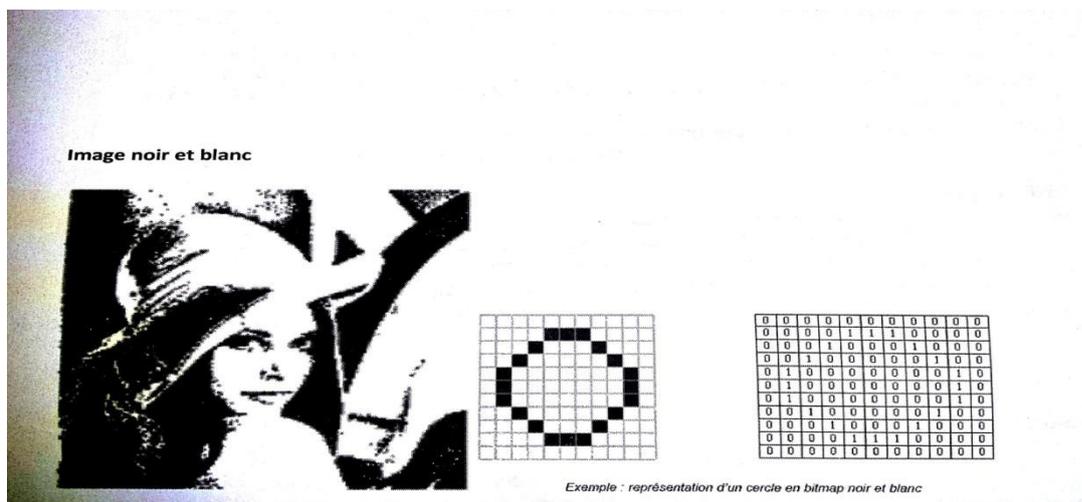


Figure I.2 : image binaire (noir et blanc)

Chapitre I

GENERALITE ET PRETRAITEMENT D'IMAGE

I.2.3 image en niveau de gris

Appelée aussi image d'intensité dans ce cas on dispose d'une échelle de niveau de gris entre la valeur '0' et la valeur '255'. En général on code le niveau de gris sur un octet soit '256' nuance de dégradé. Par convention le '0' représente le noir et la valeur '255' représente le blanc

I.2.4 image couleur

Dans ce cas des images couleur, un pixel a une valeur triple :le rouge, le vert, le bleu, (R ,V,B) c'est-à-dire une composante rouge, vert, bleu. Et chaque composante est représentée par un entier peuvent prendre la valeur entre '0'et '255'.



Figure I.3 : image couleur

I.2.5 image multi spectrale

Une image multi spectrale d'une scène est constituée d'enregistrements suivant différents canaux spectraux. Les images de la scène ont des ressemblances plus ou moins importantes selon qu'elles contiennent ou non les même informations. Dans une image multi spectrale est la lumière réfléchié par une surface et on peut ainsi accéder a des caractéristique non visibles a l'œil.

Chapitre I

GENERALITE ET PRETRAITEMENT D'IMAGE



Figure I.4 : image multi spectrale

I.2.6 image négative

Pour obtenir une image négative il suffit d'appliquer la fonction suivante : $p'(i, j) = 255 - p(i, j)$ sur l'image originale d'où :

- $p'(i, j)$ c'est les pixels de l'image négative.
- $p(i, j)$ c'est les pixels de l'image originale.
- i c'est la position de la ligne.
- j c'est la position de la colonne.

La figure suivante nous montre l'image négative :

Chapitre I

GENERALITE ET PRETRAITEMENT D'IMAGE

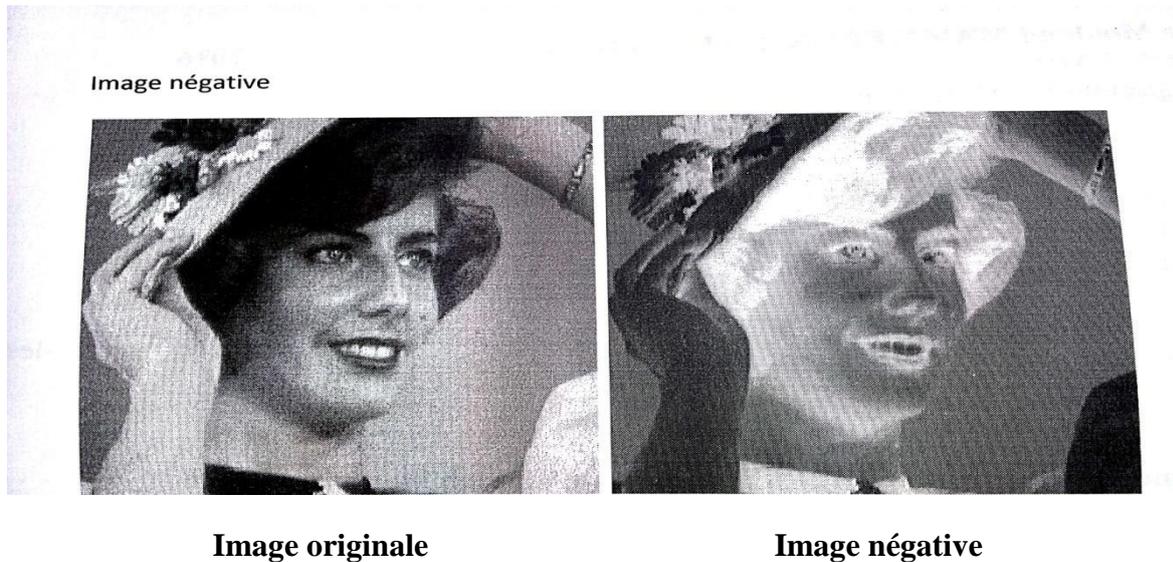


Figure I.5 : image négative

Exemple :

0	10	255	30
20	15	20	200
30	20	45	35
150	100	90	50

$$P'(i, j) = 255 - p(i, j)$$

255	245	0	225
235	240	235	55
225	235	210	220
105	155	165	205

pixels d'une image originale

pixels d'une image négative

I.3 compression d'image

La compression d'image consiste à réduire la taille de l'image afin de la stocker sans occuper beaucoup d'espace ou la transmettre plus rapidement.

Il existe deux types de compression :

I.3.1 compression avec perte :

Dans ce type de compression la restitution de l'image n'est pas totale. Elle est résumée en deux types de classifications.

Chapitre I

GENERALITE ET PRETRAITEMENT D'IMAGE

- **classification en deux classes (seuillage)**

Le seuillage permet de séparer les pixels d'une image en deux classes on utilisant un seuil. Les pixels ayant une valeur inférieure au seuil sont affectés à une classe en revanche, ceux qui ont des valeurs supérieures au seuil sont affectés à une autre classe.

- **Classification en plusieurs classes**

Les pixels de l'image sont rassemblés en un nombre limité de classes les autres de classe sont choisis de façon appropriées. Chaque pixel est affecté à la classe la plus proche en fonction de sa valeur.

I.3.2 compression sans perte

Dans ce cas la restitution de l'image est totale .et les algorithmes les plus utilisés dans la compression sans perte sont l'algorithme **RUN LENGTH ENCODING (RLE)** et l'algorithme de **HUFFMAN**.

- **Algorithme RUN LENGTH ENCODING (RLE)**

C'est un algorithme de codage de longueur. Il permet de regrouper les pixels identiques et consécutifs, et les remplace par le couple (nombre de pixel identique et consécutif ; valeur de pixels). En général cet algorithme est appliqué sur les images présentant des régions homogènes. Et dans le codage RLE on a trois orientations : (en ligne, colonnes et zig zag).

Exemple : la matrice ci-dessous représente les pixels d'une image et pour compresser cette image on utilise l'algorithme RLE. Et la taille de cette image est = 120 octets

15	15	15	15	120	120	120	120	255	255	255	255
15	15	15	120	120	120	120	255	255	255	255	100
15	15	120	120	120	120	255	255	255	255	100	100
15	120	120	120	120	255	255	255	255	255	100	100
120	120	120	120	255	255	255	255	255	100	100	100
120	120	120	120	255	255	255	255	100	100	100	35
120	120	120	255	255	255	255	100	100	100	35	35
120	120	255	255	255	255	100	100	100	35	35	35
120	255	255	255	255	100	100	100	35	35	35	35
255	255	255	255	100	100	100	100	35	35	15	35

Pixels d'une image originale

Chapitre I

GENERALITE ET PRETRAITEMENT D'IMAGE

On applique l'algorithme RLE sur l'image en donne les résultats comme suite :

Orientation en ligne : [4 ;15. 4 ;120. 4 ;255. 3 ;15. 4 ;120. 4 ; 255. 128 ; 100. 2 ; 15. 4 ; 120. 4 ; 255. 2 ; 100. 128 ; 15. 4 ; 120. 5 ; 255. 2 ; 100. 4 ; 120. 5 ; 255. 3 ; 100. 4 ; 120. 4 ; 255. 3 ; 100. 128 ; 35. 3 ; 120. 4 ; 255. 3 ; 100. 2 ; 35. 2 ; 120. 4 ; 255. 3 ; 100. 3 ; 35. 128 ; 120. 4 ; 255. 3 ; 100. 4 ; 35. 4 ; 255. 4 ; 100. 2 ; 35. 129 ; 15.35]. la taille de l'image avec cette orientation est =77 octets

Orientation en colonnes : [4 ;15. 5 ;120. 128 ;255. 3 ;15. 5 ;120. 2 ;255. 2 ;15. 5 ;120. 3 ;255. 128 ;15. 5 ;120. 4 ;255. 4 ;120. 5 ;255. 128 ;100. 3 ;120. 5 ;255. 2 ;100. 2 ;120. 5 ;255. 3 ;100. 128 ;120. 5 ;255. 4 ;100. 5 ;255. 3 ;100. 2 ;35. 4 ;255. 3 ;100. 3 ;35. 2 ;255. 4 ;100. 3 ;35. 129 ;15.255. 3 ;100. 5 ;35.] la taille de limage avec cette orientation est =73 octets

Orientation en ZIG ZAG : [10 ; 15. 30 ; 20. 42 ; 255. 24 ; 100. 12 ; 35. 129 ; 15.35.] la taille de l'image avec cette orientation est =13 octets

Donc pour cette image l'algorithme ZIG ZAG est le meilleur pour compresser cette image et la transmettre.

- **Algorithme HUFFMAN**

Il consiste à donner le code le plus court aux pixels, le plus fréquent et le code. Le plus long aux pixels le moins fréquent en passant par les différentes fréquences.il est basé sur la détermination des probabilités d'apparition des pixels.

Exemple : ce tableau représente les pixels d'une image

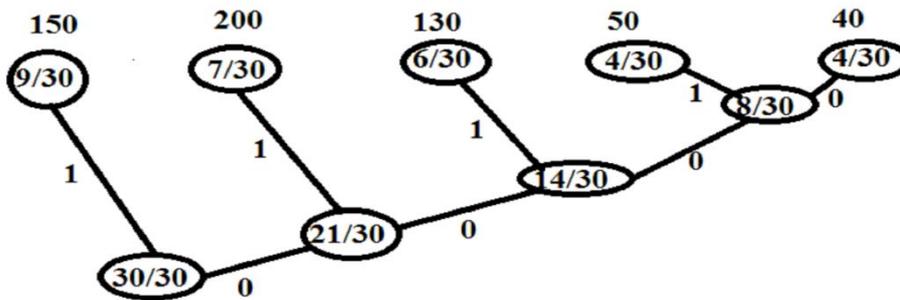
200	200	150	150	200	150
130	150	200	130	130	200
40	50	50	130	40	40
50	40	200	130	200	150
150	150	150	130	150	50

La taille de l'mage est : $6*5*8= 240$ bits =30 octets

Chapitre I

GENERALITE ET PRETRAITEMENT D'IMAGE

On applique l'algorithme HUFFMAN :



Le code de chaque pixel :

Code	Pixel	Nombre de bits
150	1	1 bit
200	01	2bits
130	001	3 bits
50	0001	4bits
40	0000	4 bits

La nouvelle taille de l'image est :

$$(9*1) + (7*2) + (6*3) + (4*4) + (4*4) = 73 \text{ bits} = 9,125 \text{ octets}$$

Donc cet algorithme nous permet de réduire l'image pour la stocker ou la transmettre.

I.4 prétraitement d'image

Dans cette partie, on s'intéresse aux opérations de prétraitement des images. On introduit les notions essentielles de traitement bas niveau des images, illustrées par des techniques classiques. L'objectif de prétraitement d'image est d'améliorer la qualité de l'image après acquisition, en la rendant plus nette par ajustement de contraste, élimination de bruit.....ect

Il existe plusieurs techniques permettant d'améliorer la qualité d'image

Chapitre I

GENERALITE ET PRETRAITEMENT D'IMAGE

I.4.1 extension de dynamique

L'extension de dynamique consiste à étaler la dynamique de l'image dont les valeurs se trouvent dans l'intervalle (minimum, maximale) sur les 256 niveaux de gris disponible. L'étalement permet de discerner les détails invisibles sur l'image originale.

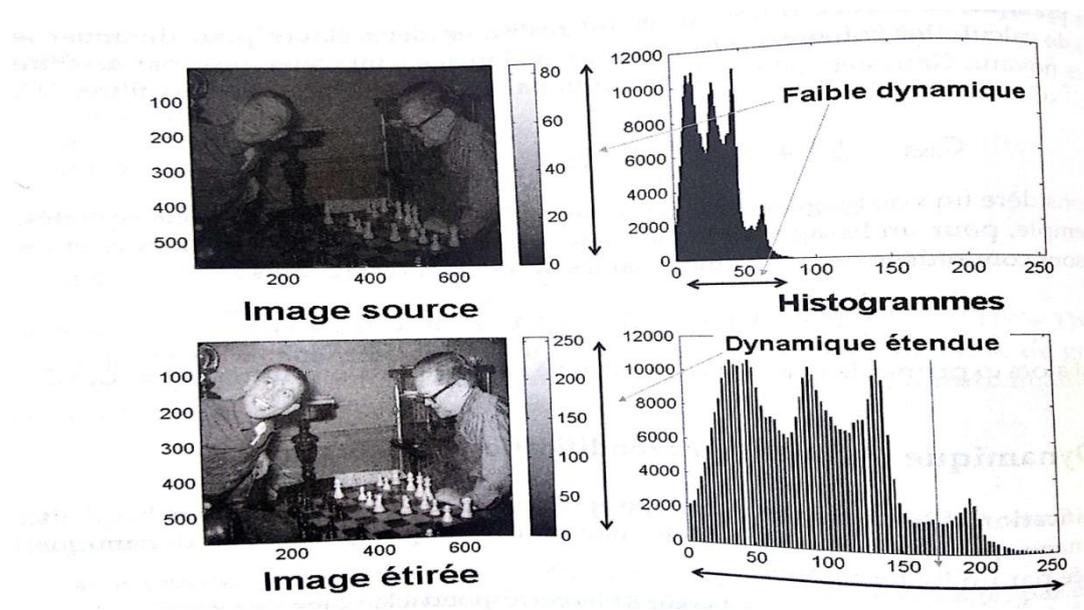


Figure I.6: extension dynamique

I.4.2 égalisation d'histogramme

C'est une méthode utilisée pour ajuster le contraste d'une image numérique. Elle cherche à transformer l'histogramme de l'image en un histogramme uniformément sur l'intervalle [0 255] en augmentant l'effectif des niveaux de gris les moins représentés dans l'image. Cette méthode est basée sur l'utilisation de l'histogramme cumulé.

Chapitre I

GENERALITE ET PRETRAITEMENT D'IMAGE

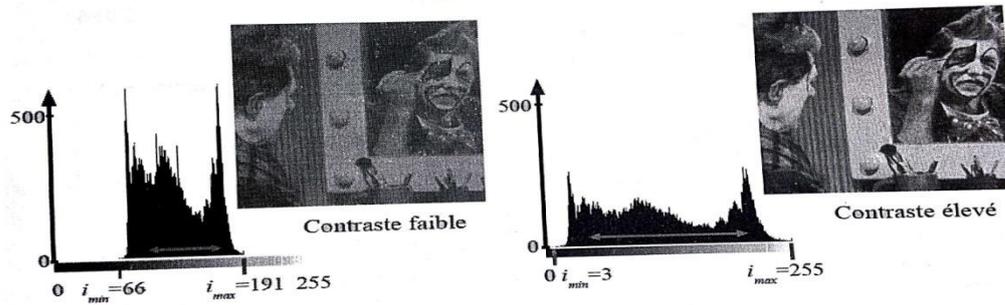


Figure I.7 : égalisation d'histogramme

I.4.3 les filtres

C'est une transformation mathématique permettant de modifier la valeur de pixel en fonction de voisinage. Le filtre est représenté par un masque (fenêtre caractérisée par ses dimensions et ses coefficients).

➤ **filtre passe bas (lissage)**

Utilisé pour éliminer le bruit de l'image et on trouve dans ce cas deux filtres : filtre moyen et filtre médian sont masque est :

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

Tableau 1 : masque de filtre passe bas (moyenne)

Chapitre I

GENERALITE ET PRETRAITEMENT D'IMAGE

Exemple sur le filtre moyen :

4	2	3	10
15	10	215	10
10	13	10	10
9	200	9	9
4	4	6	4

(4+2+3+15+10+215+10+13+10)/9=31

----->

En applique une fenêtre 3*3

4	2	3	10
15	31	31	10
10	55	54	10
9	29	29	9
4	4	6	4

Exemple sur le filtre médian :

4	2	3	10
15	10	215	10
10	13	10	10
9	200	9	9
4	4	6	4

2 3 4 10 **10** 10 13 15 215

----->

On applique une fenêtre 3*3

4	2	3	10
15	10	10	10
10	10	10	10
9	9	9	9
4	4	6	4

Le filtre moyen perturbe les contrastes de l'image alors que le filtre médian les conserve. et si on applique plusieurs fois le filtre basse pas sur une image, l'image obtenue se dégrade et elle devient floue.

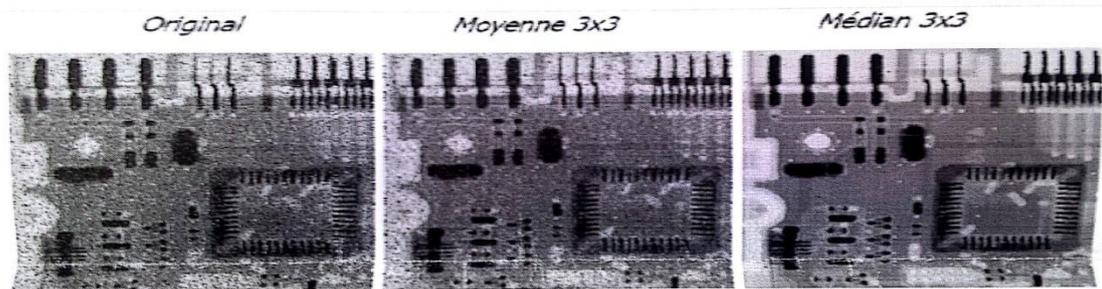


Figure I.8 : image avec filtre passe bas (lissage)

Chapitre I

GENERALITE ET PRETRAITEMENT D'IMAGE

➤ Filtre passe haut

Le filtre passe haut est contrairement au filtre passe bas. Il permet de mettre en évidence les détails de l'image son masque est :

0	-1	0
-1	5	-1
0	-1	0

Tableau 2 : masque filtre passe haut

Une fois la fenêtre est sélectionnée sur l'image on additionne les différents produit obtenus (coefficient de masque * les pixels des fenêtres de même position).

Exemple sur le filtre passe haut :

10	12	10	10	$(0*10)+(-1*12)+(0*10)+(-1*15)+(5*15)+(-1*10)+$ $(0*15)+(-1*10)+(0*11)=28$ =====>	10	12	10	10
15	15	10	12		15	28	2	12
15	10	11	13		15	0	10	13
11	10	12	12		11	5	16	12
12	12	11	10		12	12	11	10



Image originale

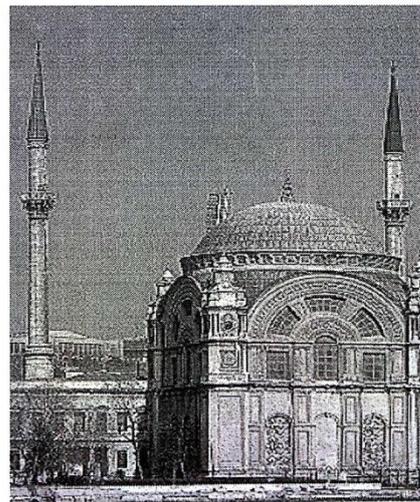


image filtrée (passe haut)

Figure I.9 : image avec filtre passe haut

Chapitre I

GENERALITE ET PRETRAITEMENT D'IMAGE

Conclusion :

Dans ce chapitre nous avons présenté les généralités et les différents types d'image existant, Par la suite nous avons donné les définitions les plus pertinentes dans l'imagerie.

Nous avons également présenté les opérations les plus utilisées dans ce domaine du traitement d'image, dont le prétraitement d'image qu'on a expliqué par des exemples. Donc l'image est un instrument révolutionnaire qui permet non seulement l'observation et la description mais aussi la manipulation indirecte d'objet qui autrement serait hors de notre portée et notre regard.

CHAPITRE II

LES METHODES D'ESTIMATION DU MOUVEMENT

Préambule

Ce chapitre est consacré à la description de différentes méthodes d'estimation du mouvement. Dans la suite de ce chapitre nous définirons le mouvement. Ensuite, nous présenterons un bref état de l'art de l'estimation de mouvement suivi des techniques d'estimation de mouvement utilisées.

L'estimation du mouvement revêt une importance capitale dans le domaine de l'analyse des séquences vidéo. L'information de mouvement résultante permet d'améliorer le codage prédictif inter-image. La littérature sur le sujet de l'estimation de mouvement étant extrêmement abondante.

Nous n'énoncerons pas toutes les méthodes existantes. Nous nous contenterons de donner un bref aperçu des méthodes les plus couramment utilisées.

II. 1 Le mouvement

Le mouvement dans une séquence d'image bidimensionnelles (2D) est perceptible grâce aux changements de la distribution spatiale, des intensités lumineuses. Le mouvement ainsi perçu est appelé (apparent), car il ne correspond pas nécessairement à la projection dans le plan de l'image du mouvement ayant lieu dans l'espace tridimensionnel (3D).

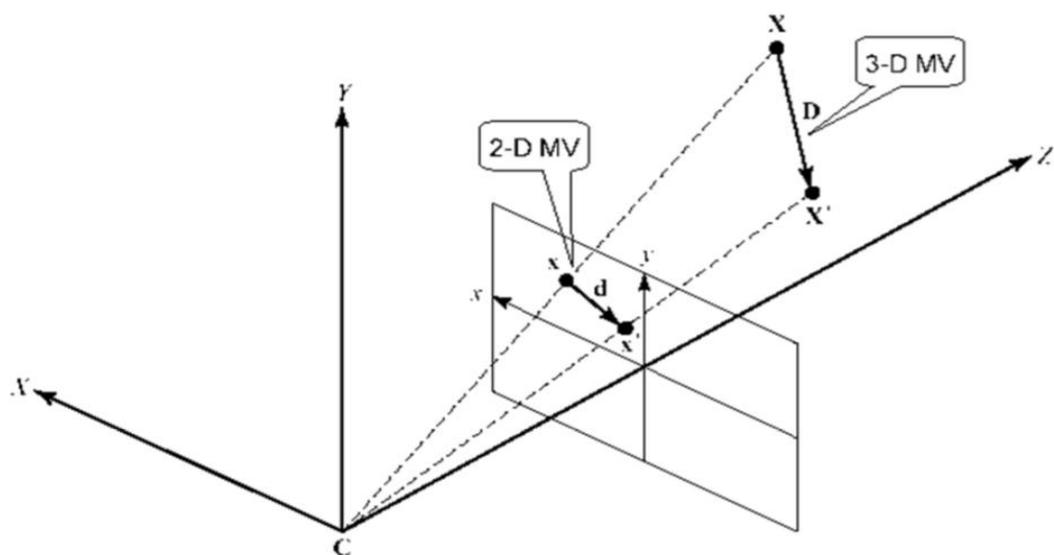


Figure II.1: projection mouvement 3D en 2D [5]

CHAPITRE II

LES METHODES D'ESTIMATION DU MOUVEMENT

II. 2 Classification du mouvement

Les images représentent souvent la projection des scènes réelles 3D. C'est pourquoi le mouvement observé (ou le mouvement apparent) dans une séquence temporelle d'images représente le plus souvent la projection du mouvement 3D, dans le plan image.

Dans ce cas, le mouvement observé correspond à la fois au déplacement dans le plan image et également à la projection sur ce plan d'un mouvement 3D.

Le but de l'estimation de mouvement est donc d'estimer le champ de mouvement à partir d'une séquence d'images. Ce champ estimé est appelé mouvement estimé. On doit donc différencier entre : [2]

- Le mouvement réel.
- Le mouvement apparent ou le mouvement observé.
- Le mouvement estimé.

II.2.1 Le mouvement réel

Le mouvement réel est le mouvement qui se produit dans le repère monde. Il anime la scène réelle, dans l'espace réel 3D. Ce mouvement réel est observé soit par l'œil humain soit par un système d'acquisition. L'inconvénient de ce mouvement est qu'il n'est pas toujours observé lors du passage à l'image. Dans le cas du mouvement réel on distingue principalement deux types de mouvement : [2]

➤ Le mouvement d'objet rigide

Par définition un objet rigide est un objet solide qui garde toujours sa forme. Par conséquent, lorsqu'un tel objet se déplace, tous les points qui le composent se déplacent de façon identique sur six degrés de liberté. Les paramètres du mouvement sont les mêmes pour tous les points de l'objet.

CHAPITRE II

LES METHODES D'ESTIMATION DU MOUVEMENT

➤ Le mouvement d'objets déformables

Un objet déformable est par contre un objet non rigide. La déformation se traduit toujours par un agrandissement ou une réduction de l'objet. Par conséquent, lorsqu'il est en mouvement chacun de ses points peut subir un mouvement différent.

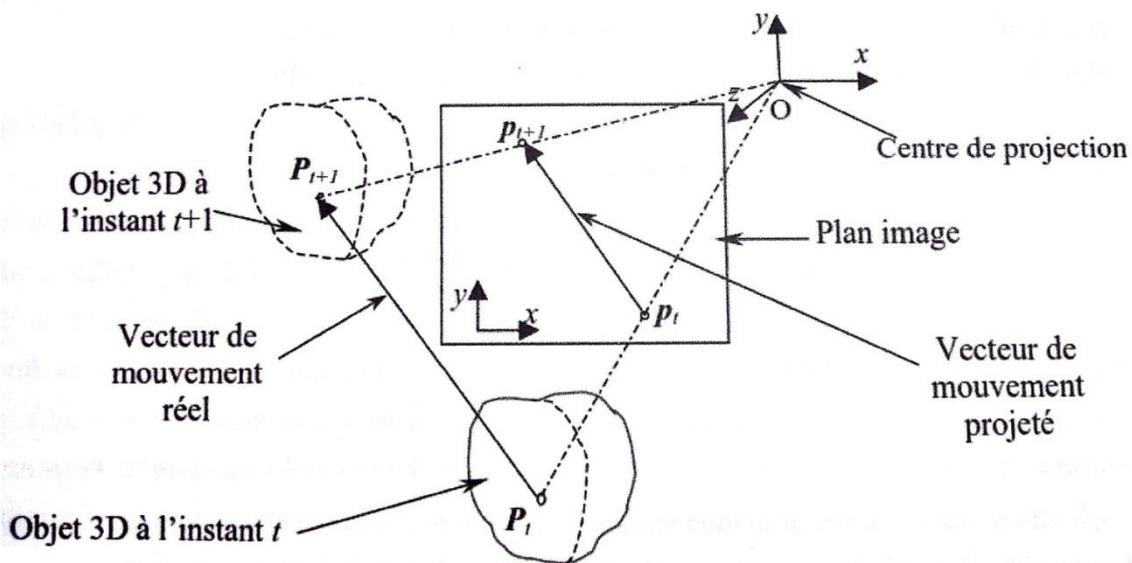


Figure II.2 : illustration des mouvements réel et apparent, dans un système optique de prise de vues [8]

II.2.2 Le mouvement apparent

Le mouvement apparent est une combinaison de deux sources de mouvement : le mouvement propre 3D des objets dans la scène et celui de la caméra. Le mouvement de la caméra correspond au mouvement apparent "global". Alors que le mouvement apparent des objets correspond au mouvement apparent "local" relatif au mouvement global.

Le mouvement apparent est le mouvement qui est observé sur l'image. Il fournit une représentation du mouvement observé sous la forme d'une séquence temporelle d'image.

CHAPITRE II

LES METHODES D'ESTIMATION DU MOUVEMENT

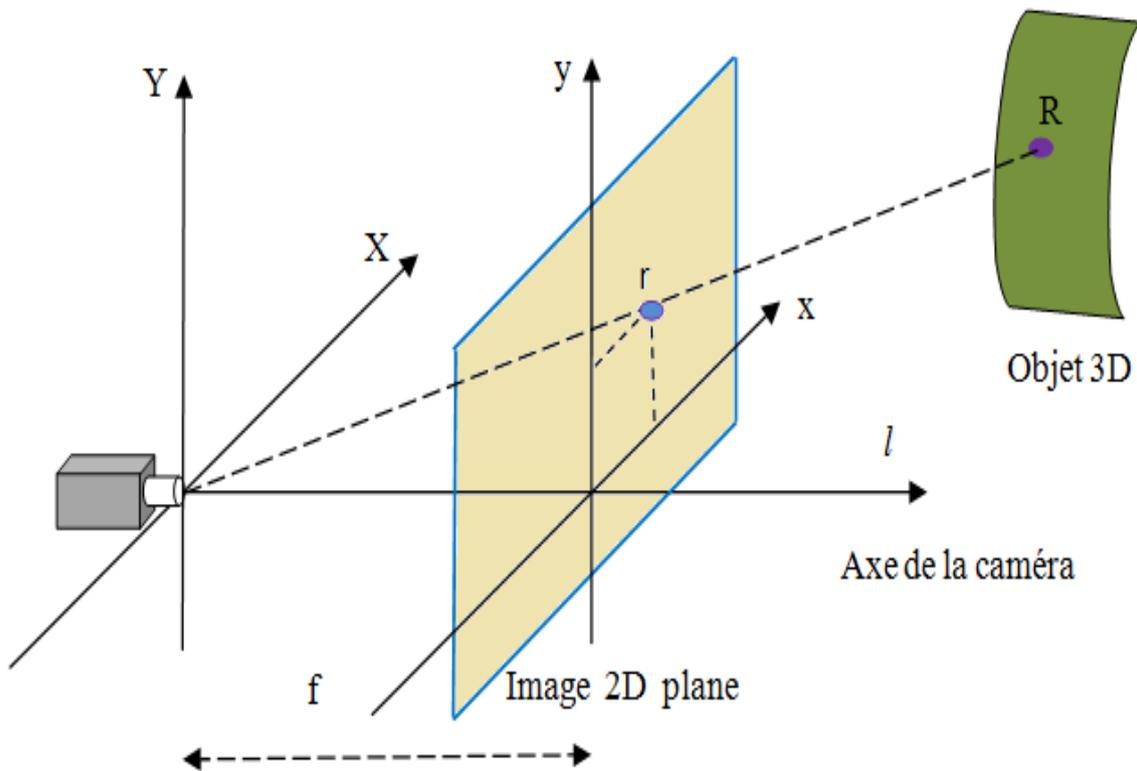


Figure II.3 : Projection d'objet 3D sur un plan 2D [5]

Le champ de mouvement apparent représente la projection du mouvement 3D dans le plan image (voire figure). C'est pourquoi, le champ de mouvement apparent s'appelle aussi .mouvement projeté Il représente une approximation du mouvement réel.

Bien que ce mouvement soit toujours observé, il n'est pas toujours dû à une réalité dans le repère monde. En effet, l'estimation d'un mouvement qui n'existe pas réellement peut avoir lieu.

II.2.3 Le mouvement estimé

L'estimation du mouvement apparent porte uniquement sur ce qui est observé. Ce mouvement estimé est caractérisé par le champ de vecteur vitesse ou par le champ de vecteurs déplacement.

Ces deux champs de vecteurs vitesse et déplacement sont identiques quand l'échantillonnage temporel de la séquence est constant.

CHAPITRE II

LES METHODES D'ESTIMATION DU MOUVEMENT

II.3 L'estimation du mouvement

L'estimation du mouvement est l'un des domaines du traitement d'image qui joue le rôle dans l'analyse des séquences d'images. Aussi c'est un procédé qui consiste à étudier le déplacement des objets dans la séquence vidéo.

Le principe de l'estimation du mouvement consiste à construire une image de prédiction à partir d'une image précédente, et d'informations relative aux déplacements subits par ses composantes.

L'estimation du mouvement n'est pas effectuée sur chaque point de l'image mais sur des macros blocs, afin de réduire au minimum la richesse d'information à transmettre.

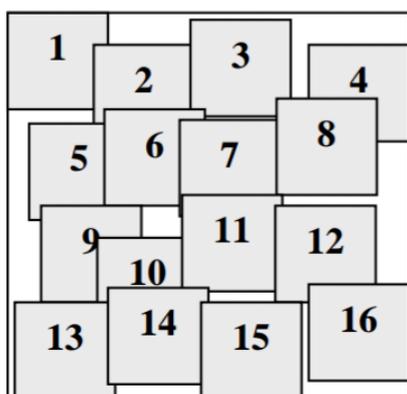
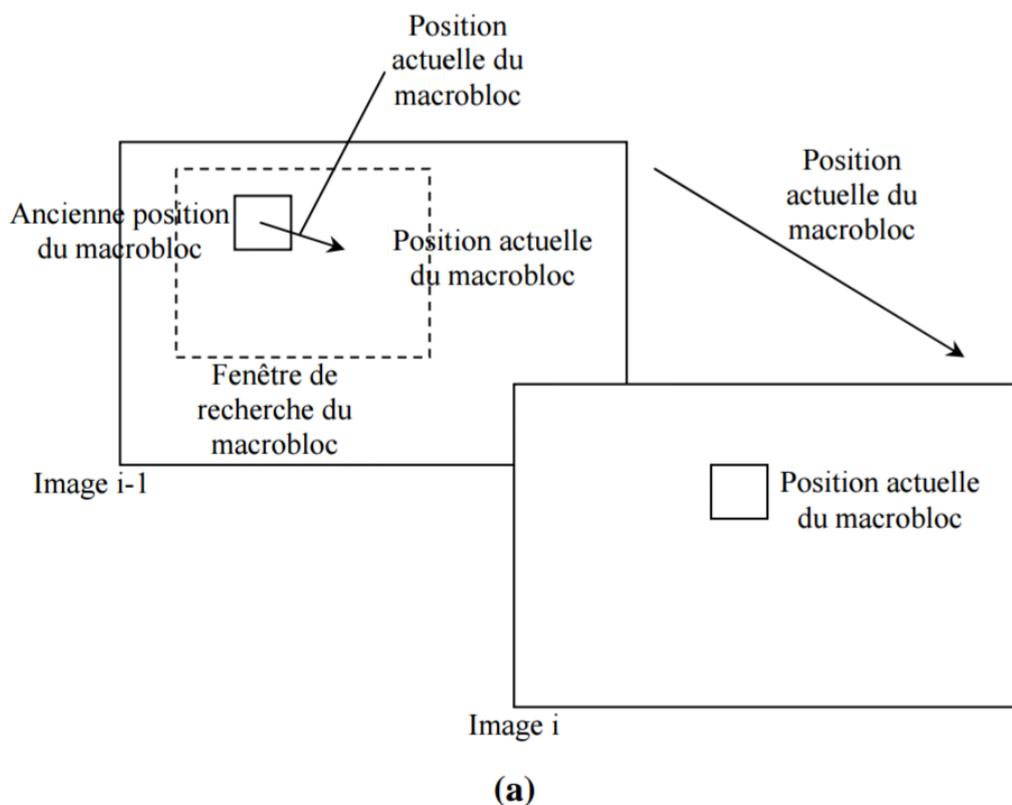
L'estimation de mouvement se décompose en cinq étapes comme suit

- Recherche des macros blocs semblables entre la nouvelle image et l'image précédente.
- Calcul des vecteurs mouvements caractérisant le déplacement des macros blocs.
- Construction d'une image prédite en utilisant ces vecteurs du mouvement.
- Comparaison de cette image prédite avec la vraie nouvelle image pour générer des données d'erreurs de prédiction.
- Codage et transmission des vecteurs et des données d'erreurs de prédiction.

Dans la suite on s'intéressera à l'estimation du champ de vecteur vitesse et vecteur déplacement.

CHAPITRE II

LES METHODES D'ESTIMATION DU MOUVEMENT



Macroblocs de l'image précédente $i-1$ utilisée pour prévoir la nouvelle image i

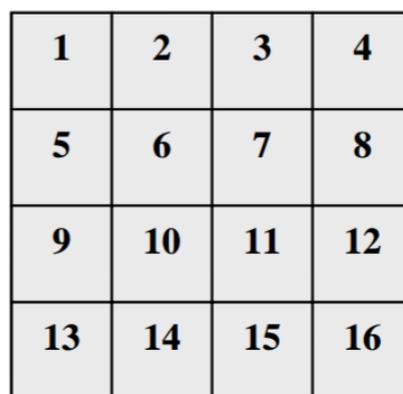


Image i prédite, après application des vecteurs mouvement sur les macroblocs de l'image $i-1$

(b)

Figure II.4 : (a) détection et (b) estimation de mouvement en MPEG [6]

CHAPITRE II

LES METHODES D'ESTIMATION DU MOUVEMENT

II.3.1 Estimation des vecteurs de déplacement

L'estimation peut être vue comme un problème d'estimation de mouvement direct ou inverse, selon que l'estimation est réalisée entre l'instant (t) et (t +1) ou entre l'instant (t) et (t -1).

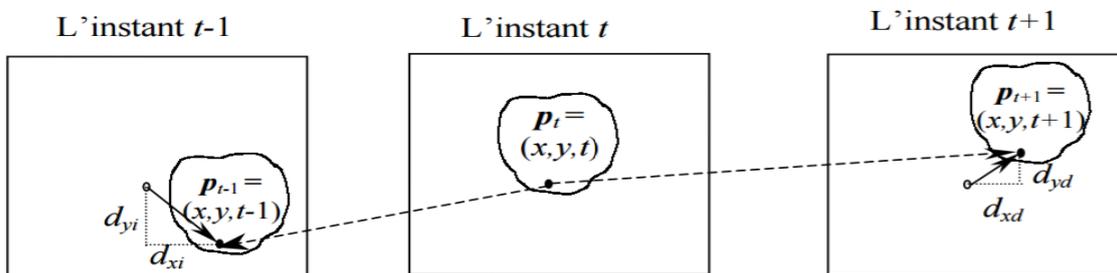


Figure II.5 : estimation directe et inverse des vecteurs déplacements [8]

En estimation de mouvement on utilise généralement l'estimation inverse (arrière). L'estimation avec compensation directe du mouvement est classiquement utilisée dans la compression prédictive de séquence d'images. Les valeurs prises par les déplacements sont souvent réelles ce qui nécessite une étape d'interpolation pour l'estimation du mouvement.

II.3.2 Estimation des vecteurs vitesse

Le champ de mouvement estimé peut être caractérisé par le champ de vecteur vitesse ou par le champs de vecteurs déplacement. Le champ de vitesse estimé et le champ de déplacement sont identiques quand l'échantillonnage temporel de la même séquence est constant.

Pour un mouvement accéléré, on doit prendre en compte plus de deux images afin d'estimer correctement le champ de vitesse. Etant donné les échantillons $\mathbf{It}(\mathbf{x}, \mathbf{y})$, on doit déterminer les vecteurs vitesse $\mathbf{Vt}(\mathbf{x}, \mathbf{y}, \mathbf{t})$. On observe que si la vitesse reste constante dans l'intervalle Δt entre deux images ; si Δt est petit, alors la vitesse estimée peut être assimilée au déplacement. [8]

$$\mathbf{Vt}(\mathbf{x}, \mathbf{y}) = d\mathbf{t}(\mathbf{x}, \mathbf{y})/\Delta t$$

CHAPITRE II

LES METHODES D'ESTIMATION DU MOUVEMENT

II.4 Les problème de l'estimation de mouvement

L'estimation de mouvement est très sensible au bruit présent dans les images qui peut être interprété comme étant les résultats d'un mouvement dans la scène réelle.

II.4.1 problème d'occlusion

Le problème de l'occlusion est un problème non résolu dans le traitement de séquence vidéo. Le phénomène d'occlusion dans une séquence vidéo est du à l'apparition et à la description, ou aux croisements d'objets, dans la vidéo. C'est le résultat du mouvement de la caméra et des objets dans la scène.

Les occlusions rendent difficiles l'estimation de mouvement apparent dans la séquence vidéo. En effet, les techniques d'estimation de mouvement entre deux images présentes à deux instants (t) et ($t+1$) d'une séquence vidéo sont basées sur les ressemblances existantes entre ces deux images.

Si la zone apparaît ou disparaît entre les deux images. La ressemblance diminue et l'estimation échoue dans cette zone. [6]

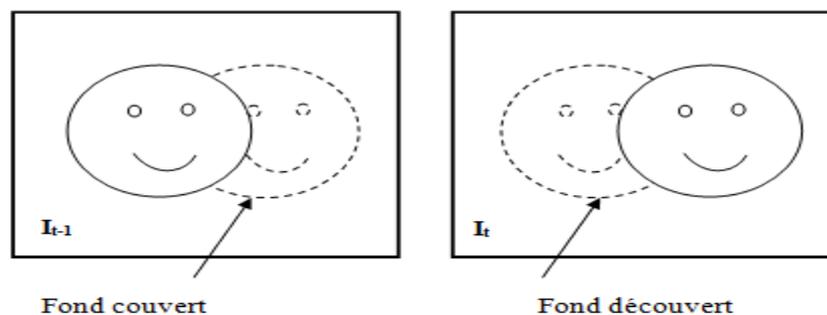


Figure II.6: problème d'occlusion

La région représentée en pointillés dans l'image I_t représente la région du fond qui a été découverte par l'objet en mouvement dans I_{t-1} . Par conséquent, pour les pixels de cette région n'auront aucun correspondant dans l'image I_{t-1} . Les algorithmes d'estimation de mouvement fournissent malgré tout une estimation de mouvement sur cette zone, mais dénuée de tout sens physique.

CHAPITRE II

LES METHODES D'ESTIMATION DU MOUVEMENT

II.4.2 Problème d'ouverture

Le problème d'ouverture représente une formulation du fait que la solution du problème d'estimation de mouvement n'est pas unique. Si on suppose les vecteurs de mouvement en chacun des pixels comme des variables indépendantes, le nombre d'inconnues sera deux fois plus grand que le nombre d'équation disponible.

Le problème d'ouverture est illustré sur la figure, on suppose que l'un des coins d'un objet est en mouvement dans la direction verticale haut. Pour estimer le mouvement on utilise un bloc noté (B1), il n'est pas possible de déterminer si l'objet est en mouvement dans la direction normale au bord de l'objet. Si pour estimer le mouvement on utilise le bloc (B2), on peut déterminer correctement le mouvement, parce que le gradient d'intensité dans ce bloc a deux composantes normales entre elle. Par conséquent, pour éviter le problème d'ouverture on doit utiliser le bloc qui contient suffisamment d'informations traduites pour des variations locales des niveaux de gris.

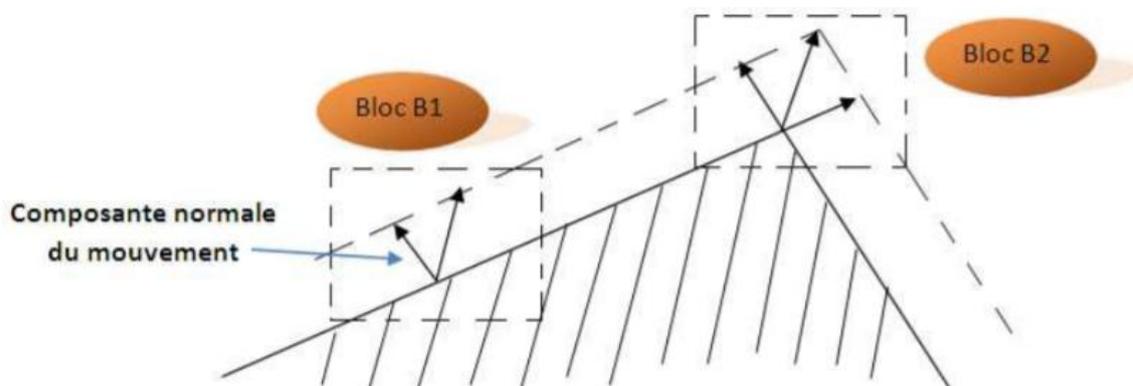


Figure II.7 : illustration du problème d'ouverture [6]

II.4.3 Problème de changement d'éclairage de la scène

La présence de variation de luminance due au changement d'éclairage de la scène est susceptible d'être interprété comme un mouvement par de nombreux algorithmes d'estimation du mouvement et cela même s'il n'existe aucun mouvement réel.

CHAPITRE II

LES METHODES D'ESTIMATION DU MOUVEMENT

II.4.4 problème du modèle de translation

Le modèle de mouvement le plus utilisé dans les méthodes d'estimation du mouvement est le modèle de mouvement constant ou modèle de translation, qui suppose que tous les pixels du bloc effectuent le même déplacement (cas d'objets rigides).

Les avantages de l'estimation de mouvement utilisant un modèle de translation du mouvement sont:

- La compression obtenue en attribuant un seul vecteur de mouvement par bloc.
- La facilité d'implantation hardware.

Par contre, un des inconvénients majeur de l'estimation de mouvement utilisant un modèle de translation du mouvement se traduit par les artefacts de type bloc, dus à un sous-échantillonnage du champ de mouvement.

II.5 Les méthodes d'estimation de mouvement

L'estimation de mouvement dans une séquence vidéo est un vaste champ de recherche qui dépasse largement le cadre de la compression et a motivé de très nombreuses études. On ne fera ici qu'évoquer les différents modèles et types de description.

Il existe différentes techniques d'estimation de mouvement et plusieurs façons d'exprimer les résultats.[4]

II.5.1 Méthodes de mise en correspondance

Les méthodes de mise en correspondance sont parmi les plus utilisées en pratique grâce à la simplicité de l'implantation physique du modèle de translation utilise le plus souvent dans ces méthodes. Elles sont utilisées dans le standards de la compression vidéo.

Parmi les méthodes de mise en correspondance on distingue deux classes :

CHAPITRE II

LES METHODES D'ESTIMATION DU MOUVEMENT

➤ Méthode de mise en correspondance dans le plan transformé

Dans cette méthode le champ de déplacement est estimé par la mise en correspondance de blocs définis dans un plan transformé. La méthode la plus répandue est la méthode de corrélation de phase obtenue par la transformée de Fourier.

➤ Méthode de mise en correspondance dans le plan image

Parmi les méthodes de mise en correspondance dans le plan image, les méthodes de mise en correspondance dans des blocs sont les plus utilisées grâce à leurs simplicités d'implantation. Ce type de méthode s'est imposé dans les standards de compression d'images.

II.5.2 Méthodes statistique

Parmi les méthodes statistiques d'estimation du mouvement les méthodes MARKOVIENNES ou BAYESIENNES sont les plus répandues. Leur principe repose sur une formulation probabiliste du champ de déplacement, l'estimation finale étant une réalisation particulière d'un champ aléatoire.

II.5.3 Méthodes pixels-récurrentes

Les méthodes pixels-récurrentes sont des estimateurs de mouvement de type prédiction correction. Les algorithmes pixels-récurrentes basés gradient produisent un champ de vecteur dense. C'est-à-dire qu'une description du mouvement sera donnée pour chaque pixel de l'image.

II.5.4 Méthode de LUCAS et KANADE

La méthode de LUCAS et KANADE (1986) est une méthode différentielle classique qui consiste à prendre en compte des hypothèses supplémentaires sur un domaine de taille réduite pour particulariser le flot optique. On minimise alors un critère sur un petit domaine, et on obtient ainsi le flot optique de ce petit domaine. Le champ de vitesse est calculé en minimisant au sens des moindres carrés la fonction définie par : [2]

$$E(x, y) = \sum_{(x, y) \in \Omega} w^2(x, y) [\nabla I(x, y, t) \cdot \mathbf{v} + I_t(x, y, t)]^2$$

CHAPITRE II

LES METHODES D'ESTIMATION DU MOUVEMENT

Ω Est le voisinage spatial du pixel de coordonnées (x, y) , dans lequel tous les pixels sont soumis à l'hypothèse du mouvement constant. Le terme $W(x, y)$ représente une fenêtre de pondération qui privilégie d'avantage les pixels situés au centre du voisinage.

II.5.5 méthode de HORN et SCHUNCK

C'est la première méthode de calcul du flot optique développée en 1980 par HORN et SCHUNCK, ce type d'approche consiste à minimiser sur le domaine entier de l'image. Cette méthode n'est pas limitée aux translations. C'est une méthode différentielle intégrative adaptée à l'estimation des petits déplacements en série de TAYLOR, des gradients spatiaux et temporels. Le but est de trouver le champ de vecteurs qui satisfait l'équation de contrainte du mouvement en chaque pixel.

II.5.6 les méthodes basées sur le flot optique

Ces méthodes reposent sur l'hypothèse que le changement de luminosité en chaque point d'une image sont uniquement dus à un déplacement.

Le flot optique est exactement équivalent au mouvement image si les hypothèses suivantes sont respectées :

- Eclairage uniforme.
- Surface à réflectivité lambertienne.
- Mouvement de translation pure, parallèle au plan de l'image.

Ces conditions ne sont bien sûr jamais rigoureusement respectées dans le cas d'images réelles.

II.5.7 les méthodes par appariement de blocs BMA (block matching algorithm)

Les méthodes par appariement de blocs BMA sont de loin les plus utilisées par les standards vidéo, du fait de leur facilité de mise en œuvre et d'un bon compromis entre complexité et efficacité de codage. [9]

CHAPITRE II

LES METHODES D'ESTIMATION DU MOUVEMENT

Elle consiste à subdiviser l'image en blocs et rechercher dans les images les plus proches, le bloc le plus ressemblant au bloc sélectionnée. (Voir la figure) C'est pourquoi on ne s'intéressera qu'à ces dernières méthodes de mise en correspondance.

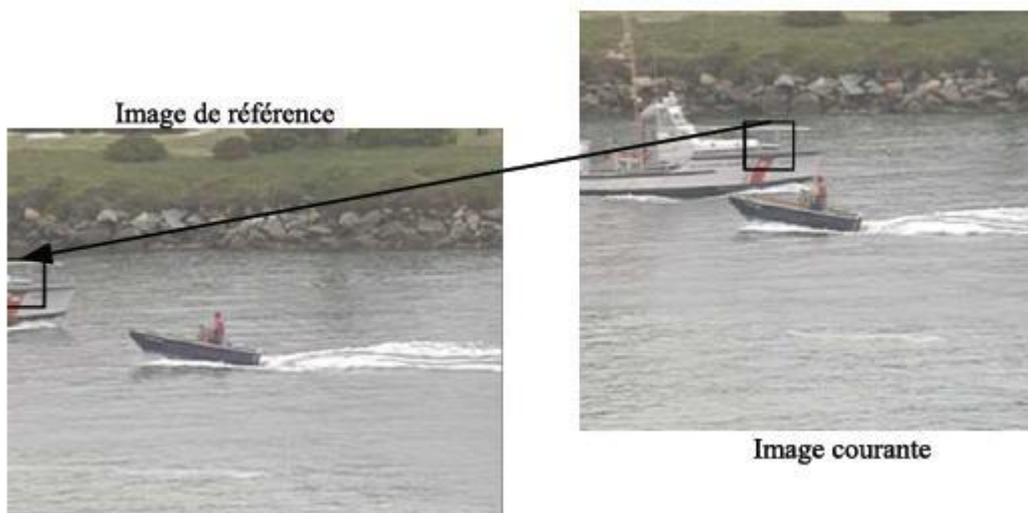


Figure II.8: block matching [9]

II.6 conclusion

Dans ce chapitre, nous avons décrit les types de mouvement représentant l'image par la suite on a défini l'estimation du mouvement ainsi que les différentes estimations de mouvement.

Ensuite nous avons cité les problèmes qui sont rencontrés et posés lors de l'estimation de mouvement. Afin d'obtenir ce dernier, plusieurs types de méthodes existent pour l'estimation de mouvement. D'où nous avons décrit ses méthodes et leur principe de fonctionnement, la méthode du block matching fera l'objet de notre étude dans le chapitre suivant.

CHAPITRE III

LES DIFFERENTES METHODE DE MISE EN CORRESPONDANCE DES BLOCS

Préambule

L'estimation du mouvement est l'un des domaines du traitement d'image. Elle consiste principalement dans l'analyse des séquences d'images et à exploiter les redondances temporelles qui existent entre les images consécutives de la séquence, afin de représenter des vecteurs caractérisant les mouvements existant dans cette séquence.

Les méthodes de block matching sont parmi les méthodes d'estimation du mouvement les plus utilisées dans la pratique. On les trouve presque dans tous les standards de compression vidéo (MPEG-1, MPEG-2, MPEG-4, ...ect). Dans la littérature, l'estimation du mouvement n'est pas utilisée uniquement pour la compression vidéo, mais elle présente aussi un élément principal de plusieurs applications vidéo.

Dans ce chapitre on s'intéressera aux différentes méthodes d'algorithme de block matching et les plus couramment utilisées.

III.1 Principe général du block matching

C'est un algorithme de correspondance de blocs. La méthode consiste à diviser l'image courante (t) en blocs non chevauchés de tailles identiques qui sont vus comme des blocs indépendants où les pixels composant chaque bloc ont le même pas de mouvement (généralement 8×8 ou 16×16 pixels). La deuxième étape consiste à trouver le meilleur bloc cible dans l'image de référence ($t-1$ ou $t+1$) par rapport à un bloc de référence choisi dans l'image courante (t). Le meilleur bloc est choisi par un algorithme minimisant / maximisant un critère de comparaison. Pour réduire le coût des calculs, on définit une fenêtre de recherche qui va limiter le nombre de blocs candidats testés dans l'image de référence. La taille de la fenêtre de recherche dépend du déplacement maximal autorisé pour chaque pixel (choisi par l'utilisateur).

On recommence le procédé avec un autre bloc jusqu'à ce qu'il n'y ait aucun bloc à tester, ou bien, jusqu'à atteindre un critère d'arrêt prédéfini. Une fois la procédure terminée, on obtient pour chaque bloc de l'image courante un vecteur de déplacement qui caractérise son mouvement.

CHAPITRE III

LES DIFFERENTES METHODE DE MISE EN CORRESPONDANCE DES BLOCS

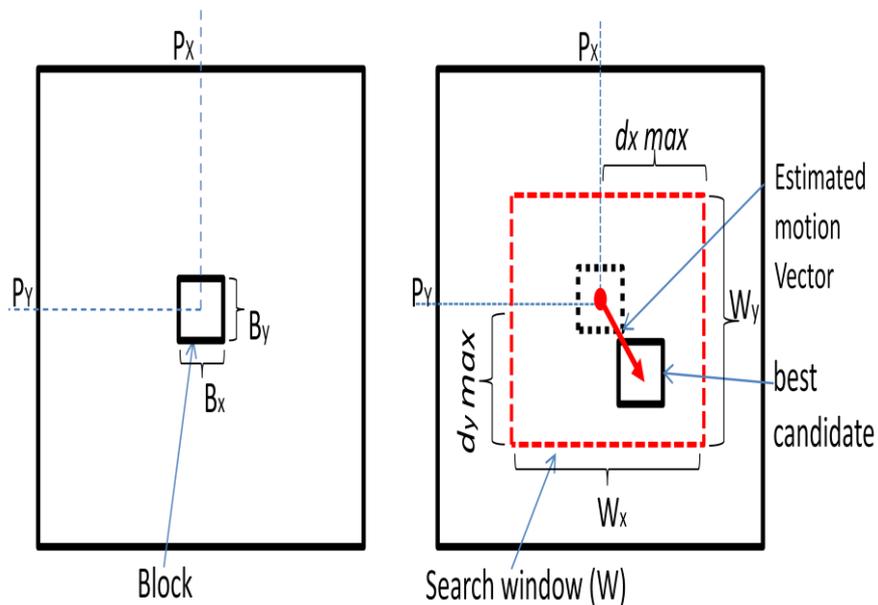


Figure III.1 principe de la méthode de mise en correspondance des blocs

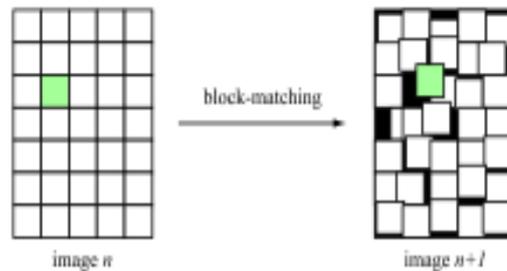
La figure III.1 donne un aperçu global sur le principe de fonctionnement de la méthode du Block Matching (BM).

III.2 Prédiction avant et arrière

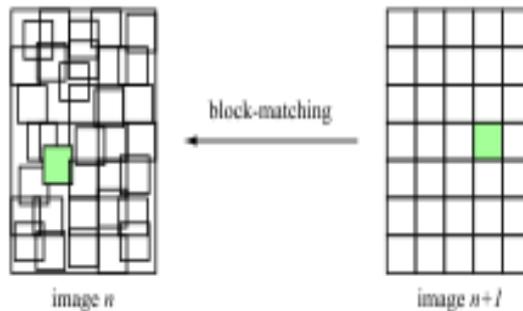
L'algorithme du block matching estime le mouvement des blocs entre deux images aux instants t et $t+1$. Le mouvement calculé va permettre de prédire les blocs de l'image à l'instant $t+1$ grâce aux blocs de l'image à l'instant t et aux vecteurs de mouvements. Deux types de prédictions sont possibles : la prédiction « avant » dite « forward » et la prédiction « arrière » dite « backward ». La prédiction forward consiste à diviser l'image n en blocs et à trouver leur position dans l'images $t + 1$. En revanche, la prédiction backward, divise l'image $t + 1$ en blocs et cherche leur position dans l'image t .

CHAPITRE III

LES DIFFERENTES METHODE DE MISE EN CORRESPONDANCE DES BLOCS



Prédiction avant (forward)



Prédiction arrière (backward)

Figure III.2 prédiction avant et arrière [3]

Les résultats sont très différents. Il apparaît par construction que l'image prédite par prédiction forward présente des « trous ». En effet, de manière générale les blocs n'ont pas un mouvement unique pour toute l'image. Lors de la prédiction certains blocs se recouvrent et laissent apparaître des zones non prédites et par conséquent noires. Ce problème est entièrement résolu par l'utilisation de la prédiction backward qui ne laisse entrevoir aucun « trou » dans la prédiction.

CHAPITRE III

LES DIFFERENTES METHODE DE MISE EN CORRESPONDANCE DES BLOCS

III.3 Critères d'évaluation

Les blocs sont assignés sur la base d'un critère d'évaluation, ce critère n'est autre qu'une mesure de ressemblance (similarité) ou de dissemblance (disparité) entre les blocs testés. Dans le cas d'un critère de ressemblance, ce dernier doit être maximisé, sinon, minimisé dans le cas d'une dissemblance. De manière générale, la plupart des articles traitant de l'estimation du mouvement, avec les algorithmes du Block Matching, considèrent que les blocs sont caractérisés par la luminance (intensité lumineuse). Ceci est dû au fait que l'œil humain est davantage sensible à l'intensité lumineuse qu'à la couleur.

Pour définir le meilleur bloc cible, l'algorithme se base donc sur des critères d'évaluation métrique qui mesurent la différence de contenu entre les deux blocs et les compare avec tous les autres blocs cibles potentiels. Voici des exemples de critères: [2]

- Som of Square Différences (SSD) : somme des différences des pixels de chaque bloc au carré, non adapté à la couleur (critère à rajouter)
- Som of Absolute différences (SAD) : Somme des différences absolues calcule la somme des valeurs absolues des différences de luminance des pixels de l'image originale et celle de référence.
- Zero-mean Normalized Sum of Absolute Differences (ZNSAD) : prend en compte les moyennes des blocs et normalise le résultat (ZNSSD pour SAD)
- Erreur absolue moyenne EAM: traite les erreurs de manière uniforme.
- Coefficient d'inter-corrélation (CC) : le critère de ressemblance statistique le plus utilisé.
- Coefficient d'inter-corrélation normalisé et centré (NCC) : c'est le CC normé et centré à la variance.
- Erreur quadratique moyenne (EQM) : Ce critère est dérivé de l'erreur moyenne absolue.

Les critères les plus couramment utilisés sont données dans le tableau. Les expressions des critères sont données pour une fenêtre de recherche de taille $L \times L$, centrée sur le pixel de coordonnées $P = (P_x, P_y)$ (position initiale) dans l'image de référence. Le déplacement d'un

CHAPITRE III

LES DIFFERENTES METHODE DE MISE EN CORRESPONDANCE DES BLOCS

bloc candidat à l'intérieur de la fenêtre est caractérisé par un vecteur $\vec{d} = (dx, dy)$ et est borné par la taille de la zone de recherche. La position du meilleur bloc candidat est notée (\hat{dx}, \hat{dy}) .

III.4 Stratégies de recherche du meilleur bloc

Il existe plusieurs méthodes du Block Matching pour le parcours du voisinage, afin de trouver le bloc qui correspond au mieux au bloc de référence, et qui optimise le critère d'évaluation. La méthode la plus fondamentale pour trouver le meilleur bloc candidat est de visiter et de tester tous les blocs de la zone de recherche. Cette stratégie, appelée recherche exhaustive, est optimale mais pourrait s'avérer très coûteuse en temps de calcul, surtout pour des amplitudes de déplacement grandes. Cela implique des zones de recherche de taille importante et donc un nombre considérable de blocs candidat à visiter. D'autres balayages non exhaustifs de la zone de recherche ont été proposés dans la littérature. L'objectif de ces techniques est de converger plus rapidement vers le minimum de la fonction de similarité. [3]

III.4.1 algorithme de recherche exhaustive (Full Search (FS))

Cette méthode consiste à faire la recherche de bloc sur l'ensemble des blocs de l'image référence. Cet algorithme est le plus simple, mais aussi le plus coûteux car il n'optimise pas la recherche: il n'y a pas de seuil pour la fenêtre de recherche. Il sert néanmoins de référence pour les autres algorithmes.

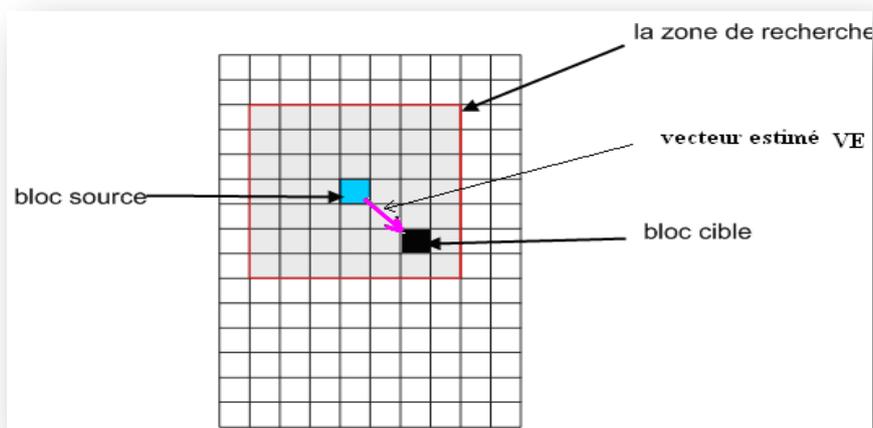


Figure III.3: Full Search Algorithm

CHAPITRE III

LES DIFFERENTES METHODE DE MISE EN CORRESPONDANCE DES BLOCS

III.4.2 algorithme de recherché en trios pas (Three Step Search (TSS))

Cet algorithme a été introduit par Koga et al en 1981. Il est devenu très populaire en raison de sa simplicité et aussi des performances robustes et presque optimales. Il recherche les meilleurs vecteurs de mouvement dans un motif de recherche grossier . L'algorithme peut être décrit comme suit: [3]

Début

- Étape 1: Un premier pas est choisi. Huit blocs à une distance de la taille du pas du centre (autour du bloc central) sont sélectionnés pour comparaison.
- Étape 2: La taille du pas est réduite de moitié. Le centre est déplacé au point avec la distorsion minimale.

Fin

Les étapes 1 et 2 sont répétées jusqu'à ce que la taille de pas soit inférieure à 1. Un chemin particulier pour la convergence de cet algorithme est montré ci-dessous:

CHAPITRE III

LES DIFFERENTES METHODE DE MISE EN CORRESPONDANCE DES BLOCS

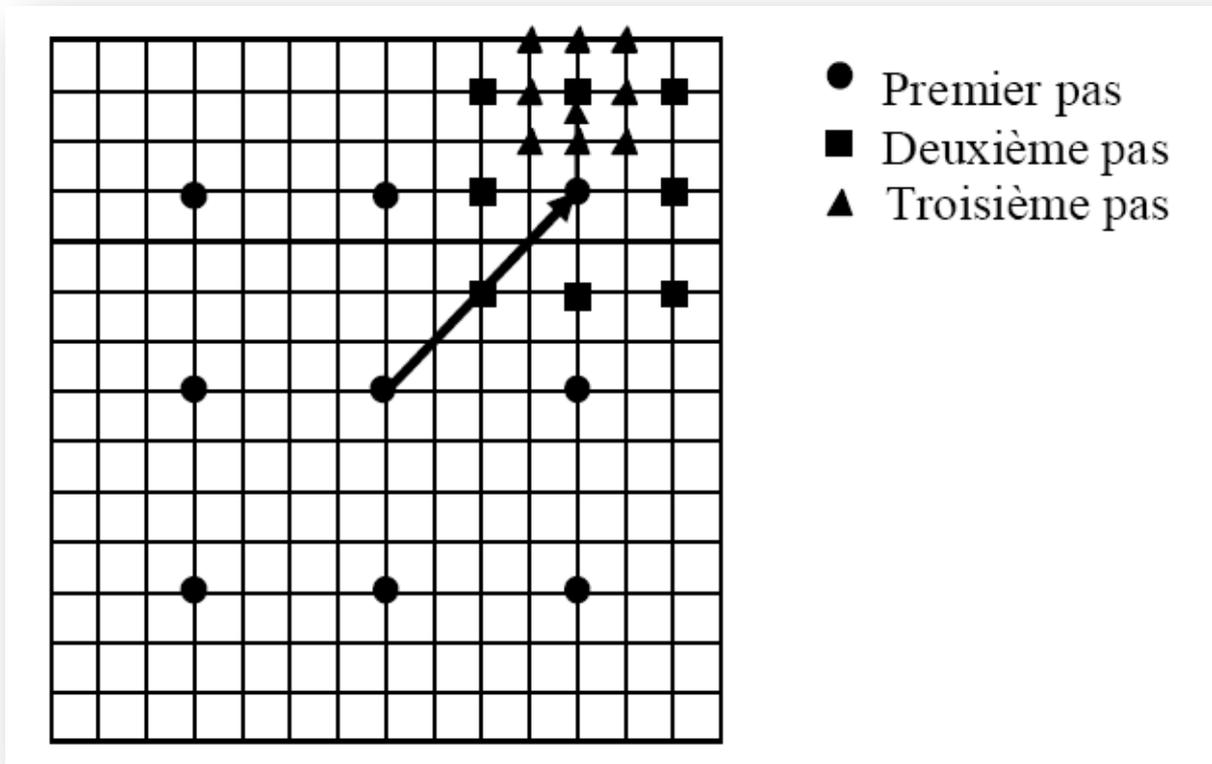


Figure III.4 Three Step Search (TSS)

Un problème qui se produit avec la recherche en trois étapes est qu'il utilise un motif de point de contrôle uniformément alloué dans la première étape, ce qui devient inefficace pour l'estimation de petits mouvements.

III.4.3 Nouvelle recherche en trois pas (New 3SS)

C'est une version modifiée de la recherche en trois pas pour s'adapter aux séquences lentes telle que la vidéo conférence. Il commence par tester neuf points autour du centre; si le minimum reste le centre, il effectue deux autres pas avec un pas de recherche égal à 1. Sinon l'algorithme traditionnel à trois pas est appliqué (figure.). Cet algorithme présente l'avantage de réduire nombre de candidats ainsi qu'une très bonne qualité en cas de simples séquences. Par contre, il risque de tomber dans les minima locaux en cas des blocs non centrés autour du (0,0).

CHAPITRE III

LES DIFFERENTES METHODE DE MISE EN CORRESPONDANCE DES BLOCS

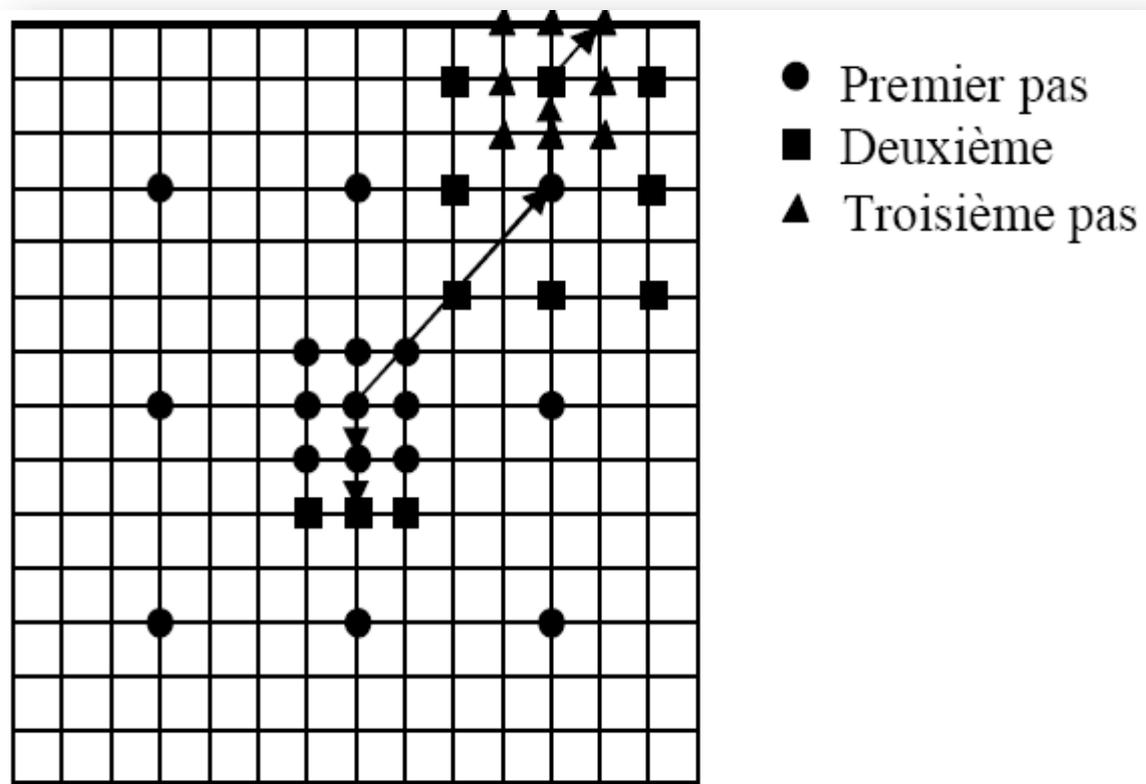


Figure III.5 New Three Step Search (TSS)

III.4.4 Recherche en quatre pas (four step search(FSS))

Cet algorithme a été proposé en 1996 par Lai-Man PO et Wing-Chung Ma. Il est basé sur la caractéristique de la séquence d'image du monde réel du mouvement polarisé au centre.

L'algorithme commence par une comparaison de neuf points ensuite les autres points de comparaison sont sélectionnés en fonction de l'algorithme suivant : [3]

Début

- Étape 1 : commence par un pas de 2. Choisissez neuf points autour du centre de la fenêtre de recherche. Calculez la distorsion et trouvez le point avec la plus petite

CHAPITRE III

LES DIFFERENTES METHODE DE MISE EN CORRESPONDANCE DES BLOCS

distorsion. Si ce point est le centre de la zone de recherche, passez a l'étape 4, si non passez a l'étape 2.

- Étape 2: Déplacez le centre vers le point avec la plus petite distorsion. La taille de pas est maintenue à 2. Le motif de recherche, cependant, dépend de la position de la distorsion minimale précédente
 - A) Si le point minimum précédent est situé au coin de la zone de recherche précédente, cinq points sont sélectionnés (comme indiqué sur la figure).
 - B) Si le point de distorsion minimum précédent est situé au milieu de l'axe horizontal ou vertical de la fenêtre de recherche précédente, trois points de contrôle supplémentaires sont sélectionnés. (Comme le montre la figure). Localisez le point avec la distorsion minimale. Si c'est au centre, passez à l'étape 4 sinon passez à l'étape 3.
- Étape 3: La stratégie de recherche est la même, mais elle passera finalement à l'étape 4.
- Étape 4: La taille du pas est réduite à 1 et les neuf points autour du centre de la recherche sont examinés.

Fin.

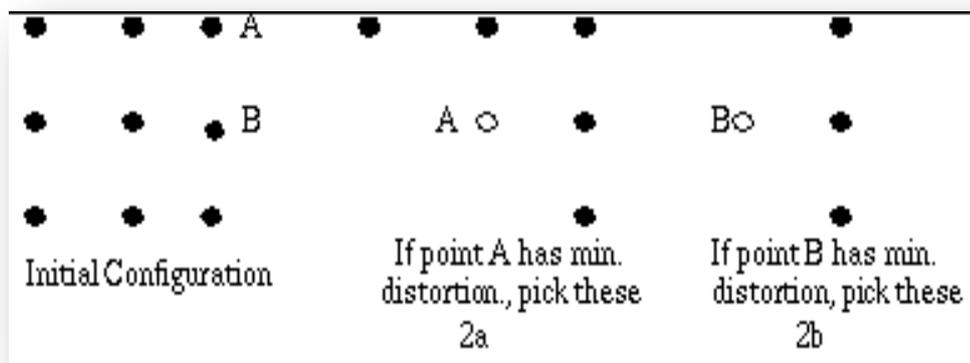


Figure III.6 Illustration de sélection des blocs en quatre étapes de recherche

CHAPITRE III

LES DIFFERENTES METHODE DE MISE EN CORRESPONDANCE DES BLOCS

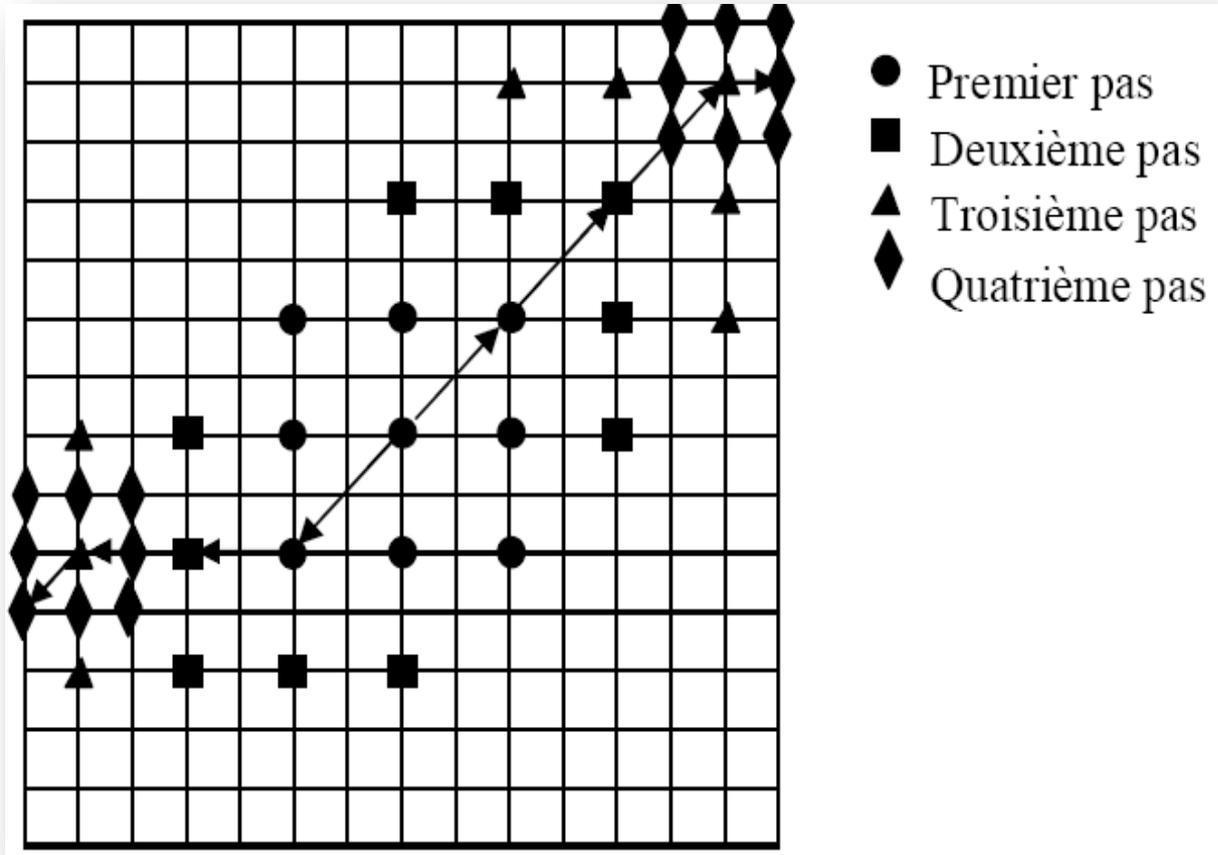


Figure III.7 four step search (FSS)

La complexité computationnelle de la recherche en quatre étapes est inférieure à celle de la recherche en trois étapes, alors que la performance en termes de qualité est aussi bonne. Il est également plus robuste que la recherche en trois étapes et il maintient sa performance pour les séquences d'images avec des mouvements complexes comme le zoom de caméra et le mouvement rapide. Il s'agit donc d'une stratégie très attractive pour l'estimation du mouvement.

III.4.5 Recherche selon le gradient (Gradient Search GS)

Cet algorithme a été proposé par Liu et Feig, 1996. Cet algorithme est basé sur l'utilisation d'un pas de recherche fixe et le passage d'un bloc candidat à l'autre se fait en

CHAPITRE III

LES DIFFERENTES METHODE DE MISE EN CORRESPONDANCE DES BLOCS

suivant la direction indiquée par la descente du gradient de la fonction de coût utilisée. C'est un algorithme qui fonctionne efficacement dans le cas des séquences stationnaires.

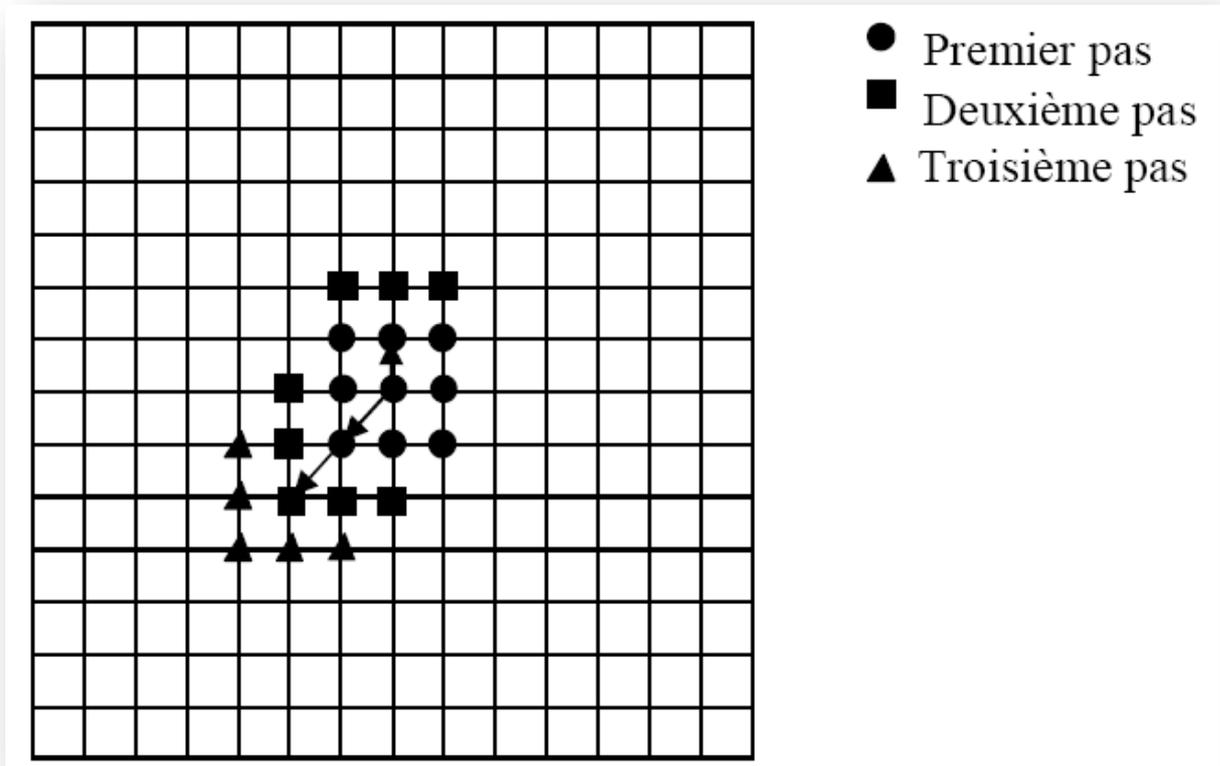


Figure III.8 recherche selon le gradient

III.4.6 algorithme de recherche en diamant

Cette algorithme est proposé par A.M. Tourapis et al, 2000 .L'algorithme de recherche en diamant est un algorithme rapide d'estimation de mouvement de correspondance de bloc, il est base sur l'étude de la distribution des vecteurs de mouvement de plusieurs séquence vidéo.

L'algorithme DS utilise deux modèles de recherche. Un modèle de recherche en grand diamant (Large diamond search pattern) de 09 points de recherche comme montré sur la figure (a). et l'autre modèle de recherche en petit diamant (Small diamond search pattern) de 05 points comme montré sur la figure (b).

CHAPITRE III

LES DIFFERENTES METHODE DE MISE EN CORRESPONDANCE DES BLOCS

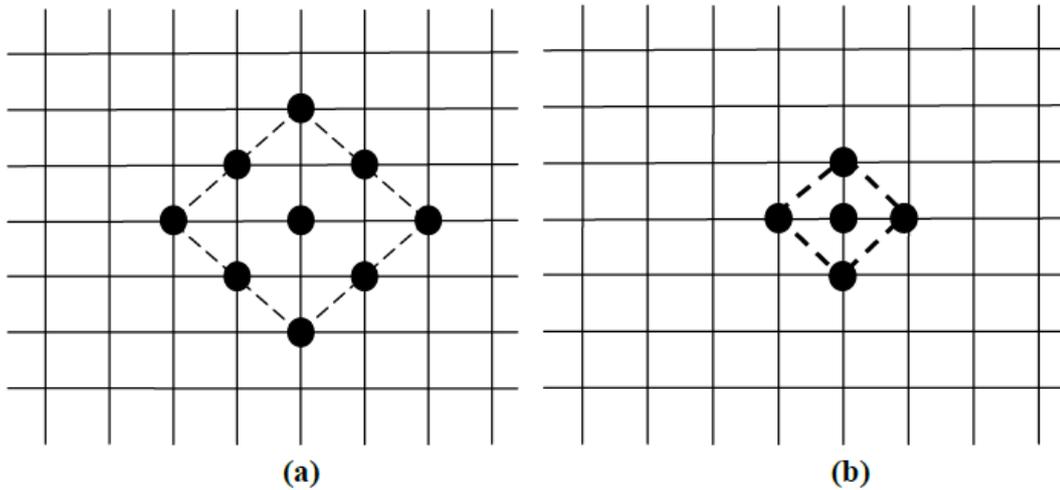


Figure III.9 (a) Modèle de recherche en grand diamant LDSP

(b) Modèle de recherche en petit diamant SDSP

Le modèle de recherche en grand diamant est toujours employé dans la procédure de recherche jusqu'à l'étape dans laquelle le point minimum de la déformation de bloc se produit au centre du diamant. Le modèle de recherche est alors passé à la recherche en petit diamant. Le point rapportant le bloc, parmi les 05 points de contrôle dans le SDSP fournit le vecteur de mouvement du bloc le mieux correspondant.

L'algorithme DS peut se récapituler comme suite :

- Le *LDSP* initial est centré à l'origine de la fenêtre de recherche, et les 9 points de contrôle de *LDSP* sont examinés. Si le point minimum du bloc calculé est situé à la position centrale, passer à l'étape 3; autrement, passer à l'étape 2.
- Le point minimum du bloc trouvé dans l'étape de recherche précédente devient le point central du nouveau *LDSP*. Si le nouveau point minimum du bloc obtenu est situé à la position centrale, passer à l'étape 3; autrement, répéter périodiquement cette étape.
- Changer le modèle de recherche de *LDSP* à *SDSP*. Le point minimum du bloc trouvé dans cette étape est la solution finale du vecteur de mouvement, donc c'est le bloc le plus correspondant.

CHAPITRE III

LES DIFFERENTES METHODE DE MISE EN CORRESPONDANCE DES BLOCS

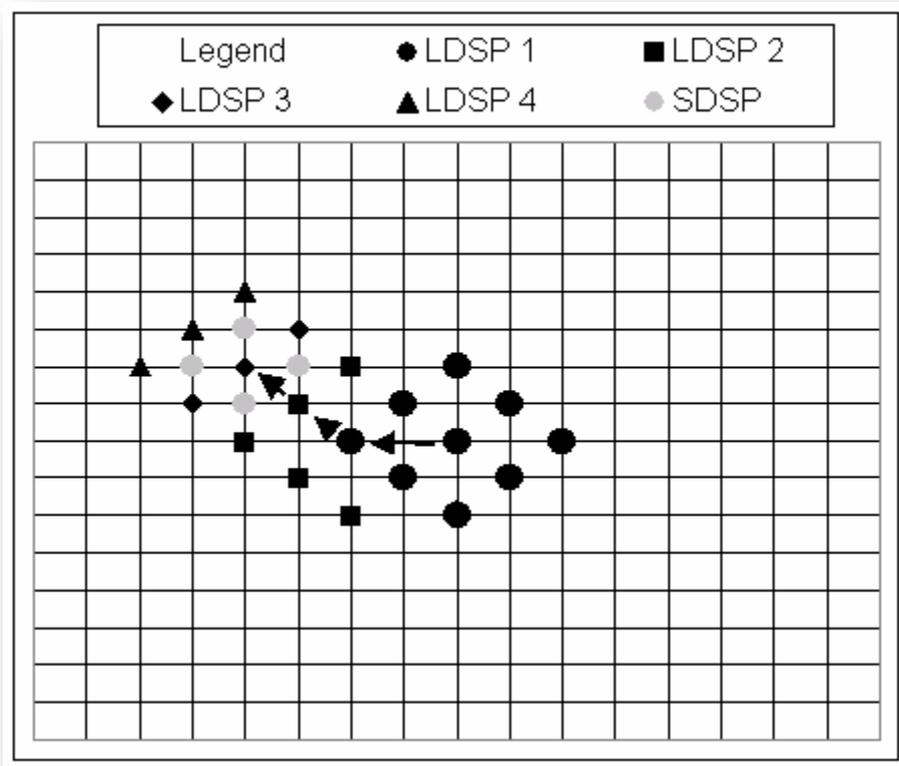


Figure III.10 diamond search algorithm

III.4.7 recherche logarithmique en deux dimensions (TDL)

Cet algorithme a été introduit par Jain & Jain autour du même temps que la recherche en trois étapes a été introduite et est étroitement liée à elle. Bien que cet algorithme nécessite plus d'étapes que la recherche en trois étapes, il peut être plus précis, surtout lorsque la fenêtre de recherche est grande. L'algorithme peut être décrit comme suit: [3]

Début

- Étape 1: Choisissez une taille d'étape initiale. Regardez le bloc au centre de la recherche et les quatre blocs à une distance de s à partir de cela sur les axes X et Y. (Les cinq positions forment un signe +)

CHAPITRE III

LES DIFFERENTES METHODE DE MISE EN CORRESPONDANCE DES BLOCS

- Étape 2: Si la position du meilleur match est au centre, divisez par deux la taille du pas. Si, cependant, l'un des quatre autres points est le meilleur match, alors il devient le centre et l'étape 1 est répétée.
- Étape 3: Lorsque la taille d'étape devient 1, tous les neuf blocs autour du centre sont choisis pour la recherche et le meilleur parmi eux est sélectionné comme le bloc requis.

Fin

Un chemin particulier pour la convergence de l'algorithme est illustré dans la figure suivante:

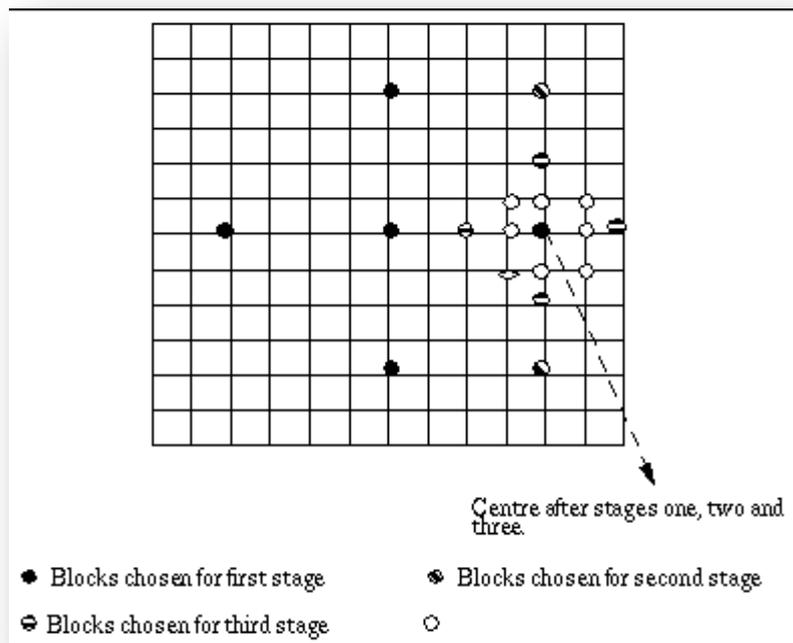


Figure III.11 Two dimensional logarithmic search

III.4.8 recherche binaires (BS)

C'est aussi l'un des algorithmes qui sont très populaires pour l'estimation de mouvement et en fait il est utilisé pour l'estimation de mouvement par MPEG-Tool. L'idée de base de cet

CHAPITRE III

LES DIFFERENTES METHODE DE MISE EN CORRESPONDANCE DES BLOCS

l'algorithme est de diviser la fenêtre de recherche en un certain nombre de régions et de faire une recherche complète uniquement dans l'une de ces régions. Il peut être décrit comme suit:

Début

- Étape 1: Le MAD est évalué sur une grille de 9 pixels qui comprennent le centre, La fenêtre de recherche est divisée en régions sur la base de ces points.
- Étape 2: Une recherche complète est effectuée dans la région correspondant au point avec le plus petit MAD.

Fin.

La convergence de l'algorithme peut être vue dans la figure c ci-dessous. Les pixels qui se trouvent entre les lignes pointillées ne sont jamais considérés. Par conséquent, bien que la recherche binaire nécessite moins de comparaisons (le pire scénario pour cette fenêtre de recherche est de 33 comparaisons), sa performance n'est pas très bonne en raison de cette zone de pixels qui n'est jamais considérée.

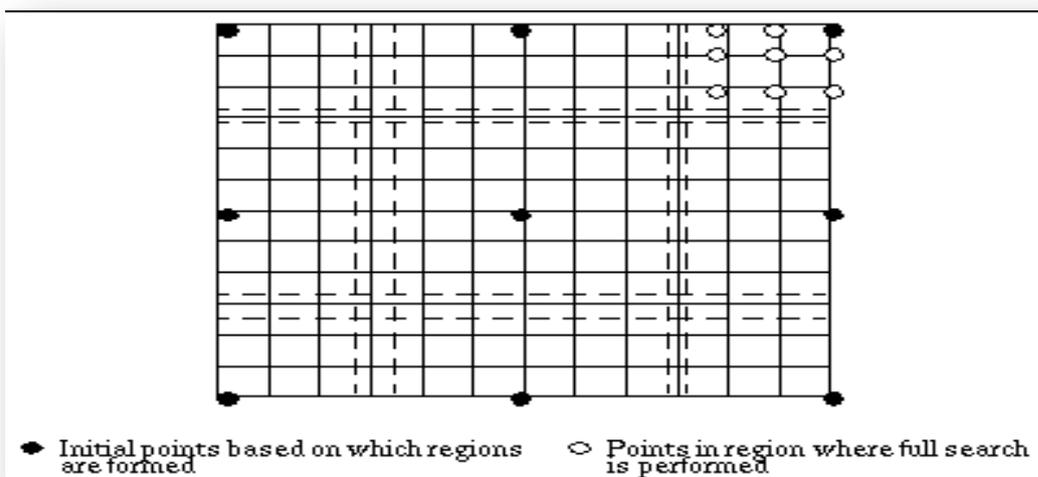


Figure III.12 binary search

CHAPITRE III

LES DIFFERENTES METHODE DE MISE EN CORRESPONDANCE DES BLOCS

III.4.9 algorithme de recherche orthogonale (OSA)

Cet algorithme a été introduit par Puri en 1987 et il est un hybride de la recherche en trois étapes et la recherche logarithmique bidimensionnelle. Il a une étape verticale suivie d'une étape horizontale pour la recherche du bloc optimal. L'algorithme peut être décrit comme suit:
[3]

Début

- Étape 1: Choisissez un pas (généralement la moitié du déplacement maximal dans la fenêtre de recherche). Prenez deux points à une distance de la taille de l'échelon dans la direction horizontale à partir du centre de la fenêtre de recherche et localisez (entre ces) le point de distorsion minimum. Déplacez le centre vers ce point.
- Étape 2: Prenez deux points à une distance de la taille du centre dans la direction verticale et de trouver le point avec la distorsion minimale.
- Étape 3: Réduire de moitié la taille de l'étape, si elle est supérieure à une, sinon arrêt. Un chemin particulier pour la convergence de l'algorithme peut être montré dans la figure suivante :

Fin.

CHAPITRE III

LES DIFFERENTES METHODE DE MISE EN CORRESPONDANCE DES BLOCS

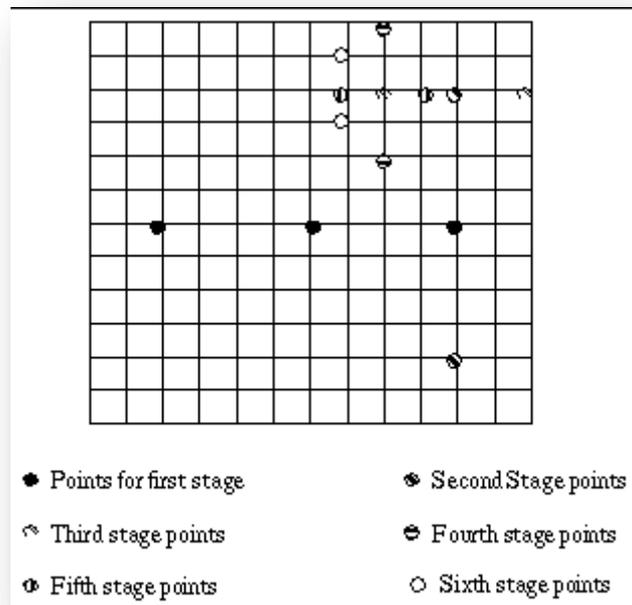


Figure III.13 Orthogonal search algorithm

III.4.10 cross search algorithm

Cet algorithme a été introduit par M. Ghanbari en 1990. L'idée de base de cet algorithme est encore une recherche logarithmique étape, cependant, la principale différence entre cette et la méthode de recherche logarithmique présentée avant est que les lieux de recherche choisis sont les points d'extrémité d'un " X " plutôt qu'un " + ". L'algorithme peut être décrit comme suit: [3]

Début

- Étape 1: Le bloc central est comparé au bloc courant et si la distorsion est inférieure à un certain seuil, l'algorithme s'arrête.
- Étape 2: Choisissez le premier ensemble de points en forme de «x» autour du centre. (La taille de pas choisie est généralement la moitié du déplacement maximal) Déplacez le centre au point de distorsion minimale.
- Étape 3: Si la taille de pas est supérieure à 1, divisez-la de moitié et répétez l'étape 2, sinon passez à l'étape 4.

CHAPITRE III

LES DIFFERENTES METHODE DE MISE EN CORRESPONDANCE DES BLOCS

- Étape 4: Si dans la dernière étape le point de distorsion minimale est le point inférieur gauche ou le point haut droit, évaluer la distorsion à 4 points de plus autour d'elle avec une zone de recherche d'un "+". Si, cependant, le point de distorsion minimal est le point supérieur gauche ou inférieur droit, évaluez la distorsion à 4 points de plus autour de lui sous la forme d'un «x».

Fin.

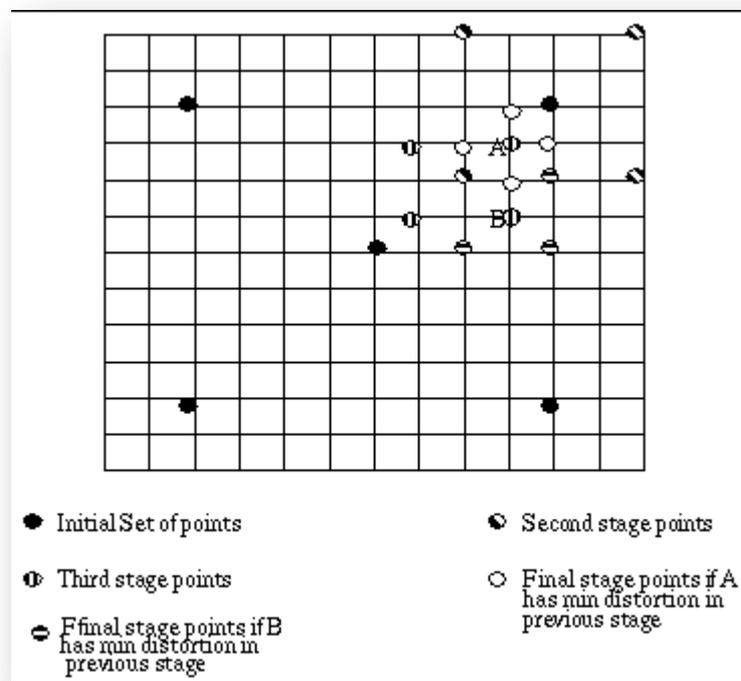


Figure III.14 Cross search algorithm

Un chemin de recherche probable pour l'algorithme est montré à la figure. L'Algorithme a une faible complexité de calcul. Ce n'est toutefois pas le meilleur en termes de compensation.

III.4.11 Spiral search (SS)

L'algorithme de recherche en spirale a été proposé par Zahariadis et Kalivas en 1995. Il cherche à combiner les idées de la recherche en trois étapes et la recherche binaire. Ce faisant, il tend à accélérer non seulement le calcul, mais aussi à supprimer le problème de la recherche

CHAPITRE III

LES DIFFERENTES METHODE DE MISE EN CORRESPONDANCE DES BLOCS

binaire, où il ya une zone de pixels qui n'est jamais évaluée. L'algorithme peut être décrit comme suit:

Début

- Étape 1: La taille du pas est choisie à la moitié du déplacement maximal dans la fenêtre de recherche. Le point de distorsion minimale se trouve parmi les neuf points choisis de la manière suivante. Cinq points sont sélectionnés en forme de "+" autour du centre de la fenêtre de recherche (à une distance de la taille des pas dans les directions verticale et horizontale) Les quatre points restants sont sélectionnés dans les coins de la fenêtre de recherche.
- Étape 2: La taille du pas est réduite et une recherche est effectuée autour du point avec la plus petite distorsion. Ceci est répété jusqu'à ce que la taille de pas tombe à 1.

Fin.

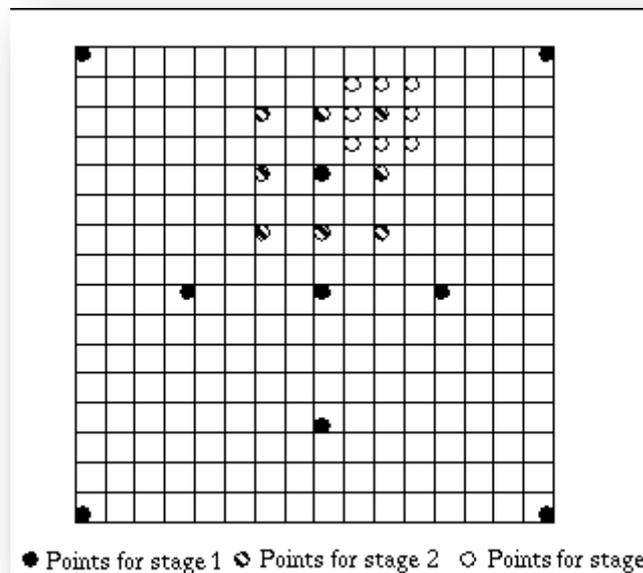


Figure III.15 spiral search

Un chemin de recherche probable pour l'algorithme peut être montré à la figure. La recherche en spirale ne surpasse pas la recherche en quatre étapes, mais cette recherche donne des

CHAPITRE III

LES DIFFERENTES METHODE DE MISE EN CORRESPONDANCE DES BLOCS

performances (en termes de qualité) comparables à la recherche en trois étapes, avec une moindre complexité de calcul.

III.5 Conclusion

Les méthodes du block matching sont parmi les méthodes d'estimation du mouvement les plus utilisées dans la pratique.

Dans ce chapitre, nous avons expliqué les principales techniques de la mise en correspondance de blocs (block matching) ainsi que le principe général de leur fonctionnement. Dans la suite de notre travail nous nous intéressons à l'amélioration de cette méthode.

CHAPITRE IV

LA METHODE PROPOSEE

Préambule

Dans les méthodes du block matching, deux blocs sont mis en correspondance suivant la répartition des informations de leurs pixels. L'hypothèse selon laquelle tous les pixels d'un bloc effectuent le même mouvement de translation constitue à la fois la force et le point faible du block matching. Cette hypothèse peut en effet s'avérer restrictive, notamment lorsqu'une rotation intervient dans l'image.

L'objectif est de développer une méthode de suivi de mouvement de translation et de l'orientation de chaque bloc dans la séquence vidéo. Il s'agit donc de coder aussi bien les orientations des blocs que les translations locales des blocs en mouvement présents dans la séquence vidéo.

Dans ce chapitre, nous présenterons une amélioration de la méthode du block matching pour l'estimation du mouvement local basé sur l'introduction des rotations des blocs.

Afin d'évaluer les performances de l'algorithme proposé ; nous avons fait une étude comparative entre les deux méthodes, la méthode du block matching(BM) et l'algorithme du block matching proposé avec les mêmes paramètres.

IV.1 L'algorithme du block matching proposé

Pour améliorer la méthode de mise en correspondance du bloc, nous avons ajouté des fonctions d'isométrie à savoir la rotation et symétries. Cette technique consiste à affecter à chaque bloc une modification du contraste ainsi que des orientations d'isométrie comme indiqué sur la figure suivante.

CHAPITRE IV

LA METHODE PROPOSEE

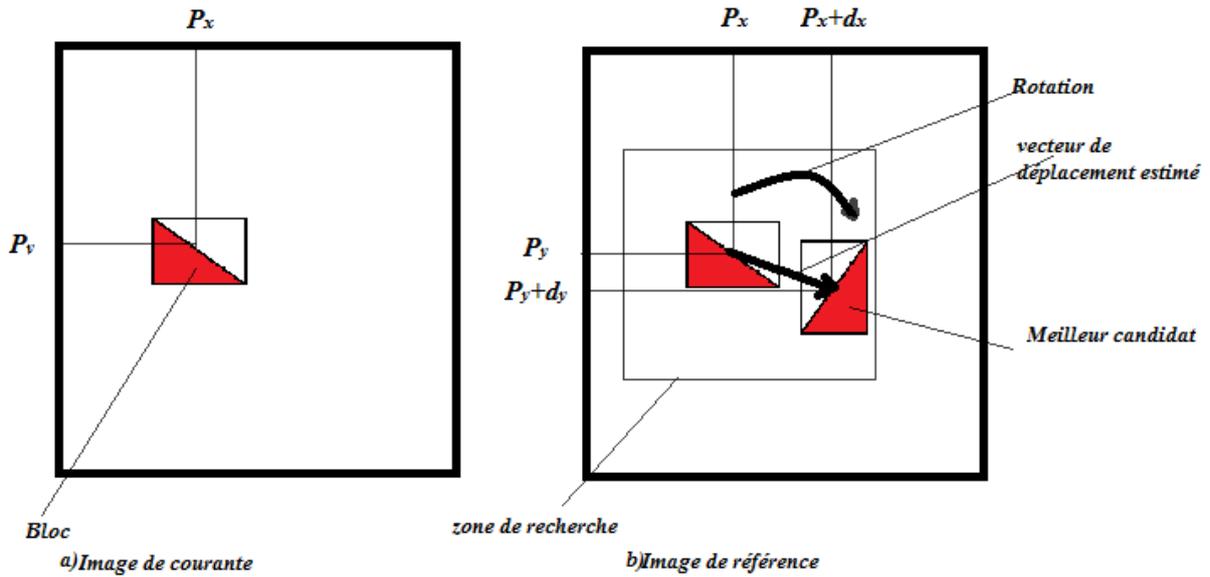


Figure IV.1 : principe de la méthode proposée

Dans notre méthode, les huit fonctions d'isométrie (rotation, symétries) employées dans l'algorithme de Jacquin[10] sont utilisées pour générer les variétés de blocs les plus similaires possibles aux blocs destinations.

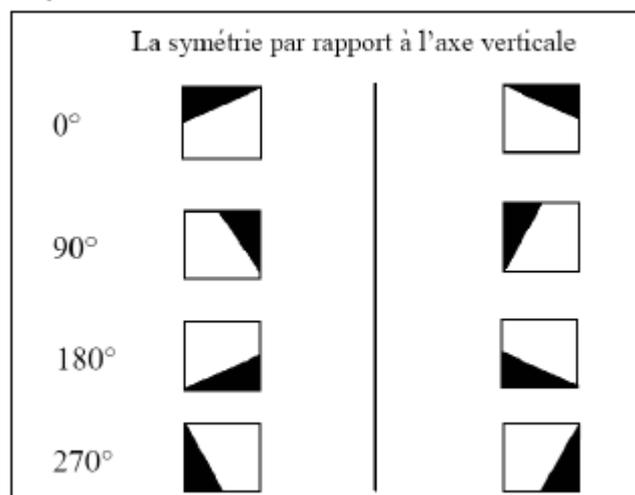


Figure IV.2 : les huit cas d'orientation d'un bloc

Les opérations de rotation ne modifient pas les valeurs de pixel ; elles désorganisent simplement les pixels dans un bloc d'une manière déterministe, nous les appelons les isométries.

CHAPITRE IV

LA METHODE PROPOSEE

Une fois la procédure terminée on obtient pour chaque bloc de l'image courante un vecteur de mouvement et un angle de rotation caractérise le déplacement de chaque bloc.

IV.2 Critères d'évaluation

Les blocs sont assignés sur la base d'un critère n'est autre qu'une mesure de ressemblance (similarité) ou de dissemblance (disparité) entre les blocs testés. Dans le cas d'un critère de ressemblance, ce dernier doit être maximisé, si non minimisé dans le cas d'une dissemblance. Cela dit, il existe plusieurs formules pour le calcule.

L'algorithme cherche pour chacun des blocs I1 de l'image de référence, le bloc I2 choisi dans l'image courante transformé par une transformation d'isométrie Ri(I2). La fonction utilisée pour déterminer le meilleur candidat correspond aux minimums de cette fonction.

$$SAD(p, d) = \sum_{x=-L/2}^{L/2} \sum_{y=-L/2}^{L/2} |I1(Px + x, Py + y) - \alpha * Ri[I2(Px + x + dx, Py + y + dy)]|$$

$$(dx, dy, Ri) = \text{argmin}(SAD(p, d))$$

Ou : I1 et I2 représentent respectivement le bloc destination et le bloc source.

L : représente la taille du bloc.

Ri : représente l'une des sept fonctions d'isométrie utilisées.

▪ L'isométrie :

Il existe huit manières différentes de coller le contenu du bloc source à celui de destination le plus similaire :

Les quatre rotations par les angles : 0°, 90°, 180°, 270° et les quatre symétries pour chaque rotation par rapport à l'axe vertical selon le tableau suivant :

CHAPITRE IV

LA METHODE PROPOSEE

Opération d'isométrie	Application a un bloc D de taille T*T	
Identité 0°	$I1 (D(i,j))=d(i,j)$	0° 
Rotation 90°	$I6 (D(i,j))=D(j,T-1-i)$	90° 
Rotation 180°	$I7(D(i,j))=D(T-1-i,T-1-j)$	180° 
Rotation 270°	$I8 (D(i,j))=D(T-1-j,i)$	270° 
Réflexion par rapport a l'axe vertical	$I2 (D(i,j))=D(i,T-1-j)$	
Réflexion par rapport a l'axe horizontal	$I3 (D(i,j))=D(T-1-i,j)$	
Réflexion par rapport a la première diagonale	$I4 (D(i,j))=D(j,i)$	
Réflexion par rapport a la seconde diagonale	$I5 (D(i,j))=D(T-1-j,T-1-i)$	

Tableau 3 : les fonctions d'isométrie (rotation, symétries)

CHAPITRE IV

LA METHODE PROPOSEE

- **Les modifications du contraste :**

Elle agit seulement sur les valeurs des pixels du bloc source I2 pour être similaire au bloc destination I1. Cette transformation consiste à multiplier les pixels du bloc source par un facteur α appelé facteur d'échelle (modification du contraste) du bloc source avec $\alpha \in [0,1]$

D'où :

$$\alpha = \frac{\sum_{i=0}^{n-1} x_i * y_i}{\sum_{i=0}^{n-1} y_i^2}$$

x_i l'ensemble des pixels du bloc référence

y_i l'ensemble des pixels du bloc courante.

IV.3 Application

Afin d'évaluer les performances de l'algorithme décrit dans ce travail, nous testerons nos technique sur deux gammes d'image : des images de synthèse et des images réelles, en utilisant pour chaque méthode les mêmes paramètres a savoir la taille du bloc et la zone de recherche. Le choix de la dimension du bloc $M*N$ et de la fenêtre de recherche W .

Un critère visuel nous a permis de retenir les tailles du bloc et de la fenêtre suivantes :

$M*N=16*16$, $W=33*33$ pour la séquence de synthèse et $M*N=8*8$, $W =17$ pour la séquence réelle.

De plus, deux critères de performance ont été utilisés pour évaluer nos résultats, à savoir, le **PSNR** s'exprime dans le cas des images en niveaux de gris par :

$$\text{PSNR} = 10 \log_{10} \frac{(255)^2}{\frac{1}{N*N} \sum_{j=0}^{N-1} \sum_{i=0}^{N-1} [p(i,j) - \widetilde{p(i,j)}]^2}$$

Où $N*N$ est la taille de l'image, $p(i,j)$ désigne un pixel de l'image de référence, $\widetilde{p(i,j)}$ désigne un pixel de l'image courante.

D'où $p(i,j) - \widetilde{p(i,j)} = \text{IE}$ (image prédite –image courante)

CHAPITRE IV

LA METHODE PROPOSEE

➤ IV.3.1 Application sur les images de synthèse

Pour tester l'algorithme du blok matching proposé, nous avons utilisé une séquence de synthèse et ce n'est qu'après la validation du fonctionnement du système sur les images de synthèse que l'on pourra passer à la phase de mise en œuvre réelle et l'utilisation d'une vraie séquence. Nous avons utilisé dans la séquence de synthèse des blocs, et afin d'appliquer la méthode du block matching proposé, nous avons fait translater et orienter les blocs dans différents sens dans la deuxième image (figure IV.3)

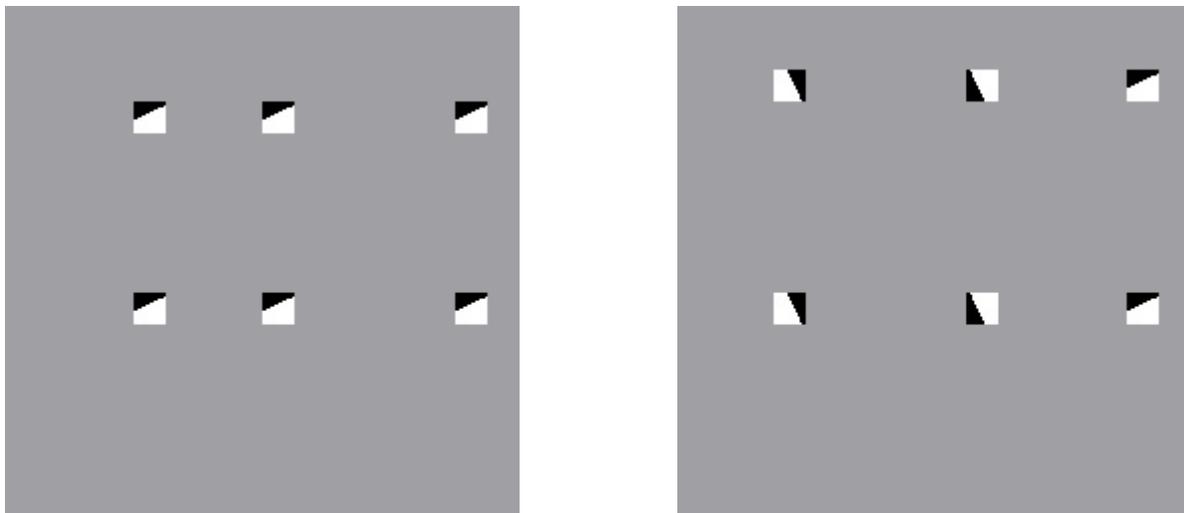


Image 1 a l'instant (t)

image 2 a l'instant (t+1)

Figure IV.3 : séquence de synthèse

IV.3.1.1 Résultats obtenus avec la méthode de block matching

CHAPITRE IV

LA METHODE PROPOSEE

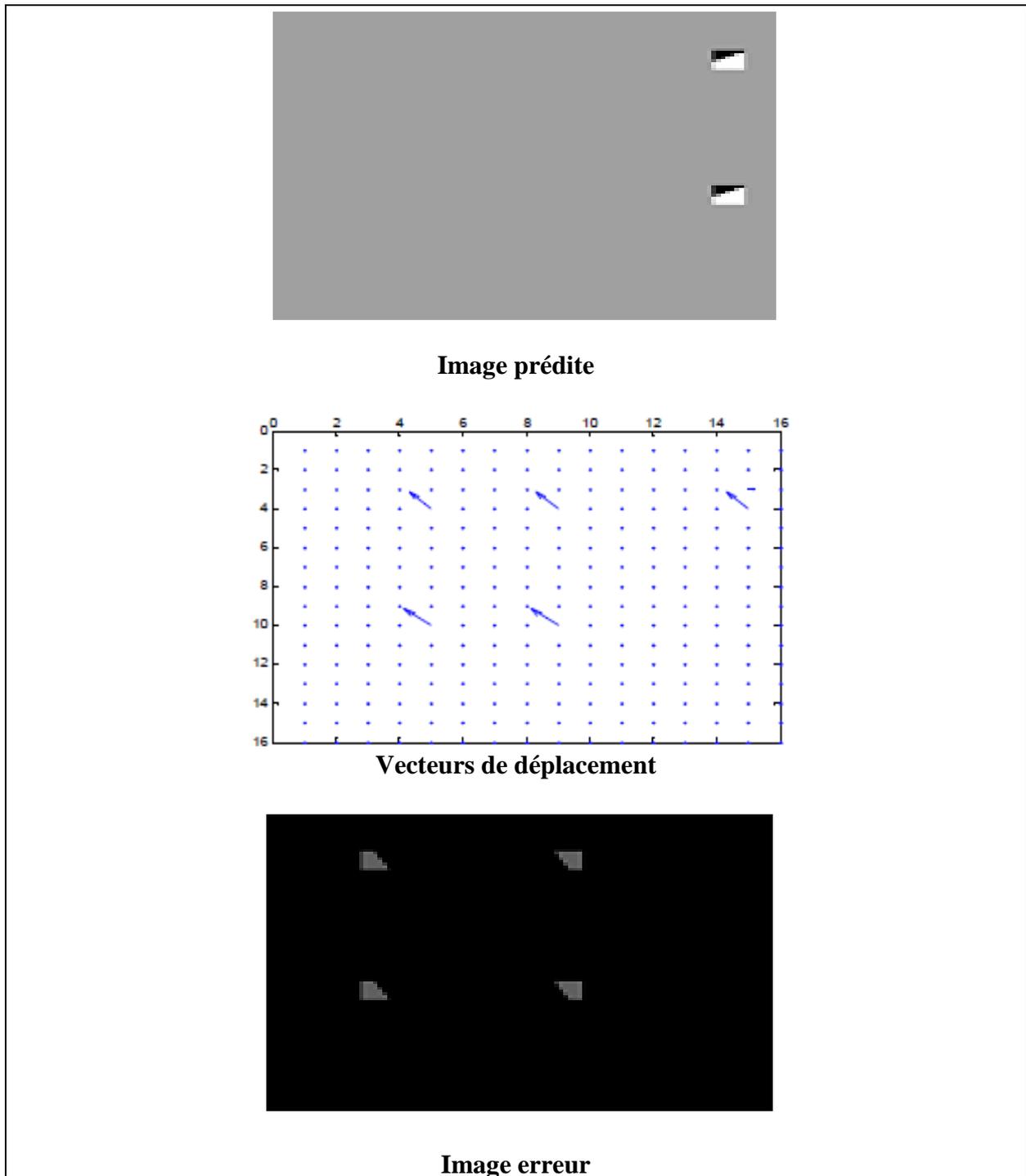


Figure IV.4 : résultats obtenus par l'algorithme de block matching classique

IV.3.1.2 Résultats obtenus avec la méthode proposée

CHAPITRE IV

LA METHODE PROPOSEE

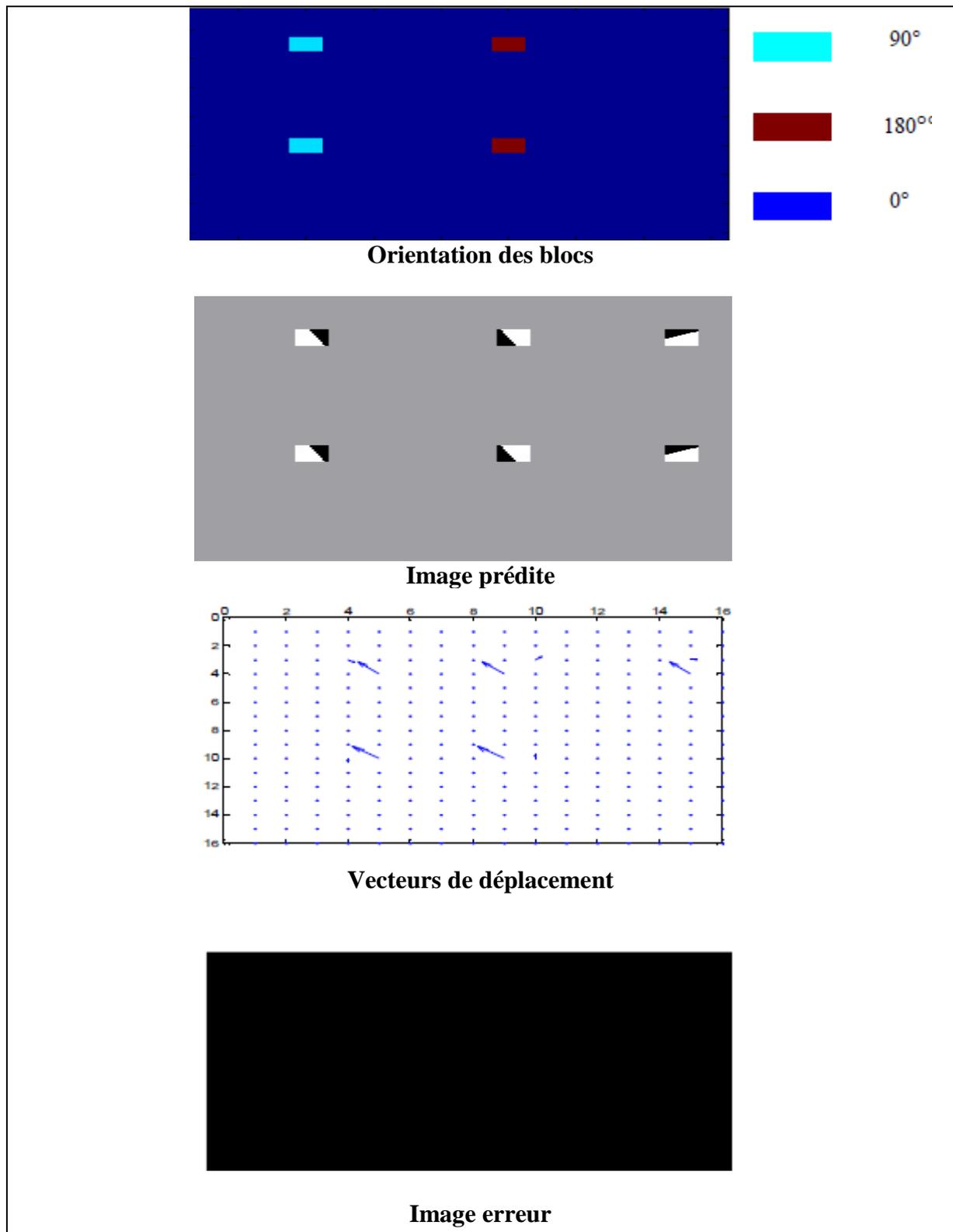


Figure IV.5 : résultats obtenus par la méthode de block matching proposée

CHAPITRE IV

LA METHODE PROPOSEE

	Block matching classique	Méthode proposée
PSNR	24.5363 db	Inf db

Tableau 4 PSNR des images de synthèse

Nous remarquons que la méthode proposée estime bien le mouvement de rotation avec PSNR= inf. l'image prédite correspond exactement à l'image courante, nous pourrions dire qu'avec la méthode du block matching proposée la direction des vecteurs suit correctement le sens du mouvement. Les différents résultats obtenus pour les images de synthèse sont très appréciables. En comparant les résultats obtenus sur les images de synthèse avec les résultats obtenus avec la méthode du block matching, nous confirmons les performances de la méthode du block matching proposée.

En effet les différentes régions de l'image prédite ont été bien reconstruites à savoir les orientations de chaque bloc. Ce qui donne une bonne estimation du mouvement de chaque bloc.

➤ IV.3.2 Application aux images réelles (séquence rubik)

Cette partie (la figure IV.6) représente l'image courante (t) et l'image référence (t+1) d'où on les utilise dans les deux méthodes (méthode BM et la méthode proposée) :



Image a l'instant (t)

CHAPITRE IV

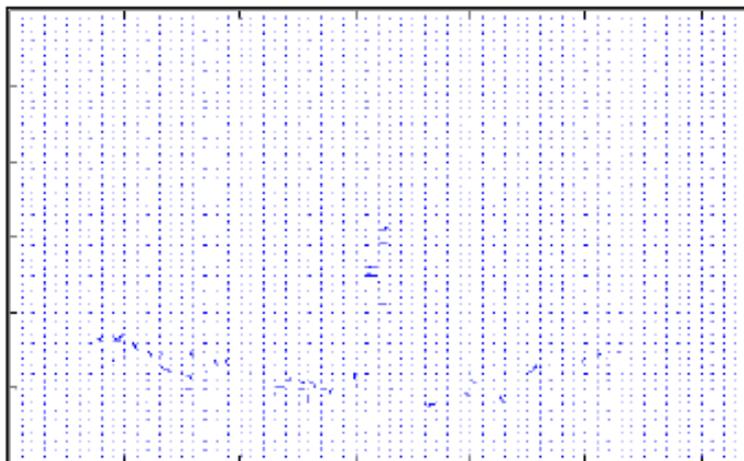
LA METHODE PROPOSEE



Image a l'instant (t+1)

Figure IV.6 image a l'instant (t) et (t+1)

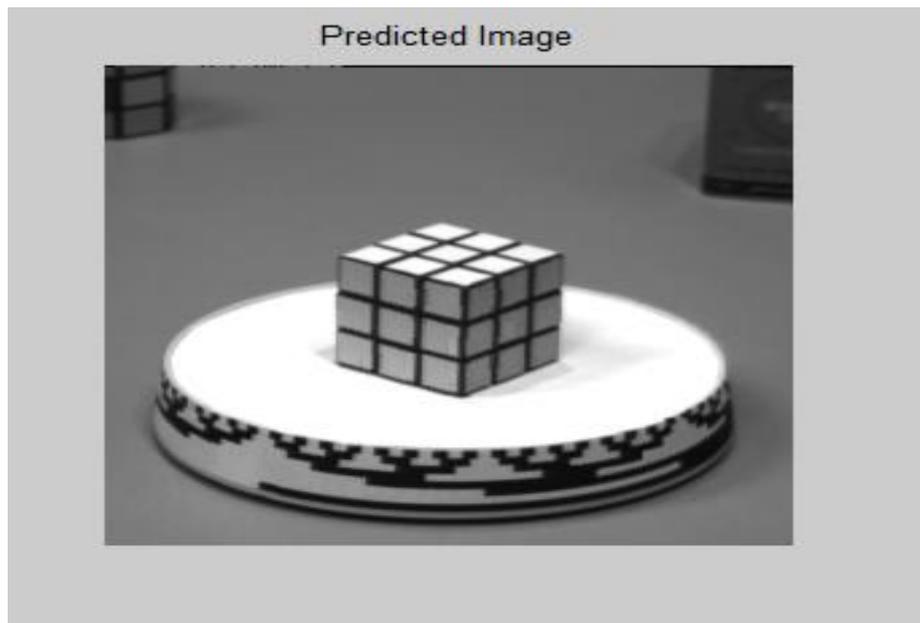
IV.3.2.1 Résultats obtenus par la méthode de block matching



1) Vecteurs de déplacement par méthode BM

CHAPITRE IV

LA METHODE PROPOSEE



2)Image prédite par méthode BM



3) Image erreur par méthode BM

Figure IV .7 les résultats obtenus par la méthode de block matching

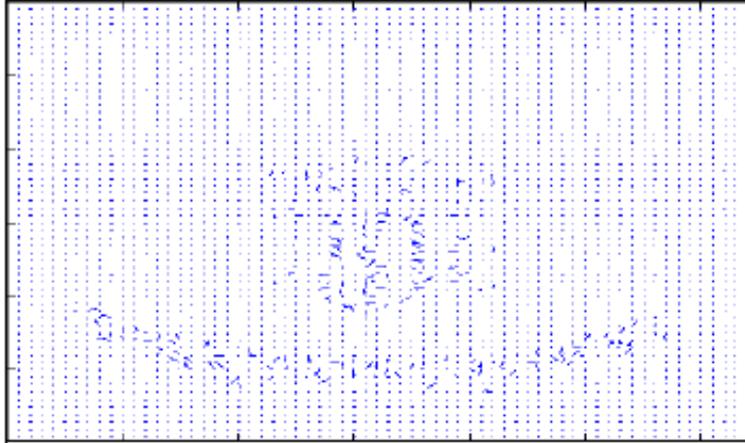
PSNR méthode BM	=35.6335db
-----------------	------------

Résultat du PSNR pour la méthode de block matching classique

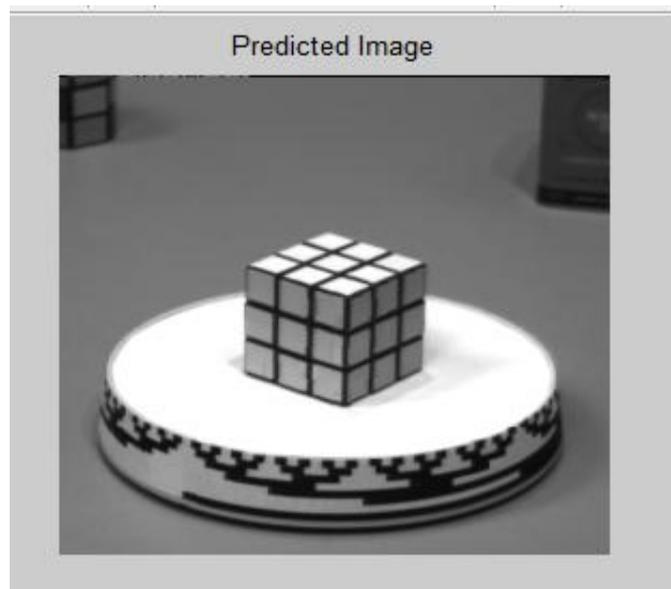
CHAPITRE IV

LA METHODE PROPOSEE

IV.3.2.2 Résultats obtenus avec la méthode proposée :



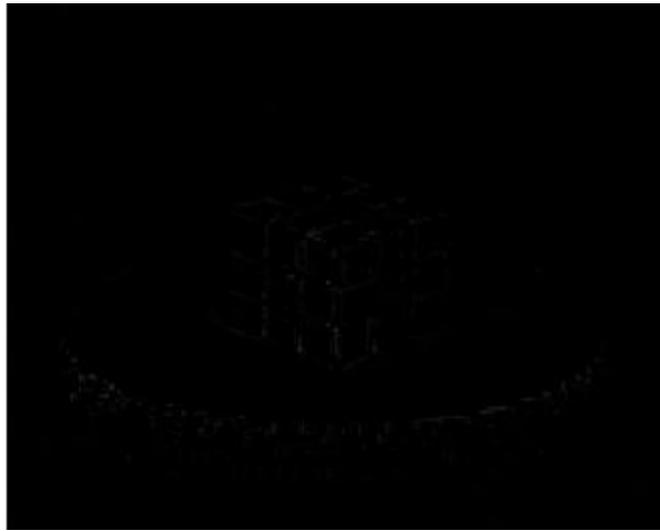
1) Vecteurs de déplacement par méthode proposée



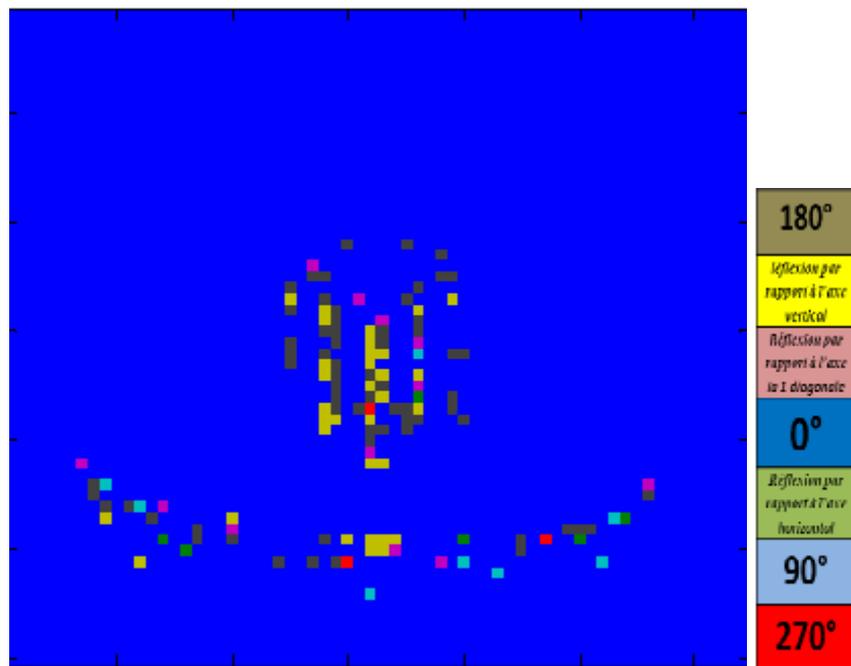
2) image prédite méthode par proposée

CHAPITRE IV

LA METHODE PROPOSEE



3) Image erreur par méthode proposée



4) Les orientations de chaque bloc

Figure IV.8 résultats obtenus avec la méthode de block matching proposée

PSNR méthode proposée	=37.3165db
-----------------------	------------

Résultats du PSNR pour la méthode de block matching proposée

CHAPITRE IV

LA METHODE PROPOSEE

Les différents résultats obtenus avec la méthode proposée montrent bien l'efficacité de cette démarche dans les images réelles. En effet, le résultat d'estimation obtenu dans la (figure IV.8) montre bien cette amélioration de prédiction de différents contours existants dans cette image.

L'image prédite représentée dans la (figure IV.8(2)) est restituée à partir des vecteurs de déplacement (figure IV .8(1)) et les angles de rotation (figure IV.8(4)) sans artefacts visibles, presque confondue avec l'image originale.

En contre partie, celle de la (figure IV.7 (2)) est restituée à partir du vecteur de déplacement avec des artefacts au niveau des régions contenant des contours, et un effet de pixellisation du bloc.

IV.4 conclusion

Dans ce chapitre, nous avons proposé une amélioration de la méthode du block matching en introduisant les orientations des blocs. Les résultats expérimentaux ont montré l'efficacité de la méthode proposée. En effet, dans nos différentes expériences, cette méthode nous a permis de reconstruire les différentes régions et d'atteindre des PSNRs de qualité dépassant les résultats obtenus avec la méthode de block matching classique.

D'après les différents résultats obtenus, nous pourrions dire que l'introduction des orientations sur les blocs donne des résultats plus performants que précédemment. De plus, nous remarquons une très bonne estimation du mouvement au niveau des contours comme le montre l'image prédite de la (figure IV. 8).

En général, nous pouvons conclure que la technique proposée permet d'atteindre des résultats et des PSNR très intéressants. Pour rendre la méthode proposée plus robuste, il est souhaitable d'intégrer d'autres stratégies de recherche du block matching. Néanmoins des améliorations peuvent être apportées afin d'augmenter le taux de calcul et une bonne estimation du mouvement.

CONCLUSION GENERALE

Dans cette étude nous sommes intéressés à l'utilisation du Block Matching pour l'estimation de mouvement de translations et d'orientations des blocs. Une étude bibliographique des différentes notions sur les images et les prétraitements d'image a été d'abord abordé dans le premier chapitre.

L'estimation de mouvement est une étape très nécessaire pour la compression des images vidéo, elle permet de réduire énormément l'espace mémoire nécessaire au stockage d'une image. L'estimation de mouvement est l'une des méthodes les plus utilisées dans ce domaine vu sa simplicité et son efficacité, qui sont abordés dans le deuxième chapitre. Chaque bloc dans la trame courante est le résultat d'un bloc de la trame précédente et un vecteur de déplacement, donc c'est une estimation locale de mouvement.

Dans le troisième chapitre nous avons introduit les principes de la méthode de block matching, par la suite nous avons abordé les différentes techniques de ce dernier. Il existe plusieurs méthodes du Block Matching pour le parcours du voisinage, afin de trouver le bloc qui correspond au mieux au bloc de référence, et qui optimise le critère d'évaluation. La méthode la plus fondamentale pour trouver le meilleur bloc candidat est de visiter et de tester tous les blocs de la zone de recherche. Cette stratégie, appelée recherche exhaustive, est optimale mais pourrait s'avérer très coûteuse en temps de calcul, surtout pour des amplitudes de déplacement grandes.

Dans le quatrième et le dernier chapitre, nous avons mis en œuvre une technique adaptative de mouvement compensé et destinée à améliorer la méthode de block matching qui prend en considération le mouvement de rotation des blocs.

L'objectif est de développer une méthode de suivi de mouvement de translation et de l'orientation de chaque bloc dans les séquences vidéo. Il s'agit donc de coder aussi bien les orientations des blocs que les translations locales des blocs en mouvements présents dans la séquence vidéo. L'estimation et le suivi de mouvement de rotation des objets dont les images sont des problèmes complexes, est encore ouvert. Il intéresse de nombreux domaines comme la robotique, la vidéo surveillance mais aussi la compression vidéo.

- ❑ L'introduction des fonctions d'isométries et la modification de contraste améliore considérablement les résultats sur les images prédites.
- ❑ La méthode proposée permet une bonne estimation du mouvement contrairement à méthode BM classique.

CONCLUSION GENERALE

- ❑ les PSNR obtenus par la méthode proposée \geq PSNR de la méthode de bloc matching classique)

ANNEXE

Table : Les critères de comparaison

Mesures de disparité :	
le meilleur candidat correspond au minimum de ces fonctions	
<p>Somme des différences absolues (SAD)</p>	$SAD(p, d) = \sum_{x=-L/2}^{L/2} \sum_{y=-L/2}^{L/2} I1(p_x + x, p_y + y) - I2(p_x + x + dx, p_y + y + dy) $ $(\hat{dx}, \hat{dy}) = \arg \min_d (SAD(p, d))$ <p>Ce critère est le même que SSD sauf qu'on somme les différences absolues des valeurs des pixels des deux blocs. Il est moins couteux en calcul vu qu'on ne lève pas la différence au carré.</p>
<p>Somme des carrés des différences (SSD)</p>	$SSD(p, d) = \sum_{x=-L/2}^{L/2} \sum_{y=-L/2}^{L/2} \left(I1(p_x + x, p_y + y) - I2(p_x + x + dx, p_y + y + dy) \right)^2$ $(\hat{dx}, \hat{dy}) = \arg \min_d (SSD(p, d))$ <p>Ce critère représente la somme des carrés des différences entre les pixels correspondants aux deux blocs.</p>
<p>Erreur absolue moyenne (EAM)</p>	$EAM(p, d) = \frac{1}{L^2} \sum_{x=-L/2}^{L/2} \sum_{y=-L/2}^{L/2} I1(p_x + x, p_y + y) - I2(p_x + x + dx, p_y + y + dy) $ $(\hat{dx}, \hat{dy}) = \arg \min_d (EAM(p, d))$ <p>Cette mesure est la plus couramment utilisée, car elle est moins couteuse en calcul que les autres méthodes.</p>

ANNEXE

<p>Erreur quadratique moyenne (EQM)</p>	$EQM(p, d) = \frac{1}{L^2} \sum_{x=-L/2}^{L/2} \sum_{y=-L/2}^{L/2} (I1(p_x + x, p_y + y) - I2(p_y + x + dx, p_y + y + dy))^2$ $(\hat{dx}, \hat{dy}) = \arg \min_d (EQM(p, d))$ <p>Ce critère est dérivé de l'erreur moyenne absolue, elle est plus coûteuse en calcul mais donne de meilleurs résultats que l'erreur absolue moyenne</p>
<p>Zero-mean Normalized Sum of Absolute Differences (ZSAD)</p>	$ZSAD(p, d) = \sum_{x=-L/2}^{L/2} \sum_{y=-L/2}^{L/2} I1(p_x + x, p_y + y) - \bar{I1} - I2(p_x + x + dx, p_y + y + dy) + \bar{I2} $ $(\hat{dx}, \hat{dy}) = \arg \min_d (ZSAD(p, d))$ <p>avec $\bar{I1}, \bar{I2}$ représentent les valeurs moyennes des pixels de la zone de recherche et respectivement dans le bloc candidat (même chose pour ZNSSD)</p>
<p>Zero-mean Normalized Sum of Absolute Differences (LSAD)</p>	$LSAD(p, d) = \sum_{x=-L/2}^{L/2} \sum_{y=-L/2}^{L/2} \left I1(p_x + x, p_y + y) - \frac{\bar{I1}}{\bar{I2}} \cdot I2(p_x + x + dx, p_y + y + dy) \right $ $(\hat{dx}, \hat{dy}) = \arg \min_d (LSAD(p, d))$ <p>avec $\bar{I1}, \bar{I2}$ représentent les valeurs moyennes des pixels de la zone de recherche et respectivement dans le bloc candidat (même chose pour LSSD)</p>
<p>Mesures de ressemblance :</p> <p>le meilleur candidat correspond au maximum de ces fonctions</p>	
<p>Coefficient d'inter-corrélation (CC)</p>	$CC(p, d) = \sum_{x=-L/2}^{L/2} \sum_{y=-L/2}^{L/2} I1(p_x + x, p_y + y) \cdot I2(p_x + x + dx, p_y + y + dy)$ $(\hat{dx}, \hat{dy}) = \arg \max_d (CC(p, d))$ <p>Le bloc candidat retenu est celui qui maximise ce critère, il permet d'avoir de bons résultats.</p>

ANNEXE

<p>Coefficient d'inter-corrélation normalisé et centré (NCC)</p>	$NCC(p, d) = \frac{\sum_{x=-L/2}^{L/2} \sum_{y=-L/2}^{L/2} (I1(p_x + x, p_y + y) - \bar{I1}) \cdot (I2(p_x + x + dx, p_y + y + dy) - \bar{I2})}{\sqrt{\sum_{x=-L/2}^{L/2} \sum_{y=-L/2}^{L/2} (I1(p_x + x, p_y + y) - \bar{I1})^2 \sum_{x=-L/2}^{L/2} \sum_{y=-L/2}^{L/2} (I2(p_x + x + dx, p_y + y + dy) - \bar{I2})^2}}$ <p>$(\hat{dx}, \hat{dy}) = \arg \max_d (NCC(p, d))$</p> <p>Ce critère est une version améliorée du coefficient d'inter-corrélation. Il permet d'avoir des résultats plus précis, mais il est plus pénalisant en calculs</p>
---	---

BIBLIOGRAPHIE

- [1] RACHID BELAROUSSI ouvrage : “traitement de l’image et la vidéo”. TRS 106.V25
- [2] CHERIFI mémoire fin d’étude : “ estimation du mouvement dans des séquences d’image”. ING.ELN.17-121/1^{ER} EX
- [3] DEEPAK TURAGA, MOHAMED ALKANNAL. IEEE: ” Search Algorithms for block-matching motion estimation “ 1998.
- [4] BENMOUSSAT mémoire en vus de l’obtention du diplôme de magister: “ étude et mis en œuvre des méthodes d’estimation du mouvement et de filtrage spatiotemporel de séquence d’image”.2000
- [5] AYOUNI BOUDJEMMA mémoire de magister en électronique :” estimation du mouvement dans une séquence d’image par approche probabiliste”..
- [6] BAILICHE MOHAMED AMINE. mémoire de magister : “ étude comparative des algorithmes d’estimation de mouvements dans l’image vidéo par la technique de block matching”. Octobre 2006
- [7] M. Lazri cours et exercice traitement d’image. 2016/2017
- [8] IEEE : méthodes classique d’estimation du mouvement.
- [9] C.P. Bernard, "Ondelettes et problèmes mal posés : la mesure du flot optique et l'interpolation irrégulière", Thèse de Doctorat, Ecole Polytechnique, CMAP: Centre de Mathématiques Appliquées, Palaiseau, France, Novembre 1999
- [10] C. Chun-Ho, and P. Lai-Man, “A novel cross-diamond search algorithm for fast block motion estimation,” IEEE Trans. On circuits and Syst. for Video Technol., vol. 12, no. 12, December2002, pp.1168-1177.

LISTE DES FIGURES

Chapitre I

Figure I.1 : représentation d'image.....	3
Figure I.2 : image binaire (noir et blanc).....	4
Figure I.3 : image couleur.....	5
Figure I.4 : image multi spectrale.....	6
Figure I.5 : image négative.....	7
Figure I.6 : extension dynamique	11
Figure I.7 : égalisation d'histogramme	12
Figure I.8 : image avec filtre passe bas (lissage)	13
Figure I.9 : image avec filtre passe haut.....	14

Chapitre II

Figure II.1: projection mouvement 3D en 2D.....	16
Figure II.2 : illustration des mouvements réel et apparent, dans un système optique de pris de vues.....	18
Figure II.3 : Projection d'objet 3D sur un plan 2D.....	19
Figure II.4 : (a) détection et (b) estimation de mouvement en MPEG	21
Figure II.5 : estimation directe et inverse des vecteurs déplacements	22
Figure II.6: problème d'occlusion.....	23
Figure II.7 : illustration du problème d'ouverture.....	24
Figure II.8: block matching.....	28

Chapitre III

Figure III.1 principe de la méthode de mise en correspondance des blocs.	30
Figure III.2 prédiction avant et arrière	31
Figure III.3: Full Search Algorithm.....	33

LISTE DES FIGURES

Figure III.4 Three Step Search (TSS).....	35
Figure III.5 New Three Step Search (TSS).....	36
Figure III.6 Illustration de sélection des blocs en quatre étapes de recherche	37
Figure III.7 four step search (FSS).....	38
Figure III.8 recherche selon le gradient.....	39
Figure III.9 (a) Modèle de recherche en grand diamant LDSP (b)Modèle de recherche en petit diamant SDSP	40
Figure III.10 diamond search algorithm.....	41
Figure III.11 Two dimensional logarithmic search.....	42
Figure III.12 binary search.....	43
Figure III.13 Orthogonal search algorithm.....	45
Figure III.14 Cross search algorithm.....	46
Figure III.15 spiral search.....	47
Chapitre IV	
Figure IV.1 : principe de la méthode propose.....	50
Figure IV.2 : les huit cas d'orientation d'un bloc.....	50
Figure IV.3 : séquence de synthèse.....	54
Figure IV.4 : résultats obtenus par l'algorithme de block matching classique.....	55
Figure IV.5 : résultats obtenus par la méthode de block matching proposée.....	56
Figure IV.6 image a l'instant (t) et (t+1).....	58

LISTE DES FIGURES

Figure IV .7 les résultats obtenus par la méthode de block matching.....	59
Figure IV.8 résultats obtenus avec la méthode de block matching proposée.....	61

Liste des tableaux

Tableau 1 : masque de filtre passe bas (moyenne).....	12
Tableau 2 : masque filtre passe haut.....	14
Tableau 3 : les fonctions d'isométrie (rotation, symétries)	52
Tableau 4 PSNR des images de synthèse.....	57

GLOSSAIRE

SSD) : somme des différences des pixels de chaque bloc au carré

SAD) : Somme des différences absolues

(ZNSAD) : prend en compte les moyennes des blocs et normalise le résultat

(ZNSSD pour SAD)

EAM : Erreur absolue moyenne

CC : Coefficient d'inter-corrélation

NCC : Coefficient d'inter-corrélation normalisé et centré

EQM : Erreur quadratique moyenne

BM : block matching algorithme

IE = image erreur