REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR
ET DE LA RECHERCHE SCIENTIFIQUE

CENTRE UNIVERSITAIRE 08 MAI 1945
GUELMA

INSTITUT DE GENIE ELECTRIQUE

MEMOIRE DE MAGISTER

Présenté par

Monsieur MESSAOUDI Kamel

En vue de l'obtention du diplôme de Magister en Electronique
Option : Informatique Industrielle et Imagerie

LA COMPRESSION DES IMAGES DYNAMIQUES PAR L'ESTIMATION DES PARAMETRES DE MOUVEMENT

Soutenu devant le jury :

M^{r} S. TOUMI	Maître de Conférences	U. Annaba	Président
M^{r} N. DOGHMANE	Maître de Conférences	U. Annaba	Rapporteur
M^{r} H. TEBBIKH	Maître de Conférences	C.U. Guelma	Examinateur
M^{r} M. BEDDA	Maître de Conférences	U. Annaba	Examinateur
M^{r} H. BOUKROUCHE	Chargé de Cours	C.U. Guelma	Examinateur

1999 - 2000

Table des Matières

Introduction générale 4

Chapitre I : Etude Bibliographique

1. Introduction 7
2. La compression de données 7
   - 2.1. La compression temporelle et spatiale 8
   - 2.2. Codage de la transformée 9
   - 2.3. La quantification vectorielle 9
3. La compression des séquences d'images 10
   - 3.1. Détection et estimation de mouvement 11
     - 3.1.1. Détection de mouvement 11
     - 3.1.2. Localisation des objets en mouvement 12
     - 3.1.3. Estimation du mouvement 13
   - 3.2. Les différents standards existants pour la compression d'images dynamiques 16

Chapitre II : La compression d'images fixes

1. Introduction 18
2. Codage direct (codage spatial) 18
   - 2.1. Le codage MIC (Modulation par Impulsion Codée) 18
   - 2.2. Le codage MICD (MIC Différentielle) 19
   - 2.3. Le codage prédictif 20
3. La compression par DCT 20
4. Le codage en sous bandes (la compression QMFB) 22
5. La norme JPEG 24
   - 5.1. La transformation DCT 25
   - 5.2. La quantification 26
   - 5.3. Codage et compression 27
   - 5.4. La méthodes Run Length Encoding (RLE) 27
6. La quantification vectorielle 28
7. La compression fractale 29
   - 7.1. Historique 29
   - 7.2. Le principe de la compression fractale 29
   - 7.3. Le partitionnement de l'image 31
   - 7.4. Nature des transformations 31
   - 7.5. Les fractales et les autres méthodes 31
   - 7.6. Résultats 31

8. Résultats et discussions 32
8.1. La compression par DCT 33
8.2. La compression QMF 37
8.3. La compression JPEG 39
9. Conclusion 41

Chapitre III : Détection et estimation du mouvement

1. Introduction 42
2. Détection de mouvement 42
2.1. Détection par différence d'images 43
2.2. Détection par test de vraisemblance 43
2.2.1. La modélisation constante par zone 44
2.2.2. La modélisation linéaire 45
3. Localisation des objets en mouvement 46
4. Estimation du mouvement 49
4.1. Bloc Matching (BM) 49
4.1.1. Algorithme de recherche à 3 étapes (3SS) 51
4.1.2. Méthode avec recherche logarithmique 2D (2DLOG) . . 52
4.1.3. Méthode avec recherche orthogonale (OSA) 53
4.1.4. Méthode avec recherche en X (CSA) 54
4.1.5. Nouvel algorithme de recherche à 3 étapes (N3SS) . . . 54
4.1.6. Algorithme de recherche à 4 étapes (4SS) 55
4.2. La méthode Pel-récursive 56
4.3. Méthodes différentielles 57
4.3.1. Méthodes à plusieurs entrées 58
4.3.2. Contraintes de lissage explicites 59
4.3.3. Regroupement de droite de contrainte 59
4.3.4. Formulation mathématique du système surdéterminé . . 61
5. Approches stochastiques pour la détection et l'estimation du mouvement 63
5.1. Rappels théoriques sur les champs de Markov 63
5.2. Les différentes méthodes de relaxation 65
5.2.1. Algorithme de relaxation stochastique 67
5.2.2. Algorithme de relaxation déterministe 67
5.2.3. La politique de visite des sites 67
5.3. Détection de mouvement par modélisation Markovienne 68
5.4. Estimation du mouvement par modélisation Markovienne . . . 71
6. Application de l'approche multirésolution pour la détection et l'estimation du mouvement 72
6.1. Introduction 72
6.2. Les différentes représentations pyramidales 73
6.2.1. La pyramide gaussienne 73

6.2.2. Représentation en arbre quaternaire 74
6.3. Estimation différentielle du mouvement sur une structure Pyramidale 75
6.4. Estimation Markovienne du mouvement sur une structure pyramidale 75
7. Résultats et discussions 77
8. Conclusion 82

Chapitre IV : La norme MPEG

1. Introduction 83
2. Le format M-JPEG 83
3. La norme MPEG 83
3.1. MPEG-1 85
3.1.1. La compression temporelle 85
3.1.2. La compression spatiale 87
3.1.3. La compression audio 87
3.2. MPEG-X 88

Conclusion générale 89

Bibliographie 91

REMERCIEMENTS

Ma reconnaissance s'adresse à Monsieur ***S. TOUMI****, Maître de Conférence à l'universté de Annaba, qui me fait l'honneur d'accepter la présidence du jury de soutenance et m'a fait bénéficier de ses aimables remarques.*

Je remercie Monsieur ***N. DOGHMANE****, Maître de Conférence à l'université de Annaba, qui m'a proposé le sujet passionnant et a suivi avec intérêt le déroulement de mes travaux ; il a su apporter à la fois une contribution judicieuse et efficace. Je le prie de bien vouloir accepter mes plus sincères remerciements.*

Je remercie vivement Monsieur ***H. TEBIKH****, Maître de Conférence à l'université de Guelma, Monsieur* ***M. BEDA****, Maître de Conférence à l'université de Annaba et Monsieur* ***A/H. BOUKROUCHE*** *Chargé de Cours à l'université de Guelma qui ont accepté de faire partie du jury et qui ont voulu me faire bénéficier de leurs aimables remarques.*

Je tiens aussi à remercier messieurs T. Bouden, M. Ramdani, M. Benouareth et R. Iken de m'avoir aider durant la réalisation de ce travail.

Introduction générale

Le traitement de l'information est une discipline qui a vu le jour, voici plusieurs décennies. Ses objectifs essentiels se résument, plus particulièrement, à l'analyse et l'extraction d'une information pertinente souvent masquée par des perturbations, mais aussi alourdie par une redondance incontournable. Cependant, avec l'avancée spectaculaire des moyens de communication et l'intérêt, sans cesse démontré, de vouloir connaître plus et comprendre davantage, ont poussé les spécialistes à proposer des méthodes, souvent numériques, pour contourner les problèmes posés par l'importance du taux d'information à gérer, à traiter, à analyser, à transmettre et/ou à stocker.

Il est évident, que si nous voulons comprendre davantage nous aurons besoin de plus d'informations. Ceci est plus particulièrement visible quant on a affaire à des images dynamiques. En effet, l'importance des images en tant qu'information nécessaire à un bon nombre d'applications pratiques (images satellitaires, multimédia, banques de données, Internet, image médicales, ...etc.) n'est plus un sujet de contradiction. Cependant, l'importance de la quantité de données à traiter, à stoker et/ou à transmettre ont obligé, et contraint encore aujourd'hui, les chercheurs à proposer des solutions permettant de contourner ce délicat problème sans altérer l'information irrémédiablement.

Pour transmettre, stocker ou tout simplement analyser une séquence d'images vidéo, par exemple celle d'un signal vidéo couleur (trois composantes RVB) selon la norme américaine NTSC avec 480x720 pixels par image, 8 bits pour coder chaque couleur et pour 30 images par seconde, il nous faut atteindre théoriquement un débit de 240 Mbits/second pour espérer transmettre convenablement et en temps réel ce signal. De même, pour stocker un film d'une heure, par exemple du même type que le signal vidéo de l'exemple précédent, il nous faut un support de 100 Goctets de mémoire au moins.

La seule alternative possible pour pallier à ce genre de difficultés est bien évidemment une compression de données tout en assurant un minimum d'altération et une meilleure reconstruction. La compression consiste alors à supprimer la redondance spatiale et spatio-temporelle présente inévitablement dans les séquences d'images dynamiques. Suivant

cette définition, on peut aboutir à une première classification des méthodes de compression, suivant que les images sont fixes ou dynamiques (appartiennent à la même scène). Suivant cette classification nous avons essayé d'organiser notre travail qui se présente, dès lors, sous la forme de deux grandes parties :

- La compression des images fixes.
- La compression des images dynamiques.

Pour la compression des images fixes, on peut distinguer deux grandes catégories de méthodes, d'une part la compression sans perte (Codage) qui se limite à supprimer autant que possible la redondance dans les données, et d'autre part la compression avec perte qui autorise une différence entre l'image originale et l'image compressée. Dans ce sens, plusieurs algorithmes (LZ77 «***Lempel-Ziv 1977***» , LZ78 «***Lempel-Ziv 1978***», PKZIP, ARJ, ...), standards et formats (TIFF «***Tagged Image File Format 1988***», GIF «***Graphics Interchange Format 1980***», JPEG «***Joint Photographic Experts Groupe 1988***», ...etc.) ont été développés, et qui sont à la base de nos machines.

En ce qui concerne les séquences d'images numériques et suivant le caractère de l'information que l'on cherche à extraire, l'analyse peut être dans l'une des quatre catégories suivantes [ORK92][BOU88][BOU95] :

- Détection de mouvement,
- Localisation des objets en mouvement,
- Estimation du mouvement,
- Segmentation du mouvement.

On voit donc que la composante essentielle qu'apporte une séquence d'images est l'appréhension directe du mouvement, que ce soit relatif aux éléments de la scène ou au capteur lui-même. D'où l'idée de l'utilisation de cette composante pour la compression spatio-temporelle. En effet, la majorité des CODEC (Codeur/Décodeur) utilisent le principe de compensation de mouvement pour la compression des séquences d'images.

L'idée de la compression des séquences d'images consiste à compresser Les images dites de référence par l'une des méthodes de compression d'images fixes. Les autres images de même scène sont représentées par les vecteurs de déplacement (les vecteurs de vitesse) des objets en mouvement.

Dans notre travail nous nous sommes intéressés à la compression des séquences d'images naturelles, tout en passant par la compression des images fixes et par l'estimation des paramètres de mouvement. Pour cela, nous avons organisé notre mémoire de la façon suivante :

- Une étude bibliographique relatant l'état de l'art quant aux recherches dans les domaines de la compression, la détection, la localisation et l'estimation du mouvement. En plus des différents standards et formats dans ces domaines.

- Une seconde partie de notre travail porte essentiellement sur l'étude et l'application des différentes méthodes de compression des images fixes. Des résultats obtenus par les différentes méthodes étudiées et mises en œuvre sont présentés, commentés et comparés.

- Une troisième partie traite les différentes méthodes de détection, de localisation et d'estimation du mouvement, dans le but de l'utiliser dans un schéma hybride de compression des séquences d'images. On a essayé aussi de montrer l'intérêt de travailler dans un environnement pyramidal.

- La quatrième partie a été réservée au format de compression MPEG. Ainsi que toutes les versions améliorées (MPEG2, MPEG3, ...etc.).

- Nous terminons notre mémoire par une conclusion et les perspectives sous-tendant notre travail.

Chapitre 1

Etude bibliographique

1. Introduction :

L'enregistrement et/ou la transmission de l'information sous sa forme canonique est difficile, voir impossible. Ceci est dû d'une part à la redondance, souvent importante, dans ce type d'information et d'autres part à la limitation des moyens techniques offerts par le coté instrumental (canaux de transmissions à bande limitée, débits de transmission assez faibles et mémoires de stockage limitées ...etc.). Cependant, une bonne compréhension d'un processus ou encore de tout autre procédé nécessite inévitablement plus d'informations. Ceci est plus particulièrement visible quant on a affaire à des images dynamiques ou le taux d'information peut atteindre des valeurs astronomiques. Nous nous heurtons donc à une sorte de contradiction (plus d'information avec des moyens de transmission et de stockage limités) qui ne peut être levée qu'avec un compromis objectif et optimal.

L'une des solutions proposées et développées dans le domaine du traitement des signaux, consiste à réduire le nombre de bits nécessaire pour la représentation de chaque symbole de l'information. Ainsi, la compression des signaux et des images devient un axe de recherche aussi bien nécessaire que prometteur.

2. La compression de données :

La compression de données fait partie des techniques de traitement de l'information. Parmi ses objectifs est d'assuré un débit de transmission rapide sans que l'intelligibilité de l'information soit sujette à une quelconque ''surenchère''.

Dans La théorie de l'information, par le terme d'entropie on entend la quantité d'information codée dans un message [KUN93][NEL94]. Plus l'entropie d'un message est élevée plus il contient d'informations.

Les premières recherches sur la compression de l'information ont commencé dans les années 40 avec *''le codage par redondance minimale''* (codage de l'entropie) [NEL94]. Ainsi, pour assurer un codage optimal d'une information (sous formes de symboles par exemple) il suffit de connaître la probabilité d'apparition de ces symboles. C'est ce que l'on appelle les

méthodes de compression sans pertes. L'un des tous premiers algorithmes opérant selon cette philosophie est celui de ***Shannon-Fano***. En fait, ***Claude Shannon*** et ***R. M. Fano*** ont développé cette méthode pratiquement au même moment [DER96]. Dans le même ordre d'idée et selon des principes assez similaires, que l'on a regroupé sous le nom de compression conservative, diverses autres méthodes ont été développées pour la compression des données en général et plus particulièrement pour les images. Les plus célèbres de ces techniques est sans nul doute l'algorithme de ***Huffman*** et le codage ***Arithmétique*** pour la compression conservative des données [NEL94][DER96], le codage des séquences de zéros ***RLE (Run Length Encoding)***, la méthode des plages pour les images numériques à deux niveaux de gris [FRA98], … etc.

- L'algorithme de ***Huffman*** (1952) est une version optimale de l'algorithme de ***Shannon-Fano***. Il a connu plusieurs développement pour le rendre évolutif dans le sens de s'adapter avec des données variantes et non stationnaires (Huffman adaptatif), ou encore pour pouvoir l'utiliser d'une manière plus répandue comme pour le cas des programmes commerciaux (Les FAX, la norme JPEG, …etc.).
- Le codage arithmétique est apparu dans les années 80. Le principe de cette méthode est de remplacer un flot de symboles par un seul nombre en virgule flottante. La sortie d'un processus de codage arithmétique est un nombre plus petit que « 1 » et plus grand ou égal à « 0 ». Le taux de compression est plus important que celui apporté par le codage de ***Huffman***. Ceci a permis, à cette technique de compression, de s'affirmer, comme un moyen sûr et robuste.

Selon une autre vision, mais pour le même objectif de compression, d'autres méthodes se sont imposées. Elles se sont basées sur la forte corrélation existante entre les échantillons voisins du signal à compresser. On observe certaines variantes opérant directement sur le signal (codage temporel et spatial) et d'autres opérant sur leurs transformées (codage de la transformée). Ces techniques peuvent être également classées selon le type de codage utilisé. On observe donc :

- Le codage scalaire (quantification échantillon par échantillon).
- Le codage vectoriel (quantification des vecteurs).

2.1 La compression temporelle et spatiale :

La compression directe, peut être déterministe ou non. La compression déterministe consiste à utiliser un codage PCM (***Pulse Coded Modulation***), appelé également MIC (***Modulation par Impulsion Codée***) ou encore un codage différentiel DPCM (***Différential Pulse***

Coded Modulation). Ces deux méthodes ne nécessitent pas un traitement complexe mais leurs utilisations restent restreinte à certains types d'applications [SAA94][SAM98].

L'idée du codage DPCM est simple, il utilise le même principe de quantification que le codage MIC, en travaillant non plus sur le signal $x(n)$ mais sur la différence entre un échantillon $x(i)$ et l'échantillon qui le précède $x(i-1)$. Cette méthode est basée sur la forte corrélation entre les échantillons voisins de l'image. La différence entreprise sur les amplitudes des échantillons voisins nous assurera de faibles valeurs (ce qui est en lui-même une compression).

Le codage prédictif fait partie des méthodes de compression non déterministe. Il s'inspire du codage DPCM, mais au lieu de quantifier la différence entre deux valeurs successives, il quantifie la différence entre la valeur courante $x(i)$ et sa valeur prédite $x_e(i)$.

Généralement, la compression directe est associée à d'autres méthodes de compression dans le but d'augmenter encore plus le taux de compression.

2.2 Codage de la transformée :

La compression indirecte, opérant sur la transformée du signal, s'appuie sur les transformations orthogonales : la DFT (***Discret Fourier Transform***), la DCT (***Discret Cosinus Transform***) et la WLT (***Wavelet Transform***). L'avantage d'une image transformée est qu'elle est concentrée dans une zone particulière à la fois étroite et basse fréquence. Pour assurer encore plus de compression, on divise cette zone en deux, voire plusieurs parties distinctes. Celle présentant plus d'information, généralement la plus proche de la fréquence nulle, est codée convenablement (par exemple en utilisant l'une des méthodes de codage conservatives). Les autres parties de l'image peuvent être quantifiées avec un nombre de bits minimal.

Parmi ces techniques, nous pouvons citer le codage par DCT, la compression par les filtres QMFB (***Quadrature Mirror Filter Band***) et la compression par la transformation en ondelettes (WLT). Ces méthodes, mieux adaptées pour les images, permettent d'atteindre, en combinaison avec les méthodes spatiales et le codage entropique, des résultats satisfaisants. Elles sont à la base de plusieurs standards et formats de compression (JPEG , H261/262, MPEG «***Moving Pictures Experts Group***», ...etc.) [VER84][WAL91][DER96].

2.3 La quantification vectorielle :

La quantification vectorielle (QV) est une généralisation de la quantification scalaire aux vecteurs de dimension k. Elle s'intéresse à des groupes d'échantillons subissant une compression

d'ensemble. Cette méthode utilise un dictionnaire dynamique qui doit être stocké et/ou transmis avec l'information compressée, ce qui conduit à un surcoût de compression. Elle est basée aussi sur une règle d'apprentissage, donc un temps de calcul très important. Malgré ces deux inconvénients, la quantification vectorielle atteint des taux de compression importants et inégalables [ANT92][SAM98].

A partir des années 90, une nouvelle technique est apparue, à savoir la compression Fractale. Cette technique identifie les parties de l'image qui sont à la fois des copies réduites et transformées d'autres parties de l'image. Malgré la limitation imposée par un temps de calcul très prohibitif cette technique assure un taux de compression jugé des plus intéressants [FRA98].

3. La compression des séquences d'images :

Les séquences d'images sont des prises de vue fournit par les capteurs à un intervalle de temps bien déterminé, soit 25 images par seconde. Ce qui est équivalant approximativement à un débit de 200Mbits/second. Ceci représente un taux d'information à transmettre relativement important même avec une technologie de pointe. La solution envisageable est d'introduire une compression dynamique.

Le principe préliminaire de ce type de méthodes est de compresser chaque image de la séquence indépendamment des autres, en utilisant par exemple la méthode de compression JPEG ou bien la méthode QMFB. Ceci devrait permettre d'obtenir des rapports de compression variant entre 7 et 27 [DER96].

Quant à la deuxième génération des méthodes de compression d'image dynamique, elle est basée sur l'estimation du mouvement (estimation des paramètres de mouvement). En effet, il est inutile de considérer chaque image de la séquence comme une nouvelle image indépendante des autres. Il suffit de se baser sur une première image et de décrire les suivantes par rapport à celle-ci, en tenant compte uniquement de ce qui a changé. Les objets en mouvement, se trouvant dans une scène dynamique, seront représentés uniquement par leurs paramètres de mouvement. L'exemple type de cette catégorie de méthodes est celle pratiquée par la norme MPEG.

Pour la détection et l'estimation du mouvement plusieurs méthodes ont été développées. Cependant, aucune de ces différentes techniques ne peut dominer toutes les applications et être une méthode standard. Ceci nous a poussé, dans le troisième chapitre, à présenter un état de l'art en ce qui concerne les méthodes de détection et d'estimation du mouvement.

3.1 Détection et Estimation du mouvement :

L'estimation du mouvement peut être vue comme un moyen pour l'élimination de la redondance spatio-temporelle présente habituellement dans les séquences d'images. Ceci a encouragé les scientifiques et les chercheurs à œuvrer dans ce sens pour développer des algorithmes plus puissants assurant une estimation et une détection plus raffinée, plus robuste même en présence de contraintes importantes (nombre d'objets en mouvement très important, présence de bruits aussi bien temporel que spatial, présence d'occultation, objet rigide et déformable, temps de calcul prohibitif ...etc.). Plusieurs approches et méthodes ont été développées afin d'estimer les vecteurs vitesses, ou du moins une certaine approximation. En effet, par la compression on cherche avant tout à diminuer le taux d'information tout en assurant un minimum de dégradation (la reconstruction de l'image devra être optimale). Ainsi, une approximation des paramètres de mouvement pourra, dans la quasi-totalité des cas, suffire à atteindre ces objectifs.

L'estimation du mouvement est souvent précédée par une détection et/ou une localisation des objets mobiles.

3.1.1 Détection de mouvement :

La détection d'objets en mouvement présents dans une scène dynamique revient souvent à une discrimination entre les pixels présentant un changement d'intensité lumineuse, avec l'hypothèse d'invariance des ces intensités (les changements locaux d'intensité lumineuse dans les images sont dus uniquement au mouvement). La recherche des variations temporelles de l'intensité s'effectue en deux temps : le calcul de la dérivée de la fonction intensité selon l'axe du temps et la binarisation des images résultats.

Plusieurs méthodes sont proposées pour le calcul de la dérivée, soit par le calcul direct de la différence de l'intensité entre deux images, soit par l'appelle à des différences métriques plus complexes. La méthode qui parait la plus simple est de calculer la différence d'images. Elle présente l'avantage d'être très rapide mais avec une limitation de taille due à son caractère très sensible aux bruits. Plusieurs articles ont étudié et proposé des méthodes similaires tout en assurant une nette amélioration quant à la réduction de l'effet du bruit. Nous citons, à titre indicatif :

- La détection par la méthode des dérivées partielles,
- Méthode de détection à base du modèle d'éclairement,
- Méthode de détection par le test de vraisemblance,

Cette dernière méthode offre un bon compromis entre le temps de calcul et la sensibilité aux bruits. Ceci lui a value une utilisation fréquente dans le domaine de la détection [HEN93][CAP97][BOU98-99].

3.1.2 Localisation des objets en mouvement :

Plusieurs méthodes ont été proposées pour la localisation des objets en mouvement. Nous pouvons classer ces techniques dans deux grandes familles :

- La première utilise la notion ***«d'image de référence »***, qui désigne l'image qui représente la scène exempte de tout objet mobile. Parmi les travaux qui ont contribués aux développements de certaines de ces techniques nous citons [ORK92-a][ORK92-b] :
  - ***Jain*** (1979) et ***Orkisz*** (1990) proposent l'idée de remplacer les objets mobiles par un fond pour chaque image de la séquence.
  - ***Orkisz*** dans son article (1989) propose l'idée d'utiliser une accumulation des parties invariantes entres images voisines.
  - ***Duncan*** dans son article (1989) utilise des filtres particuliers pour localiser les objets en mouvement dans l'image centrale.
  - ... etc.

  L'inconvénient majeur, des méthodes à base d'image de référence, réside dans le retard introduit pour localiser les objets en mouvement dans une séquence. Il faut attendre l'arrivée de plusieurs images de la séquence pour prendre la décision des objets mobiles dans l'image courante.
- Quant à la deuxième famille, elle s'intéresse à la localisation des frontières des objets en mouvement dans une séquence. Cette approche est appelée méthode par coïncidence des frontières. L'idée de cette approche est de chercher les frontières présentes dans l'image courante et absentes dans l'image précédente. Sachant que dans une séquence d'images obtenues à l'aide d'une caméra fixe, les éléments statiques de la scène apparaissent d'une image à l'autre aux même endroits.

La coïncidence de frontières peut être déterminée de plusieurs manières. Les pionniers dans ce domaine sont ***Leung*** et ***Yang [BOU95]***. Ces deux chercheurs proposent de réaliser la coïncidence par l'opération ***"ET"*** logique appliquée sur les gradients binaires de l'image courante et l'image de différence. L'inconvénient de cette méthode est qu'elle est très sensible aux bruits spécialement ceux d'origine spatial. Pour éviter ce problème, ***Haynes*** et ***Jain*** (1979) ont affirmé qu'il est nécessaire de repousser la binarisation à la fin du traitement, et de réaliser la

coïncidence par l'opération de multiplication. Dans le même cadre, d'autres améliorations ont été proposées par ***stelmaszyk***, dans son article [STE80], pour rendre la localisation plus robuste. L'inconvénient de ces méthodes, utilisant deux images seulement, est qu'elles détectent des frontières parasites (les frontières des objets désocculté) en plus des véritables frontières mobiles. La solution proposée, pour résoudre ce problème, est d'utiliser une troisième image. En effet, ***Vieren*** dans son article [VIE89] a proposé une nouvelle méthode de détection utilisant les gradients de deux différences d'images. Cette méthode résout le problème de désoccultation, mais elle reste incapable de localiser les véritables frontières mobiles, en plus du terme de retard introduit par la troisième image.

L'un des opérateurs les plus récents est celui proposé par ***M. Orkisz*** [ORK92-b]. Cette méthode permet de détecter les véritables frontières en mouvement. De même elle est de loin la plus rapide. Ceci est dû à l'utilisation de l'opération de comparaison au lieu de la multiplication pour réaliser la coïncidence des frontières.

3.1.3 Estimation du mouvement :

Le développement technologique de ces dernières années a contribué, d'une manière ou d'une autre, à l'apparition de plusieurs applications de l'imagerie aussi bien dans le domaine grand public que dans le domaine professionnel. Les images dynamiques et tout ce qui concerne le mouvement ont eu la part du ''Lion'' dans ces applications. En effet, à partir des années 80 plusieurs approches et méthodes ont été développées pour la modélisation des paramètres de mouvement. On s'intéresse, dans notre travail, aux méthodes adaptées aux petits déplacements par rapport à la taille des objets.

Ces méthodes sont souvent désignées par le terme générique d'estimation du flot optique. Plusieurs définitions du flot optique ont été proposées. Il est défini, par exemple, dans [HOR81][SCH85] comme étant la distribution de la vitesse apparente du mouvement de l'intensité lumineuse dans une image. De même, [AGG88][ORK92-a] définit le flot optique comme étant le champ bidimensionnel des vitesses instantanées des intensités lumineuses dans le plan de l'image. Quant à ***Bouthemy,*** dans son article [BOU88-a], il le compare à un champ associé aux variations spatio-temporelles de la fonction intensité ... etc.

Ainsi, sous certaines conditions, le flot optique peut être vu comme la projection des vitesses réelles des objets. Cependant, dans un cadre plus général, le flot optique résulte aussi des mouvements relatifs, le problème des ombres et des occultations ...etc. Ces phénomènes provoquent des différences entre le flot optique et le véritable champ de projection des vecteurs

de vitesse. Ces différences ne sont pas aussi importantes dans le domaine du codage d'image ou l'objectif de la compression est d'atteindre la meilleure reconstruction et non d'assurer une estimation garantit des paramètres de mouvement.

La plupart des méthodes, existantes pour l'estimation du flot optique, prennent en considération le mouvement propre de la caméra. Mais elles exigent un éclairage, non seulement constant dans le temps, mais aussi homogène dans l'espace. Après certaines modélisations mathématiques, on obtient ce que l'on appel la fonction DFD (***Displaced Frame Différence***). Le développement en séries de ***Taylor*** de cette fonction nous ramène à résoudre une équation spatio-temporelle appelée Equation de Contrainte de Mouvement (ECM). Les techniques actuelles consistent soit :

- à minimiser la DFD (Les méthodes de mise en correspondance (les méthodes par similarité de blocs) et les méthodes pel-récursives),
- à résoudre l'équation de contrainte de mouvement (méthodes différentielles),
- à exploiter les propriétés spectrales de cette équation (méthodes fréquentielles),
- à minimiser l'écart entre les observations et un modèle stochastique du mouvement (méthodes stochastiques).

La méthode Pel-récursive, proposée par ***A.N. Netravali*** et ***J.D. Robbins*** (1979), pour l'estimation du mouvement, est une méthode de prédiction et de correction. Elle vise à effectuer une bonne prédiction plutôt qu'une bonne estimation du mouvement réel. Elle consiste à minimiser d'une manière itérative la fonction DFD^2 pour chaque pixel. Ces méthodes, utilisant des calculs itératifs, restent très coûteuses de point de vue temps de calcul. Néanmoins, le point fort de ces techniques se résume aux bons résultats qu'elles peuvent assurer si le mouvement est purement translationnel.

Les méthodes de mise en correspondance «BM» (***Bloc Matching***) est l'une des méthodes les plus simples et les plus utilisées en pratique. Ceci aussi bien pour des implantations matérielles que pour des implantations logicielles. Pratiquement, tous les standards existants (H261/262/263 de CCITT «***Consultive commité for International Télégraphique and Téléphone***», MPEG de l'ISO «***International Standards Organization***») utilisent cette méthode pour la compensation du mouvement entre les images successives d'une séquence vidéo.

Le principe de cette méthode se base sur une hypothèse supplémentaire, selon laquelle les pixels voisins effectuent le même mouvement [BOU88][ORK92-a][CHE98]. L'image à l'instant « t » est considérée comme une grille de blocs (MxN pixels), la méthode consiste alors à trouver le bloc le plus semblable dans l'image à l'instant «t+dt». Etant donné l'hypothèse des petits

déplacements, la recherche est limitée à une fenêtre de dimension ((2*d+M)x(2*d+N) pixels) appelée fenêtre de recherche (d étant le déplacement maximum du bloc).

La mesure de ressemblance, la stratégie de balayage dans la fenêtre de recherche et les tailles des fenêtres (N, M et d), constituent tout ensemble des paramètres de régularisation de l'algorithme. Elles sont choisit en fonction de l'application. Les mesures de ressemblance les plus utilisées sont : la somme des différences au carrée, la somme des différences absolues et la mesure du degré de corrélation.

La différence entre les algorithmes existants réside dans la stratégie de recherche. Elle peut être exhaustive, multigrille ou bien orientée. Le balayage exhaustif donne des bons résultats mais il est très long, pour compenser l'inconvénient du temps de calcul, plusieurs stratégies heuristiques ont été proposées afin de localiser plus vite le maximum de ressemblance. Elles sont effectivement beaucoup plus rapides, mais elles peuvent échouer dans le cas de l'existence de plusieurs maxima locaux. On trouve entre autres, l'algorithme Three Step Search (3SS 1981), l'algorithme 2D Logarithmic Search (2DLOG 1981), l'algorithme Orthogonal Search (OSA 1987), l'algorithme New Three Step Search (N3SS 1994), ... etc. [CHE98].

Les méthodes de mise en correspondance sont adaptées dans le domaine de codage d'image, où une légère différence entre le mouvement réel et le mouvement détecté est autorisée. Par contre pour des problèmes d'analyse des scènes dynamiques, le calcul des véritables vecteurs de déplacements est important. Pour cela l'approche différentielle est apparue. Elle consiste à exploiter une relation entre les déplacements d'une part et les dérivées spatiales et temporelles de l'intensité lumineuse d'autre part [PEL93].

Les méthodes différentielles consistent alors à résoudre l'équation de contrainte de mouvement, avec l'ajout des contraintes supplémentaires, ce qui donne un système d'équations, à deux variables, pour chaque pixel de la séquence, où la solution de ce système n'est pas toujours possible. Ces méthodes donnent des bons résultats, en dehors des zones homogènes, pour l'estimation des véritables vecteurs vitesses. Mais elles demandent des temps de calcul très importants.

Plus loin des approches déterministes, l'approche stochastique est utilisée, à partir des années 80, initialement pour la détection et plus tard pour l'estimation du mouvement. Les méthodes stochastiques consistent à estimer un champ aléatoire de primitives (Mobile/immobile dans le cas de détection et le vecteur de déplacement dans le cas de l'estimation) à partir d'un champ d'observations (la séquence d'images). Dans ce cadre, la combinaison des champs markoviens avec la théorie de l'estimation bayesienne, conduit à la minimisation d'une fonction

d'énergie globale non convexe, ce qui conduit, à leur tour, à des algorithmes de relaxation itératifs. Nous décrivons dans le chapitre III les deux algorithmes les plus utilisés : l'algorithme de relaxation déterministe ICM (avec l'inconvénient des fausses estimations) et celle de relaxation stochastique (algorithme de Métropolis qui demande des temps de calcul très long) [HEN93].

En plus des fausses estimations, dues à l'utilisation des algorithmes déterministes, l'utilisation des algorithmes itératifs, dans un espace de configuration extrêmement vaste (l'espace des images de taille NxM), peut donner lieu à des optimisations longues et coûteuses. Ce qui a motivé l'introduction des techniques multirésolution et multiéchelle pour l'estimation du mouvement. Ces techniques permettent, en effet, l'accélération des algorithmes de relaxation et d'éviter les minimums locaux. Dans ce cadre, plusieurs travaux ont été réalisés. On trouve entre autres [HET91][ZER91][PER92-a] [PER92b][HET94][MEM95][ODO95][LAF98], où les auteurs proposent, pour la détection et l'estimation du mouvement, d'utiliser l'algorithme ICM dans un environnement pyramidale. ***A. Caplier*** et ***F. Luthon***, dans [CAP96] et [Cap97], proposent une approche multirésolution spatio-temporelle avec la mise en œuvre sur un DSP, où la composante temporelle est prise en compte pour la construction de la pyramide. Malheureusement, il n'existe pas un support théorique permettant d'associer la stratégie multigrille et la modélisation markovienne. La méthode utilisée consiste à prendre le même modèle de Markov à toutes les résolutions.

L'approche fréquentielle [ORK92-a][SPI94] est fondée sur l'équation de contrainte du mouvement représentée dans l'espace des fréquences spatio-temporelles (w_x , w_y , w_t). Elle est basée essentiellement sur l'utilisation des filtres spatio-temporels de Gabor [PEL95]. Divers algorithmes et méthodes ont été proposées pour cette approche, relativement récente, promettant ainsi une évolution certaine dans un avenir proche.

3.2 Les différents standards existants pour la compression d'images dynamiques :

La standardisation des techniques de compression vidéo est devenue une priorité compte tenu de la révolution que l'on connaît actuellement dans le domaine des télécommunications et qui doit assurer une harmonie entre tous les utilisateurs potentiels. En effet, un standard dans ce domaine permet :

- La réduction des coûts élevés de CODEC (Codeur-décodeur).
- La résolution des problèmes de compatibilité entre des équipements provenants de constructeurs différents.

Le CCITT a débuté la standardisation des algorithmes de compression vidéo dans les domaines de la téléconférence et de la visiophonie, avec le nouveau né H.261. Le H.261 décrit les versions 625 lignes 50 trames/s et 525 lignes 60 trames/s du CODEC, il est connu sous le nom du standard à Px64 Kbits/s pour ses débits variants de 64 Kbits/Second à 2 Mbits/Second [DER96].

Le H.261 utilise un codage hybride, il combine les techniques de codage en transformée (DCT) et le codage prédictif. Le codage en transformée est utilisé pour la réduction de la redondance spatiale (codage intra-trame), tandis que le codage prédictif est utilisé pour la réduction de la redondance temporelle (codage inter-trames).

Les efforts développés par les équipes CCITT pour le H.261 sont utilisés comme point de départ pour le développement d'un standard de codage d'images animées par l'ISO : il s'agit du MPEG. L'activité du comité MPEG commença en 1988, pour donner après deux ans le MPEG-1. Le MPEG-1 est spécifique à la compression vidéo pour un débit de 1,5 Mbits/s.

En 1991, le comité MPEG lance une nouvelle norme MPEG-2 pour la compression audio et vidéo avec un débit de l'ordre de 10 Kbits/Second. MPEG-2 a comme objectif l'obtention d'une qualité d'image supérieure à celle des standards ***NTSC***, ***PAL*** ou ***SECAM***, avec les débits suivants :

- 60 Mbits/Second pour HDTV.
- 15 Mbits/Second pour NTSC, PAL ou SECAM.
- 4 à 15 Mbits/Second pour CCIR-601.

MPEG-3 étant destinée pour la télévision haute définition à des débits de 20 à 40 Mbits/Second et avec des résolutions allant jusqu'à 1020x1080x30 Hz. Tandis que l'objectif de la norme MPEG-4 est d'être adaptée aux applications audiovisuelles de très bas débits comme la vidéo-téléphonie ou les résolutions vont jusqu'à 176x144x10 Hz et pour des débits variants entre 4800 et 64 000 bits par second.

Récemment, certains chercheurs ont orienté leurs travaux vers l'utilisation de la géométrie fractale pour la compression des séquences vidéo. Malheureusement, ces méthodes, du moins pour l'instant, présentent les inconvénients du temps de calcul élevé et la complexité algorithmique.

Chapitre 2

La compression d'images fixes

1. Introduction :

La compression des images numériques est un axe de recherche très sollicité depuis déjà un bon nombre d'années. Ceci a conduit à une grande diversité d'approches et d'algorithmes. Cette multitude de techniques et de méthodes est due en grande partie aux caractères souvent différents des images à compresser. Cette différence réside aussi bien dans les applications qui ont donné naissance à ces images qu'aux objectifs à atteindre. Le choix d'un algorithme, pour une application donnée, dépend donc de plusieurs facteurs :

- La nature de l'image.
- Le but de la compression.
- Le taux de compression.
- L'utilisation finale de l'image compressée.
- …etc.

Dans ce chapitre, nous allons exposer l'essentiel des méthodes de compression d'images fixes. Nous essayerons, par ailleurs, de faire sortir les avantages et les inconvénients de chacune, tout en étudions les possibilités d'une extrapolation de ces techniques dans le cas des images dynamiques.

2. Codage Direct (Codage Spatial) :

Dans ce type de compression on agit directement sur les échantillons (pixels) de l'image. Pour cela on s'appuie sur la forte corrélation existante entre les pixels voisins (redondance spatiale). La redondance spatiale part du principe que les pixels voisins d'une image ont des valeurs de luminance et de chrominance très proches. En fonction des techniques utilisées, le codage direct peut être déterministe (MIC, MICD), ou bien non déterministe (Codage prédictif).

2.1 Le codage MIC (Modulation par Impulsion Codée) :

Dans le codage MIC, encore appelé PCM, le code binaire correspondant à chaque niveau d'un signal x(t) est transmis vers le récepteur chargé de restituer ces niveaux (étape de décodage).

Soit une image de taille NxM, quantifiée sur 2^k niveaux, le nombre de bits B à transmettre est donné par :

$$B = N * M * k \qquad (2\text{-}1)$$

Pour réduire le nombre de bits transmis on procède à un codage entropique, qui consiste à transmettre des codes de longueurs variables entre l'émetteur et le récepteur [GOU94] [SAA94][SAM98].

2.2 Le codage MICD (Modulation par Impulsion Codée Différentielle) :

Le codage MICD (DPCM) utilise le même principe de quantification que le codage MIC. Cependant, il est appliqué sur un signal différence obtenu à partir d'opérations de soustraction entre chaque pair d'échantillons voisins. En effet, compte tenu de la redondance, souvent importante, entre les échantillons voisins on est sûr qu'une différence entre ces échantillons conduira à des échantillons résultats possédant des amplitudes relativement faibles. Le code binaire affecté à ces échantillons résultats comportera moins de bits d'où l'intérêt de cette méthode de compression.

La figure suivante présente le schéma synoptique d'un système de transmission, appliqué sur un signal unidimensionnel, basé sur la méthode MICD :

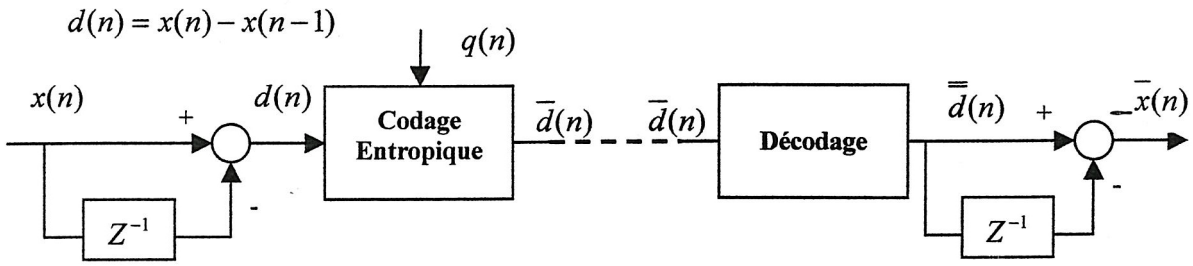


Fig. II.1 : Principe d'un CODEC de type DPCM

Dans la pratique, cette technique donne des taux de compression relativement faibles essentiellement dans le cas des images.

2.3 Le codage prédictif :

Le codage prédictif est une méthode de compression non déterministe. Elle s'inspire du codage DPCM. Seulement la quantification est effectuée sur la différence entre l'échantillon présent $x(i)$ et sa prédiction $x_e(i)$.

$$d(i) = x(i) - x_e(i) \qquad (2\text{-}2)$$

$d(i)$: L'erreur de prédiction

$x_e(i)$ est la valeur estimée de $x(i)$ obtenue par un prédicteur linéaire d'ordre n (exemple du type RIF à 04 coefficients (***Fig. I.2***)).

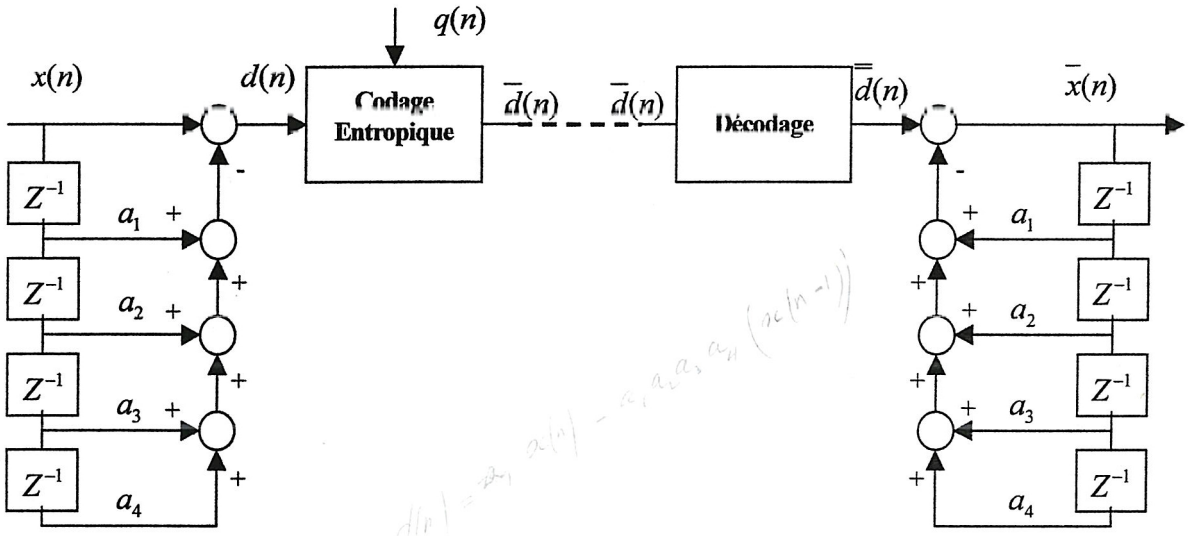


Fig. II.2 : Principe du codage prédictif à l'aide d'un filtre RIF à 4 coefficients.

Cette méthode permet d'avoir des rapports de compression jusqu'à 4 pour des coefficients de prédiction bien choisis, avec une qualité du signal restauré acceptable [SAA94].

3. La compression par DCT :

La DCT est la transformation la plus utilisée pour la compression des données. Elle assure une concentration de la quasi-totalité de l'énergie dans la gamme des basses fréquences. Cette transformation orthogonale est très proche de celle de ***Karkhunen-Loeve*** (KLT). Cependant, son principal avantage est la simplicité de sa mise en œuvre ainsi que l'existence d'un algorithme rapide permettant l'obtention de ses coefficients (déduit à partir de l'algorithme FFT).

La compression par DCT s'effectue en deux temps : la transformation suivit par la quantification (le codage). La figure suivante expose le principe de fonctionnement d'un système de transmission à codage par DCT.

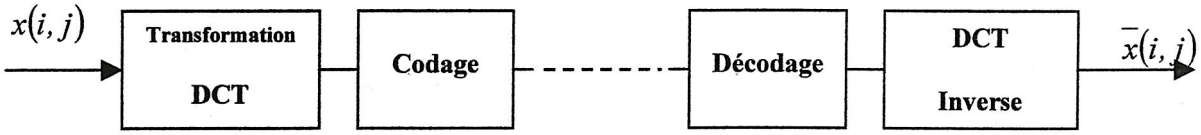


Fig. II.3 : Schéma synoptique d'un CODEC à base de DCT

La DCT est une transformation conservatrice donnant à la sortie une matrice de coefficients représentant l'énergie totale répartie sur les différentes fréquences du signal source.

La quantification est l'étape non conservative de la méthode. Elle consiste soit à réaliser une opération de seuillage sur les coefficients de la DCT pour ne garder que ceux qui sont représentatifs, soit à affecter un nombre de bits sensiblement faible aux coefficients les moins représentatifs (par exemple les coefficients hauts fréquences) d'où l'idée de la compression. En effet, la propriété essentielle de la DCT est la concentration des coefficients dans les basses fréquences. Dans ce sens plusieurs méthodes de quantification ont été proposées :

- La norme JPEG, qui consiste à réduire de plus en plus la précision des coefficients haute fréquences.
- La troncature des blocs (Block Troncation Coding), pouvant être effectué de deux manières différentes :
  - Troncature par la méthode de seuil (***Threshold Coding***) : Seul les coefficients dont l'amplitude est plus importante que ce seuil seront conservés.
  - Troncature par zone (***Zonal Coding***) : Cette technique consiste à utiliser un masque qui couvre la région d'intérêt et élimine les coefficients restants. La figure suivante donne deux types de masque :

a : masque rectangulaire de taille 4* 4

b : masque diagonal à 5 lignes

Fig. II.4 : Principe de la troncature par zone

4. Le codage en sous bandes (La compression QMFB) :

Le codage en sous bandes exploite le fait qu'un signal ne possède pas une représentation fréquentielle uniforme, d'où l'idée de le découper en sous bandes adjacentes de longueurs égales ou non, et d'affecter à chaque bande une méthode de compression suivant leur importance dans le signal d'entrée. Dans ce cadre, plusieurs méthodes ont été mises au point, telle que le codage en sous bandes par ''QMFB'', le codage en sous bandes à "Banc de filtres DFT" ...etc.

La compression par les ''QMFB'' est devenue, depuis quelques années, très populaire. Elle a encouragé les chercheurs à proposer des versions améliorées de cette méthode tel que la méthode QMFB prédictive (PQMFB).

Le schéma de principe de cette méthode, appliquée à un signal unidimensionnel x(n), est donné par la figure suivante :

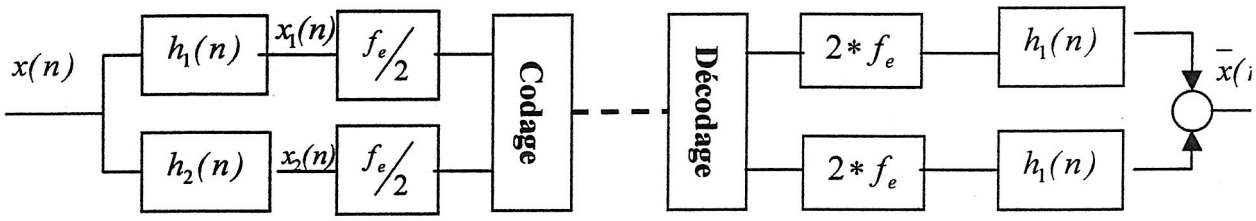
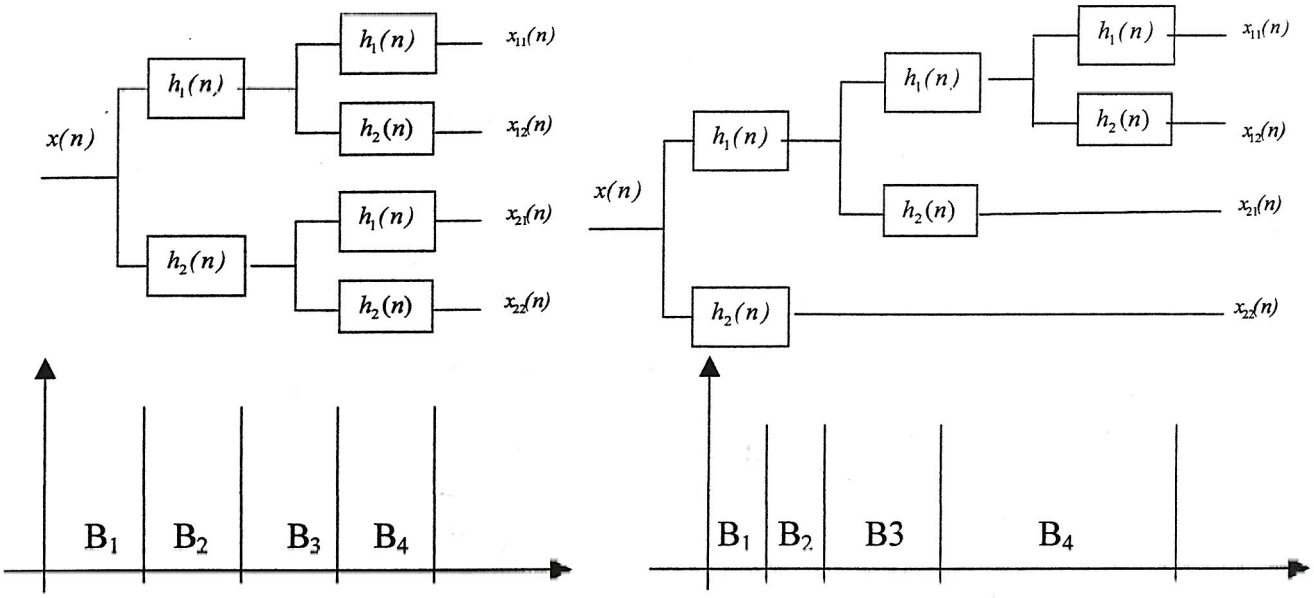


Fig. II.5* : *Schéma de principe de la méthode QMF appliquée sur un signal unidimensionnel

La méthode QMFB consiste à filtrer le signal source x(n) à travers deux filtres miroirs, l'un passe bas $h_1(n)$ et l'autre passe haut $h_2(n)$. Pour avoir deux signaux complémentaires $x_1(n)$ et $x_2(n)$, caractérisé chacun par un sur-échantillonnage dû à l'opération de limitation de fréquence causée par les deux filtres. Ceci nous permet de faire subir à $x_1(n)$ et $x_2(n)$ une opération de décimation dans un rapport de deux. Le résultat final de filtrage et de décimation sont deux signaux, dont le nombre total de point est le même que celui du signal original. Le nombre de bandes peut s'étendre à des valeurs supérieures à 2 par l'application du même principe pour les signaux $x_1(n)$ et $x_2(n)$. La figure suivante montre deux types de subdivisions du signal source avec leur répartition fréquentielle en 4 bandes.



a : Décomposition uniforme sur 4 bandes fréquentielle

b : Décomposition pyramidale sur 4 bandes fréquentielle

Fig. II.6* : *Décomposition d'un signal en sous bandes

L'opération de subdivision en sous bandes est conservative, si on suppose que le filtrage est idéal, elle n'apporte rien de point de vue compression (le nombre de point reste constant). L'opération de codage survient juste après la subdivision en sous bandes, en faisant subir à chaque bande le codage qui lui est approprié suivant leur importance dans la constitution du signal. L'exemple le plus simple est l'utilisation du codage MIC mais avec des nombres de bits variables (exemple : 6 bits pour la bande B_1, 4 bits pour la bande B_2, 2 bits pour les bandes B_3 et B_4). On peut aussi utiliser un codage prédictif différentiel pour les bandes de plus grandes importances et un codage MIC pour le reste des bandes…etc.

La reconstruction du signal original à partir de $x_1(n)$ et $x_2(n)$ nécessite, en premier lieu, une opération d'interpolation (rajouter un échantillon nul entre deux échantillons voisins) pour ramener les deux signaux au même niveau fréquentiel que celui du signal initial. Ensuite une opération de filtrage à travers les deux filtres $k_1(n)$ et $k_2(n)$, suivie par l'addition des deux signaux nous assure une approximation du signal original. Les conditions pour que $y(n) \approx x(n)$ sont [SAA94] :

- N « l'ordre du filtre » soit pair,
- $h_1(n)$ symétrique,
- $H_1^2(w) + H_1^2(w + {}^{w}/_{w_c}) = 1,$ (2-3)
- $H_2(Z) = H_1(-Z)$; $K_1(Z) = H_1(Z)$; $K_2(Z) = -H_2(Z)$,

Pour les signaux bidimensionnels, plusieurs chercheurs ont montré que les filtres miroirs en quadratures bidimensionnels possèdent la propriété de séparabilité, qui leurs permet de s'écrire sous forme de deux filtres unidimensionnels, opérant chacun selon une direction spatiale [VER84][SAA94] :

$$H_{ij}(w1,w2) = H_i(w1) * H_j(w2) \qquad (2\text{-}4)$$

La figure suivante montre l'application de la méthode QMFB sur un signal bidimensionnel :

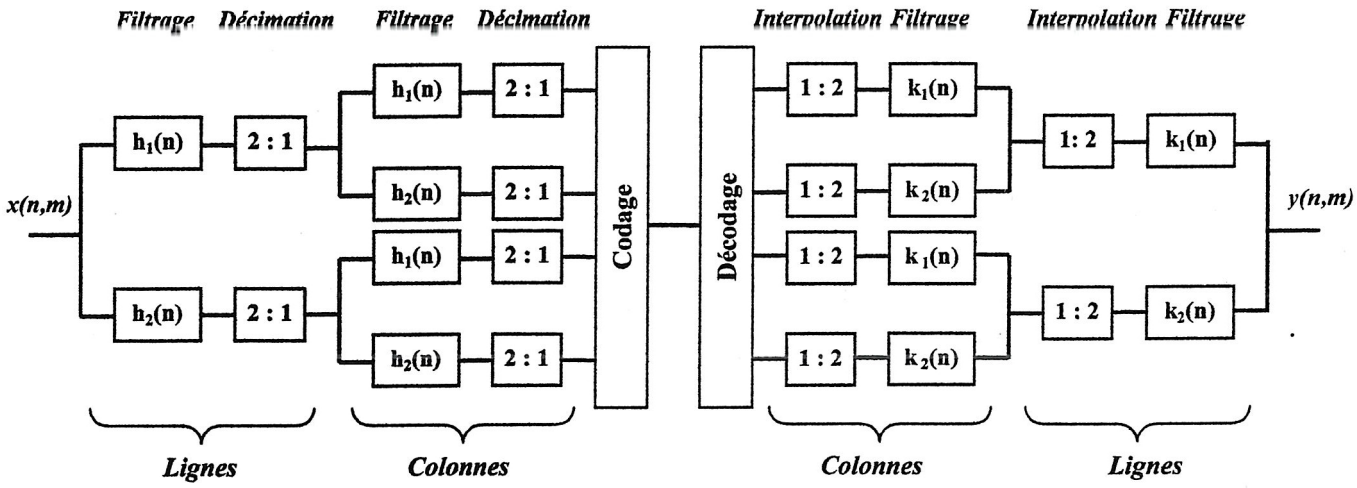


Fig. II.7 : Codage-Décodage d'un signal bidimensionnel Par la méthode QMFB

5. La norme JPEG :

La norme JPEG, issue de la collaboration du CCITT et de l'ISO, est née en 1988. La JPEG tient son nom du groupe chargé de l'élaboration de la norme *: le Joint Photographic Experts Groupe*. Elle est destinée à la compression des images couleur et à niveau de gris en vue de leur stockage sur les supports numériques et leur transmission à travers un réseau à faible débit.

Les techniques définies par la norme JPEG se divisent en deux classes : les méthodes de compression sans pertes utilisant le codage MICD suivi d'un codage entropique. La seconde classe, concerne les méthodes de compression avec pertes, basée sur la transformation DCT

suivie d'une quantification et d'un codage entropique. Evidemment, l'image compressée sera de qualité moindre que l'originale. Cependant, il est possible de régler la qualité en fonction du choix de l'utilisateur. On notera que le rapport moyen de compression entre le JPEG et le GIF est de 1 pour 4. L'algorithme de compression JPEG se compose en trois parties :

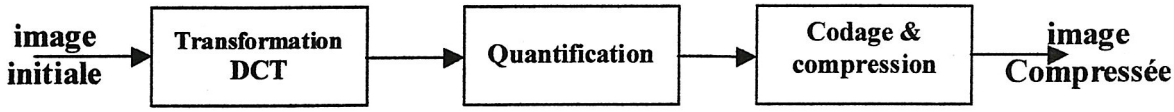


Fig. II.8 : Principe de codage utilisé par le groupe JPEG

L'image est avant tout scindée en blocs de taille 8x8, pour éviter le nombre très élevé des opérations si on prenait la DCT de l'image tout entière. En effet, le calcul de chaque élément de la matrice DCT dépend de la matrice utilisée. Les blocs 8x8 permettent de calculer la DCT plus rapidement et avec un minimum de pertes.

5.1 La transformation DCT :

La transformation DCT est la clé de la compression JPEG. La formule mathématique de la DCT, appliquée sur un bloc de taille NxN, est donnée par :

$$X(k,l) = DCT\left(x(i,j)\right) = \frac{1}{\sqrt{2N}} C(k)C(l) \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} x(i,j) * \cos\left(\frac{(2i+1)k\pi}{2N}\right) * \cos\left(\frac{(2j+1)l\pi}{2N}\right) \quad (2\text{-}5)$$

pour $k = 0 \ldots N-1$

$l = 0 \ldots N-1$

avec :

$$C(x) = \begin{cases} 1/\sqrt{2} & \text{si } x = 0 \\ 1 & \text{si } x > 0 \end{cases}$$

La première propriété de la DCT est qu'elle assure la concentration de l'énergie vers les basses fréquences. En effet, les composantes continues sont les plus importantes pour la reconstruction du signal, et au fur et à mesure que l'on s'éloigne de ses composantes, les coefficients deviennent moins importants dans la composition de l'image. La deuxième propriété de la DCT est qu'elle est réversible, la fonction inverse est notée IDCT, ce qui nous permet de reconstruire l'image dans la phase de décodage.

$$x(i,j) = IDCT(X(k,l)) = \frac{1}{\sqrt{2N}} \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} C(k)C(l)X(k,l) * Cos\left(\frac{(2i+1)k\pi}{2N}\right) * Cos\left(\frac{(2j+1)l\pi}{2N}\right)$$
$$i = 0 \;..\; N\text{-}1 \qquad (2\text{-}6)$$
$$j = 0 \;..\; N\text{-}1$$

Pour la norme JPEG, N est pris égal à 8. La fonction DCT est appliquée sur l'image Bloc par Bloc suivant une stratégie de balayage exhaustive (ligne par ligne).

5.2 *La quantification :*

D'après la formule de la DCT, cette transformation est conservative de point de vue compression, mais elle prépare à la deuxième étape non conservative du processus : la quantification.

La quantification est l'action de réduire le nombre de bits dans la matrice DCT par la diminution des précisions des entiers en fonction de leurs positions, nous pouvons réduire la précision du coefficient au fur et à mesure que nous nous éloignons du coefficient continu. Le JPEG utilise, pour son algorithme, une matrice de quantification (8x8) appelée *Quautum* ou bien matrice *Taille*. Les éléments de cette matrice sont calculés, en fonction d'un facteur (***facteur de qualité***) introduit par l'utilisateur, par la formule suivante [NEL94] :

$$Quautum(i,j) = Q(i,j) = 1 + (1 + i + j)quality \qquad (2\text{-}7)$$

Pour déterminer les valeurs de la matrice Q, il suffit de donner une seule valeur (*quality*), qui doit être dans l'intervalle [1 .. 25]. Un facteur de 1 correspondant à la plus grande précision, au fur et à mesure que l'on augmente le facteur de qualité, les coefficients de la DCT sont de plus en plus arrondis. La valeur quantifiée est donnée par :

$$Valeur_quantifiée(i,j) = \frac{DCT(i,j)}{Quautum(i,j)} \quad arrondi\ à\ l'entier\ le\ plus\ proche \qquad (2\text{-}8)$$

L'opération inverse est donnée par

$$DCT(i,j) = Valeur_quanlifiée(i,j) * Quautum(i,j) \qquad (2\text{-}9)$$

5.3 Codage et compression :

Nous pouvons remarquer que suite à la quantification, un grand nombre de coefficients est arrondi à zéros. Dès lors, le JPEG utilise deux types de codages différents : un algorithme de codage de séquence (RLE) pour les coefficients nuls, et un codage entropique pour les autres coefficients (***Huffman*** ou ***Arithmétique***). Dans le but d'augmenter la longueur des séquences de zéros, JPEG utilise la technique zigzag pour convertir la matrice 8x8 en un vecteur de 64 éléments.

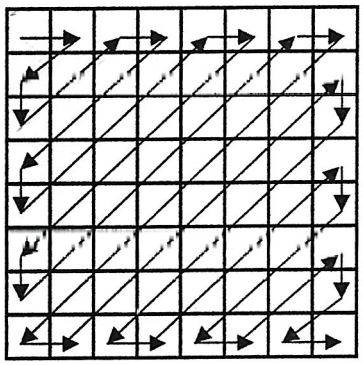


Fig. II.9 : La structure zigzag utilisée par la norme JPEG

5.4 La méthode Run Length Encoding (RLE) :

La méthode RLE est la méthode de codage la plus utilisée pour la compression des séquences de zéros. Les coefficients à coder sont vus comme étant une séquence de plusieurs ensembles constitués d'une suite de zéros et d'un coefficient *AC* qui la termine. Chaque suite est codée par deux symboles [FRA98] :

- ***Symbole_1 = (RUNLENGHT , SIZE) ;***
- ***Symbole_2 = (AMPLITUDE) ;***

La première partie (***Symbole_1***) contient deux types d'information :

- ***RUNLENGHT*** : Nombre de zéros consécutifs qui précèdent immédiatement le coefficient non nul (*AC*), sa longueur est limitée à 4 bits.
- SIZE : Nombre de bits sur lequel ***Symbole_2*** est représenté, sa longueur est limitée sur 4 bits.

La deuxième partie (***Symbole_2***) contient la valeur du coefficient non nul qui termine la suite à coder, elle est codée sur SIZE bits.

Le codage RLE représente généralement une partie complémentaire dans les mécanismes de codage. Il a comme but d'augmenter le taux de compression. Il est, ainsi, utilisé dans plusieurs applications (la norme JPEG, la norme FAX, ...etc.).

6. La quantification vectorielle :

La quantification vectorielle (QV) est une généralisation de la quantification scalaire aux vecteurs de dimension k. Elle consiste à représenter tout vecteur X de dimension « k » par un autre vecteur Y_i (vecteur représentant) de même dimension. Cependant ce dernier appartient à un ensemble fini Y de N vecteurs, appelé dictionnaire (*Code Book*). Sous forme mathématique la quantification vectorielle s'écrit :

$$\begin{aligned} Q \;:\; & R^k \longrightarrow Y \\ & x \in R^k \longrightarrow Q(x) = y_i \end{aligned} \qquad (2\text{-}10)$$

$Y = \{y_i \in R^k \;;\; i = 1...N\}$: L'ensemble de vecteurs de reproduction

Donc au lieu de stocker le vecteur X, il suffit de stocker la position de son image Y_i dans le dictionnaire Y. La mesure des distorsions entre X et Y_i s'effectue à travers une distance $d(X, Y_i)$, qui peut être par exemple la distance Euclidienne.

$$Y_i = Q(X) \quad si \quad d(X,Y_i) < d(X,Y_j) \quad pour \quad j = 1...N \quad et \quad i \neq j \qquad (2\text{-}11)$$

L'organisation du dictionnaire passe nécessairement par une phase d'apprentissage destinée à créer les vecteurs de reproduction Y_i. Plusieurs algorithmes ont été développés dans ce sens, on trouve entre autres l'algorithme ***LBG***, l'algorithme ***K-Means*** et celui de ***Kohonen*** [BAR94][SAM98]. La figure suivante montre le principe de fonctionnement de la méthode de compression basée sur la quantification vectorielle :

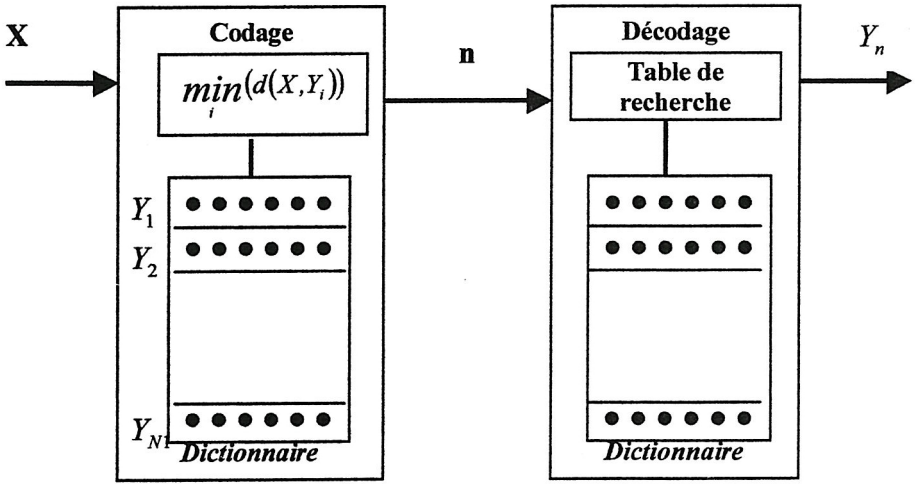


Fig. II.10 : ***Schéma de principe du QV***

7. La compression Fractale :

7.1 Historique :

Le terme « Fractale » revient du mot latin « ***Fractus*** », équivalent au verbe « ***to break*** » [BOU99]. La théorie Fractale, inventée par ***Benoit Mandelbrot*** en 1982, a été depuis, développée dans plusieurs domaines, en particulier dans la génération et le codage des images numériques. En effet, en 1988 ***Michael Barnsley*** a publié un article dans lequel il propose une nouvelle technique de compression des images basée sur l'utilisation de transformations affines contractives : Les IFS (Iterated Function System). Plus tard, ***Pentland*** propose une méthode de traitement d'images utilisant la géométrie Fractale, dans le but d'améliorer la segmentation et l'extraction des contours [FRA98]. Ceci a permis une évolution certaine de ce domaine.

7.2 Le principe de la compression fractale :

Les redondances dans l'image sont des parties qui se ressemblent ou qui sont proches. Expérimentalement, on montre que la plupart des images possèdent ce type de redondances internes. Nous citerons, par exemple, les images d'arbres, les visages, les montagnes, les nuages, ...etc.

La transformation Fractale consiste à décomposer une image en plusieurs parties de formes quelconques, et à chercher les parties qui sont similaires entre elles à une transformation affine près. D'une autre manière, la compression Fractale consiste à identifier les parties de l'image qui sont des copies réduites transformées de l'ensemble de l'image. Dans le domaine de génération des images, par exemple, on peut, à partir d'une feuille et d'une tige, générer une branche, et de cette branche on peut générer un arbre.

La méthode Fractale est donc basée sur le principe de la méthode IFS, qui est applicable pour les images en deux couleurs, ou bien sa version modifiée PIFS (Partitioned IFS).

La méthode PIFS exploite une forme particulière de self-similarité : certaines parties de l'image sont des copies transformées par rapport à d'autres parties de la même image. Cette méthode est à la base des machines à copie réduites multiples (MCRM), qui permettent, à chaque itération, de reproduire sur une copie plusieurs exemplaires réduits de l'image originale.

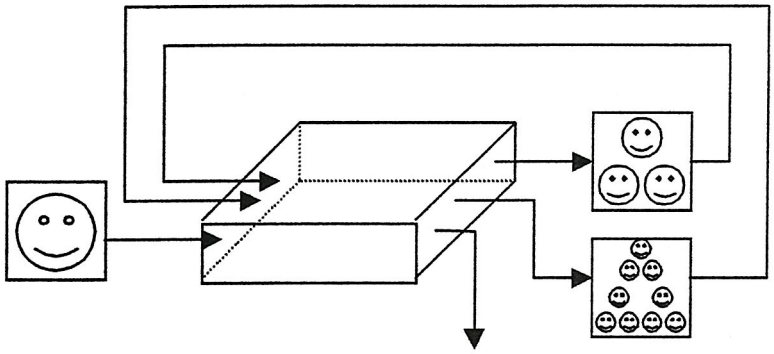


Fig. II.11 : ***La première et la deuxième copie d'une MCRM***

La pratique a montrée, qu'après un certain nombre d'itérations l'image reproduite converge vers une même image (L'attracteur de l'IFS), et cela quelle que soit l'image initiale. Ceci explique le théorème de point fixe (T(f) = f).

La compression Fractale consiste à décomposer l'image initiale en plusieurs Blocs « Domaines » D_i de taille *(DxD)* (ces blocs peuvent se chevaucher) et à décomposer (sans chevauchement) la même image en blocs « ranges » R_i de taille *BxB*, avec $D > B$. En pratique D est choisi égal à *2*B.*

Le processus de codage consiste à trouver un ensemble de transformation T_i, définie chacune sur une région D_i de l'image vers une cellule range R_i de même image. La condition sur T_i est de minimiser une fonction de distorsion $d_{L2}(R_J, T_i(D_i))$. Aucune information sur l'image à coder n'est stockée, seules les transformations et les coordonnées des régions aux quelles elles sont appliquées sont nécessaires pour la reconstruction de l'image.

Le processus de décodage consiste tout simplement à itérer ces transformations à partir d'une image initiale quelconque. La séquence d'image obtenue converge vers une approximation de l'image originale (théorème du point fixe).

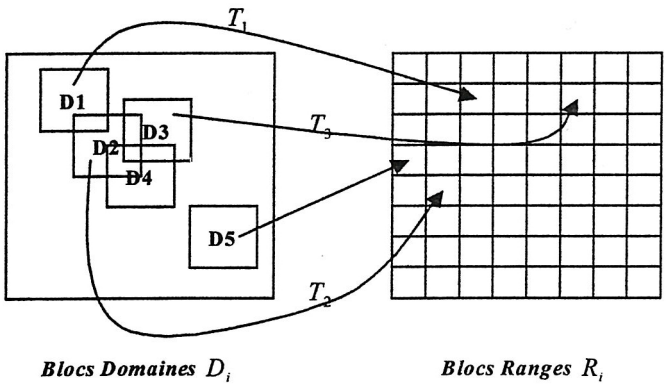


Fig. II.12 : ***Schéma de principe de la compression Fractal***

7.3 Le partitionnement de l'image :

Plusieurs types de découpage de l'image ont été étudiés de façon à minimiser le nombre de transformations locales et à coder au mieux les similarités inter-blocs. Chaque type de découpage possède ses avantages et ses inconvénients, on trouve :

- Le partitionnement carré, qui découpe l'image en blocs de même taille *BxB*.
- Le partitionnement quadtree, dans lequel la partition exhibe des blocs de tailles différentes.
- Le partitionnement rectangulaire.
- Le partitionnement triangulaire adaptatif, qui découpe l'image en deux triangles, chacun est à son tour subdivisé en quatre triangles.

7.4 Natures des transformations T_i :

Les transformations utilisées par les experts du Fractale, sont en général composées de deux transformations : Géométrique G_i et Massique M_i (soit $T_i = M_i \circ G_i$). La transformation géométrique est réalisée par le remplacement des blocs de taille *DxD* par d'autres blocs de taille *BxB*, avec (*D>B*). les transformations massiques utilisées sont [FRA98] :

- Translation de luminance.
- La modification du contraste.
- Les isométries : les transformations modifiant l'orientation du bloc (rotation, réflexion ou bien orthogonale).

7.5 Les Fractales et les autres méthodes :

Actuellement des chercheurs utilisent la théorie des IFS pour des schémas hybrides fondés sur la transformée en Ondelettes ou en Cosinus Discrète. Plusieurs travaux, opérant de la même manière, ont été élaborés afin d'améliorer les résultats obtenus par les méthodes de la première génération. L'exemple type est celui de la norme JPEG qui a été améliorée et révisée plus d'une fois avec des résultats assurant à la fois des hautes qualités d'images restaurées et des taux de compression importants.

7.6 Résultats :

La compression Fractale assure des taux de compression de l'ordre de 1000 (pour certains types d'images). Néanmoins, elle présente l'inconvénient majeur de nécessiter un temps de calcul prohibitif. L'exemple type de la compression fractale est celui appliqué sur l'image de Sierpinski de résolution (256x256), le taux de compression pour cette image est 113 [FRA98].

8. Résultats et discussions :

Pour comparer les résultats des différents algorithmes, nous avons choisi l'image de ***Lena*** avec une résolution de 256x256 points (***Fig. II.13***). Les échantillons de l'image sont codés sur 8 bits, soit 256 niveaux de gris. On peut mesurer le degré de compression soit par le calcul du rapport de compression (RC), soit par le taux de compression, avec :

$$RC = \frac{I}{\hat{I}} = \frac{taille\ de\ l'image\ Initiale}{taille\ de\ l'image\ Finale} \qquad (II\text{-}12)$$

la formule du taux de compression en terme de pourcentage est donnée par :

$$Taux\ \% = 100 * \frac{(RC-1)}{RC} \qquad (II\text{-}13)$$

Le degré d'altération des images (exprimé en terme du rapport signal à bruit SNR) est donné par la formule suivant :

$$SNR = \frac{\sum_{i=0}^{N-1}\sum_{j=0}^{M-1} I^2(i,j)}{\sum_{i=0}^{N-1}\sum_{j=0}^{M-1} e^2(i,j)} \qquad (II\text{-}14)$$

avec :

$$e(i,j) = I(i,j) - \hat{I}(i,j) \qquad (II\text{-}15)$$

Fig. II.13 : L'image originale de Lena

8.1 La compression par DCT :

Le tableau II.1 présente les résultats obtenus à l'aide de la méthode de compression par DCT (appliquée sur des blocs de tailles 8x8), avec une troncature par zone (masque carré). La figure (***Fig. II.14***) illustre les résultats obtenus.

	Taille du masque					
	2	**3**	**4**	**5**	**6**	**7**
RC	*16,00*	*7,11*	*4,00*	*2,56*	*1,78*	*1,31*
SNR (db)	*96,90*	*112,90*	*126,98*	*139,99*	*155,3*	*171,81*

Tableau II.1 : Variation du RC et SNR en fonction de la taille du masque carré

Les résultats de la méthode DCT, sur des blocs de tailles 8x8, avec une troncature par zone (masque diagonal), sont résumés sur le tableau II.2. Les résultats obtenus sont présentés sur la figure (***Fig. II.15***).

	Taille du masque							
	2	**3**	**4**	**5**	**6**	**7**	**8**	**10**
RC	*21,33*	*10,67*	*6,40*	*4,27*	*3,05*	*2,29*	*1,78*	*1,31*
SNR (db)	*92,60*	*104,20*	*114,19*	*124,75*	*134,90*	*144,51*	*156,35*	*181,04*

Tableau II.2 : Variation du RC et SNR en fonction du nombre de diagonal

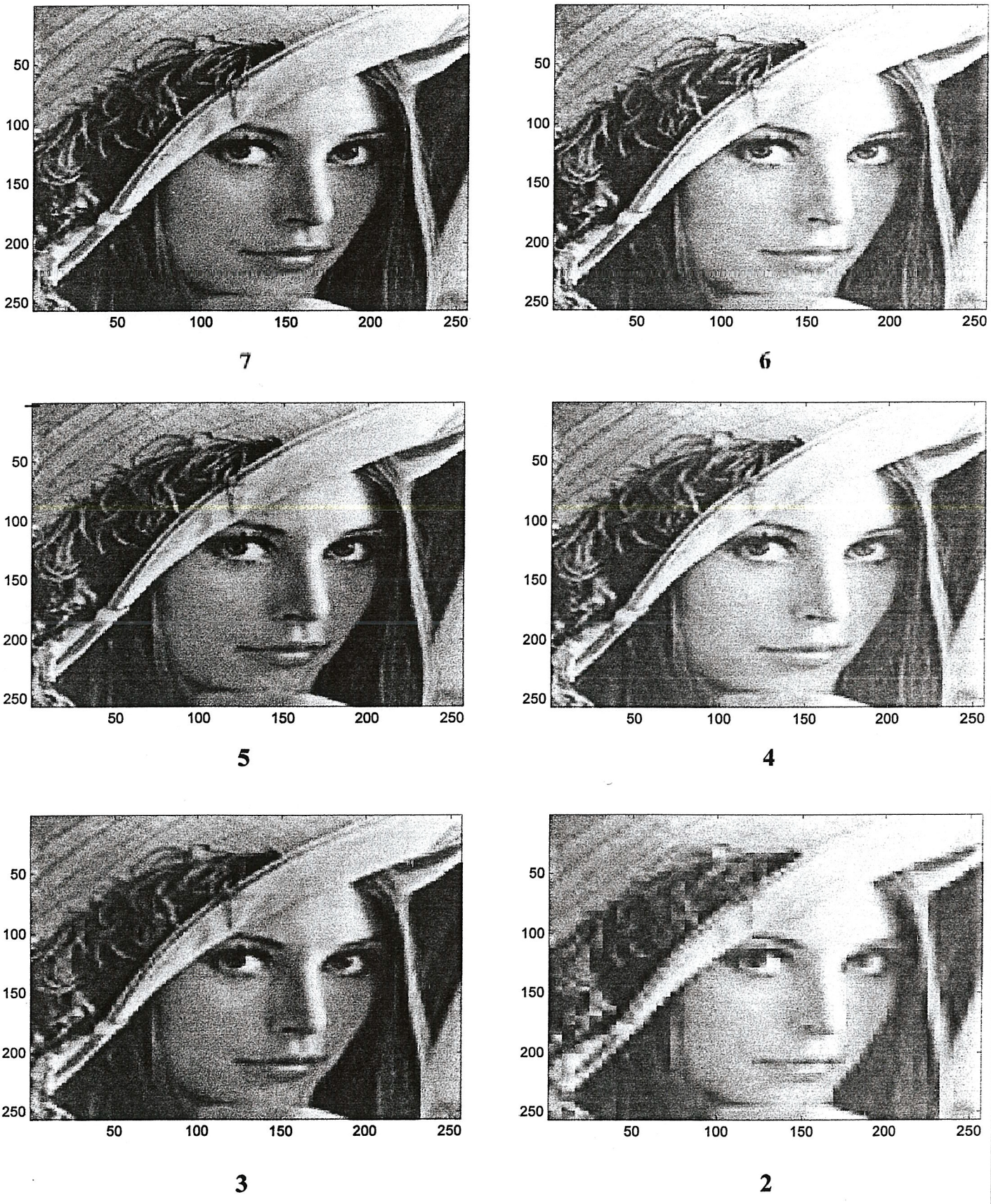
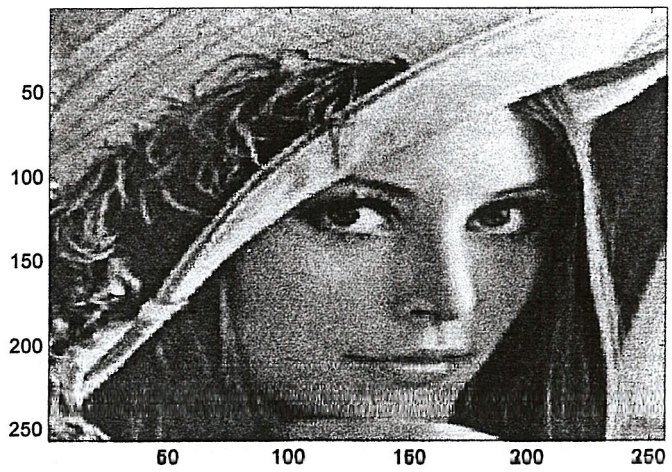
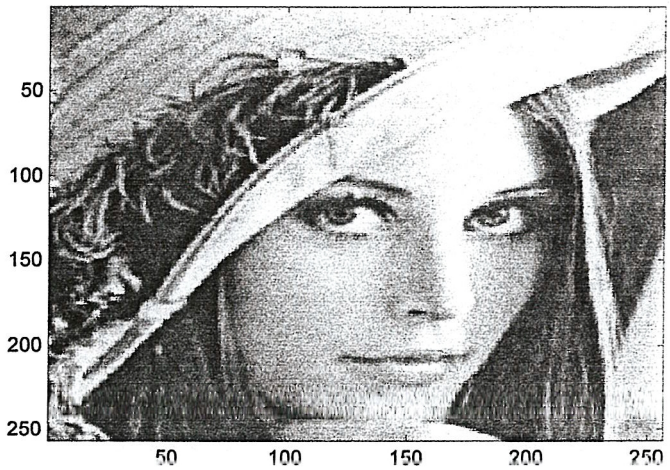


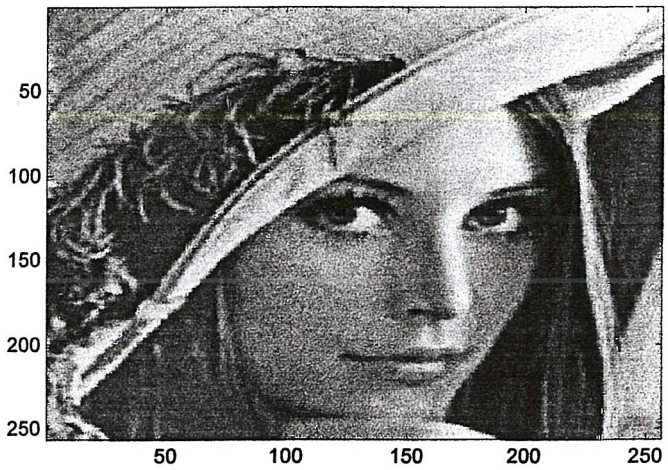
Fig. II.14 : Résultats de la méthode DCT sur l'image de Lena (Masque Carré)


50
100
150
200
250
50
100
150
200
250

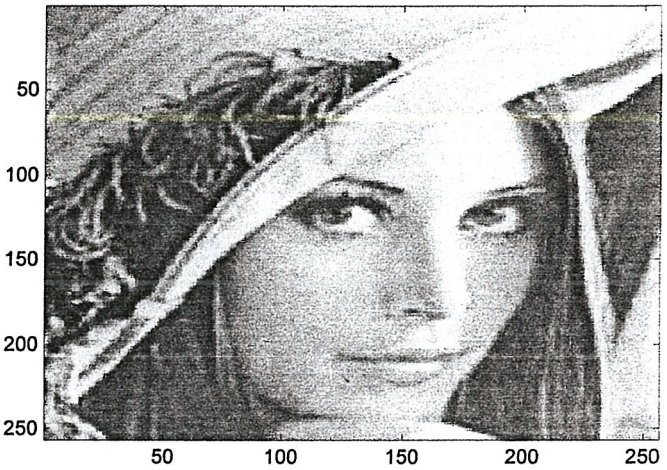
8


50
100
150
200
250
50
100
150
200
250

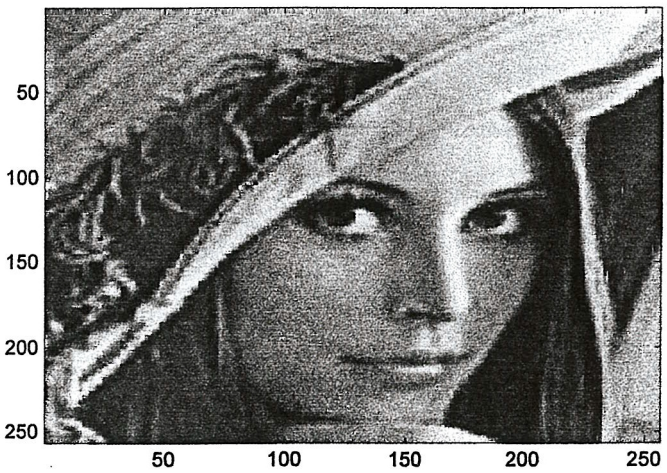
7


50
100
150
200
250
50
100
150
200
250

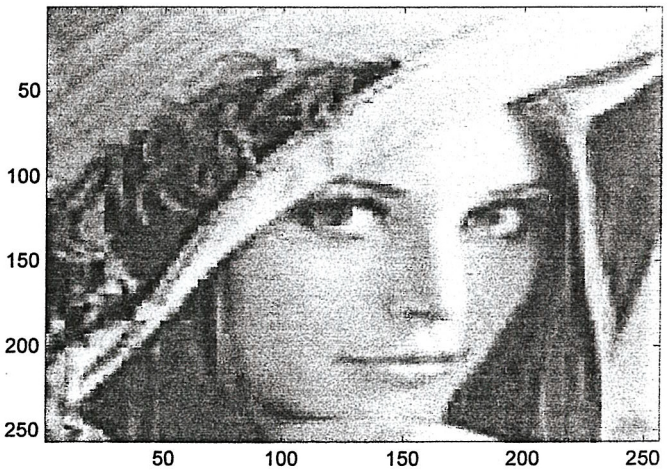
6


50
100
150
200
250
50
100
150
200
250

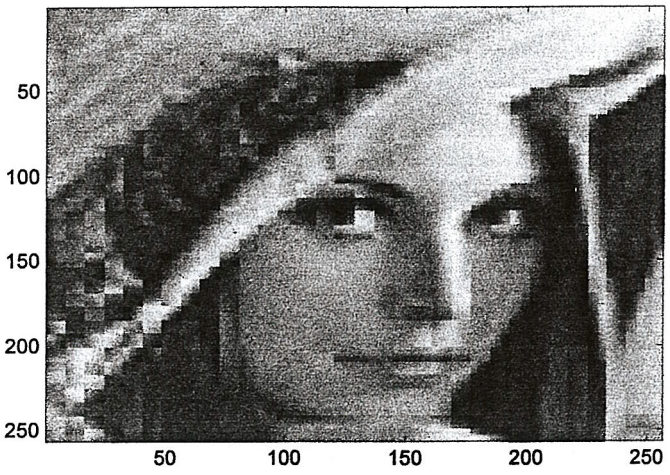
5


50
100
150
200
250
50
100
150
200
250

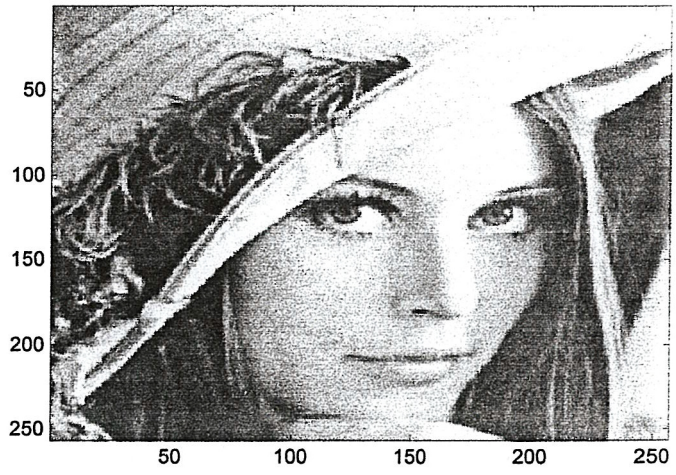
4


50
100
150
200
250
50
100
150
200
250

3



2



10

Fig. II.15 : Résultats de la méthode DCT sur l'image de Lena (Masque diagonal)

RC = 64

SNR = 78,42 db

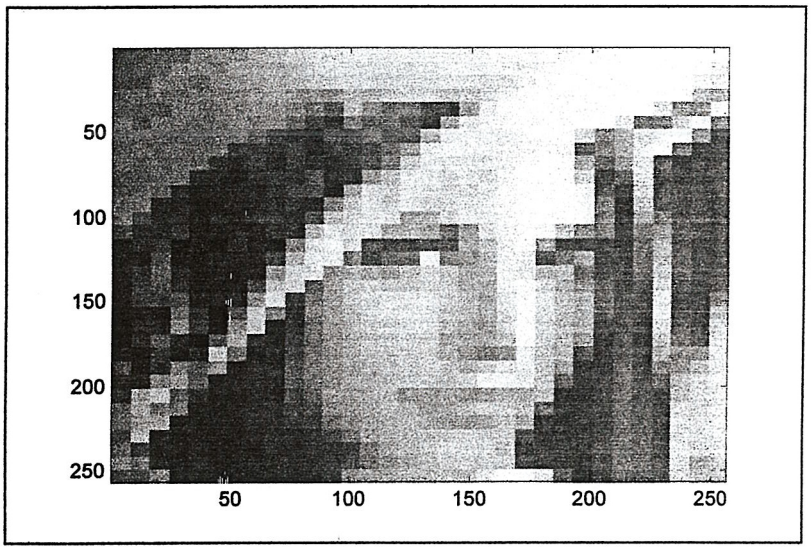


Fig. II.16 : Résultats de la méthode DCT sur l'image de Lena (Masque Carré de taille 1x1)

Les résultats obtenus (RC et SNR) montrent l'efficacité de la méthode de compression par DCT. Elle permet d'atteindre des taux de compression de l'ordre de 10 tout en maintenant un SNR assez élevé. Ce qui nous permet de conclure que l'utilisation d'un masque diagonal donne les meilleurs résultats (rapport de compression maximum de l'ordre de 20). Pour les deux masques, la qualité de l'image diminue considérablement au-delà d'un rapport de compression égal à 7 (masque carré) et au-delà de 10 (masque diagonal).

Pour des taux de compression assez élevés, nous remarquons deux effets néfastes :

- L'effet du flou : à cause des hautes fréquences éliminées par l'opération de troncature.
- L'effet des blocs : Cet effet est bien clair dans la figure (***Fig. II.14***) (masque carré de taille 2x2) et dans la figure (***Fig. II.15***) (masque diagonal D(2)). Il résulte de l'application du DCT sur des blocs et non pas sur l'image tout entière.

L'utilisation d'un masque carré (diagonal) de taille 1x1, nous permet d'obtenir un rapport de compression RC=64. La qualité de l'image restaurée est objectivement (SNR = 78,42 db) et subjectivement (effets du flou et des blocs) dégradée (***Fig. II.16***).

8.2 La compression QMF :

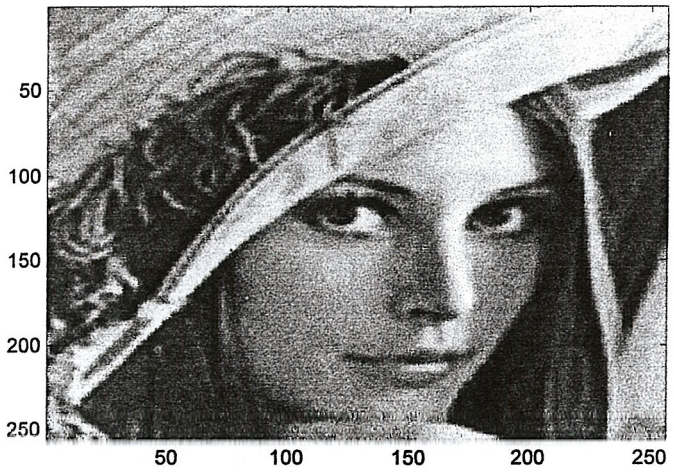
Le tableau II.3 résume les résultats obtenus par la méthode QMF, ou deux cas de filtres miroirs sont utilisés (avec 16 et 32 coefficients). Le signal image est décomposé en quatre bandes fréquentielles (BB, BH, HB et HH). Chaque bande peut être codée par l'une des méthodes de compression classique (PCM, DPCM, DCT). Les résultats correspondants sont représentés sur la figure (***Fig. II.17***).

			1				*2*				*3*				*4*			
		Bande	***B1***	***B2***	***B3***	***B4***	***B1***	***B2***	***B3***	***B4***	***B1***	***B2***	***B3***	***B4***	***B1***	***B2***	***B3***	***B4***
		Type De codage	PCM (6bits)	PCM (2bits)	PCM (1bit)	PCM (1bit)	PCM (4bits)	PCM (2bits)	PCM (1bit)	PCM (1bit)	*	PCM (2bits)	PCM (1bit)	PCM (1bit)	**	PCM (2bits)	PCM (1bit)	PCM (1bit)
Ordre du filtre	16	RC	3,20				4,00				8,00				11,64			
		SNR	119,44				116,98				101,04				85,8993			
	32	RC	X				4				X				11,64			
		SNR	X				117,81				X				83,72			

* : *Compression par DCT (masque carré de taille 4x4)*

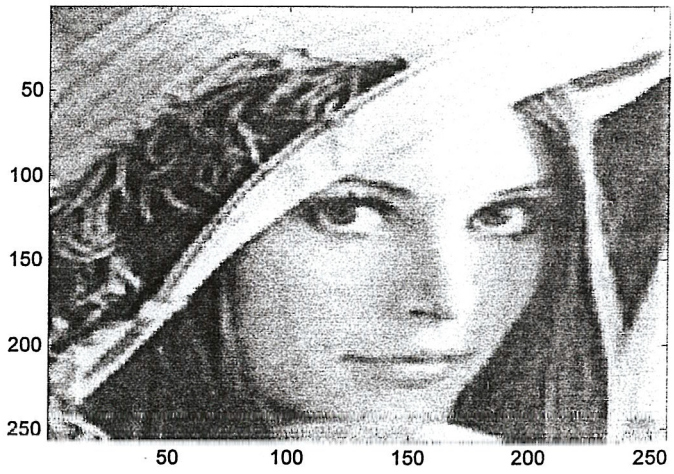
** : *Compression par DCT (masque diagonal D=3)*

***Tableau II.3** : Variation du RC et SNR en fonction du type de codage pour chaque bande et de l'ordre du filtre*

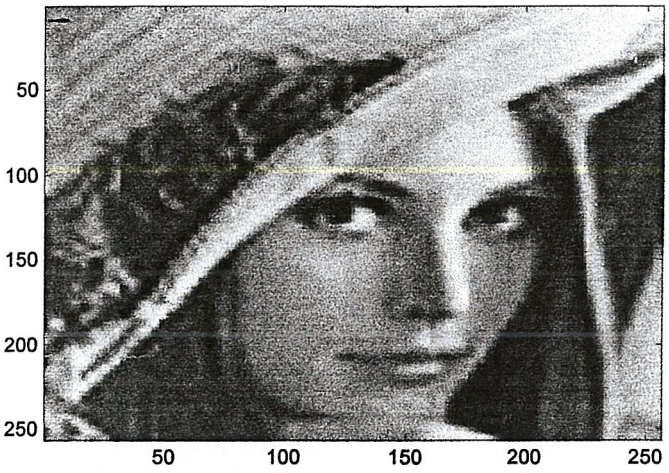


Codage par PCM
(B1 : 6 ; B2 : 2 ; B3 : 1 ; B4 : 1)

16 Coefficients

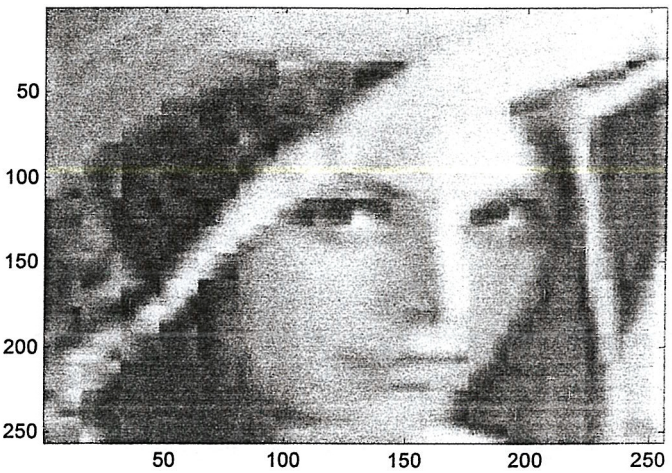


Codage par PCM
(B1 : 4 ; B2 : 2 ; B3 : 1 ; B4 : 1)

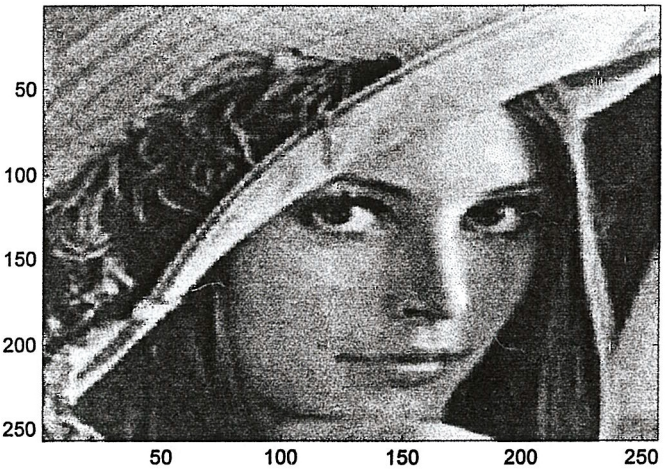


Codage par PCM (B2 :2 ; B3,B4 :1)
Codage par DCT (Masque 4x4) : B1

16 Coefficients

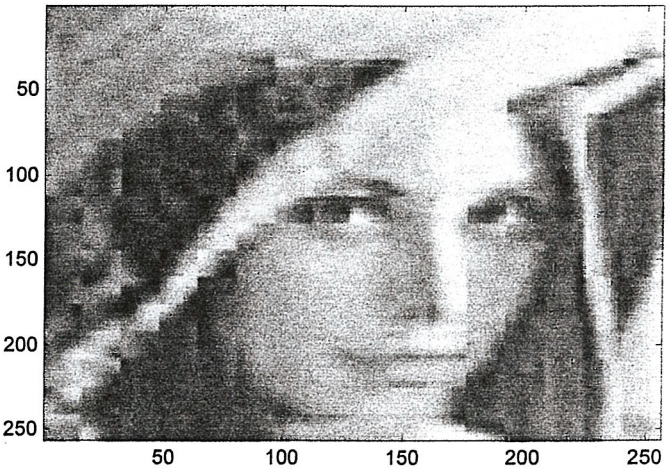


Codage par PCM (B2 :2 ; B3,B4 :1)
Codage par DCT (Masque D(3)) : B1



Codage par PCM
(B1 : 4 ; B2 : 2 ; B3 : 1 ; B4 : 1)

32 Coefficients



Codage par PCM (B2 :2 ; B3,B4 :1)
Codage par DCT (Masque D(3)) : B1

Fig. II.17 : ***Résultats de la méthode QMF appliquée sur l'image de Lena***

L'utilisation de la méthode QMF offre une diversité de combinaisons avec d'autres méthodes. Dans le but d'obtenir un bon compromis entre le temps de calcul, le taux de compression et la qualité de l'image restaurée, l'utilisateur a le choix entre le codage PCM, le codage DPCM et le codage par DCT, pour coder les différentes bandes fréquentielles. L'utilisation d'un codage PCM, avec un nombre de bits variable en fonction de l'importance de la bande à coder, reste le meilleur de point de vue temps de calcul, le rapport de compression pour cette méthode est de l'ordre de 4, qui est très faible par rapport aux autres méthodes à un SNR constant.

La solution la plus efficace pour cette méthode est d'utiliser le codage par DCT soit pour les bandes fréquentielles les plus importantes (les autres bandes sont codées par la méthode PCM / DPCM), soit pour toutes les bandes avec un masque variable.

8.3 La compression JPEG :

Pour la compression JPEG, le tableau II.4 montre la variation du rapport de compression (RC) et du rapport signal sur bruit (SNR) en fonction du facteur de qualité (Quautum). Les résultats de cette méthode sont présentés sur la figure (***Fig. II.18***).

	Facteur de qualité (Quautum)							
	1	2	5	10	15	20	25	40
RC	1,87	2,76	4,97	7,54	9,55	11,23	12,61	15,64
SNR (db)	164,08	145,71	127,45	115,37	108,57	103,82	100,56	93,03

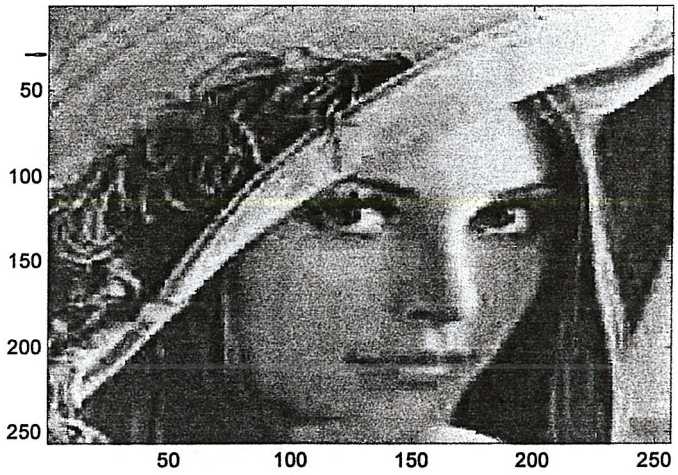
Tableau II-4 : Variation du RC et SNR en fonction du facteur de qualité (Quautum)

D'après les résultats nous pouvons affirmer que la compression JPEG donne les meilleurs résultats. Elle est basée essentiellement sur les inconvénients de la méthode de compression par DCT. En effet, cette dernière élimine les hautes fréquences (l'opération de troncature) sans tenir compte de leur importance dans la reconstruction de l'image. Par contre, le principe de la méthode JPEG se limite à diminuer la précision des échantillons hautes fréquences, et non pas à les éliminer, ce qui donne des images restaurées de meilleures qualités.

La qualité des images restaurées, pour la méthode JPEG, est acceptable jusqu'à une valeur de 25 pour le facteur de qualité (Quautum =25), au-delà duquel la dégradation devient assez visible. Nous remarquons sur la figure (***Fig. II.18*** Quautum = 40) de petits carrés (effets des blocs), cela vient du fait que le JPEG fonctionne sur des blocs de tailles 8x8.

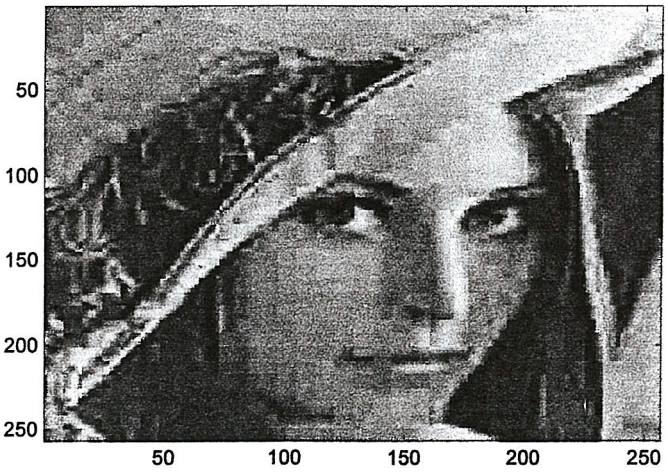
Quautum = 1

Quautum = 5

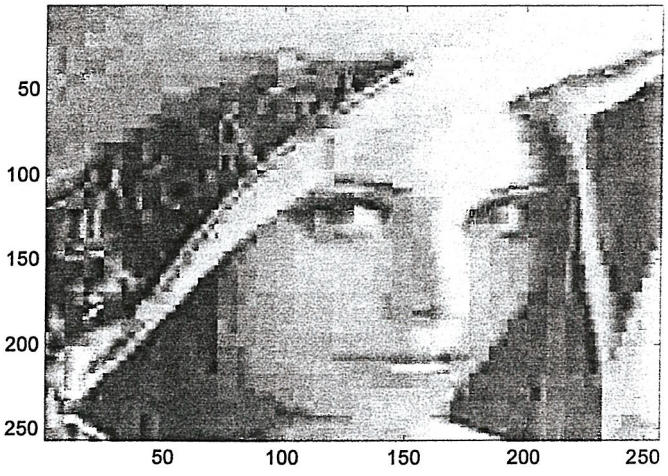


Quautum = 15

Quautum = 20



Quautum = 25



Quautum = 40

Fig.II-18 : ***Résultats de la méthode JPEG appliquée sur l'image de Lena***

9. Conclusion :

D'après l'étude théorique et les résultats obtenus nous pouvons conclure que :

- Le codage de l'entropie ne donne pas de bons résultats, essentiellement dans le cas des images. Mais il peut servir comme une étape complémentaire pour les autres méthodes dans le but d'augmenter le taux de compression.
- Les méthodes classiques de compression (codage spatial et temporel) donnent des résultats ''objectivement'' et ''subjectivement'' acceptables, mais avec des rapports de compression assez faibles par rapport aux autres méthodes de compression (inférieurs ou égales à '4' pour le codage PDPCM).
- Les méthodes de compression par transformée assurent les meilleurs résultats. Elles offrent un bon compromis entre le temps de calcul, le rapport de compression et la sensibilité au bruit. La méthode de compression par DCT, à titre d'exemple, atteint un rapport de compression égale à '10' avec un temps de calcul peu élevé, ce qui lui permet d'être utilisé souvent dans la compression à temps réel. En effet, la méthode de compression par DCT reste la base de plusieurs normes et standards (JPEG, MPEG, les FAX, ...etc.).

  D'après les résultats nous pouvons déduire aussi que la compression par les filtres QMFB et la compression par les ondelettes WLT [SAA94] combinées avec la DCT permet d'aboutir à des taux de compression assez importants, avec l'inconvénient d'un temps de calcul prohibitif.
- Les résultats de la compression vectorielle [SAM98] et de la compression fractale [FRA98] assurent, sans aucun doute, les taux de compression les plus élevés. Ces méthodes permettent d'aboutir, pour le même ordre de compression, à des signaux de qualité supérieure à ceux du codage par transformée. Néanmoins, l'inconvénient du temps de calcul handicap lourdement ces méthodes et limite leurs utilisations à des applications « off lines ».
- Les méthodes de compression par transformée, combinées avec les méthodes d'estimation du mouvement, sont donc les mieux adaptées à la compression des séquences d'images compte tenu du compromis qu'elles offrent entre le temps de calcul, le taux de compression et la bonne reconstitution de l'image.

Chapitre 3

Détection et estimation du mouvement

1. Introduction :

Nous nous intéressons, dans cette partie de notre travail, à présenter l'état de l'art concernant les différentes méthodes de détection et d'estimation du mouvement. Pour la détection, plusieurs méthodes ont été proposées [BOU95]. Nous avons mis l'accent sur deux des méthodes les plus utilisées : la technique de détection par différence d'images suivie d'un seuillage et celle de la détection par le test de vraisemblance. La seconde méthode étudiée (détection par le test de vraisemblance) dispose d'un atout traduit par un compromis entre temps de calcul et sensibilité aux bruits. Pour l'estimation du mouvement nous avons jugé que l'approche directe (méthodes de mise en correspondance) est la meilleure au moins dans le cas du codage d'images (compression spatio-temporelle). Aussi, nous avons étudié plus succinctement cette méthode en balayant la quasi-totalité des techniques qui lui sont dérivées. Dans l'ensemble cette technique, ainsi que ses améliorations, visent plus particulièrement à accélérer l'algorithme de recherche. Cependant, nous ne sommes pas limités à une seule famille de technique. D'autres approches ont été exposées en insistant sur leurs avantages, leurs limitations et éventuellement les points communs qu'elles ont entre elles.

Parmi ces techniques nous pouvons citer, entre autres :

- L'approche différentielle permettant d'estimer les véritables vecteurs de vitesse, mais dont l'inconvénient majeur se résume au temps de calcul souvent important.
- L'approche stochastique, appliquée initialement pour la détection des véritables objets en mouvement, a été généralisée pour l'estimation du mouvement. Elle offre encore des axes de recherche très intéressants, particulièrement pour des applications dans un environnement multirésolution.

2. Détection de mouvement :

La détection des objets en mouvement n'est pas un but à atteindre, du moins dans notre cas, mais elle peut servir comme une étape intermédiaire à l'estimation du mouvement. La détection garantit l'identification des points de l'image ayant subit une variation de l'intensité

lumineuse. Par ailleurs, elle leur affecte l'étiquette « 1 », au détriment des autres points qui auront l'étiquette « 0 ». Sous certaines conditions (invariance des intensités, éclairage constant, ...etc.), la détection des changements temporels d'intensité lumineuse est équivalente à la détection du mouvement.

2.1 Détection par différence d'images :

Pour détecter immédiatement un changement temporel de la fonction intensité il est intéressant d'effectuer la différence ''temporelle'' entre deux images successives de la même séquence, soit :

$$\forall\ p \quad FD(p,t) = I(p,t) - I(p,t-dt) \qquad ; \quad p = (x,y) \qquad (3\text{-}1)$$

La seconde étape de cette méthode consiste à comparer la valeur absolue de FD(p,t) à un seuil λ. Ceci nous ramène à une prise de décision sur la présence éventuelle d'un mouvement.

$$\forall\ p \quad CT(p,t) = \begin{cases} 1 & si \quad |FD(p)| \geq \lambda \\ 0 & si\ non \end{cases} \qquad (3\text{-}2)$$

FD : Différence d'image (Frame Différence).

CT : L'image binaire.

Où λ est un seuil fixé arbitrairement ou suivant une procédure de seuillage choisie. L'avantage de cette méthode est sa rapidité. Son inconvénient majeur est sa grande sensibilité aux bruits temporels. Nous nous sommes donc intéressés plus particulièrement à cette méthode, pour ses nombreux avantages, quant à l'estimation des paramètres du mouvement.

2.2 Détection par le test de vraisemblance :

Dans cette méthode on considère deux fenêtres A_1 et A_2 de deux images consécutives centrées respectivement en (x_0,y_0,t) et $(x_0,y_0,t+dt)$. Ainsi, tout un ensemble de pixels voisins du pixel central (x_0,y_0) sera pris en compte. Le test de vraisemblance est basé sur deux hypothèses :

Hyp_0 : Les deux fenêtres ont la même distribution de niveau de gris : il n y a pas de changement temporel.

Hyp_1 : Les deux fenêtres ont des distributions, de niveau de gris, différentes : des changements temporels ont eu lieu entre les deux instants t et t+dt.

Le niveau de gris de chaque fenêtre peut être modélisé de différentes manières [HEN93][BOU95]:

2.2.1 La modélisation constante par zone :

Pour chaque région (fenêtre) de l'image, l'intensité est supposée suivre le modèle suivant [BOU95]:

$$I_{région} = \mu + bruit \qquad (3\text{-}3)$$

où :

- Le bruit est supposé gaussien, centré et de variance σ^2.
- μ est la moyenne des intensités des N pixels de la région.

Les hypothèses Hyp_0 et Hyp_1 se traduisent alors par :

Hup_0 : A_1 et A_2 suivent la même distribution : Loi normale (μ_0, σ).

Hyp_1 : A_1 et A_2 suivent des distributions différentes : Lois normales (μ_1, σ) et (μ_2, σ).

On définit alors une fonction de vraisemblance "L" en fonction des paramètres μ, σ et N. On doit choisir aussi un seuil de binarisation adapté au contraste de l'image.

$$L = \left(\frac{1}{2\pi\sigma^2}\right)^{\frac{N}{2}} \exp^{-\frac{\sum_{x=0}^{N}(I(p_x-\mu)^2}{2\sigma^2}} \qquad (3\text{-}4)$$

Le rapport de vraisemblance s'écrit alors sous la forme :

$$R_1 = \frac{\left(\frac{1}{2\pi\sigma^2}\right)^{\frac{N}{2}} \exp\left(-\frac{\sum_{A_1}(I(p)-\mu_1)^2}{2\sigma^2}\right) * \left(\frac{1}{2\pi\sigma^2}\right)^{\frac{N}{2}} \exp\left(-\frac{\sum_{A_2}(I(p)-\mu_2)^2}{2\sigma^2}\right)}{\left(\frac{1}{2\pi\sigma^2}\right)^{\frac{N}{2}} \exp\left(-\frac{\sum_{A_1\cup A_2}(I(p)-\mu_0)^2}{2\sigma^2}\right)} \qquad (3\text{-}5)$$

Cette expression peut être simplifiée pour donner l'expression suivante :

$$R = Ln(R_1) = \frac{1}{2\sigma^2}\left\{\sum_{A_1\cap A_2}(I(p)-\mu_0)^2 - \sum_{A_1}(I(p)-\mu_1)^2 - \sum_{A_2}(I(p)-\mu_2)^2\right\} \qquad (3\text{-}6)$$

Le critère est alors optimisé selon

$$\overline{R} = \max_{\mu_1 \mu_2} \min_{\mu_0} R \qquad (3\text{-}7)$$

La décision d'un changement temporel s'effectue selon le critère :

$$\frac{\sqrt{N}}{2\mu}\left|\overline{\mu}_1 - \overline{\mu}_2\right| \begin{matrix} > H_1 \\ < H_2 \end{matrix} \quad Seuil \qquad (3\text{-}8)$$

les valeurs de μ_0, μ_1 et μ_2 sont données par :

$$\begin{cases} \mu_0 = \sum_{A_1 \cup A_2} \dfrac{I(p)}{2N} \\ \mu_1 = \sum_{A_1} \dfrac{I(p)}{N} \\ \mu_2 = \sum_{A_2} \dfrac{I(p)}{N} \end{cases} \qquad (3\text{-}9)$$

Le critère de décision se réduit alors après développement à :

$$\frac{n}{2\mu_0}\left|\mu_1 - \mu_2\right| \begin{matrix} > H_1 \\ < H_0 \end{matrix} \quad \lambda \qquad (3\text{-}10)$$

La carte binaire des changements temporels est alors élaborée :

$$\begin{cases} B(x,y) = 0 \quad si \quad Hyp_0 \\ B(x,y) = 1 \quad si \quad Hyp_1 \end{cases} \qquad (3\text{-}11)$$

2.2.2 La modélisation linéaire :

Sur chaque région, la distribution des niveaux de gris est supposée être linéaire :

$$I = B_1 + B_2 * \Delta x_i + B_3 * \Delta y_j + bruit \qquad (3\text{-}12)$$

avec

Δx_i et Δy_j représente les variations de coordonnées par rapport à un point de référence dans la région .

De même que pour la modélisation constante, on définit une fonction de vraisemblance suivant les deux hypothèses Hyp_0 et Hyp_1. Après un calcul simple mais assez long [BOU95], on aboutit à une règle de décision proche de celle du cas précédent. Dès lors, on déduit la carte binaire des changements temporels. Cette méthode est un bon compromis entre la sensibilité au bruit et un faible coût de calcul.

3. Localisation des objets en mouvement :

L'opérateur de ***M. Orkisz*** (1992) est l'un des plus récents parmi les méthodes, de localisation, dites classiques opérant avec trois images. Il permet d'éliminer tous les contours indésirables, tout en étant plus rapide que l'opérateur de ***Vieren***. En effet, l'opération de multiplication, utilisée par ce dernier, est remplacée par une simple opération de comparaison. La formule de localisation utilisée par ***Orkisz*** est donnée par :

$$\forall_{x,y} \quad M_n(x,y) = Max\left(G_{n-1}(x,y), G_n(x,y), G_{n+1}(x,y)\right) - Max\left(G_{n-1}(x,y), G_{n+1}(x,y)\right) \qquad (3\text{-}13)$$

G_k : désigne le gradient de l'image « k ».

Ce qui donne la formule binaire de ***Orkisz*** :

$$\forall_{x,y} \quad M_n(x,y) = E_n(x,y) \bullet \overline{E_{n-1}}(x,y) \bullet \overline{E_{n+1}}(x,y) \qquad (3\text{-}14)$$

De point de vue algorithmique l'opérateur de ***Orkisz*** est réalisé par :

$$S = Max\left(G_{n-1}(x,y), G_{n+1}(x,y)\right)$$
$$si \quad G_n(x,y) > S \quad alors \quad M_n = G_n(x,y) - S$$
$$si\ non \quad M_n = 0.$$

La technique de seuillage est souvent utilisée pour la binarisation des résultats, en raison de sa simplicité. Cependant, un seuil permettant d'éliminer le bruit, peut faire disparaître certaines frontières correspondant aux zones de faible contraste. On peut également utiliser d'autres méthodes de seuillage, comme par exemple l'amincissement multiniveaux ou bien le seuillage à hystérésis, qui devront faire ressortir les frontières éliminées [ORK91][KUN93].

Pour le calcul du gradient, plusieurs opérateurs ont été proposés (***Canny***, ***Deriche***, ***Robert***, ***Sobel***, ...etc.)[KUN93]. Ces méthodes consistent à effectuer deux opérations : un lissage et le calcul de la dérivée. L'opération de lissage permet de déterminer l'épaisseur des frontières ainsi extraites. Pour notre cas, nous avons utilisé l'opérateur de ***Sobel*** qui reste l'un des meilleurs pour ce type d'applications :

$$S_1 = \begin{vmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{vmatrix} \quad ; \quad S_2 = \begin{vmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & 1 \end{vmatrix} \qquad (3\text{-}15)$$

L'application de différents algorithmes sur une séquence d'images réelles, prise au sein de notre institut, est donnée sur la figure (***Fig. III.1***) :

D'après ces résultats on peut conclure que :

- Pour la localisation des véritables frontières mobiles, certaines méthodes font ressortir des frontières parasites : désoccultées (méthode de ***Stelmaszyk***) ou cachées dans l'image courante mais visibles dans ses deux voisines (méthode de ***Vieren***). On peut conclure que la méthode de ***Orkisz*** reste la meilleure de point de vue détection de véritables frontières mobiles.

- De point de vue rapidité, la méthode de ***Orkisz*** reste la meilleure à cause du nombre réduit d'opérations par pixels ainsi que la possibilité de réutiliser les résultats pour les images suivantes.

- On peut affirmer aussi que la méthode de ***Orkisz*** offre un bon compromis entre la détection des véritables frontières mobiles et la sensibilité aux bruits.

conclusion

- La localisation des objets en mouvement, tout comme la détection, peut être considérée comme une étape préliminaire à l'estimation du mouvement. Avec la remarque que les vecteurs vitesses sont souvent perpendiculaires aux frontières des objets en mouvement.

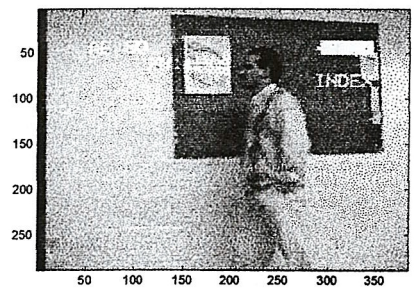


Image 1

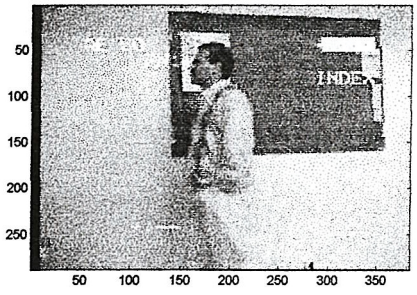


Image 2

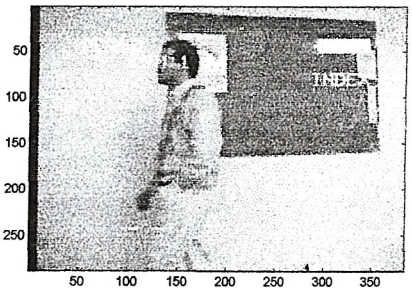
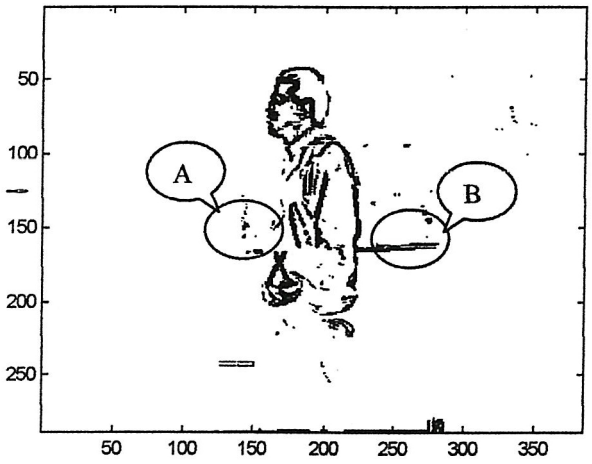
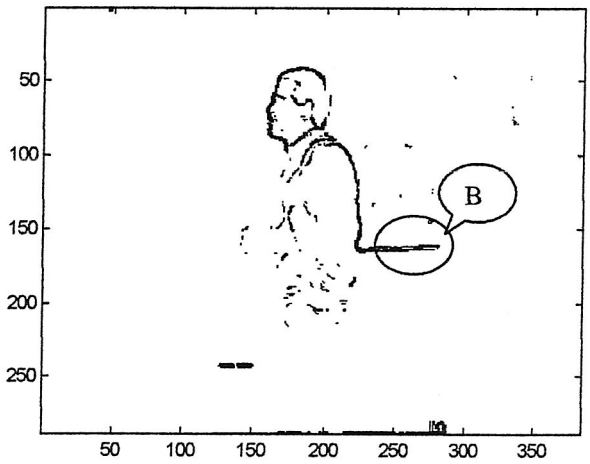


Image 3

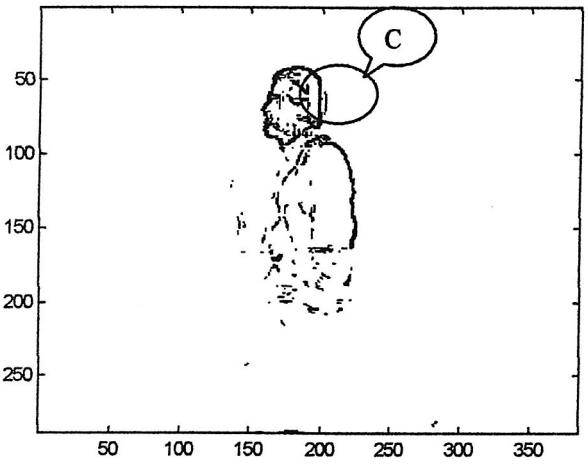


Méthode de Haynes et Jain

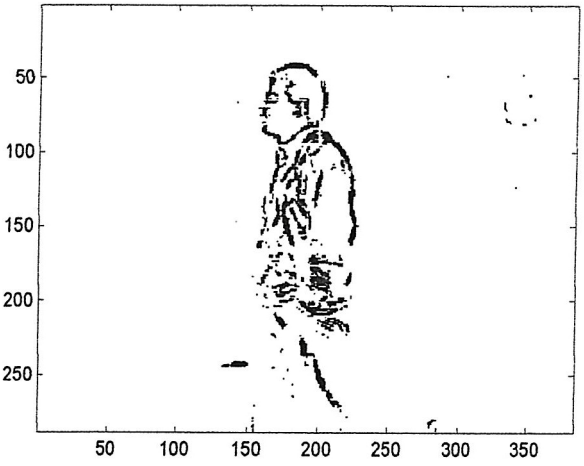


Méthode de Stelmaszyk

Méthodes à base de deux images



Méthode de Vieren



Méthode de Orkisz

Méthodes à base de trois images

Zone A : Problème de bruit

Zone B : Problème des frontières disocculter

Zone C : Problème des frontières absentes dans l'image courante et présentes dans les deux

Fig. III.1 : Localisation des frontières des objets en mouvement

4. Estimation du mouvement :

La majorité des méthodes d'estimation du mouvement (estimation du flot optique) prennent en compte le mouvement propre de la caméra. Mais elles exigent un éclairage, non seulement constant dans le temps, mais aussi homogène dans l'espace. La deuxième hypothèse suppose que la brillance d'un point dans l'espace (et par-là, même la luminance au point projeté dans l'image) est invariante dans le temps, du moins entre les deux instants successifs. Supposons qu'un point de coordonnées p=(x,y) et d'intensité lumineuse I(x,y,t) = I(p,t) se déplace de d=(dx,dy) de l'instant « t » à l'instant « t+dt », alors nous pouvons écrire :

$$\begin{cases} DFD(p,d) = I(x+dx, y+dy, t+dt) - I(x,y,t) \\ DFD(p,d) = I(p+d, t+dt) - I(p,t) \end{cases} \qquad (3.16)$$

DFD : Displaced Frame Différence.

dx : Déplacement suivant l'axe x.

dy : Déplacement suivant l'axe y.

d : Vecteur de déplacement, considéré identique au vecteur de vitesse $w = (u,v) = (\frac{dx}{dt}, \frac{dy}{dt})$ tant qu'il s'agit des données discrétisées. Ils seront confondus au facteur « dt » près ($d = wdt$).

Afin de tenir compte du bruit, des erreurs de quantification et des ombres ...etc., il est inutile de chercher à annuler la DFT, il faut en revanche la minimiser. Pour cet effet, plusieurs méthodes ont été proposées.

4.1 La méthode de mise en correspondance (Estimation par similarité des blocs) :

La méthode de mise en correspondance (Block Matching BM) consiste à considérer l'image à l'instant « t » comme étant une grille de blocs de taille NxM pixels, et de chercher le bloc qui lui ressemble le plus dans l'image « t+dt », alors le vecteur de déplacement (vecteur de vitesse) est la différence entre les positions des deux blocs. Suivant la séquence à traiter, un déplacement maximum est fixé « d », ce qui limite la recherche à une fenêtre de dimension ((2*d+N)x(2*d+M)) appelée fenêtre de recherche. la figure suivante montre le principe de cette méthode :

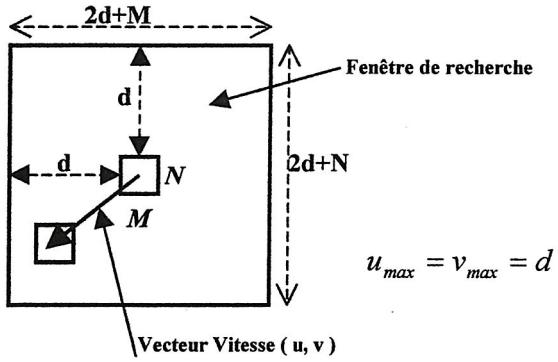


Fig. III.2 : Principe de base d'estimation du mouvement par la méthode de mise en correspondance

Pour la mesure de ressemblance trois critères peuvent être utilisés : l'erreur quadrature moyenne (MSE), l'erreur absolue moyenne (MAE) et le critère MPC, où la ressemblance est inversement proportionnelle à l'erreur mesurer. A cause de sa simplicité, le critère MAE et le plus sollicité pour les implémentations matérielles. Quant au critère MSE il est le mieux adapté aux implémentations logicielles. L'erreur entre les deux fenêtres (k : k+N-1, l : l+M-1 , t) et (k+u : k+u+N-1, l+v : l+v+M-1 , t+dt) est donnée par :

- ***Mean Square Error (MSE) :***

$$MSE(k,l,u,v) = \frac{1}{M*N}\sum_{i=0}^{N-1}\sum_{j=0}^{M-1}\left[I_t(k+i,l+j) - I_{t+dt}(k+i+u,l+j+v)\right]^2 \qquad (3\text{-}17)$$

- ***Mean Absolute Error (MAE) :***

$$MAE(k,l,u,v) = \frac{1}{M*N}\sum_{i=0}^{N-1}\sum_{j=0}^{M-1}\left|I_t(k+i,l+j) - I_{t+dt}(k+i+u,l+j+v)\right| \qquad (3\text{-}18)$$

- ***Maximum matching Pel Count (MPC) :***

$$MPC(k,l,u,v) = \sum_{i=0}^{N-1}\sum_{j=0}^{M-1} T(k+i,l+j,u,v) \qquad (3\text{-}19)$$

avec

$$T(s,t,u,v) = \begin{cases} 1 & \text{si } \left|I_t(s,t) - I_{t+dt}(s+u,t+v)\right| \le \lambda \\ 0 & \text{si non} \end{cases} \qquad (3\text{-}20)$$

Pour la stratégie de balayage, trois structures peuvent être utilisées : le balayage exhaustif, le balayage orienté et le balayage multigrilles. L'algorithme standard (BMA) utilise un balayage exhaustif qui consiste à tester la ressemblance pour toutes les positions possibles (Full

Search Algorithm FSA). Il assure de bons résultats, avec toujours l'handicapes du temps de calcul relativement élevé (pour chaque bloc, on effectue $(2*d+1)^2$ mesures de ressemblance). Plusieurs autres algorithmes (basés sur la structure orientée et la structure multigrille) ont été proposés pour réduire le temps de calcul. Ces algorithmes restent, de loin, les meilleurs de point de vue temps de calcul, mais ils sont souvent piégés par des maxima locaux.

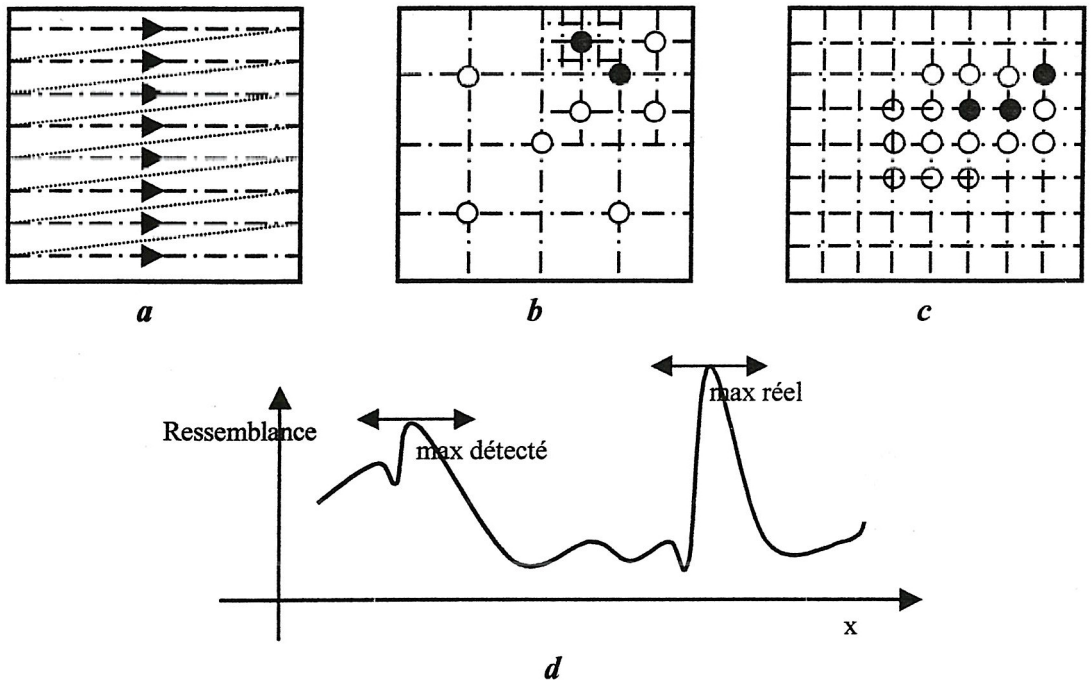


***Fig. III.3** : Stratégie de balayage et le problème De maximum local*

a : balayage exhaustif b : balayage multigrille

c : balayage orienté d : problème de maximum local

4.1.1 Algorithme de recherche à 3 étapes 3SS (Three-Step Search) :

Cet algorithme, proposé par ***T. Koga*** en 1981, consiste à utiliser une stratégie multigrille de trois étapes, avec un déplacement décroissant. Le déplacement initial est pris égal à d/2 (arrondi à l'entier le plus grand).

Si on prend l'exemple de d=7 et M=N=3. pour chaque étape on a 8 blocs de l'image I(t+dt)) à comparer avec le bloc central (de l'image I(t), sauf pour l'étape 1, où le nombre de blocs est égal à 9. Le nombre de blocs total balayés est égal à (9+8+8 = 25 blocs).

Pour le même exemple, et si on utilise un balayage exhaustif, le nombre des blocs balayés sera égal à ($(2*7+1)^2 = (15)^2 = 225$ *blocs*).

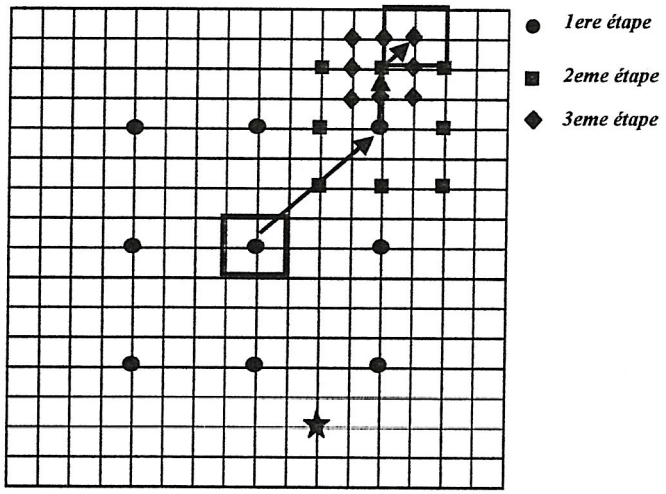


***Fig. III.4** : Principe de la méthode 3SS*

Cette méthode permet de réduire le temps de calcul, mais elle peut échouer dans le cas de l'existence de plusieurs maxima locaux (si le maximum se trouve à la position (*) sur la figure (***Fig. III.4***)).

On peut, éventuellement, généraliser l'algorithme à « N » étapes (NSS) selon la valeur de d. Le nombre de blocs balayés est égal à «$1+8*Log_2(d+1)$» [CHE98].

- ***d = 15*** : ***4SS*** : ***33 Blocs***
- ***d = 31*** : ***5SS*** : ***41 Blocs***

4.1.2 Méthode avec recherche logarithmique 2D (2DLOG) :

Cette méthode proposée par ***J. R. Jain*** et ***A. K. Jain*** en 1981, utilise une recherche orientée et croisée (en +) sur plusieurs étapes. L'étape initiale utilise un déplacement de (d/4). D'une étape à l'autre, la valeur de déplacement est divisée par 2 si le bloc cherché est le bloc central de l'étape ou bien si on arrive aux limites de la fenêtre de recherche, sinon la valeur de déplacement reste identique à l'étape précédente. Lorsque la valeur de déplacement devient égale à 1, l'algorithme utilise les 8 voisins pour la recherche du minimum. La figure suivante montre le principe de cette méthode :

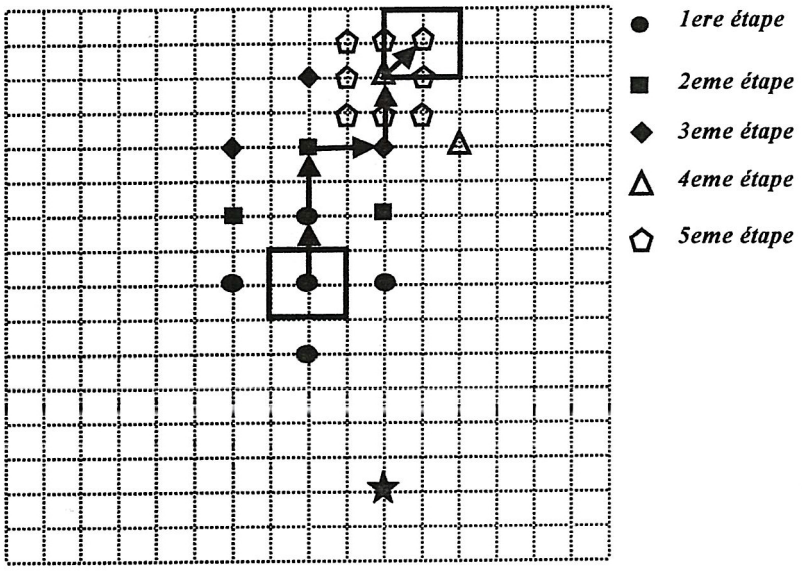


***Fig. III.5** : Principe de la méthode 2DLOG*

Pour cet exemple, le nombre de blocs balayés est égal à (5+3+3+2+8=21 blocs).

4.1.3 Méthode avec recherche Orthogonale (OSA) :

La méthode (OSA) proposée par ***A. Puri*** en 1987, consiste à utiliser un balayage orienté avec des étapes horizontales/verticales. La valeur de déplacement, pour chaque étape, est divisée par deux, avec une étape initiale de déplacement égale à d/2.

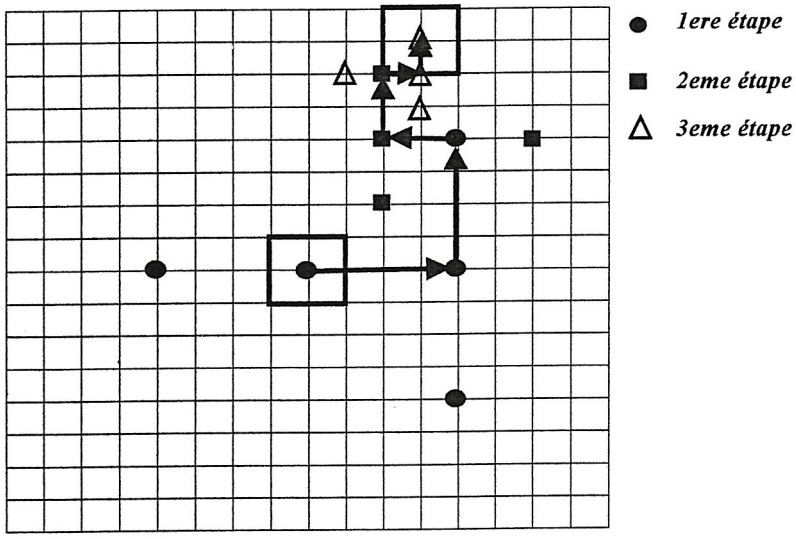


***Fig. III.6** : Principe de la méthode OSA*

4.1.4 Méthode avec recherche en X (Cross Search Algorithm CSA) :

L'algorithme CSA proposé en 1990 par ***M. Ghanbari***, consiste à utiliser un balayage multigrille diagonale (dans les deux directions diagonales X). Pour chaque étape, la valeur de déplacement est divisée par 2 avec un déplacement initial égal à d/2.

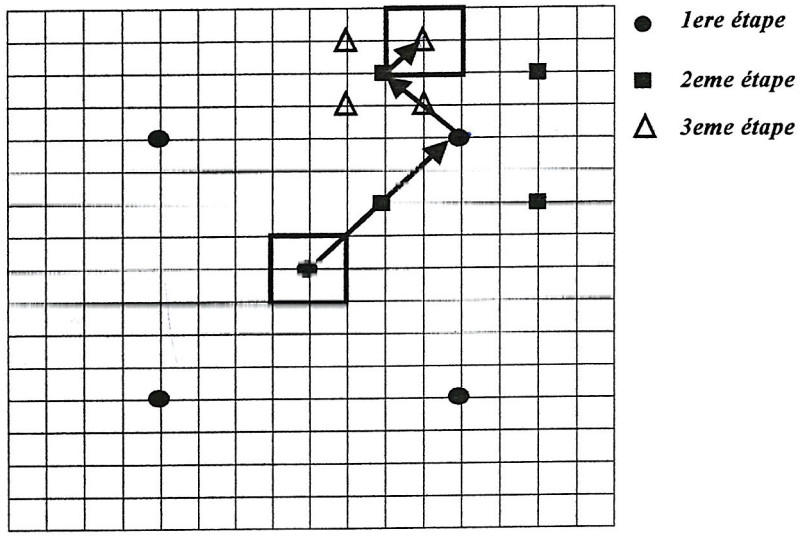


Fig. III.7 : Principe de la méthode CSA

4.1.5 Nouvel algorithme de recherche à 3 étapes (N3SS) :

Le N3SS, proposé par ***R. Li, B. Zeng*** et ***M. L. liou*** en 1994, est la version modifiée de l'algorithme 3SS pour la détection des petits déplacements dans les séquences d'images. Cette méthode utilise deux chemins de recherche pour éviter les maxima locaux, un pour les petits déplacements, et l'autre pour les grands déplacements. Le premier chemin utilise un balayage orienté avec un maximum de déplacement égal à d/2. Le deuxième utilise un balayage multigrille (la même stratégie que l'algorithme 3SS).

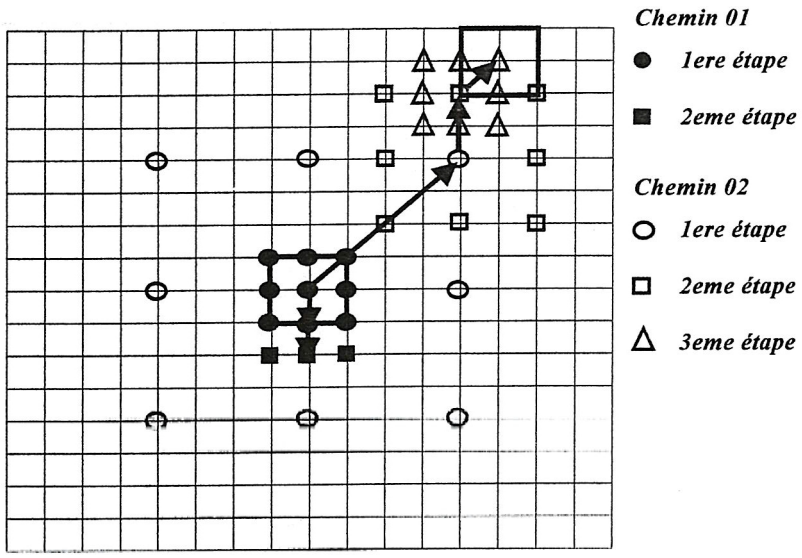


***Fig. III.8* : Principe de la méthode N3SS**

4.1.6 Algorithme de recherche à 4 étapes (4SS) :

L'algorithme 4SS, proposé par ***L. M. Po*** et ***W. C. Ma*** en 1996, utilise un petit déplacement initial (d/4). Il est effectué en 4 étapes avec un balayage multigrille. D'une étape à l'autre, la valeur du déplacement est divisée par deux si le bloc à chercher est le bloc central de l'étape, ou encore si on arrive à la limite de la fenêtre de recherche.

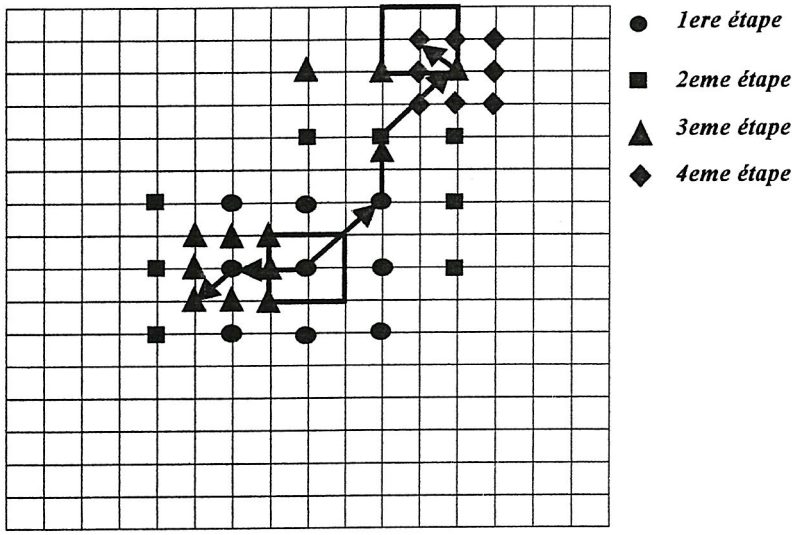


***Fig. III.9* : Principe de la méthode 4SS**

Dans cet exemple, deux chemins différents ont été exposés :

- Chemin 01 : pour des grands déplacements

  Le nombre de blocs balayés est égal à : (9+5+3+8 =25 Blocs).

- Chemin 02 : pour des petits déplacements

  Le nombre de blocs balayés est égal à : (9+3+8 =20 Blocs).

Le tableau suivant montre le nombre, des blocs balayés, nécessaires pour la mesure de ressemblance de chaque algorithme, avec d = 7 et N = M = 3.

	FS	3SS	2DLOG	OSA	CSA	N3SS	4SS
Nombre d'op. min.	**255**	**25**	**13**	**13**	**17**	**20 ! 33**	**17**
Nombre d'op. max	**255**	**25**	**23**	**13**	**17**	**20+33**	**25**

Tableau III.1 : Nombre de mesure de ressemblance maximum et minimum de chaque algorithme

4.2 La méthode Pel-récursive pour la minimisation de la DFD² :

La méthode Pel-récursive, proposée par ***A.N. Netravali*** et ***J.D. Robbins*** (1979), pour l'estimation du mouvement est une méthode de prédiction et de correction. Elle vise à effectuer une bonne prédiction plutôt qu'une bonne estimation du mouvement réel. Elle consiste à minimiser, d'une manière itérative, la fonction DFD² pour chaque pixel. Le déplacement d_i estimé dans une itération « i », est donné par la formule suivante [TZI88][ORK92][CHE98] :

$$d_i = \begin{bmatrix} u_i \\ v_i \end{bmatrix} = d_{i-1} - \varepsilon * DFD(p, d_{i-1}) * \nabla I(p - d_{i-1}, t - dt) \qquad (3\text{-}21)$$

ε : Le pas d'adaptation de l'algorithme (de l'ordre de 10^{-3}).

$$\nabla I(p - d_{i-1}) = \begin{pmatrix} I_x(x - u_{i-1}, y - v_{i-1}, t - d) \\ I_y(x - u_{i-1}, y - v_{i-1}, t - d) \end{pmatrix}$$

Si le mouvement est une translation pure sur le plan image, cet algorithme converge vers le vrai vecteur de déplacement, mais cette convergence est lente. Les efforts des chercheurs sont alors focalisés essentiellement sur l'accélération de la convergence de cet algorithme. En 1983,

C. Cafforio et ***F. Rocca*** [TZI88] ont proposé l'utilisation d'un pas d'adaptation variant suivant l'activité spatiale dans l'image, où ε est donné par :

$$\varepsilon = \frac{1}{\lambda + I_x^2(p - d_{i-1}, t - dt) + I_y^2(p - d_{i-1}, t - dt)} \qquad (3\text{-}22)$$
$$\lambda = 100$$

Dans le même sens, ***D. Walker*** et ***K. Rao*** (1984) [TZI88] ont proposé l'utilisation du pas d'adaptation suivant :

$$\varepsilon = \frac{0{,}5}{I_x^2(p - d_{i-1}, t - dt) + I_y^2(p - d_{i-1}, t - dt)} \qquad (3\text{-}23)$$

Comme toutes les autres méthodes récursives, les méthodes Pel-récursives souffrent de point de vue temps de calcul en plus du problème d'initialisation. En effet, il faut généralement plusieurs pixels pour remettre à jour l'estimateur.

4.3 Méthodes différentielles pour l'estimation du mouvement :

Les méthodes de mise en correspondance adaptées dans le domaine de codage d'image, où une légère différence entre le mouvement réel et le mouvement détecté est autorisée. Par contre, pour des problèmes d'analyse des scènes dynamiques, le calcul du véritable vecteur déplacement est important. Si l'on suppose que la fonction intensité possède des propriétés de continuité et de dérivabilité jusqu'à un ordre requis, on peut alors considérer le développement de Taylor de la fonction intensité au point (x,y,t) :

$$I(x + dx, y + dy, t + dt) = I(x, y, t) + dx\frac{\partial I}{\partial x} + dy\frac{\partial I}{\partial y} + dt\frac{\partial I}{\partial t} + \theta^2 \qquad (3\text{-}24)$$

Si l'on néglige les termes d'ordres supérieurs, alors :

$$DFD(p, d, t) = dx\frac{\partial I}{\partial x} + dy\frac{\partial I}{\partial y} + dt\frac{\partial I}{\partial t} = 0 \qquad (3\text{-}25)$$

Ce qui donne l'équation de contrainte du mouvement (ECM) qui relie les gradients spatiaux et temporels de l'intensité lumineuse :

$$\frac{\partial I}{\partial x}\cdot\frac{dx}{dt}+\frac{\partial I}{\partial y}\cdot\frac{dy}{dt}+\frac{\partial I}{\partial t}=0 \quad \text{ou bien} \quad \nabla I\cdot w=-I_t \qquad (3\text{-}26)$$

avec :

$w=(u,v)=(\frac{dx}{dt},\frac{dy}{dt})$ Vecteur de vitesse.

$\nabla I=(\frac{\partial I}{\partial x},\frac{\partial I}{\partial y})$ Vecteur gradient de l'image.

L'approche différentielle consiste alors à exploiter une relation entre les déplacements d'une part, et les dérivées spatiales et temporelles de l'intensité lumineuse d'autre part. L'équation de contrainte du mouvement avec deux inconnues (problème indéterminé), ne permet de déterminer que la projection du vecteur de vitesse sur le vecteur gradient spatial de l'intensité lumineuse, ce qui traduit formellement le problème d'ouverture (aperture problem) [VER90][BOU95][BIM95].

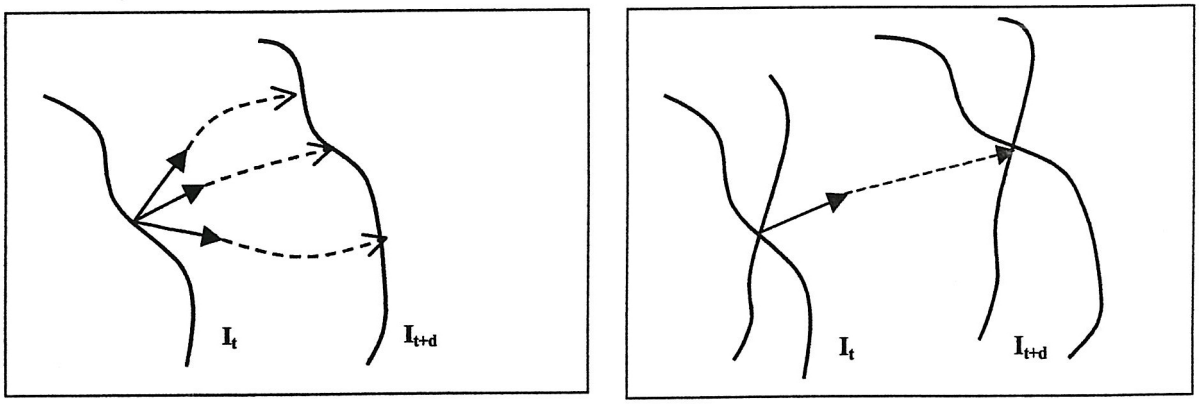


***Fig. III.10* : Synoptique du problème d'ouverture**

Pour résoudre ce problème, diverses approches ont été élaborées dans le but d'introduire des contraintes supplémentaires (***Fig. III.10-b***).

4.3.1 Méthodes à plusieurs entrées :

Dans cette méthode, on peut considérer conjointement plusieurs observations de mouvement. Par exemple, si le signal capté est formé de trois composantes de couleur RVB, le système devient surdéterminé [BOU88] :

$\nabla R \cdot w = -R_t$ Equation de contrainte du mouvement pour le signal rouge.

$\nabla V \cdot w = -V_t$ Equation de contrainte du mouvement pour le signal vert.

$\nabla B \cdot w = -B_t$ Equation de contrainte du mouvement pour le signal Bleu.

On peut aussi utiliser les dérivées secondes de la fonction intensité comme observations supplémentaires.

4.3.2 Contraintes de lissage explicites :

Cette méthode est basée sur une hypothèse supplémentaire, selon laquelle les champs des vecteurs de vitesse sont lisses, où les variations du champ de vitesse d'un point à un autre est a priori faible [TZI88][BOU88][ORK93]. L'erreur totale à minimiser devient alors :

$$\iint (w \cdot \nabla I + I_t)^2 + \alpha^2 (u_x^2 + u_y^2 + v_x^2 + v_y^2) \qquad (3\text{-}27)$$

$$(u_x, u_y) = (\frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}) \quad ; \quad (v_x, v_y) = (\frac{\partial v}{\partial x}, \frac{\partial v}{\partial y})$$

α^2 est un facteur à travers lequel est exprimé les poids relatifs attribués à l'observation.

Plusieurs chercheurs (***Tziritas88***, ***Anandan89***, ***Nagel90***, ...) ont proposé des améliorations pour la détection des discontinuités du champ des vitesses, et pour éviter le piège du minimum local. Cependant, ces méthodes souffrent des fausses estimations dues au lissage à travers les frontières entre objets différents (occlusion boundaries), et aux régions homogènes de l'image (problème d'ouverture)[BOU95]. D'autre part, l'implantation numérique complète de ces méthodes est relativement lourde (calculs itératifs).

Dans le même but de minimiser la fonction DFD, plusieurs méthodes ont été proposées tout en tenant compte de l'orientation du gradient (précédé l'estimation par l'opération de localisation des frontières mobiles).

4.3.3 Regroupement de droites de contrainte :

Cette méthode proposée par ***Schunk*** (1989) [ORK92-a], utilise le reflet graphique de l'équation de contrainte du mouvement dans le plan des composantes orthogonales du vecteur de vitesse. En effet, l'équation de contrainte du mouvement décrit une droite dans l'espace des vitesses ($\boldsymbol{u}$, $\boldsymbol{v}$).

$$d = \frac{I_t}{\sqrt{I_x^2 + I_y^2}} = \frac{|I_t|}{|\nabla I|}$$

$$\theta = \begin{cases} arctg(I_x, I_y) & pour\ I_t \geq 0 \\ arctg(-I_x, -I_y) & pour\ I_t < 0 \end{cases}$$

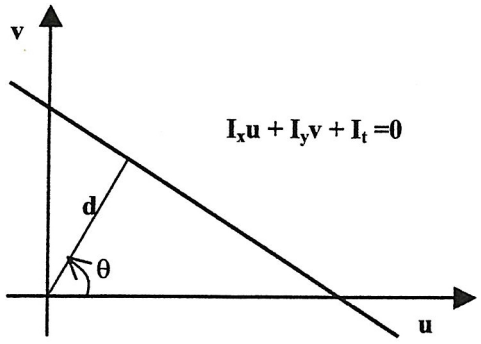


Fig. III.11 : ***Représentation graphique de l'équation de contrainte du mouvement dans le plan (u , v)***

L'orientation de cette droite étant dépendante de l'orientation du gradient spatial pour le point considéré. La méthode de ***Schunk*** consiste à déterminer l'ensemble des intersections de la droite de contrainte du point considéré avec des droites de contrainte des points appartenant à un voisinage donné et à chercher le centre de la partie la plus dense de cet ensemble [ORK93]. Généralement, on utilise un voisinage d'ordre 2 donné par la figure (***Fig. III.12-a***).

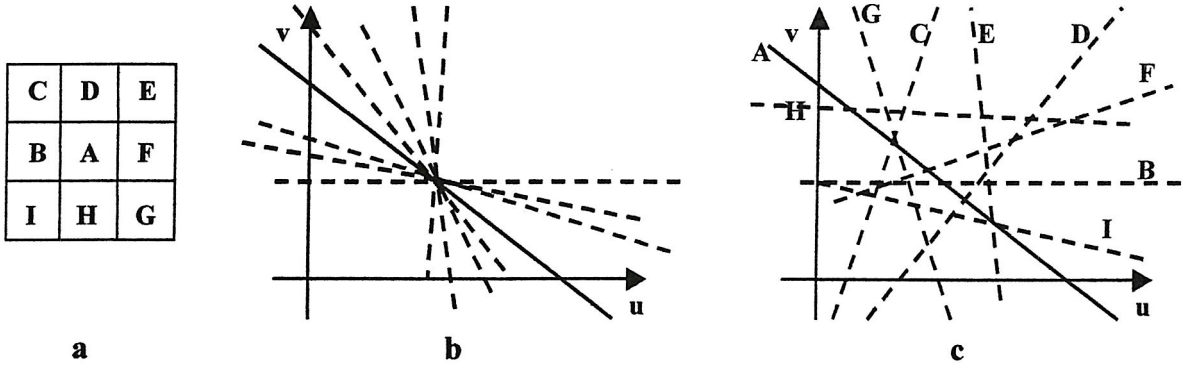


Fig. III.12 : ***Principe de la méthode de regroupement des droites de contraintes***
a : voisinage de 8 pixels b : cas idéal c : cas réel

Tout comme les autres méthodes, le regroupement de droites de contrainte souffre du point de vue problème des régions homogènes et du problème d'ouverture. Pour prendre en compte ces problèmes, ***Schunk*** conseille de prendre un voisinage suffisamment grand. Seulement, ceci peut nous amener à un nuage de points d'intersections bimodales proches de deux objets en mouvement.

Dans le but de minimiser le temps de calcul et de réduire les effets du bruit, ***M. Orkisz*** et ***D. Pellerin*** ont proposé une méthode d'estimation du mouvement par une approche mixte [ORK93]. Cette méthode consiste à calculer un maximum de vecteurs de vitesse à l'aide de l'équation de contrainte, en déterminant auparavant les zones de validité de cette équation. Hors les zones de validité de l'équation de contrainte, les deux auteurs proposent d'utiliser la mise en correspondance et de ne pas estimer le mouvement dans les zones homogènes ou aucune méthode ne peut donner un résultat fiable.

4.3.4 Formulation mathématique du système surdéterminer :

La prise en compte des équations de contrainte de mouvement des points appartenant à un voisinage donné, ou bien la prise en compte de plusieurs entrées, nous ramène à résoudre un système surdéterminé de la forme suivante, ou N représente le nombre d'entrée (le nombre de voisinage) pris en compte :

$$\begin{cases} I_x^1 u + I_y^1 v = -I_t^1 \\ I_x^2 u + I_y^2 v = -I_t^2 \\ \quad .... \\ \quad .... \\ I_x^N u + I_y^N v = -I_t^N \end{cases} \qquad (3\text{-}28)$$

Sous forme matricielle, le système s'écrit :

$$-\begin{bmatrix} I_x^1 & I_y^1 \\ I_x^2 & I_y^2 \\ . & . \\ . & . \\ I_x^N & I_y^N \end{bmatrix} * \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} I_t^1 \\ I_{t.}^{21} \\ . \\ . \\ I_t^N \end{bmatrix} \qquad (3\text{-}29)$$

Ou bien :

$$G * W = T \qquad (3\text{-}30)$$

Avec :

$$G = -\begin{bmatrix} I_x^1 & I_y^1 \\ I_x^2 & I_y^2 \\ . & . \\ . & . \\ I_x^N & I_y^N \end{bmatrix} \quad ; \quad W = \begin{bmatrix} u \\ v \end{bmatrix} \quad ; \quad T = \begin{bmatrix} I_t^1 \\ I_{t.}^{21} \\ . \\ . \\ I_t^N \end{bmatrix}$$

Dans le cas idéal, ce système présente une solution unique correspondante au vecteur de vitesse. Graphiquement, ce cas est vérifié si l'intersection de toutes les droites de contrainte s'effectue en un seul point (***fig. III.11-b***). Cette possibilité est rarement réalisée. Dans la majorité des cas plusieurs intersections sont effectuées (***Fig.III.11-c***). Une estimation du vecteur de vitesse $\overline{W}$ peut être calculée par la minimisation d'une norme d'erreur « e », définie par :

$$\|e\| = \left\|T - G * \overline{W}\right\| \qquad (3\text{-}31)$$

La solution du système d'équations sous forme matricielle est donnée par :

$$W = \left(G^t * G\right)^{-1} * G^t * T \qquad (3\text{-}32)$$

Ou bien :

$$\begin{bmatrix} u \\ v \end{bmatrix} = -\begin{bmatrix} \sum_{j=1}^{N}\left(I_x^j\right)^2 & \sum_{j=1}^{N} I_x^j * I_y^j \\ \sum_{j=1}^{N} I_x^j * I_y^j & \sum_{j=1}^{N}\left(I_y^j\right)^2 \end{bmatrix} * \begin{bmatrix} \sum_{j=1}^{N} I_x^j * I_t^j \\ \sum_{j=1}^{N} I_y^j * I_t^j \end{bmatrix} \qquad (3\text{-}33)$$

Après certaines simplifications, la solution devient :

$$\begin{bmatrix} u \\ v \end{bmatrix} = -\frac{1}{AD - BC}\begin{pmatrix} D & -B \\ -C & A \end{pmatrix}\begin{pmatrix} T_i \\ T_2 \end{pmatrix} \qquad (3\text{-}34)$$

avec :

$$A = \sum_{j=1}^{N}\left(I_x^j\right)^2 \quad ; \quad D = \sum_{j=1}^{N}\left(I_y^j\right)^2$$

$$B = \sum_{j=1}^{N} I_x^j * I_y^j \quad ; \quad C = B$$

$$T_1 = \sum_{j=1}^{N} I_x^j * I_t^j \quad ; \quad T_2 = \sum_{j=1}^{N} I_y^j * I_t^j$$

On remarque que la solution existe sauf si la matrice $(G^t * G)$ est irréversible, ou bien :

$$\Delta = AD - BC \neq 0 \qquad (3\text{-}35)$$

5. Approches stochastiques pour la détection et l'estimation du mouvement :

Les champs de markov sont connus par les statisticiens depuis les années 70. Vers les années 80, les champs de markov ont fait une apparition remarquée dans le domaine du traitement d'images, d'abord appliqués à des problèmes de restauration, et ensuite dans des domaines aussi variés (l'extraction de contours, la segmentation, la reconstruction de surface et même pour la détection et l'estimation du mouvement) [HET91][PER92].

Comme n'importe qu'elle autre nouvelle méthode, la modélisation Markovienne ait un vocabulaire spécial et assez étrange pour des personnes non spécialisées dans le domaine. Des termes comme sites, étiquettes, voisinage, modèle de Gibbs ...etc., sont utilisés sans vergogne. Pour résoudre ce problème, on propose une étude théorique de l'utilisation des modèles stochastiques, soit pour la détection des changements temporels soit pour l'estimation des paramètres du mouvement.

5.1 Rappels théoriques sur les champs de Markov :

On adopte les notations suivantes :

- s=(x,y,t) : « site », désigne un pixel (groupe de pixel) de coordonnées (x,y,t)

  $S=\{ s[1...N_x]x[1...N_y]x[1...N_t] \}$: est l'ensemble des sites.

- Les données disponibles pour chaque site (l'intensité lumineuse I_n, l'image de différence FD(p,t), ...etc.) désigne le champ d'observations.

  $O=\{ O(s) , s \in S \}$ champ d'observations.

  $o=\{ o(s) , s \in S \}$ est une réalisation particulière du champ O.

- Les primitives à extraire de la séquence d'images sont les étiquettes à attribuer aux sites, en partant de ces observations et d'un modèle du phénomène étudié.

  $E=\{ E(s) , s \in S \}$ champ d'étiquettes.

  $e=\{ e(s) , s \in S \}$ est une réalisation particulière du champ E.

  Exemples d'étiquettes suivant leurs applications :

  - Mobile/Immobile : Localisation des objets en mouvement.
  - Numéros des régions ou des objets : Problème de poursuite.
  - Couple (Module, Orientation) de vecteur déplacement : Estimation du mouvement.

- $V=\{ V_s , s \in S \}$ une collection de voisinages sur S, avec :

  $$s \in V_s$$

  $$r \in V_s \quad \Rightarrow \quad s \in V_r$$

V_s : le voisinage du site s.

V_r : le voisinage du site r.

L'ordre du champ détermine la taille du voisinage considéré, on trouve :

- Le voisinage d'ordre 1 : composé de 5 pixels.
- Le voisinage d'ordre 2 : composé de 9 pixels.

A chaque voisinage, on associe un ensemble de cliques (sous-ensemble de pixels inclue dans un voisinage). Une clique comprend toujours le pixel central, et deux pixels d'une même clique sont nécessairement voisins (***Fig. III.13***).

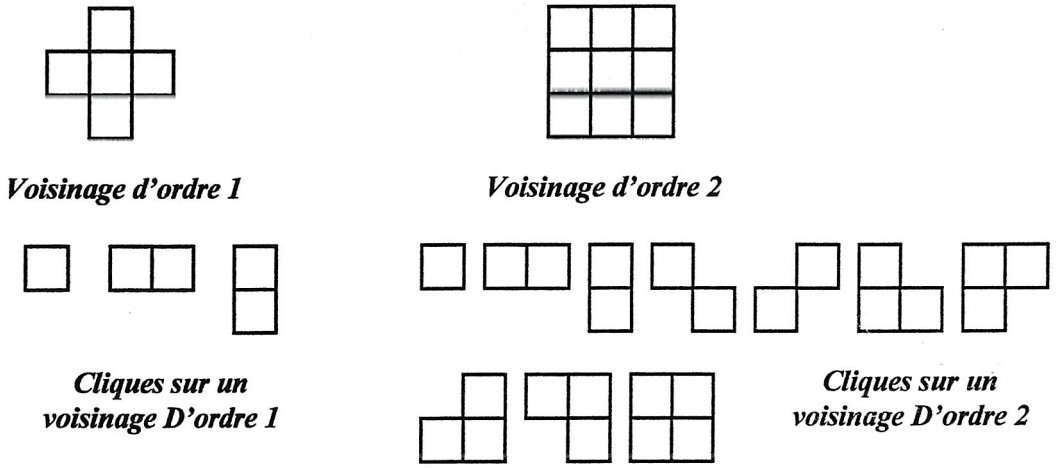


***Fig. III.13** : Voisinages et cliques associés aux voisinages*

- $V_c(e)$: Potentiel élémentaire associé au clique c=(s,r).
- U(e) : La fonction d'énergie associée à e, elle se décompose en une somme de fonctions locales $V_c(e)$,dites potentiels, définies sur les cliques.

$$U(e) = \sum V_C(e) \qquad (3\text{-}36)$$

- Le Modèle : Un modèle a priori du champ d'étiquettes égale à l'ensemble des cliques et des potentiels.
- A : L'ensemble discret des étiquettes possibles, ce qui nous donne toutes les configurations possibles de E définis par Ω, avec le cardinale de Ω est donné par $|\Omega| = A^{|S|}$.

Exemple :

A = { 0,1 } = { Mobile, Immobile } : cas de détection de mouvement.

$|\Omega| = 2^{|S|}$

- Le champ de Markov : Un champ d'étiquettes est considéré comme Markovien lorsque la probabilité qu'à un site donné correspondre une étiquette particulière dépend des étiquettes des sites voisins, mais ne dépend pas de l'image tout entière. C'est-à-dire :

  - $\forall e \in \Omega \quad et \quad \forall s \in S \quad : \quad \Pr(E = e) > 0$
  - $$\Pr[E(s) = e(s) / E(r) = e(r) \; ; r \neq s \; ; r \in S] = \Pr[E(s) = e(s) / E(r) = e(r) \; ; r \neq s \; ; r \in V] \qquad (3\text{-}37)$$

Avec

$\Pr[X = x]$ = Probabilité que X=x.

➢ ***Distribution de Gibbs :***

$$Pr[E(s) = e(s)] = \frac{1}{Z_T} exp\left(\frac{-U(e)}{K_0 T}\right) \qquad (3\text{-}38)$$

c'est la mesure de Gibbs associée à l'énergie U(e) et à la température T.

Z_T : Constante de normalisation appelée fonction de partition, définie par :

$$Z_T = \sum_{e \in \Omega} \exp\left(-\frac{U(e)}{K_0 T}\right) \qquad (3\text{-}39)$$

5.2 Les différentes méthodes de relaxation :

Parmi l'ensemble des réalisations possibles, on cherche le champ « e » des étiquettes qui maximise la probabilité $\Pr(E = e / O = o)$. Le critère le plus utilisé est celui du Maximum A Posteriori (MAP), leur principe est simple : étant donné une réalisation « o » du champ d'observations, on cherche la réalisation « $\hat{e}$ » du champ d'étiquettes la plus probable au sens de la distribution a posteriori, soit :

$$\hat{e} = \arg\max[pr(E = e / O = o)] \qquad (3\text{-}40)$$

Notons que l'on peut trouver d'autres critères dans la littérature. L'avantage du critère MAP est qu'il offre la possibilité d'utiliser la règle de Bayes :

$$\Pr[E = e / O = o] = \frac{\Pr[O = o / E = e] * \Pr[E = e]}{\Pr[O = o]} \qquad (3\text{-}41)$$

Le terme $\Pr(O=o)$ est considéré comme un constant vis-à-vis du problème de maximisation. On suppose en plus que le champ E est Markovien relativement à un voisinage V, et d'après le théorème de Hammersly et Clifford, E suit une distribution de Gibbs :

$$\Pr[E=e]=\frac{1}{Z}\exp(-U_2(e))$$
$$U_2(e)=\sum_{c\in V}V_C(e(s)) \qquad (3\text{-}42)$$

Le terme $\Pr(O=o/E=e)$ définit la relation entre les observations O et les primitives E, qui est de type :

$$O=F(E)+n \qquad (3\text{-}43)$$

F : La fonction caractéristique.

n : bruit gaussien centré de variance σ^2.

posant $U_1(e,o)=-\ln(\Pr(O=o/E=e))$, on voit donc que la maximisation de la probabilité conditionnelle $\Pr(E=e/O=o)$ est équivalente à la minimisation d'une fonction d'énergie globale $U(e,o)$ constituée de deux termes :

$$U(e,o)=U_1(e)+U_2(e,o) \qquad (3\text{-}44)$$

d'où la relation du MAP :

$$\hat{e}=\arg\min(U(e,o))=\arg\min(U_2(e)+U_1(e,o)) \qquad (3\text{-}45)$$

La fonction d'énergie totale $U(e,o)$ est composée de deux termes : un premier terme ($U_1(e,o)$) qui tire la solution vers les observations (l'énergie d'adéquation), et un deuxième terme ($U_2(e)$) qui tire la solution vers le modèle a priori (l'énergie spatio-temporelle ou encore l'énergie de régularisation)[ORK92-a]. Cette énergie n'est pas nécessairement convexe, pour la minimiser, on doit utiliser des algorithmes de relaxation qui peuvent être déterministes, tel que l'algorithme ICM (Itérated Conditional Modes), ou bien stochastique tel que l'algorithme de Métropolis.

5.2.1 Algorithme de relaxation stochastique :

Dans ce type de méthodes, on fait introduire un facteur T (Température) pour assurer la convergence rapide vers une configuration « ê » optimale et pour éviter les minimums locaux. Parmi les algorithmes stochastiques on trouve l'algorithme de Métropolis :

- Soit E_k = e l'étiquette à l'itération k.
- A l'itération k+1:
  - On choisit de manière aléatoire une nouvelle configuration « é » proche de « e ».
  - On calcule la variation d'énergie $\Delta U = U(é) - U(e)$.

    Si $\Delta U < 0$ *alors* $E_{k+1} = é$

    Si $\Delta U \geq 0$ *alors* $E_{k+1} = é$ *avec la probabilité* $e^{-\Delta U}$

    $E_{k+1} = e$ *avec la probabilité* $1 - e^{-\Delta U}$

    En pratique, une valeur $\alpha \in [0\ ,\ 1]$ est choisie de manière aléatoire est sert de seuil.

    $E_{k+1} = é \quad si \quad e^{-\Delta U} < \alpha$

    $E_{k+1} = e \quad si \quad e^{-\Delta U} \geq \alpha$

    L'avantage de cet algorithme c'est qu'il conduit de façon certaine à la solution optimale, mais il présente l'inconvénient du temps de calcul.

5.2.2 Algorithme de relaxation déterministe :

On cherche en chaque point « s » de l'image, l'étiquette qui minimise l'énergie U(e,o). Cet algorithme peut se trouver piéger dans des minimums locaux, l'initialisation est donc très importante, elle doit se situer dans une plage proche du minimum global.

- On choisit un site « s ».
- On calcule l'énergie pour toutes les étiquettes possible de ce site.
- On retient l'étiquette qui a l'énergie la plus faible.

5.2.3 La politique de visite des sites :

L'image peut être parcourue de trois façons différentes :

- Séquentielle : les points de l'image sont traités dans l'ordre. Cette méthode est très rapide mais elle introduit une causalité dans l'image.

- Aléatoire : Les pixels à traiter sont tirées au sort à chaque itération.
- La pile d'instabilité : dans cette méthode, on cherche les pixels instables, c'est-à-dire tous ceux dont l'énergie diffère de l'énergie minimale possible localement. Ces points instables sont alors classés par ordre d'instabilité décroissante (pile d'instabilité). Lors d'une itération, seul le point le plus instable est modifié.

5.3 *Détection de mouvement par modélisation Markovienne :*

La détection du mouvement repose sur l'analyse des variations de la fonction de luminance entre deux instants « t –dt » et « t ». Les observations sont les différences absolues inter-images :

$$o_t(s) = |I(x,y,t) - I(x,y,t-dt)| \qquad (3\text{-}46)$$

A ces observations quantitatives sont ajoutées des observations symboliques binaires $\overline{o_t}(s) \in \{0,1\}$, indiquant la présence ou l'absence de variations significatives de luminance du site « s ». Ces observations binaires sont obtenues soit par un simple seuillage (avec l'inconvénient d'être sensible aux bruits), soit par le test de vraissemblance sur un modèle constant ou bien linéaire [BOU95].

Les étiquettes, quant à elles, prennent leurs valeurs dans $A = \{0,1\}$, pour indiquer l'absence ou la présence de mouvement dans chaque site à l'instant t :

- e(s) = 1 : si le pixel appartient à un objet mobile.
- e(s) = 0 : si le pixel appartient à un fond fixe.

Donc, pour la détection du mouvement à l'instant t, on utilise les images à l'instant t-dt, t et t+dt de la séquence. On suppose en plus que le champ des étiquettes suit la propriété de Markov relativement au voisinage spatio-temporel de la figure suivante :

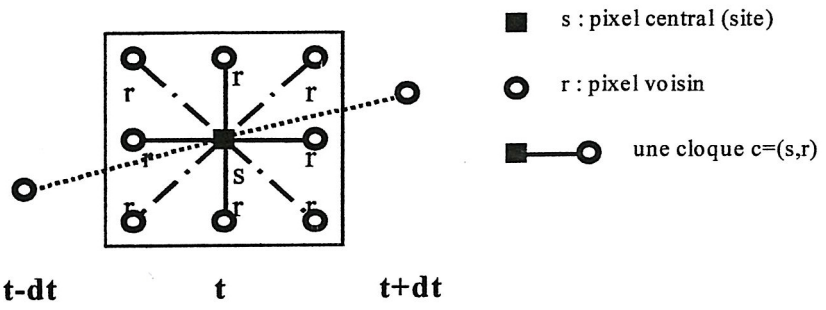


Fig. III.14 : Voisinage spatio-temporel pour la détection du mouvement

L'utilisation d'un voisinage temporelle, en plus du voisinage spatial (utilisé pour le traitement des images fixes), a pour but d'intégrer explicitement la dimension temporelle dans la modélisation. La fonction d'énergie globale se décompose alors en deux termes :

➢ ***L'énergie spatio-temporelle :***

$$U_2(e) = \sum_{c \in V} V_c(e(s)) \qquad (3\text{-}47)$$

Le potentiel d'interaction spatial sert à éliminer les points isolés et assure la cohésion spatiale des objets, par contre l'énergie temporelle sert à supprimer l'écho.

$$V_c(e(s), e(r)) = \begin{cases} -\beta & si \quad e(s) = e(r) \\ +\beta & si \quad e(s) \neq e(r) \end{cases} \qquad (3\text{-}48)$$

Avec β dépend de la nature de la clique considérée :

$$\beta = \begin{cases} \sqrt{2}\beta_s & \text{pour les cliques spatiales verticales et horizontales} \\ \beta_s & \text{pour les cliques spatiales diagonales} \\ \beta_f & \text{pour les cliques temporelles futures} \\ \beta_p & \text{pour les cliques temporelles passées} \end{cases}$$

➢ ***L'énergie d'adéquation :***

$$U_{11}(e_t, o_t, o_{t+dt}) = \sum_{s \in S} \left[\frac{1}{2\sigma^2} (o_{t+dt}(s) - F(e_t(s)))^2 + \frac{1}{2\sigma^2} (o_t(s) - F(e_t(s)))^2 \right] \qquad (3\text{-}49)$$

Ou

$$F(e_t(s)) = \begin{cases} 0 & si\ e(s) = e(r) \\ \mu & si\ e(s) \neq e(r) \end{cases} \qquad (3\text{-}50)$$

σ^2 est une variance supposée connue, et μ est un paramètre réel positif représentant en quelque sorte la variation moyenne de luminance en un site où se produit un mouvement.

D'autres auteurs préfèrent utiliser un troisième terme U_{12} en fonction des observations binaires. Ce terme permet d'éviter la prise en compte des zones d'occlusion (zones apparaissant ou disparaissant entre deux instants) [PER92].

$$U_{12}(e_t, \bar{o}_t, \bar{o}_{t+dt}) = \sum_{s \in S} W(e_t, \bar{o}_t, \bar{o}_{t+dt}) \qquad (3\text{-}51)$$

Les valeurs de la fonction W étant données par la table suivante où β_1 est un paramètre réel positif.

$e_t(s)$	$\bar{o}_t(s)$	$\bar{o}_{t+dt}(s)$	$W(e_t(s),\bar{o}_t(s),\bar{o}_{t+dt}(s))$
0	**0**	**0**	$-\beta_1/2$
0	**0**	**1**	$-\beta_1/2$
1	**0**	**0**	β_1
1	**0**	**1**	β_1
0	**1**	**0**	$-\beta_1/2$
0	**1**	**1**	β_1
1	**1**	**0**	β_1
1	**1**	**1**	$-\beta_1/2$

Tableau III.2 : Les valeurs de la fonction W

Comme l'on a dit auparavant, le problème de maximisation de la probabilité conditionnelle $\Pr(E=e/O=o)$ revient à minimiser la fonction d'énergie $U(e,o)=U_{11}(e_t,o_t,o_{t+dt})+U_{12}(e_t,\bar{o}_t,\bar{o}_{t+dt})+U_2(e_t)$. Dans la pratique les paramètres sont fixés comme suite :

$$\beta_s=15 \;;\; \beta_f=50 \;;\; \beta_p=10 \;;\; \beta_1=100 \;;\; \mu=10$$

la figure suivante montre le principe, utilisé par la majorité des auteurs, de détection de mouvement par modélisation Markovienne :

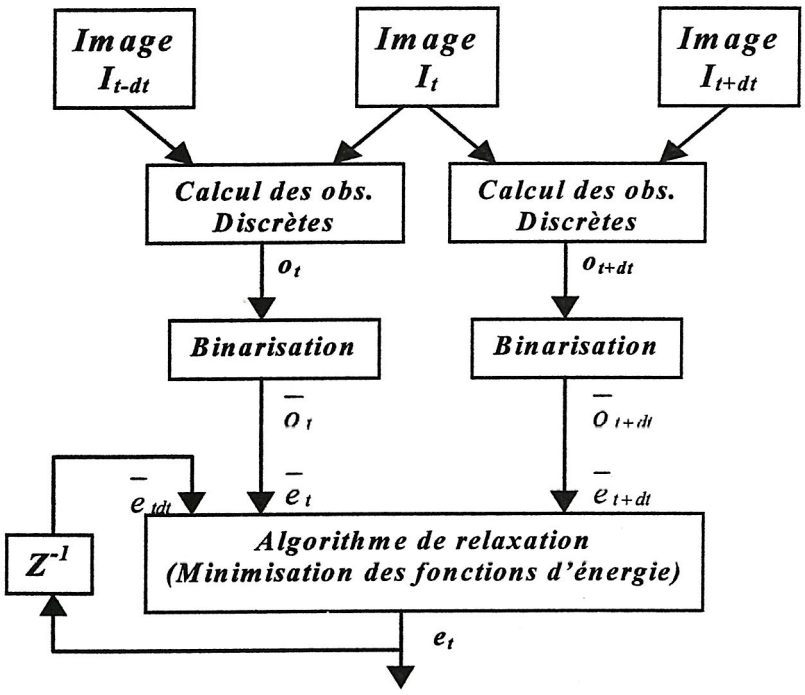


Fig. III.15 : Synoptique de l'algorithme de relaxation stochastique pour la détection de mouvement

5.4 Estimation du mouvement par modélisation Markovienne :

L'application des champs de Markov pour l'estimation du mouvement consiste à effectuer aux sites s=(x,y,t) de l'image des étiquettes $d_t = \{d_t(s); s \in S\}$ (vecteur déplacement). Les luminances observées sont les uniques observations utilisées pour l'estimation. Le vecteur $d_t(s)$ exprime que le point physique situé au site « s » à l'instant « t » se trouve au site « $s + d_t(s)$ » à l'instant « t+dt ». On limite de plus l'espace des configurations en imposant aux composantes de ces vecteurs d'appartenir à $\{-d_{\max}...d_{\max}; -d_{\max}...d_{\max}\}$, ce qui donne :

$$|A| = (2d_{\max} + 1)^2 \qquad (3\text{-}52)$$

La fonction d'énergie globale se décompose en deux termes :

$$\begin{aligned} U(d_t, I_t, I_{t+dt}) &= \sum_{s \in S} W(d_t, I_t, I_{t+dt}) + \sum_{c \in V} V_c(d_e) \\ &= U_1(d_t, I_t, I_{t+dt}) + U_2(d_t) \end{aligned} \qquad (3\text{-}53)$$

où :

- Le premier terme correspond à la DFD et exprime l'hypothèse de luminance constante :

$$\forall\, s \in S\,,\, W(d_t(s), I_t(s), I_{t+dt}(s)) = \left[I_t(s) - I_{t+dt}(s + d_t(s))\right]^2 \qquad (3\text{-}54)$$

- Le second terme dit de « régularisation » favorise les champs de déplacement « lisses » et pénalise donc les variations brutales entre deux sites voisins.

$$\forall\, c \in V;\quad V_c(d_t) = \begin{cases} \alpha \left\| d_t(s_1) - d_t(s_2) \right\|^2 & si \quad c = \{s_1; s_2\} \\ 0 \quad si\ non \end{cases} \qquad (3\text{-}55)$$

Où α est un paramètre réel positif permettant de régler le poids relatif des deux énergies U_1 et U_2.

La fonction DFD présente une forte non-linéarité, ce qui est justifié par la présence de nombreux minima locaux. L'utilisation des algorithmes de relaxation déterministes peut nous faire piéger dans un de ces minima locaux. De même, les algorithmes stochastiques demandent un temps de calcul très long, essentiellement avec le grand cardinal de l'ensemble A. Les auteurs proposent, pour résoudre ces problèmes, d'utiliser les approches multirésolution et multiéchelle pour avoir un bon compromis entre le temps de calcul et la précision d'estimation des véritables vecteurs vitesses.

6. Application de l'approche multirésolution pour la détection et l'estimation des paramètres de mouvement :

6.1 Introduction :

Dans le but de minimiser le temps et d'augmenter la précision des calculs, l'approche multirésolution est apparue, initialement utilisée pour la segmentation et ensuite appliquée au codage d'images et à l'estimation du mouvement. Pour cette dernière, l'utilisation d'un algorithme monorésolution exige la condition supplémentaire sur la fréquence d'échantillonnage spatiale, qui doit être adaptée à la taille et à la vitesse des objets en mouvement (il est difficile d'estimer un grand déplacement d'un petit objet). D'une autre manière, la bande des fréquences spatiales doit être adaptée aux amplitudes de déplacement des objets filmés. Cette adaptation peut être réalisée par l'utilisation des algorithmes multirésolution. En effet, à cause de sous-échantillonnage, la fréquence spatiale est divisée par deux dans chaque niveau de la pyramide.

L'approche multirésolution consiste à décomposer l'image initiale (la résolution la plus fine) en une pyramide passe-bas par filtrage puis sous-échantillonnage, on obtient alors une succession d'espaces de configuration dont la résolution est décroissante (du niveau le plus bas « la résolution la plus fine » au niveau le plus haut « la résolution la plus grossière »). A la limite, toute l'image se réduirait en un point au sommet de la pyramide. La figure suivante montre la représentation pyramidale d'une image sur 4 échelles.

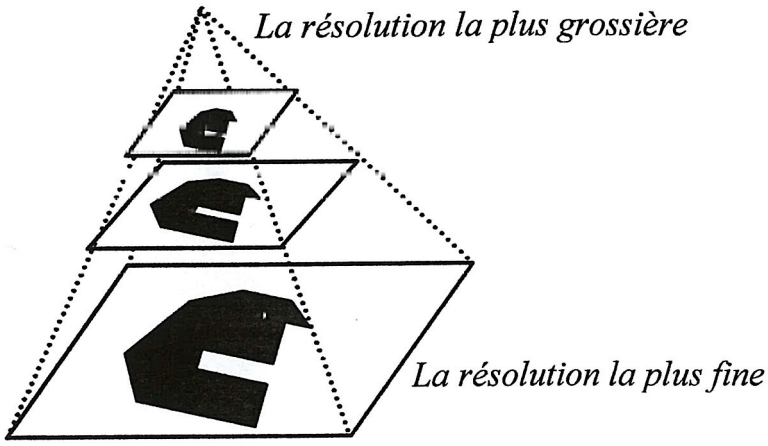


Fig. III.16 : Structure pyramidale d'une image

La structure multirésolution pyramidale est caractérisée par une réduction de dimension de résolution d'un facteur égale à 2 d'un niveau à l'autre :

- La résolution la plus fine (d'ordre L) : la taille de l'image est égale à $NxN = 2^L x2^L$
- La résolution d'ordre L-1 : la taille de l'image est égale à $\frac{N}{2}x\frac{N}{2} = 2^{L-1}x2^{L-1}$
- ....
- La résolution la plus grossière (d'ordre 0) : la taille de l'image est égale à $1x1 = 2^0x2^0$

6.2 Les différentes représentations pyramidales :

Les représentations pyramidales les plus utilisée sont la pyramide Gaussienne et la représentation quaternaire.

6.2.1 La pyramide Gaussienne :

On peut avoir la pyramide gaussienne comme un empilage des différentes versions de filtrage passe-bas et de reéchantillonnées du signal initial. Les filtres passe bas utilisés ont la même forme, avec une réponse impulsinnelle s'approchant de la forme d'une courbe de Gauss, d'où le nom de la pyramide.

La figure suivante montre la représentation mono-dimensionnelle du processus qui génère la pyramide gaussienne. L'image G_L est filtrée et ensuite sur-échantillonnée pour former l'image du niveau suivant G_{L-1} (G_{L-1} est la version réduite de G_L). D'une façon similaire, nous formons l'image G_{L-2} qui est la version réduite de G_{L-1}, et ainsi de suite jusqu'au résolution la plus fine.

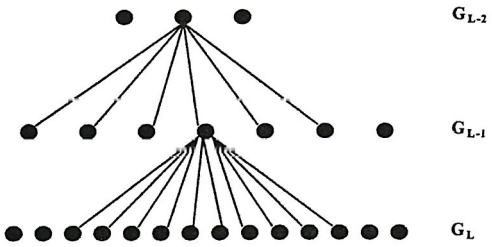


Fig. III.17 : Structure de la pyramidale Gaussienne

A partir de la pyramide de gausse, on peut construire une autre pyramide dite de Laplace (L_L, L_{L-1}, ..., L_0), beaucoup plus utilisée dans le domaine de codage des images. Chaque résolution de la pyramide de Laplace est calculée à partir de la différence entre deux résolutions de la pyramide gaussienne :

$$L_i(x,y) = G_i(x,y) - G'_{i-1}(x,y) \qquad (3\text{-}56)$$

Où G'_{i-1} est une extension de l'image G_{i-1} obtenue par interpolation.

6.2.2 Représentation en arbre quaternaire :

Cette méthode consiste en une décomposition récursive d'un plan divisé en régions carrées de tailles différentes (typiquement de taille 2x2).

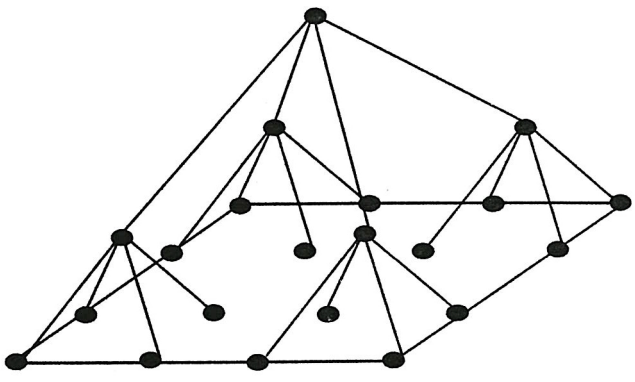


Fig. III.18 : Structure d'un arbre quaternaire (Quadarbre)

On peut utiliser la simple moyenne des quatre nœuds du niveau « l » pour avoir un nœud au niveau « l-1 ».

$$V(i,j,l-1)=\frac{1}{4}\sum_{m=0}^{1}\sum_{n=0}^{1}V(i+m,j+n,l) \qquad (3\text{-}57)$$

6.3 Estimation différentielle du mouvement sur une structure pyramidale :

F. Meyer dans son travail [MEY93] a utilisé un modèle affine avec six paramètres pour décrire le vecteur vitesse dans chaque résolution. Pour construire la pyramide, Meyer a utilisé une méthode classique de type filtrage descendant appliqué sur deux images de la séquence (à l'instant "t" et à l'instant "t+dt"). Les paramètres sont initialement estimés au niveau le plus grossière de la pyramide, où le nombre d'opérations est minimum. Ces estimations servent comme étant des valeurs initiales pour le niveau suivant de la pyramide.

6.4 Estimation markovienne du mouvement sur une structure pyramidale :

Rappelons qu'il existe deux méthodes de relaxation : la méthode stochastique à travers l'algorithme de Métropolis et la méthode déterministe (ICM). Les techniques classiques de minimisation d'une fonction d'énergie consiste alors soit à l'application de l'algorithme ICM auquel cas le système peut être bloqué dans un minimum local, soit à l'utilisation d'une technique stochastique de recuit simulé qui assure la convergence asymptotique vers le minimum global mais qui demande à priori un coût de calcul très élevé.

L'application des algorithmes de relaxation sur des champs d'observations pyramidales permet de réduire le temps de calcul et d'éviter le problème des maxima locaux. En effet, le volume des données traitées est d'autant plus faible que la résolution est grossière, et l'initialisation des différents niveaux (par les résultats du niveau précédent) peut ramener l'algorithme vers le minimum global même avec l'utilisation d'un algorithme déterministe.

Dans la pratique, les auteurs utilisent la stratégie multirésolution associée à un algorithme déterministe (ICM), où la seule information dont on dispose est le champ d'observations défini sur une grille « S » correspondant à l'échelle la plus fine. La stratégie multirésolution consiste à :

- Construire les observations binaires aux différentes échelles.
- Appliquer la méthode de relaxation (ICM) à l'échelle « l » pour calculer le champ d'étiquettes « e_l ».
- Projeter le champ « e_l » à l'échelle suivant « l+1 » pour avoir le champ d'étiquettes non optimale « $\hat{e}_{l+1}$ ».

- Le champ « $\hat{e}_{l+1}$ » sert comme valeur initiale pour l'algorithme de relaxation à l'échelle « l+1 ».
- Appliquer les même étapes jusqu'à l'arrivé à l'échelle la plus fine.

La figure suivante montre la propagation de l'information d'un niveau à l'autre :

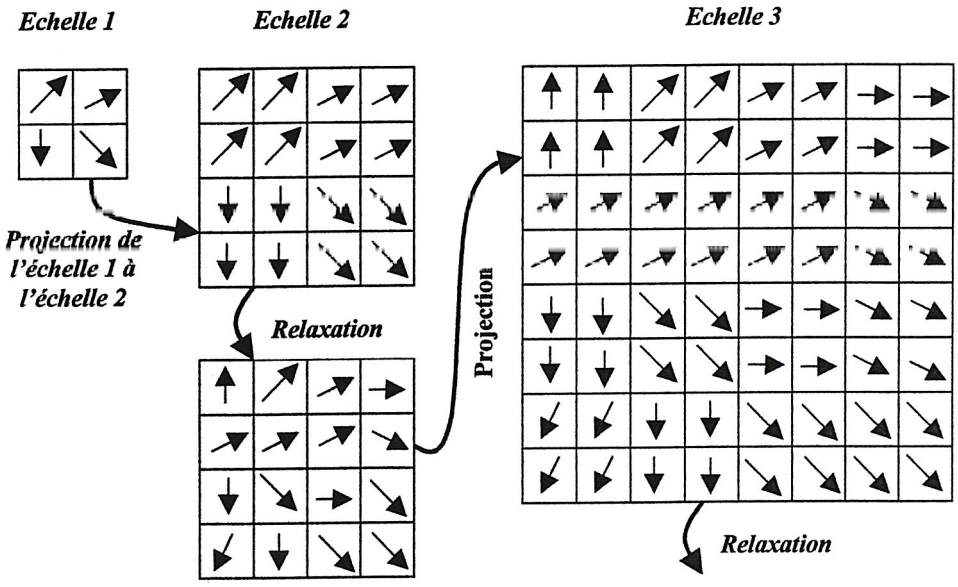


Fig. III.19 : Approche multirésolution pour l'estimation du mouvement

L'avantage du multirésolution est la prise en compte des pixels plus éloignés pour l'estimation des primitives. En effet, on considère l'influence des points de plus en plus distants au fur et à mesure que l'on progresse dans la pyramide, alors que les calculs restent locaux (la dimension du voisinage aux différents niveaux reste identique).

La majorité des auteurs affirment qu'il est suffisant d'utiliser l'algorithme ICM avec une structure pyramidale de trois niveaux. Dans l'article [PER92] les deux auteurs utilisent un filtre QMF séparable, caractériser par les coefficients suivants : [0,01995 ; -0,04271 ; -0,05224 ; 0,29271 ; 0,056458 ; 0,29271 ; -0,05224 ; -0,04271 ; 0,01995]

Dans le même travail, une comparaison a été effectuée entre les différentes méthodes appliquées sur plusieurs séquences d'images :

- Estimation monorésolution par relaxation déterministe (ICM).
- Estimation monorésolution par relaxation stochastique (l'échantillonneur de Gibbs).
- Estimation multirésolution classique (avec une relaxation ICM).
- Estimation mutiéchelle.

7. Résultats et discussions :

Les résultats d'estimation du mouvement, par la méthode de mise en correspondance, sont présentés sur les figures suivantes :

- *Fig. III.21* : Une séquence synthétique.
- *Fig. III.22* : La séquence d'une artiste dans le cirque (mouvement de rotation).
- *Fig. III.23* : La séquence d'une artiste dans le cirque (mouvement de rotation et de translation).

Sur la figure (***Fig. III.20***), nous avons présenté deux organigrammes de la méthode de mise en correspondance. Le premier organigramme (***Fig. III.20-a***) représente l'algorithme standard, cependant il présente l'inconvénient d'être non applicable sur les zones homogènes (Zone1) et pour les zones où l'objet glisse sur lui-même (Zone2). Pour cela nous avons présenté, sur la figure (***Fig. III.20-b***) un autre organigramme basé sur la détection de mouvement. En effet il est inutile de calculer les vecteurs de déplacement là où le mouvement n'existe pas, ce qui a donné l'idée de précéder l'estimation par l'opération de détection.

Nous remarquons que l'application de l'algorithme 2 sur des images synthétiques donne de bons résultats même avec la présence du bruit (***Fig. III.21***).

Il est à remarquer que les images de la figure (***Fig. III.22***) et la figure (***Fig. III.23***) sont le résultat d'un cycle de codage-décodage par la méthode JPEG, ce qui explique les dégradations figurées sur les images initiales ainsi que les résultats d'estimation. Néanmoins, l'application de l'organigramme 1 sur ce type de séquences donne des résultats acceptables, surtout dans le cas où ($N=M=4$).

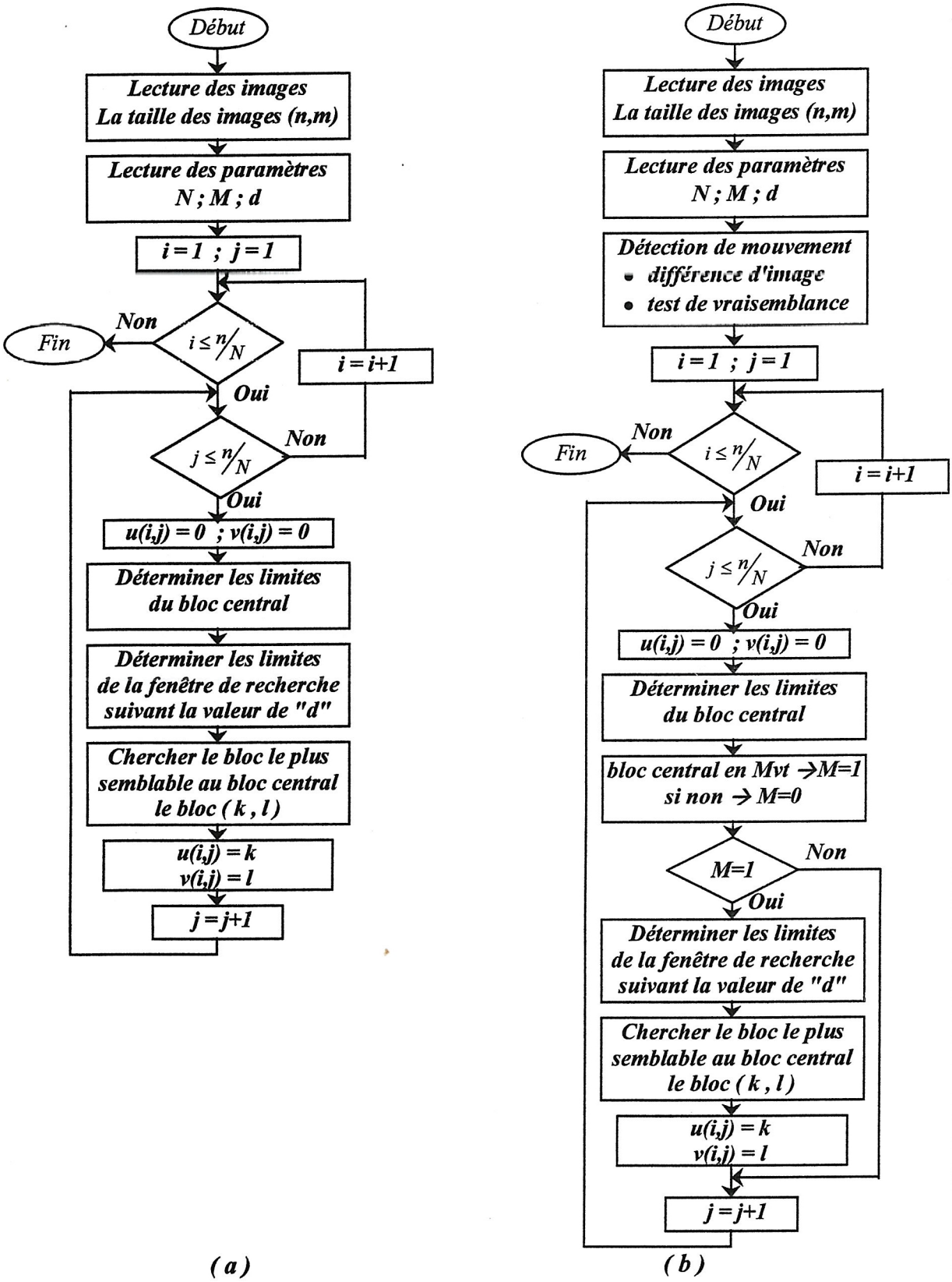


Fig. III.20 : Organigrammes pour l'estimation du mouvement

Par la méthode de mise en correspondance

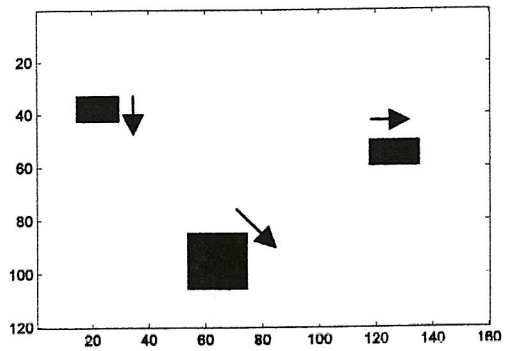


Image 1

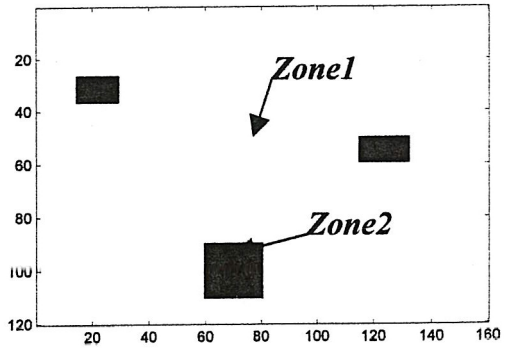
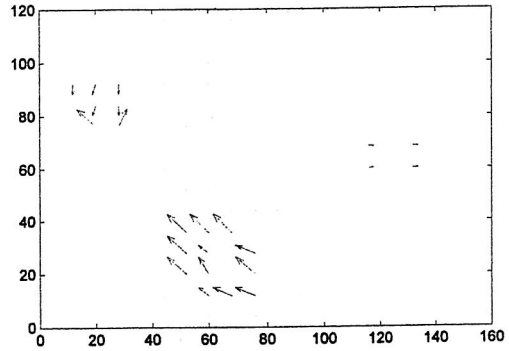
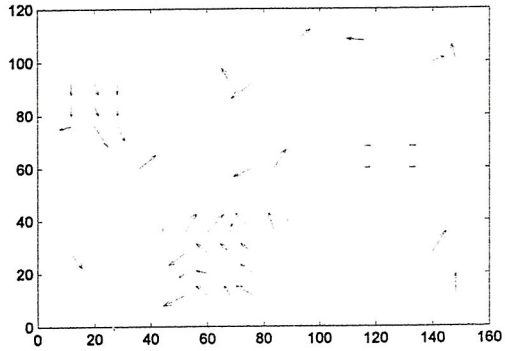


Image 2



Estimation sans bruit



Estimation avec bruit

Fig. III.21 **:** ***Résultats de l'estimation du mouvement (BM) Appliquée sur une séquence synthétique***

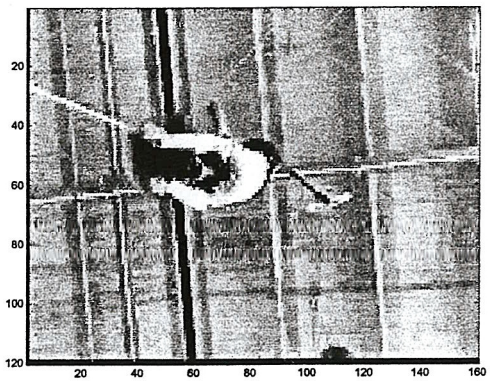


Image 1

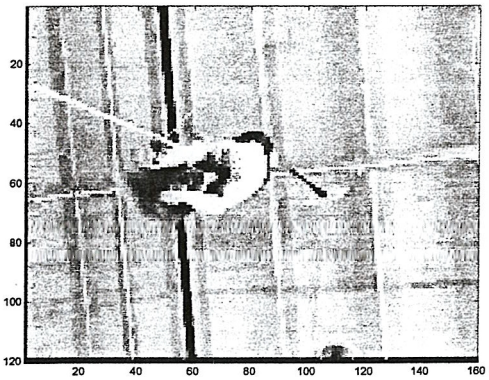
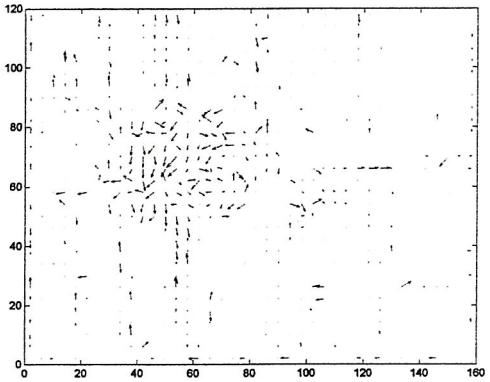
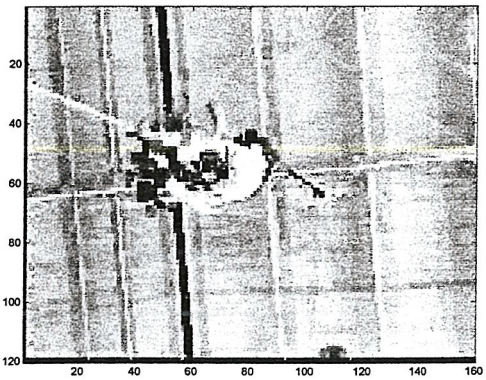
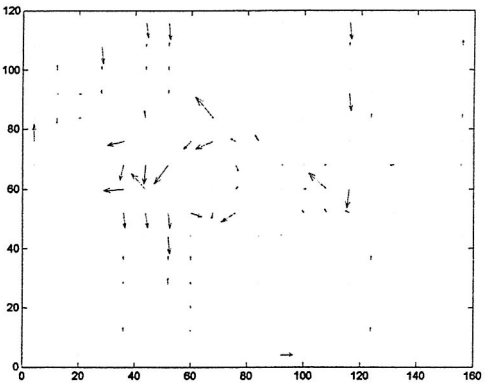


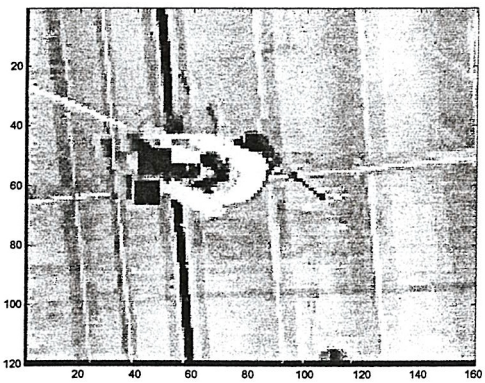
Image 2





n = 120 ; m=160 ; N=M=4 ; d=11





n = 120 ; m=160 ; N=M=8 ; d=11

Fig. III.22 **:** ***Résultats de l'estimation du mouvement (BM)***
Appliquée sur une séquence « Pirouette »

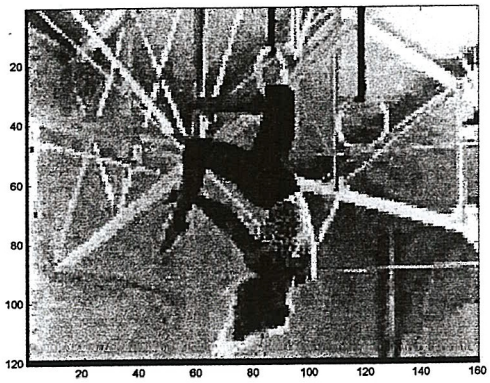
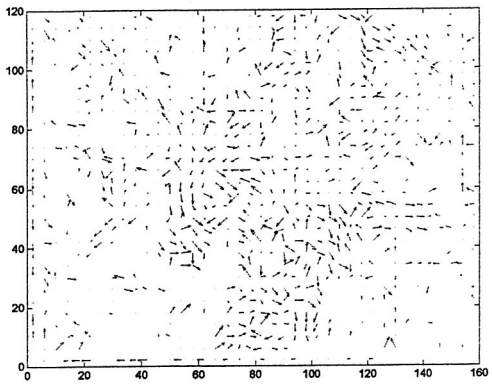
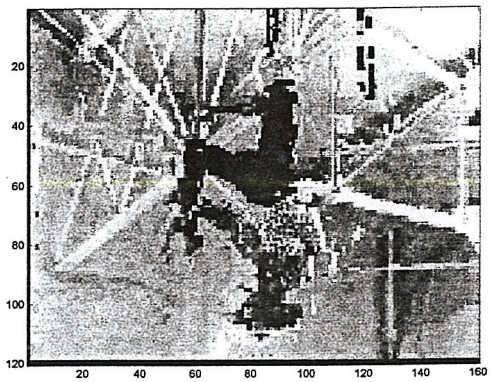


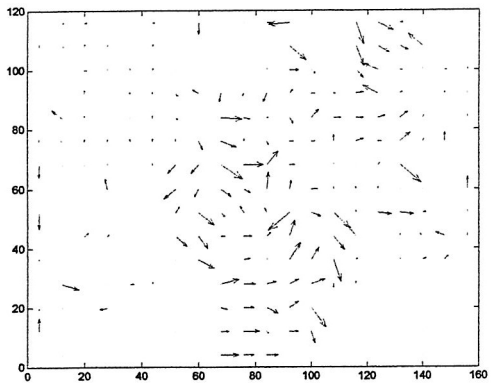
Image 1

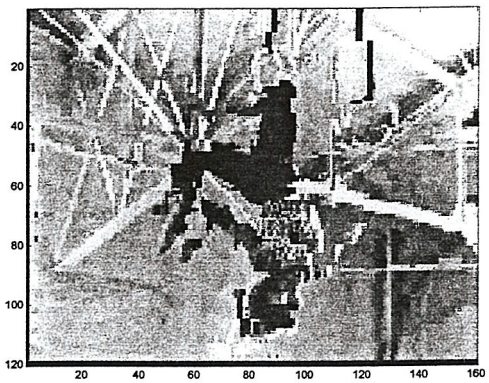
Image 2





n = 120 ; m=160 ; N=M=4 ; d=11





n = 120 ; m=160 ; N=M=8 ; d=11

Fig. III.22 **:** ***Résultats de l'estimation du mouvement (BM) Appliquée sur une séquence «Tourbill »***

8. Conclusion :

Nous avons présenté dans ce chapitre les différentes méthodes de détection, de localisation et d'estimation du mouvement dans le but de les utiliser dans un schéma hybride de compression spatio-temporelle.

D'après l'étude théorique et les résultats obtenus, nous pouvons conclure que, moyennant certaines hypothèses (éclairage uniforme, mouvement de translation ...etc.) et un choix judicieux des paramètres d'exécution des algorithmes (N, M, d ...etc.), la méthode de mise en correspondance, combinée avec les méthodes de compression d'images fixes, donne les meilleurs résultats pour la compression des séquences d'images.

Néanmoins, il est possible de l'utiliser en parallèle avec les autres méthodes dans le but d'augmenter l'efficacité d'estimation et pour éviter les fausses estimations (approches mixtes [PEL93]).

Chapitre 4

La norme MPEG

1. Introduction :

Vers les années 80, plusieurs compagnies ont essayé de mettre un standard des techniques de compression vidéo valable pour tous les cas. En effet, la vidéo numérique concerne plusieurs secteurs de l'industrie (téléconférence, Visiophonie, l'électronique grand-public, ...etc.). Pour satisfaire tous ces différents secteurs, le standard idéal doit donc être indépendant d'une application donnée, mais sans ignorer les besoins de celle-ci.

2. Le format M-JPEG :

Le format M-JPEG (***Motion JPEG***) est un format de compression d'images animées. Ce format est en fait un simple stockage successif d'images compressées au format JPEG, il permet d'avoir des débits situés entre 7 et 28 Mbits/s, ce qui reste très élevé pour la majorité des applications. Ce format reste idéal pour les applications de montage vidéo, et permet surtout d'obtenir des cartes d'acquisition temps réel moins chères que les cartes MPEG. L'avantage de cette méthode est qu'elle ne pose pas des conditions sur le type et la valeur du mouvement, elle est applicable pour tous les cas (mouvement de translation, rotation, zoom, ... etc.).

3. La norme MPEG :

En 1988, les deux comités ISO et CCITT se sont réunis pour former le MPEG (***Moving Picture Expert Groupe***). Ces deux comités sont basés essentiellement sur les travaux des trois organisations internationales :

- ***JPEG : (Joint Photographique Expert Groupe).***
- ***CMTT/2 : A travers ses normes H21 et H22.***
- ***CCITT : à travers sa norme H261.***

Les experts travaillant dans le groupe MPEG ont fixé un débit de 1,5 Mbits/s (pour la vidéo et l'audio). Ce débit est parfaitement adapté à la majorité des unités de stockage et de canaux de télécommunication, on trouve entre autres :

- Les CD-ROM.
- Les cassettes numériques DAT.
- Les réseaux d'ordinateurs(le réseau Internet comme exemple).
- Les réseaux de télécommunications.
- ...etc.

L'idée de la compression MPEG est que dans une même scène, d'une image à la suivante, les changements ne sont pas très importants. Il est alors inutile de considérer chaque image comme une nouvelle image indépendante de la précédente. Il suffit de se baser sur une première image et de décrire les suivantes par rapport à celle-ci en indiquant uniquement ce qui a changé. Quand la scène filmée contient des objets en mouvement, il suffit de les décrire. Or, ceci implique d'abords la reconnaissance des objets mobiles, ensuite la mesure de leur vitesse et enfin la représentation de cette information.

La norme MPEG utilise, pour l'estimation du mouvement, la méthode par similarité des blocs (Blocks Maching). Dans cette méthode on suppose qu'une forme carrée dans une première image et à une position donnée, s'est déplacée dans l'image suivante à une autre position. On cherche cette nouvelle position pour mesurer le mouvement. La figure suivante montre le principe de cette méthode :

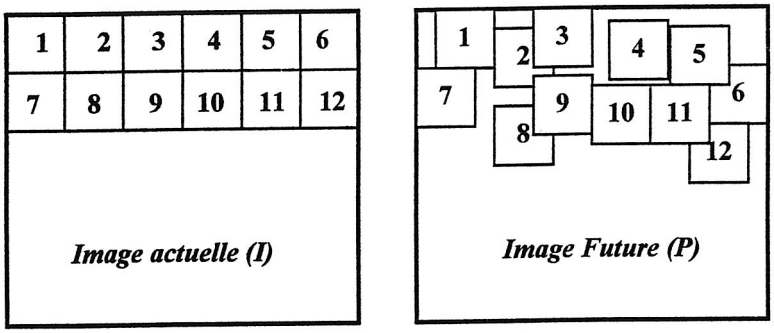


Fig. IV.1 : Estimation du mouvement par similarité de blocs (BM)

Il est aussi possible de prédire les déplacements pour d'autres images, pour l'image 3 par exemple ou bien pour des images intermédiaires entre l'image 1 et l'image 2, Ce qui augmente considérablement le taux de compression.

Les activités du groupe MPEG ne sont pas limitées à la compression vidéo, l'audio joue un rôle très important dans la diffusion des vidéos. La synchronisation doit être aussi intégrée

dans la norme. MPEG s'est donc divisé en trois groupes distincts, chacun s'occupant de la standardisation de l'une des composantes du signal vidéo [DER96] :

- ***MPEG-Vidéo*** s'occupe de la compression des signaux purement vidéo pour un débit de 1,5 Mbits/s environ.
- ***MPEG-Audio*** s'occupe de la compression des signaux purement audio pour des débits de 64, 128 et 192 Kbits/s par canal.
- ***MPEG-System*** s'occupe de la synchronisation et du multiplexage des différents signaux audio et vidéo.

3.1 MPEG-1 :

Pour la compression vidéo, MPEG-1 utilise la prédiction avec compensation de mouvement pour la réduction de la redondance temporelle et la transformée en cosinus discrets (DCT) pour la réduction de la redondance spatiale.

3.1.1 La compression temporelle :

Dans le cadre du codage d'images, sauter des images au codeur et les interpoler au décodeur constitue une technique de compression très intéressante. Cette technique permet de réduire, d'une manière significative, le débit d'information nécessaire à la transmission du signal vidéo. Pour cela, les experts du groupe MPEG-1 divise les images de la séquence en trois types :

- L'image de repère I (Intra-image).
- Les images prédites P.
- Les images intermédiaires B.

La séquence originale est sous-échantillonnée dans la direction temporelle, en laissant tomber, par exemple, une image sur deux ou bien deux images sur trois (les images de type B).

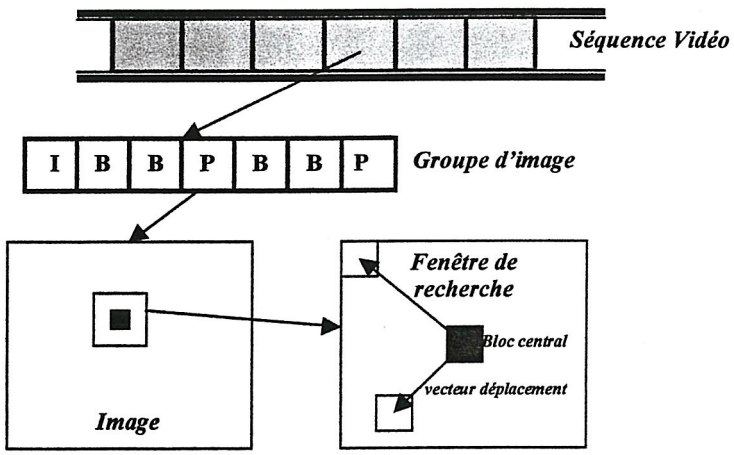


Fig. IV.2 : Le principe de décomposition d'une séquence d'images Et d'estimation du mouvement par la méthode BM.

La compression du mouvement pour les images prédites est réalisée par la méthode de similarité des blocs (Block Matching), avec des blocs carrés (16x16). L'estimation du mouvement pour la troisième image (deuxième image de type P) est effectuée par une extrapolation. En effet, si le mouvement du bloc est linéaire, la prédiction sera juste, si ce n'est pas le cas, la différence entre l'image, dont le mouvement a été prédit, et l'image actuelle sera en moyenne faible. On peut alors décrire la séquence d'images avec les données de la prédiction et de l'erreur de cette dernière. On parle dans ce cas de différence de trames déplacées ou encore de compensation de mouvement. Donc à chaque bloc (16x16) de la première image prédite on associe deux valeurs de déplacement (Horizontale et Verticale). Par contre, pour les autres images prédites on associe à chaque bloc les deux termes correctifs.

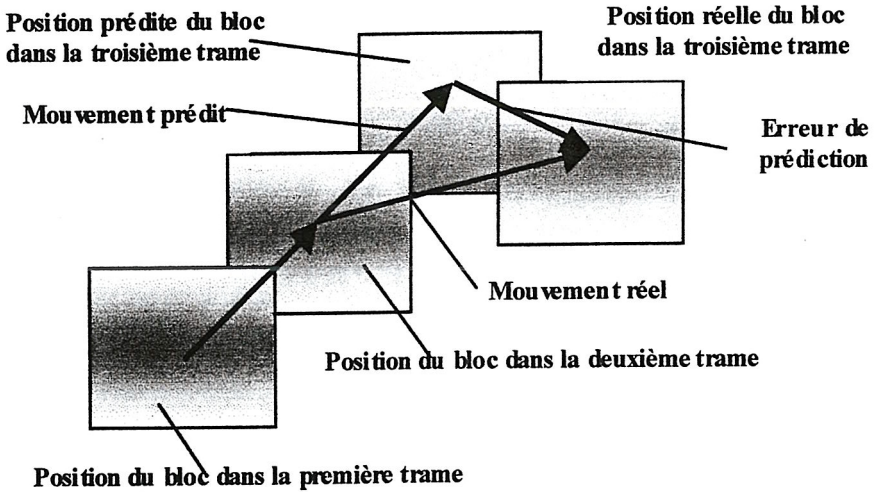
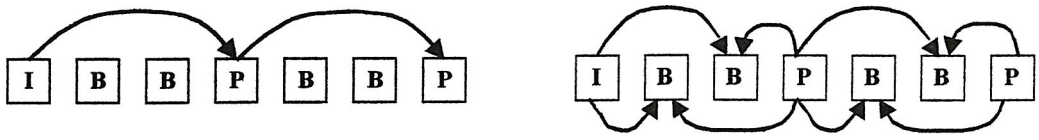


Fig. IV.3 : Principe de compensation du mouvement

A l'étape de décodage, les images prédites sont reconstruites à partir de l'image de repère I et des valeurs de déplacement associées à chaque bloc. Les images intermédiaires B sont reconstruit par interpolation (insertion d'images supplémentaires entre les images I et B). Différentes techniques d'interpolation ont été proposées, on trouve entre autres : l'interpolation par recopie de l'image la plus proche, l'interpolation linéaire et l'interpolation par compensation de mouvement qui donne les meilleurs résultats.



a : Reconstruction de s images (P) à partir des vecteurs de déplacement

b : Interpolation des images manquantes (B)

***Fig. IV.4* : Reconstruction et interpolation des images utilisées par le groupe MPEG**

Le nombre M d'images B entre deux images P et I, et le nombre N d'images P entre deux images I, constituent les paramètres d'exécution de l'algorithme, pour faire varier le taux de compression en fonction du rapport signal à bruit SNR. Typiquement M est égal à 1 ou 2, et N est de l'ordre de 15.

3.1.2 La compression spatiale :

La compression spatiale utilisée par MPEG-1, appliquée sur les images de type I, est pratiquement celle utilisé par JPEG, avec l'utilisation du DCT sur des blocs de taille 8x8.

3.1.3 La compression audio :

La technique de compression du son utilisée dans MPEG est la technique ADPCM (Adaptative Différential Pulse Code Modulation). On note aussi, que MPEG-1 utilise des techniques (MPEG Système) pour la synchronisation et le multiplexage des différents signaux audio et vidéo.

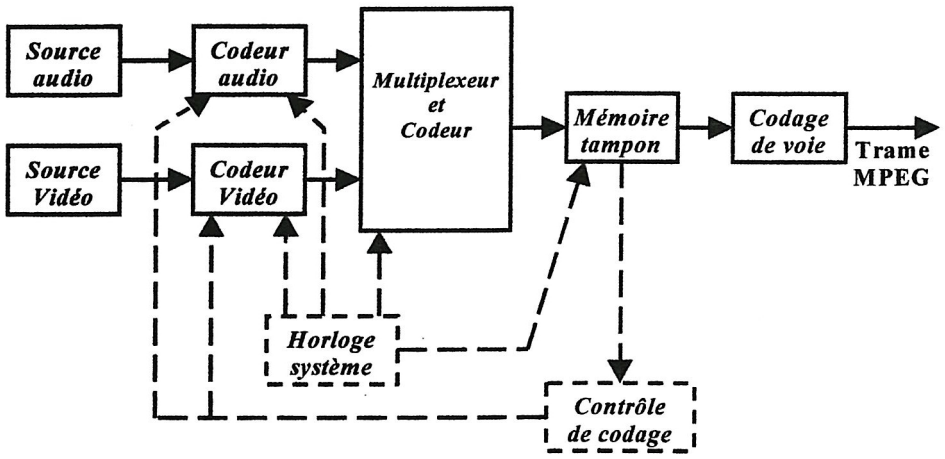


***Fig. IV.5* : Production d'une trame au format MPEG**

3.2 MPEG-X :

En 1991 (à 1994) Le comité MPEG a développé un autre standard de compression d'images dynamiques MPEG-2, plus flexible que la norme MPEG-1 pour pouvoir être étendu à de nouveaux standards et pour avoir des qualités d'image supérieures à celles des standards NTSC, PAL ou SECAM. MPEG a développé aussi le standard MPEG2 Système pour spécifier le format de codage pour le multiplexage de l'audio, vidéo, et autres données dans une forme pratique pour la transmission et le stockage. La majorité des auteurs affirment que l'avenir appartient au MPEG-2, et que les autres standards (MPEG-3 et MPEG-7) ne sont que des dérivés de ce dernier.

La norme MPEG-3 visait presque exclusivement les applications TVHD (***la Télévision Haute Définition***) avec des résolutions allant jusqu'à 1920x1080x30 Hz, et pour des débits allants de 20 à 40 Mbits/Seconde. Avec la remarque que la norme MPEG-2 offrait suffisamment d'atouts techniques pour traiter les données TVHD.

Par contre, les objectifs de la norme MPEG-4 sont de définir une norme adaptée aux applications audiovisuelles de très bas débits comme, par exemple, la vidéo téléphonie. Les résolutions vont jusqu'à 176x144x10 Hz et pour des débits entre 4800 et 64000 bits par seconde.

Conclusion générale

Nous avons présenté, voir développé, dans ce mémoire plusieurs algorithmes concernant la compression des images et l'estimation du mouvement où de nombreuses expériences sur des images fixes, des séquences synthétiques et des séquences réelles sont exposées.

Dans une première partie de notre travail, nous avons passé en revue sur les différentes méthodes de compression des images fixes. Ce qui nous a permet d'établir une étude comparative. A l'issue de l'étude bibliographique et aux résultats obtenus, nous pouvons conclure que les méthodes de codage par transformée restent les mieux adaptés pour la majorité d'applications, qui nécessitent la compression, à cause du bon compromis entre le temps de calcul et le rapport de compression. Elles peuvent également donner de bons résultats en combinaison avec les autres méthodes de compression (le codage en ondelette, la QV, la compression fractale, ...etc.).

Ensuite, notre travail a été orienté vers l'estimation du mouvement dans le but d'une compression spatio-temporelle. L'idée d'estimation consiste à effectuer à chaque pixel (méthodes différentielles) ou bien à chaque groupe de pixels (la mise en correspondance) deux valeurs de déplacement (horizontale et verticale).

La méthode de mise en correspondance est la plus utilisée et la plus adaptée, au moins pour l'instant, pour la compression des séquences d'images. Elle est combinée, sur des schémas hybrides, avec les méthodes de compression par transformée des images fixes pour augmenter le taux de compression spatio-temporelle. En effet, la majorité des standards existant (MPEG, H261, ...etc.) consistent à compresser la première image par l'une des méthodes de compression des images fixes, et de décrire les suivantes (entre 1 et 15) par les vecteurs de déplacement calculés par l'estimation du mouvement.

Néanmoins, la méthode de mise en correspondance peut présenter quelques inconvénients pour quelques types d'images. Nous pouvons citer à titre d'exemple :

- La méthode est adaptée pour des mouvements de translation et aux petits déplacements.
- Elle exige la condition d'homogénéité d'éclairage pour toute la séquence.
- Elle peut causer des défauts dans les images décompressées (effet des blocs, les fausses estimations, ...etc.).

Par contre, les auteurs affirment que les autres méthodes (différentielles, stochastiques et fréquentielles) donnent les meilleurs résultats, pour l'estimation des véritables vecteurs de déplacement, avec l'inconvénient du temps de calcul qui est très long par rapport à la méthode de mise en correspondance.

D'un autre coté, toutes les méthodes partagent le problème des fausses estimations dues aux zones homogènes et aux zones d'accultations. La méthode de mise en correspondance souffre, en particulier, de ces deux problèmes. Nous proposons, pour résoudre ce problème, de précéder l'estimation par une détection de mouvement, et de n'estimer le mouvement que sur les zones où un changement temporel est détecté. De cette façon on limite les fausses estimations et on diminue considérablement le temps de calcul.

Nous avons remarqué, durant notre étude, que l'utilisation des filtres QMFB permet de réduire la résolution des images à compresser, ce qui rend l'estimation du mouvement plus rapide (estimation multirésolution du mouvement). Dans ce sens, les principales perspectives de notre travail seront les suivantes :

- ✓ Utilisation des filtres QMFB pour la compression spatiale et les techniques d'estimation du mouvement dans un environnement pyramidal pour la compression temporelle.

- ✓ Combinaison des différentes méthodes de compression (par DCT, ondelette, fractale, ...etc.) et d'estimation du mouvement pour formé un standard de compression d'images dynamiques.

- ✓ Application des méthodes d'estimation du mouvement aux différentes régions de l'image et aux différents types d'images.

BIBLIOGRAPHIE

[AGG88] ***J. K. Aggarwal*** et ***N. Nandhakumar***, ''On the coputation of otion from séquence of images – a review'', IEEE, PP917-935, 1988.

[ANT92] ***M. Antonini, M. Barlaud, P. Mathien*** et ***I. Danbechies***, ''Image coding using wavelet transform'', IEEE, Image proc. Trans., Vol 01, N° 02, Avril 1988.

[BAR91] ***P. Baraldi, M. Tislarelli, G. Sandini*** et ***F. Gandolfo***, ''Spatio-temporel filtring for visual motion estimation from real images'', SPIE, Vol. 1613 Mobile robots VI, 1991.

[BAR92] ***J. L. Barron, D. J. Fleet, S. S. Beauchemin*** et ***T. A. Burkitt***, ''Performance of optical flow techniques'', IEEE 1992.

[BAR94] ***M. Barlaud***, ''Wavelets in image communication'', Advances in image communication 5, ELSEVIER 1994.

[BAR95] ***J. L. Barron*** et ***R. Eagleson***, ''Motion and structure from time varying optical flow'', Vision Interface, Mai 1995.

[BIM95] ***A. D. Bimbo, P. Nesi*** et ***J. Sang***, ''Analysis of optical flow constraints'', IEEE, Vol.04, N°04, PP460-469, Avril 1995.

[BOU87-a] ***P. Bouthemy***, ''Estimation et structuration d'indices spatio-temporels pour l'analyse du mouvement dans une séquence d'images'', Traitement du signal, Vol.04, N°03, 1987.

[BOU87-b] ***P. Bouthemy*** et ***J. S. Rivero***, ''Segmentation en régions selon des critères de mouvement dans une séquence d'images'', Pro. AFCET, PP105-114, 1987.

[BOU88] ***P. Bouthemy***, ''Modèles et méthodes pour l'analyse du mouvement dans une séquence d'image'', Avril 1988.

[BOU95] ***T. Bouden***, ''Détection et localisation du mouvement dans une séquence d'image'', Mémoire de Magister, Université de Annaba, Institut d'électronique, 1995.

[BOU98-99] ***T. Bouden*** et ***N. Doghmane***, ''Approches déterministes et stochastiques pour la détection des objets mobiles en imagerie'', Proc. SNAPSEA98, PP162-168, Proc. SNAS99 PP156-161.

[BOU99] ***A. Boukelif***, ''Adaptative fractal image copression based on fisher classification and two level quadtree partioning'', Proc. SNAS99, Université de ANNABA, PP25-46, Novembre 1999.

[CAP96] ***A. Capilier***, ***C. Dumotier***, ***F. Luthon*** et ***P. Coulon***, ''Algorithme de détection de mouvement par modélisation markovienne : mise en ouevre sur DSP'', Traitement du signal, Vol. 13, N° 02, PP177-191, 1996.

[CAP97] ***A. Capilier***, ***F. Luthon***, ''Approche spatio-temporelle pour l'analyse de séquence d'image. Application en détection de mouvement'', Traitement du signal, Vol. 14, N° 02, PP195-208, 1997.

[CHE98] ***C. K. Cheung***, ''Fast motion techniques for vidéo compression'', Master of Philosophy, Université de HONG KONG, juillet 1998.

[DER96] ***M. Derridj***, ''La compression de données'', Rapport de cours EPITF, 1996.

[FRA95] ***E. François*** et ***B. Chupeau***, ''Segmentation d'un champ de mouvement dense. Application au codage très bas débit'', 1995.

[FRA98] ***Fractales98***, ''Proceding FRACTALES98'', Séminaire national sur les fractales dans la compression des images, Alger 1998.

[GOU94] ***J. P. Gourret***, ''Modélisation d'image fixes et animées'', Masson 1994.

[HEN93] ***C. Hennebert***, ''Etude de plusieurs approches multirésolutions associées à un modèle markovien pour la détection du mouvement dans des séquences d'images'', DEA (signal, image, parole) ENSERG, Juin 1993.

[HET91] ***F. Heitz*** et ***E. Memin***, ''Parallélisation des algorithmes d'analyse d'images par champs Markoviens dans un contexte pyramidal'', troisième colloque gretsi-Juan-les-pins, PP845-846, Sept. 1991.

[HET94] ***F. Heitz***, ***P. Perez*** et ***P. Bouthemy***, ''Multiscale minimization of global energy functions in some visual recovery problems'', CVGIP, Vol.59, N°01, PP125-134, Jan. 1994.

[HOF93] ***T. Huffman***, ***D. Muller*** et ***C. Vogt***, ''Technique de compression des signaux vidéo pour la communication multimédia'', Revue des télécommunication 1993.

[HOR81] ***B. K. P. Horn*** et ***B. G. Schunk***, ''Détermining optical flow'', Proc. Artificial Intelligence (AILMIT), PP185-202, USA 1981.

[KUN93] ***M. Kunt***, ***G. Granlund*** et ***M. Kocher***, ''Traitement numérique des images'', Vol. 02, 1993.

[LAF98] ***J. M. Laferté, F. Heitz*** et ***P. Pérez***, ''Modèles markoviens d'images et algorithmes d'estimation non linéaire sur le quadarbre'', Traitement du signal, Vol. 15, N° 03, PP213-230, 1997.

[MAU95] ***M Maurizot, P. Bouthemy, B. Delyon, A. Louditski*** et ***T. M. Odebez***, ''Caractérisation du mouvement déformable dans une séquence d'images par la localisation de points singuliers'', Quinzième colloque Gretsi-Juan-Les-Pins, PP805-808, sept. 1995.

[MEM95] ***E. Mémin*** et ***P. Pérez***, ''Champs de markov multirésolution et algorithme multigrille pour l'estimation de mouvement'', Quinzième colloque Gretsi-Juan-Les-Pins, PP813-816, sept. 1995.

[MEY93] ***F. Meyer***, ''Suivi de région et analyse des trajectoires dans une séquence d'image'', test de l'université de RESSES-1, Juin 1993.

[NEL94] ***M. Nelson***, ''La compression de données : Texte – Image – Son'', 1994.

[ODO95] ***J. M. Odobez*** et ***P. Bouthemy***, ''Estimaion robuste multiéchelle de modèles paramétrés de mouvement sur des scènes complexes'', Pro. Traitement du signal, Vol. 12, N°02, 1995.

[ORK91] ***M. ORKISZ***, ''Localisation d'objets mobiles dans des scènes naturelles filmées par une caméra fixe'', Traitement du signal, Vol. 09, N° 04, PP 325-346, 1991.

[ORK92a] ***M. ORKISZ***, ''Détection et estimation du mouvement dans des séquences d'images'' Rap. De TIRF ; INPG, Juin 1992.

[ORK92b] ***M. Orkisz***, ''Localisation d'objets mobiles dans des scènes filmées par une caméra fixe'', Traitement du signal, Vol.09, N°04, Juillet 1992.

[PEL93] ***D. Pellerin*** et ***M. Orkisz***, ''Estimation du mouvement par approche mixte : Droite de contrainte / mise en correspondance'', Quatorzième colloque gretsi-Juan-les-pins, PP855-858, Sept. 1993.

[PEL95] ***D. Pellerin, A. Spinai*** et ***A. Guerin-Dugue***, ''Estimation du mouvement par filtrages spatio-temporels récursifs'', Quinzième colloque Gretsi-Juan-Les-Pins, PP841-844, Sept. 1995.

[PER92-a] ***P. Perez*** et ***F. Heitz***, ''Une approche multiéchelle à l'analyse d'image par champs Markoviens'', Traitement du signal Vol.09, N°6, PP459-492, 1992.

[PER92-b] ***P. Pèrez, F. Heitz***, ''Multiscale markov ramdom fields and constrained relaxation in low leval image analysis'', IEEE, PP61-64, 1992.

[SAA94] ***M. Saaidia***, " Compression des signaux mono et bidimentionnels, application à la parole aux images'', mémoire de magister, Université de Annaba, Institut d'électronique, 1994.

[SAM98] ***D. Samai***, ''Contribution à l'étude et au développement d'algorithme en compression d'image numérique fixe'', mémoire de magister, Université de Annaba, Institut d'électronique, 1998.

[SCH85] ***B. Schunck***, ''Image flow : Fondamentals and future research'', IEEE, PP560-571, 1985.

[SPI94] ***A. Spinei***, ''Etude de filtres orientés de type Gabor. Apllication à l'estimation du mouvement dans une séquence d'images'', Rapport de stage (INPG), Fev. 1994.

[STE80] ***P. Stelmaszyk***, ***P. Bonnet*** et ***J. C. Postaire***, ''Analyse de scènes dynamiques par recherche de contours en mouvement'', Cinquième congrès RFIA Grenoble, PP1181-1189, Nov. 1985.

[TZI88] ***G. Tziritas***, ''Estimation récursive-itérative du mouvement apparent dans une séquence d'images'', Traitement du signal Vol. 05, N°4, PP305-310, 1988.

[VER84] ***M. Vetterli***, ''Multidimentionnel sub-band coding : some théory and algorithms'', Signal processing, Vol. 06, PP92-112, Avril 84.

[VER90] ***A. Verri***, ***F. Girosi*** et ***V. Torre***, ''Differential techniques for optical flow'' Pro. J. Opt. Sco. A. A. , Vol. 07, n°05, PP912-922, Mai 1990.

[VIE89] ***C. Vieren***, ''Détection du contour extérieur d'objets en mouvementsur un fond non uniforme'', deuxième colloque Gretsi-Juan-Les-Pins, PP621-624, Juin 1989.

[WAL91] ***G. K. Wallace***, " The JPEG picture compression standart'', IEEE Dec. 1991.

[ZER91] ***J. Zeruba*** et ***F. Ployette***, ''Détection de contours et lissage d'image par des algorithmes déterministes de relaxation : Mise en œuvre sur la CM'', treizième colloque Gretsi-Juan-Les-Pins, PP11181-1184, sept. 1991.