



MEMOIRE

Présenté à

L'École Nationale d'Ingénieurs de Sfax

en vue de l'obtention du

MASTERE

Dans la discipline informatique

***Nouvelles Technologies des Systèmes Informatiques Dédiés
(option SoC)***

Par

Ismail KTATA

(Ingénieur – Génie électrique)

**Techniques de traitement d'image dédiées aux
flux des diapositives lors d'une visioconférence**

Soutenu le 24 Juillet 2008, devant le jury composé de :

M. Chokri BEN AMAR

Président

M. Moncef BOUSSELM

Membre

M. Mohamed ABID

Membre

Remerciements

Je voudrais exprimer ma gratitude et mes remerciements les plus sincères à l'égard de toutes les personnes qui m'ont aidé aussi bien par leur soutien moral que par leur savoir et savoir-faire pour mener à bien ce projet :

Je voudrais remercier Mr. Mohamed ABID, Professeur à l'ENIS, pour m'avoir accueilli au sein de son équipe de recherche CES, pour m'avoir fait l'honneur d'encadrer mes travaux, pour ses directives fructueuses et astucieuses, pour la confiance particulière qu'il m'a accordé, pour son aide, sa patience et son dévouement durant toute la période de mon stage.

Mes remerciements s'adressent pareillement à Mr Chokri Ben AMAR, Maître de conférence à l'ENIS, pour l'intérêt qu'il a porté à ce travail en acceptant de me faire l'honneur de présider le jury de ma soutenance.

Je tiens également de remercier Mr Moncef BOUSSELMI, Maître Assistant à la Faculté des Sciences de Sfax, d'avoir bien voulu me faire l'honneur de participer à ce jury.

Mes remerciements distingués pour Mr Tarek OUNI et Mr Walid Ayadi, chercheurs dans notre unité de recherche CES, qui ont été les meilleurs compagnons durant la période du projet et qui m'ont donné beaucoup de soutien par leurs idées inspirées et par leurs enthousiasmes communicatifs... je vous souhaite une grande carrière car vous le méritez.

Merci à tous les membres du laboratoire CES de l'ENIS pour leur ambiance agréable, leur soutien et l'accueil chaleureux dont j'ai profité.

Avant-propos

L'étude menée dans cette mastère s'inscrit au début dans le contexte industriel de traitement vidéo d'une présentation lors d'une conférence. Il s'agit initialement d'une collaboration entre le laboratoire « CES » de l'ENIS et l'équipe « Recherche et développement » de la société française « EMPREINTE MULTIMEDIA ».

Il a été recherché puis démontré lors d'une première étude de faisabilité [EMP] [OUN06] que l'emploi d'un algorithme adapté au flux des diapositives de la présentation du conférencier est très bénéfique pour ce contexte d'usage. Cette solution a présenté les défaillances et les problèmes que posent les systèmes actuels et a proposé une solution adéquate qui consiste à mettre en place un module d'acquisition d'images [OUN06] permettant la récupération des diapositives directement depuis un signal vidéo analogique ou numérique provenant du PC du conférencier.

Cependant, le système réalisé a montré des limites, il s'agissait plutôt d'un système d'acquisition et pas vraiment d'un système de traitement adapté, il traite le flux de diapositives en tant que suite d'images et par la suite il élimine les animations. Il a donc fallu refaire la spécification des traitements à faire et disposer d'algorithmiques intelligents qui permettent des opérations pertinentes et adaptées à ce contexte.

Il nous a été confié donc de concevoir et valider un algorithme vidéo qui devra permettre la récupération de la présentation du conférencier et assurer une compression adaptée des diapositives en faisant appel aux nouvelles technologies et en tenant compte des limites des systèmes actuels.

Table des Matières

Table des Matières	1
Liste des figures	4
Liste des tableaux	5
Introduction générale.....	6
Chapitre I. Les Techniques de codages Image et Vidéo	9
1. Introduction	9
2. Principes de base	10
2.1. Les espaces de représentation de la couleur.....	10
2.1.1. L'espace RGB/RVB.....	10
2.1.2. Les espaces dédiées à la télévision (YCbCr, YUV, et YIQ).....	12
2.2. Représentation numérique.....	13
2.2.1. La notion de pixels	13
2.2.2. Définition et résolution.....	13
2.2.3. Le codage de la couleur.....	14
2.2.4. Poids d'une image.....	15
2.2.5. Les différents types de formats graphiques.....	16
3. Systèmes de codage vidéo	16
3.1. Nécessité de la compression.....	16
3.2. Types de compression	17
3.3. Redondances dans une vidéo.....	18
3.4. Techniques de codage	19
3.5. Codage sans perte.....	19
3.5.1. Codage de Huffman	19
3.5.2. Codage RLE (Run Length Encoding)	20
3.5.3. Codage Lempel-Ziv.....	20
3.6. Codage avec perte	21
3.6.1. Transformation	22
3.6.2. Quantification.....	23
3.7. Codage d'images fixes : standards JPEG & PNG.....	23
3.7.1. JPEG.....	24
3.7.1.1. Principe.....	24

3.7.1.2. DCT.....	24
3.7.1.3. Quantification.....	25
3.7.1.4. Réorganisation en zig-zag.....	25
3.7.1.5. Codage entropique.....	25
3.7.1.6. Limite du standard JPEG.....	26
3.7.2. PNG.....	26
3.7.2.1. Caractéristiques du format PNG	27
3.7.2.2. Structure d'un fichier PNG	28
3.7.3. JPEG vs PNG	29
3.7.4. Choisir le format le plus approprié.....	30
3.8. Normes de codage vidéo	31
3.8.1. Le MJPEG	31
3.8.2. Norme MPEG.....	32
3.8.2.1. Principe de la norme MPEG.....	33
3.8.2.2. MPEG-1	34
3.8.2.3. MPEG-2	35
3.8.2.4. MPEG-4	35
3.8.2.5. MPEG-7	36
3.8.3. Normes UIT-T.....	36
4. Mesures de qualité des codeurs.....	36
5. Conclusion.....	38
Chapitre II. Visioconférence et le flux des diapositives : étude et spécifications	39
1. Introduction	39
2. Contexte général : la visioconférence	40
2.1. Positionnement par rapport au projet de l'équipe	40
2.2. Traitement des transparents dans les systèmes de visioconférence	42
2.2.1. Acquisition	43
2.2.2. Traitement : Image ou vidéo ?.....	44
2.2.2.1. Séquence vidéo.....	44
2.2.2.2. Suite d'images fixes	47
2.3. Réflexions et synthèse.....	48
3. Etude et spécification	49
3.1. Caractéristiques d'une présentation multimédia	49
3.1.1. Hétérogénéité des objets média.....	50

3.1.2. Qu'est-ce qu'un diaporama ?.....	51
3.2. Points d'intérêt	52
3.2.1. Méthodes de détection de changement de plan	54
3.2.2. Méthodes de détection de texte	57
4. Approche proposée.....	60
4.1. Phase d'initialisation	62
4.2. Module de détection et localisation du changement	63
4.3. Module filtrage : élimination du bruit	67
4.4. Module de codage sélectif.....	68
5. Conclusion.....	71
Chapitre III. Expérimentation & résultats.....	72
1. Introduction	72
2. Approche conceptuelle.....	72
3. Environnement de développement	74
4. Différentes étapes d'implémentation.....	76
4.1. Module de détection de changement.....	76
4.2. Détection de texte.....	79
4.3. Module de codage	81
4.4. Phase de réception et décodage.....	83
5.5. Mesure de qualité	84
5.6. Synthèse	86
5- Conclusion	87
Conclusion générale & perspectives	88
Bibliographie.....	90

Liste des figures

Figure 1.1 : Architecture du système proposé	7
Figure 1.2 : Le cube de couleurs RVB	11
Figure 1.3 : Exemple de pixels.....	13
Figure 1.4 : Profondeur de couleur	15
Figure 1.5 : Flot de compression d'image/vidéo	17
Figure 1.6 : Arbre binaire de Huffman	20
Figure 1.7 : Schéma classique d'un système de compression avec perte	22
Figure 1.8 : Schéma-bloc de l'encodeur JPEG	24
Figure 1.9 : Dégradation d'image JPEG à un taux de compression élevé.....	26
Figure 1.10 : Séquence d'images I, P, B de la MPEG.....	34
Figure 1.11 : Courbe de distorsion-débit.....	37
Figure 2.1 : Description globale du système	41
Figure 2.2 : Acquisition et traitement des diapositives.....	42
Figure 2.3 : capture de la projection	45
Figure 2.4 : Illustration de la segmentation et la fonction g.....	45
Figure 2.5 : Exemple de diapositives	51
Figure 2.6 : Structure d'un flux de transparents.....	52
Figure 2.7 : Exemple de texte animé dans un PowerPoint	53
Figure 2.8 : Les Coefficients DCT	59
Figure 2.9 : Diagramme synoptique de l'algorithme.....	61
Figure 2.10 : Module de codage/décodage.....	68
Figure 2.11 : Comparaison de différents formats de fichiers images.....	69
Figure 2.12 : Images présentant des pics en PNG par rapport au JPEG.....	70
Figure 3.1 : Module de capture d'écran	75
Figure 3.2 : Etapes d'expérimentation	75
Figure 3.3 : Schéma blocs des modules de la technique proposée	76
Figure 3.4 : Module de détection des blocs modifiés.....	77
Figure 3.5 : Détection par comparaisons de trames successives.....	79
Figure 3.6 : Détection par comparaisons de trames non-successives (pour même seuil)	79
Figure 3.7 : Détection de texte par calcul d'histogramme	80
Figure 3.8 : Détection de texte dans le domaine compressé.....	81
Figure 3.9 : Module du CODEC JPEG	83
Figure 3.10 : Décodage en JPEG : cumulation des erreurs.....	84
Figure 3.11 : Mesure de PSNR pour le flux codé en JPEG seul.....	85
Figure 3.12 : Mesure de PSNR pour le flux codé en JPEG et PNG	85
Figure 3.13 : Qualité du diapositives entre le JPEG et notre méthode	86

Liste des tableaux

<i>Table 1.1. Méthodes de représentation de couleurs.....</i>	<i>15</i>
<i>Table 1.2 : Poids des images non compressées.....</i>	<i>16</i>
<i>Table 1.3 : Différentes formats d'image et leurs méthodes de compression.....</i>	<i>16</i>
<i>Table 1.4 : Code de Huffman.....</i>	<i>20</i>
<i>Table 1.5 : Comparaison des méthodes de compression sans pertes.....</i>	<i>21</i>
<i>Table 1.6 : Comparaison entre PNG et JPEG.....</i>	<i>29</i>
<i>Table 1.7 : Comparaison des usages typiques entre quatre formats d'images.....</i>	<i>30</i>
<i>Table 1.8 : Exemples de codeurs vidéo avec domaines d'application.....</i>	<i>31</i>
<i>Table 1.9 : Compression des transparents : MPEG vs MJPEG.....</i>	<i>34</i>
<i>Table 2.1 : Techniques de compression du flux des transparents.....</i>	<i>49</i>
<i>Table 2.2 : Différentes méthodes de détection de changement de plan.....</i>	<i>57</i>
<i>Table 3.1 : Taux de transfert.....</i>	<i>78</i>



Introduction générale

Très déployée, au niveau mondial, par de très grandes entreprises, la visioconférence couvre un champ vaste d'outils, de la visiophonie améliorée aux conférences grand public, téléréunions, formation continue interactive, travail collaboratif...

Les progrès incessants des technologies lui ouvrent des champs d'applications plus larges mais dans le même temps en font un élément parmi d'autres de systèmes plus complexes, utilisant plus complètement les possibilités d'Internet [GEM00].

Les usages de la visioconférence concernent le travail coopératif (exp: réunions de travail), la communication (exp: événements ponctuels) ou encore la formation. Historiquement, les applications spécifiques telles que la télé médecine et l'enseignement à distance ont alimenté l'intérêt pour la visioconférence. De nos jours, la plupart des entreprises choisissent cette technologie pour communiquer en interne ou s'étendre aux fournisseurs, partenaires et clients. L'évolution des techniques (« streaming video » par exemple) a fortement rapproché la visioconférence et l'audiovisuel, incitant à réfléchir à la « mise en boîte », à la réutilisation de certains passages et à la post-production. Toutefois, sa mise en œuvre est principalement portée par la volonté technologique, l'analyse des besoins des utilisateurs, ainsi que les besoins en terme de traitement de l'application elle-même [GEM00]. Il s'agit principalement d'un effort commercial des divers producteurs des outils et des matériels qui ont ainsi facilement trouvé un nouveau créneau « porteur ». D'autant que nous sommes conscients de l'enjeu important que représente les techniques de visioconférence sur Internet en terme d'efficacité et de coût, nous pensons qu'un recul ainsi qu'une révision des techniques utilisées sont souhaitables afin de mettre en place des dispositifs taillés à ce type d'application.

Dans ce contexte, nous nous intéressons, afin d'améliorer la qualité de services offerte par ce système, au traitement d'un flux assez particulier qui représente les transparents (ou diapositives) de l'exposé du conférencier. Notre travail consiste donc à définir une méthode spécifique et adaptée aux caractéristiques d'un tel flux.

Pour cela, il a fallu penser à des nouvelles solutions autres que celles usuelles. Généralement, ces solutions sont réalisées selon l'une des deux façons :

- Adapter les techniques et les solutions existantes à la problématique en question ;
- Construire une solution adaptée suite à une étude des spécifications de cette problématique.

Néanmoins, le traitement du flux des transparents d'un conférencier (Fig.1.1 en bleu) dans les systèmes de visioconférence existants ne se basait pas sur une analyse judicieuse des caractéristiques et de la nature de l'information à traiter. Il s'agit principalement de trouver la meilleure adaptation ou exploitation de l'existant en terme de technique, technologie et norme. L'efficacité de ces techniques semble être limitée dans certaines études de cas telles que la notre.

Pour cette raison, nous reprenons l'étude de ce contexte en partant d'une analyse objective du besoin et en tenant compte des spécificités de ce flux et des contraintes (coût, qualité...) exigées pour la mise en œuvre du système global.

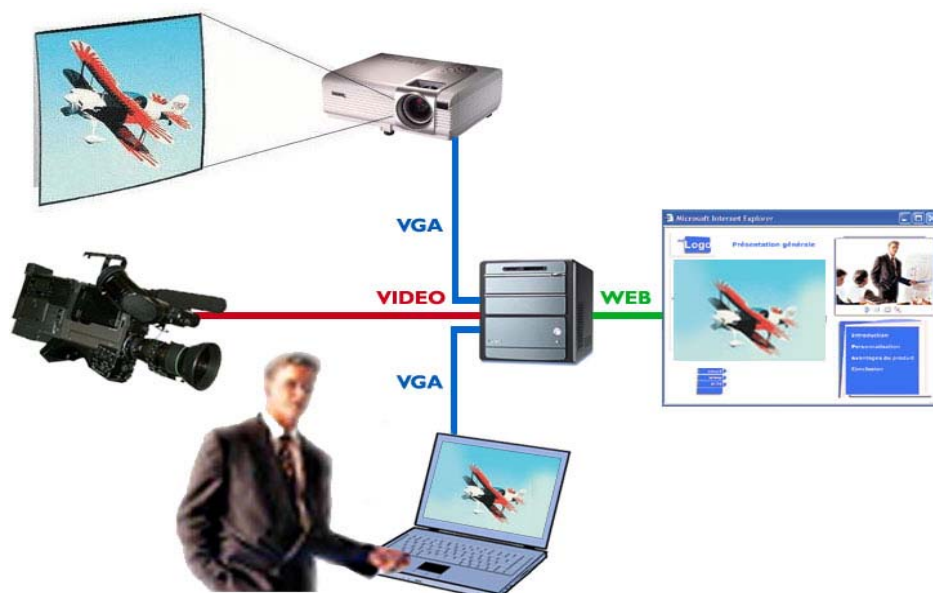


Figure 1.1 : Architecture du système proposé

Cette solution comporte la définition et la conception d'un algorithme qui devra permettre la :

- détection d'image dans le flux VGA,
- traitement des images extraites du flux VGA,
- diffusion et restitution des images,

et qui doit satisfaire les contraintes suivantes :

- limitation de la capacité de mémoire et de la performance du calcul du processeur ;
- limitation de la bande passante de transmission ;

Ce mémoire s'articule autour de trois chapitres dans lesquels nous présentons respectivement : l'état de l'art des systèmes de compression des données numériques en mettant l'accent sur leur performance et leur types d'applications (chapitre 1), les systèmes de visioconférence de point de vue architecture et surtout les techniques utilisées pour le traitement du flux des diapositives ainsi que le développement théorique de notre approche proposée (chapitre 2), et les résultats expérimentaux des testes élaborées pour la validation des différents modules implémentés (chapitre 3).

Finalement en conclusion, un bilan est dressé sur les aspects de notre contribution et nous proposons différentes pistes et perspectives de travail pour l'avenir.

Chapitre I. Les Techniques de codages Image et Vidéo

1. Introduction

La visioconférence constitue un système très complexe, composé de plusieurs éléments fonctionnels (codeurs/décodeurs, interfaces d'acquisition, serveurs, applications...), il implique plusieurs disciplines (électronique, traitement d'image/vidéo, réseau, web...).

Cette richesse et diversité de données impose l'utilisation d'outils et techniques spécifiques qui fiabilisent leur transmission à travers des réseaux informatiques et facilitent leur manipulation. Pour s'y mettre, il a fallu faire appel à plusieurs techniques de traitement d'image et de vidéo afin de pouvoir produire, gérer et diffuser ces différentes informations audiovisuelles (vidéo, son, diapositives). Certaines techniques ont dû être adaptées afin de pouvoir gérer certaines données spécifiques à l'instar du flux des transparents traité en tant que vidéo dans certains cas et en tant qu'une suite d'images dans d'autres. La question qui se pose est qu'on doit utiliser des techniques existantes et les faire adapter ou construire une nouvelle technique qui tient compte des caractéristiques des données traitées ?

Pour répondre à cette question, nous présenterons, dans ce premier chapitre, une étude sur le concept de codage afin d'avoir une vision détaillée sur les principes employés dans les différents types de système de compression d'image et de vidéo. Par la suite nous allons nous baser sur ces différents éléments pour tenter répondre à notre problématique.

2. Principes de base

La couleur est le résultat de la perception sur la rétine des rayons lumineux visibles d'une lumière incidente. La région visible du spectre a des longueurs d'ondes comprises entre 400 nm et 700 nm [SCH01].

La perception de couleur est un phénomène sensoriel qui ne dépend que de nos yeux et de notre cerveau. En fait, la rétine humaine possède deux types de cellules photoréceptrices, les bâtonnets et les cônes. Alors que les bâtonnets ne réagissent qu'à l'intensité de la lumière et souvent à la vision nocturne, les cônes servent à la vision diurne et à la discrimination des couleurs [LEF02].

La couleur détectée par le système visuel humain est une fonction qui dépend de la quantité de lumière incidente $E(\lambda)$, la réflectance sur une surface d'un objet $S(\lambda)$ et la sensibilité des trois différents pigments photorécepteurs $h_R(\lambda)$, $h_V(\lambda)$, $h_B(\lambda)$ [SCH01]. Ainsi la réponse du système visuel à une quantité de lumière $E(\lambda)$ réfléchi sur une surface $S(\lambda)$ est donnée par :

$$R = \int S(\lambda) \cdot E(\lambda) \cdot h_R(\lambda) \cdot d\lambda$$

$$V = \int S(\lambda) \cdot E(\lambda) \cdot h_V(\lambda) \cdot d\lambda$$

$$B = \int S(\lambda) \cdot E(\lambda) \cdot h_B(\lambda) \cdot d\lambda$$

2.1. Les espaces de représentation de la couleur

Le codage de la couleur dans des images numériques peut être effectué en utilisant différents espaces de représentation, appelés souvent espaces de couleur [LEF05].

Des expériences psycho-visuelles d'égalisation (voir KUNT et al. [KUN93]) ont montré qu'en combinant trois stimuli de longueur d'ondes particulières, il est possible de synthétiser presque toutes les couleurs existantes. Cette expérience peut s'effectuer à l'aide d'un système de reproduction de couleurs de type additif ou de type soustractif.

Bien que de nombreux systèmes de représentation de la couleur aient été proposés, nous nous limitons à la représentation des principaux espaces définis dans la littérature.

2.1.1. L'espace RGB/RVB

L'espace RGB (Red, Green, Blue, pour Rouge Vert Bleu, en français RVB), mis au point en 1931 par la CIE, est le système de représentation couleur le plus fréquemment utilisé dans les télévisions et les écrans d'ordinateurs.

Il consiste à représenter l'espace des couleurs à partir de trois rayonnements monochromatiques de couleurs :

- Rouge (de longueur d'onde égale à 700,0 nm),
- Vert (de longueur d'onde égale à 546,1 nm),
- Bleu (de longueur d'onde égale à 435,8 nm).

Dans cet espace le noir et le blanc sont représentés respectivement par les triplets ($R=0$, $V=0$, $B=0$) et ($R=255$, $V=255$, $B=255$) si l'on considère le cas des images numériques couleur codées sur 24 bits. Généralement, ce modèle est représenté graphiquement par un cube dont chacun des axes correspond à une couleur primaire (Figure 1.2 [LUK07]) :

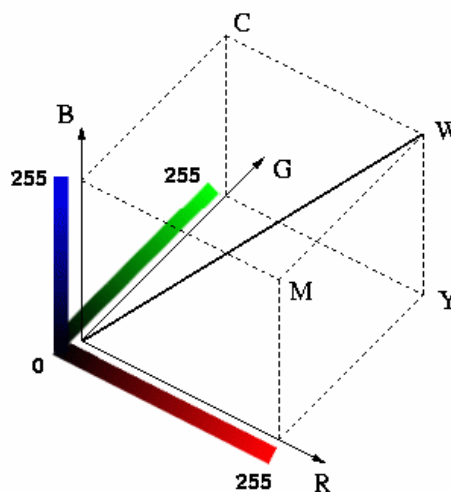


Figure 1.2 : Le cube de couleurs RVB

Ce modèle de représentation propose de coder sur un octet chaque composante de couleur, ce qui correspond à 256 nuances pour chacune soient 16777216 possibilités théoriques de couleurs différentes, c'est-à-dire plus que ne peut en discerner l'œil humain (environ 10 millions) [WIL08]. On conviendra donc que les 256 nuances de chaque primaire sont suffisantes.

Pour stocker une image dans un ordinateur, on affecte une couleur à chacun de ses pixels. Cette couleur est codée en RVB, c'est à dire qu'elle est décrite par la proportion de rouge, de vert et de bleu qu'elle contient. Comme chacune des couleurs est codée sur un octet, il faut trois octets pour décrire une couleur (ou un pixel), ceci bien sûr si l'image n'est pas compressée.

Cet espace présente l'avantage qu'il est parfaitement adapté à l'informatique. De plus, il contient un grand nombre de couleurs, 16 millions. Cependant, il n'est pas intuitif pour l'utilisateur, il est notamment difficile de choisir une couleur rien qu'avec les coefficients

RGB. Enfin, le RGB n'est pas universel et adapté pour tous les systèmes, notamment pour l'impression [CHE05].

2.1.2. Les espaces dédiées à la télévision (YC_bC_r, YUV, et YIQ)

Historiquement, ces espaces ont été adoptés afin de permettre de transmettre des informations colorées aux téléviseurs couleurs, tout en s'assurant que les téléviseurs noir et blanc existant continuent d'afficher une image en tons de gris. Ils sont également très importants dans le monde du multimédia, à mi-chemin entre le codage RGB et la perception humaine. En effet, dans ces espaces, on sépare la luminance Y (c'est-à-dire l'information en noir et blanc) de la chrominance (c'est-à-dire l'information sur la couleur) ce qui s'avère très utile vu que l'oeil humaine est sensible aux écarts de lumière et de couleur, mais pas à des écarts de rouge, vert ou bleu qu'il ne sait trop appréhender. Ainsi, il est plus facile d'agir sur ces composants de luminance et de chrominance pour pouvoir compresser et coder l'image et la vidéo de façon plus adaptée.

Ce passage est effectué en transformant linéairement les composantes trichromatiques RGB. Pour tous les systèmes, la luminance est la même. En contrepartie, les composantes de chrominance Cr₁ et Cr₂ sont obtenues à partir de combinaisons linéaires des valeurs R, B et Y pondérées par des facteurs. Ces facteurs dépendent du standard de télévision utilisé.

La forme générale des composantes chromatiques est donnée par [BER03] :

$$Cr1 = \alpha1 (R - Y) + \beta1 (B - Y), \quad Cr2 = \alpha2 (R - Y) + \beta2 (B - Y).$$

En image, le passage de RGB à YC_bC_r est la base du codage JPEG. En vidéo, le MPEG et bien d'autres formats utilisent le YUV.

Voici les relations liant ces espaces aux composantes de l'espace RGB :

- YC_bC_r : télévision numérique, JPEG :

$$\begin{pmatrix} Y \\ C_b \\ C_r \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.16875 & -0.33126 & 0.5 \\ 0.5 & -0.41869 & -0.08131 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

- YUV : PAL (standard allemand) :

$$\begin{pmatrix} Y \\ U \\ V \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.436 \\ 0.615 & -0.515 & -0.100 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

- YIQ : NTSC (standard américain) :

$$\begin{pmatrix} Y \\ I \\ Q \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ 0.595716 & -0.274453 & -0.321263 \\ 0.211456 & -0.522591 & 0.311135 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

Ces espaces sont parfaits pour être utilisés dans le codage de l'image et de la vidéo. Ils permettent de transmettre aisément de la vidéo sur les téléviseurs en noir et blanc.

Cependant, ces espaces ne sont vraiment adaptés qu'au multimédia, et ne peuvent donc pas réellement être utilisés de façon optimale ailleurs. Ils ne permettent également pas de décrire la couleur avec précision, et ne suivent donc pas vraiment le modèle de vision des couleurs de l'œil [CHE05].

2.2. Représentation numérique

Une image numérique est composée d'un ensemble de pixels (picture element).

Elle se définit par sa résolution, son nombre de couleurs (profondeur) et son format.

2.2.1. La notion de pixels

Une image est constituée d'un ensemble de points appelés pixels (pixel est une abréviation de PICTure ELement). L'ensemble de ces pixels est contenu dans un tableau à deux dimensions constituant l'image :

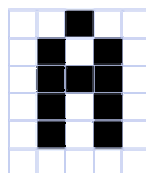


Figure 1.3 : Exemple de pixels

Etant donné que l'écran effectue un balayage de gauche à droite et de haut en bas, on désigne généralement par les coordonnées [0,0] le pixel situé en haut à gauche de l'image [DUN05]

2.2.2. Définition et résolution

On appelle définition le nombre de points (pixel) constituant l'image, c'est-à-dire sa « dimension informatique » (le nombre de colonnes de l'image que multiplie son nombre de

lignes). Une image possédant 640 pixels en largeur et 480 en hauteur aura une définition de 640 pixels par 480, notée 640x480.

La résolution, terme souvent confondu avec la « définition », détermine par contre le nombre de points par unité de surface, exprimé en points par pouce (PPP, en anglais DPI pour Dots Per Inch). Une résolution de 300 dpi signifie donc 300 colonnes et 300 rangées de pixels sur un pouce carré ce qui donne donc 90000 pixels sur un pouce carré [DUN05].

2.2.3. Le codage de la couleur

Une image est donc représentée par un tableau à deux dimensions dont chaque case est un pixel. Pour représenter informatiquement une image, il suffit donc de créer un tableau de pixels dont chaque case contient une valeur. La valeur stockée dans une case est codée sur un certain nombre de bits déterminant la couleur ou l'intensité du pixel, on l'appelle profondeur de codage (parfois profondeur de couleur). Il existe plusieurs standards de codage de la profondeur [GIP07] :

- Bitmap noir et blanc : en stockant un bit dans chaque case, il est possible de définir deux couleurs (noir ou blanc).
- Bitmap 16 couleurs ou 16 niveaux de gris : en stockant 4 bits dans chaque case, il est possible de définir 16 dégradés de gris allant du noir au blanc ou bien 16 couleurs différentes.
- Bitmap 256 couleurs ou 256 niveaux de gris : en stockant un octet dans chaque case, il est possible de définir 56 dégradés de gris allant du noir au blanc ou bien 256 couleurs différentes.
- Palette de couleurs (colormap ou LUT) : grâce à cette méthode, il est possible de définir une palette, ou table des couleurs, contenant l'ensemble des couleurs pouvant être contenues dans l'image, à chacune desquelles est associé un indice. Le nombre de bits réservé au codage de chaque indice de la palette détermine le nombre de couleurs pouvant être utilisées. Ainsi en codant les indices sur 8 bits il est possible de définir 256 couleurs utilisables, c'est-à-dire que chaque case du tableau à deux dimensions représentant l'image va contenir un nombre indiquant l'indice de la couleur à utiliser. On appelle ainsi image en couleurs indexées une image dont les couleurs sont codées selon cette technique.

Table 1.1. Méthodes de représentation de couleurs

	Palette de couleurs	Valeurs de couleurs
BMP	x	x
JPEG		x
GIF	x	
PNG	x	x

- « Couleurs vraies » (True color) ou « couleurs réelles » : cette représentation permet de représenter une image en définissant chacune des composantes RGB. Chaque pixel est représenté par un entier comportant les trois composantes, chacune codée sur un octet, c'est-à-dire au total 24 bits (16 millions de couleurs). Il est possible de rajouter une quatrième composante permettant d'ajouter une information de transparence ou de texture, chaque pixel est alors codé sur 32 bits.



16 couleurs



256 couleurs



16 M de couleurs

Figure 1.4 : Profondeur de couleur

2.2.4. Poids d'une image

Pour connaître le poids (en octets) d'une image, il est nécessaire de compter le nombre de pixels que contient l'image, cela revient à calculer le nombre de cases du tableau, soit la hauteur de celui-ci que multiplie sa largeur. Le poids de l'image est alors égal à son nombre de pixels que multiplie le poids de chacun de ces éléments.

Voici le calcul pour une image 640x480 en TrueColor :

- Nombre de pixels : $640 \times 480 = 307200$
- Poids de chaque pixel 24 bits / 8 = 3 octets
- Le poids de l'image est ainsi égal à : $307200 \times 3 = 921600$ octets, $921600 / 1024 = 900$ Ko.

Voici quelques exemples (en considérant que l'image n'est pas compressée) (Tab.1.2) :

Table 1.2 : Poids des images non compressées

Définition de l'image	Noir et blanc (1 bit)	256 couleurs (8 bits)	65000 couleurs (16 bits)	True color (24 bits)
320x200	7.8 Ko	62.5 Ko	125 Ko	187.5 Ko
640x480	37.5 Ko	300 Ko	600 Ko	900 Ko
800x600	58.6 Ko	468.7 Ko	937.5 Ko	1.4 Mo
1024x768	96 Ko	768 Ko	1.5 Mo	2.3 Mo

2.2.5. Les différents types de formats graphiques

Pour sauvegarder une image, Il existe un grand nombre de formats de fichiers. Parmi les formats de fichiers graphiques voici les plus utilisés (Tab.1.3) [STE06] et [MIA99] :

Table 1.3 : Différentes formats d'image et leurs méthodes de compression

Format	Compression	Dimensions maximales	Nb de couleurs max.
BMP	aucune / RLE	65 536 x 65 536	16 777 216
GIF	LZW	65 536 x 65 536	256
IFF	aucune / RLE	65 536 x 65 536	supérieur à 16 777 216
JPEG	JPEG (RLE, Huffman, DCT)	65 536 x 65 536	supérieur à 16 777 216
PCX	aucune / RLE	65 536 x 65 536	16 777 216
PNG	LZ / Huffman	65 536 x 65 536	supérieur à 16 777 216
TGA	aucune / RLE	65 536 x 65 536	supérieur à 16 777 216
TIFF/TIF	RLE / JPEG / LZW / UIT-T	$2^{32}-1$	supérieur à 16 777 216

3. Systèmes de codage vidéo

Les technologies actuelles d'acquisition d'information et de communication sont à l'origine de l'augmentation importante des volumes de données à gérer, à emmagasiner, à traiter et à transférer.

Prenons l'exemple d'un cas concret : une seconde de vidéo codée en 24 bits possédant une résolution spatiale de CIF (352×288) et une résolution fréquentielle de 25 images/s contient au minimum 60 Mbits. Ainsi, un débit de 60 Mbits/s est nécessaire afin de transmettre cette vidéo en temps réel. La nécessité d'une compression s'impose alors non seulement pour le stockage de données mais aussi pour la transmission de celles-ci à travers un réseau dont le débit est restreint [TOT07].

3.1. Nécessité de la compression

Le procédé inhérent à la compression d'image consiste à réduire la dimension du fichier de données en ne retenant que les informations essentielles. Le taux de compression est spécifié

en utilisant un coefficient de compression CR (compression rate), défini comme étant le rapport des dimensions du fichier non comprimé et du fichier compressé. Les meilleurs procédés de compression sont ceux qui offrent le meilleur rapport entre un CR élevé et les informations essentielles retenues. Évidemment, cela est fonction de l'application visée. Par exemple, dans une image binaire comportant seulement des informations textuelles, l'information pertinente n'implique que le texte, comparativement à une image médicale dont l'information pertinente à extraire peut consister en des détails très fins.

Ainsi, une compression se présente en tant qu'une réduction maximale de la taille des données, donc du débit binaire lors d'une transmission continue, avec le minimum de perte de qualité. Celle-ci est dépendante de trois paramètres principaux : la vitesse de codage et décodage, le taux de compression et la qualité de compression [MEN06].

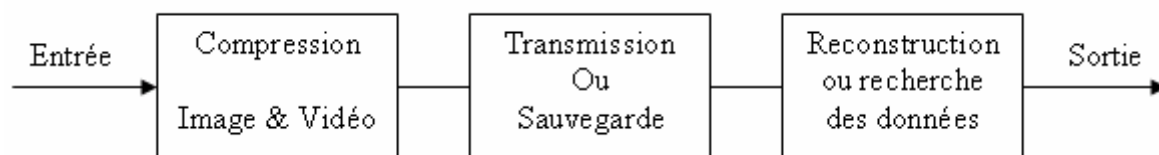


Figure 1.5 : Flot de compression d'image/vidéo

Une méthode de codage d'images ne se juge pas uniquement par ses résultats bruts en terme de taux de compression, mais également par son aptitude à s'adapter aux différents cas d'usages. Ainsi, outre l'efficacité du codage, les principales fonctionnalités qui peuvent être associées à une technique de codage particulière sont [REN04] :

- une bonne qualité visuelle des images à bas débits,
- l'unicité de l'algorithme pour du codage avec et sans pertes,
- un codage par régions d'intérêt,
- une protection des images,
- une complexité réduite,
- un contrôle du débit et de la qualité,
- une résistance aux erreurs.

3.2. Types de compression

Il existe plusieurs méthodes de compression. Une première méthode consiste simplement à réduire la taille de chaque image. Dans une image de dimension 320x240, on ne dénombre qu'un quart des pixels composant une image de dimension 640x480. Une deuxième méthode consiste à réduire le nombre d'images par seconde de la vidéo. Une vidéo comprenant 15

img/s occupe moitié moins d'espace disque qu'une vidéo comprenant 30 img/s. Cependant, ces méthodes de compression élémentaires ne fonctionnent pas si la vidéo doit être affichée sur un moniteur de télévision, avec une résolution et un nombre d'images par seconde optimisés. Il nous faut donc aborder le problème de la compression sous un autre angle.

Il s'avère que l'œil humain est plus sensible aux changements survenus dans la luminance d'une image qu'à ceux survenus dans sa couleur. La plupart des schémas de compression vidéo tirent parti de la particularité de la perception humaine, en éliminant une bonne partie des informations relatives aux couleurs de l'image. Tant que ce type de compression n'est pas trop élevé, il est généralement indiscernable. En réalité, même dans les vidéos de qualité optimale utilisées par les chaînes de télévision, certaines informations d'origine sur les couleurs ont été éliminées.

Lorsque chaque image est compressée séparément, on parle de compression spatiale. Par ailleurs, certains systèmes de compression vidéo utilisent la compression inter-trame ou « interframe ». Ce type de compression temporelle tire parti de la probable similitude entre une image donnée de la vidéo et celles qui la suivent. De ce fait, au lieu de stocker la totalité de l'image, il suffit de stocker les différences entre celle-ci et l'image précédente. On parle ici de l'élimination des redondances [ADO00].

3.3. Redondances dans une vidéo

La réduction des redondances dans la vidéo permet un meilleur taux de compression.

Dans ce domaine, il existe différents types de redondances :

- La redondance spectrale : définit la corrélation entre les différentes composantes de la couleur.
- La redondance spatiale : est la corrélation entre les pixels voisins.
- La redondance temporelle : correspond à la corrélation entre les images de la séquence.
- La redondance psycho-visuelle : est l'exploitation de propriétés de la vue humaine.

Ces redondances sont présentes dans les images naturelles. La compression d'une image nécessite l'exploitation de la redondance spectrale, spatiale et psycho-visuelle. On parle alors d'une compression spatiale.

En ce qui concerne la vidéo, toutes les redondances sont étudiées, dans ce cas on parle d'une compression spatio-temporelle, spatio-fréquentielle ou hybride.

3.4. Techniques de codage

Le codage de source (pour l'élimination de la redondance) et le codage de canal (pour la protection contre les erreurs de transmission) sont les principes de base dans la théorie de l'information [TOT07].

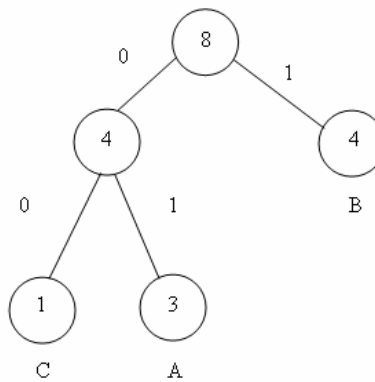
On s'intéresse particulièrement au codage de source. Le codage de source est appelé aussi compression d'information ou compression de données. La transmission du contenu d'une source d'information nécessite des traitements. Les traitements peuvent être divisés en deux catégories : sans perte d'information ou avec perte d'information.

3.5. Codage sans perte

Avec les méthodes sans perte, aucune donnée n'est perdue lors de la phase de compression. Ces méthodes sont nécessaires pour certaines applications telles que les images médicales. Parmi ces techniques, notons les codages Huffman, Run-Length, Lempel-Ziv-Welch et arithmétique. Pour des images complexes, il est très difficile d'obtenir un taux de compression plus élevé que trois avec de telles méthodes. Pour les images simples telles que les images composées uniquement de texte, les méthodes de compression de données sans perte peuvent donner un taux de compression plus élevée [CRI99].

3.5.1. Codage de Huffman

Le codage de Huffman consiste à coder les symboles par une représentation de bits à longueur variable. Les symboles ayant la probabilité d'apparition forte sont codés avec des chaînes de bits plus courtes, tandis que les symboles dont la probabilité d'apparition est faible sont codés par des chaînes plus longues. Le code d'un symbole ne doit pas être le préfixe d'un autre code. Cette propriété est admise afin que la reconnaissance soit possible. Pour représenter le codage de Huffman, on utilise l'arbre binaire. Soit un message à coder « ABBBBBAAC ». La fréquence d'apparition ainsi que le code Huffman correspondant sont donnés dans le Tab.1.4 et représentés par la Fig. 1.6.

**Figure 1.6 : Arbre binaire de Huffman****Table 1.4 : Code de Huffman**

Symbole	A	B	C
Fréquence d'apparition	3	4	1
Code Huffman	01	1	00

3.5.2. Codage RLE (Run Length Encoding)

Plutôt que de coder seulement le message lui-même, il est plus intéressant de coder un message contenant une suite d'éléments répétitifs par « un couple répétition et valeur ». Le codage RLE (parfois noté RLC pour Run Length Coding) consiste en effet à coder un élément du message par sa valeur de répétition. Considérons le message « AAAAAABBBBBBCCC », le code RLE correspondant est « 6A5B3C », ce qui permet d'obtenir un gain de $(14-6)/14$, soit 57%. On s'aperçoit que plus la suite est longue, plus le débit est grand. Pour autant, s'il n'y a pas de répétition d'éléments, la technique ne donne pas de résultats satisfaisants. Voici par exemple un message à coder « ABCABC », le code RLE correspondant est « 1A1B1C1A1B1C », ce qui conduit à un taux de compression négatif $(6-16)/6$, soit -166%. On s'aperçoit que le taux de compression est négatif. Ainsi pour éviter cela, le codage RLE introduit un système de contrôle (bits) pour réaliser l'encodage. Il réalise le codage s'il y a répétition successive d'éléments (minimum égal à 4). Dans le cas contraire, il insère les bits contrôle (00).

3.5.3. Codage Lempel-Ziv

C'est une technique de codage qui utilise un dictionnaire. On cherche dans le fichier les chaînes qui se répètent, puis on mémorise dans le dictionnaire. Ensuite, le codage consiste à

remplacer les chaînes mémorisées par leur indice construit dans le dictionnaire. L'élaboration du dictionnaire ainsi que la recherche de chaîne répétée sont différentes selon la version de l'algorithme. Il en existe trois versions :

- LZ77, version originale, la recherche s'effectue par une fenêtre glissante ;
- LZ78, la recherche s'effectue sur tout le fichier. La taille du dictionnaire est limitée en fonction du mode de codage (16, 32, ou 64 bits) ;
- LZW, introduite en 1984 et brevetée par la société Unisys, est une amélioration de la LZ78. Le dictionnaire, initialement construit, contient l'ensemble des codes ASCII. Il est élaboré au fur à mesure, ce qui permet de changer la taille du dictionnaire au cours du codage.

Le tableau 1.5 [TCO02] montre une comparaison entre quelques exemples de codages sans perte.

Table 1.5 : Comparaison des méthodes de compression sans pertes

	RLE	LZW & LZ77
Avantages	Non destructif, très simple à implémenter, rapide en exécution	Non destructif pour image en 4, 8, 24 bits/pixel
Désavantages	Très faible compression	Faible compression
Format de fichier	BMP, PCX, RLE	GIF, TIF, PNG (pour LZ77)
Images supportées	2, 4, 8, 24 bits/pixel	4, 8 ou 24 bits/pixel
Taux de compression	Typique = 2	Typique ≥ 2 Non prog.

3.6. Codage avec perte

Les méthodes de compression avec perte impliquent la perte d'une certaine quantité de données; par conséquent, l'image finale ne comporte pas toutes les informations initiales qui étaient présentes dans l'image originale. Parmi les techniques de compression de données avec perte, notons les codages « gray-level run-length », « block truncation », « vector quantization », « differential predictive » et le « transform coding ». Avec ces différentes techniques, il est possible d'obtenir un taux de compression de 20 pour des images complexes et pouvant atteindre 200 pour des images simples [CRI99].

Le cas typique est celui du JPEG : un standard de compression normalisée ISO pour image en millions de couleurs (24bits) qui présente, à fort taux de compression des effets de mosaïque.

La Fig. 1.7 représente le schéma classique d'un système de compression avec perte.

Dans un premier temps, afin de mieux compacter l'information, la source est transformée en groupe de coefficients. Les transformations les plus utilisées, que ce soit pour les images fixes

ou les séquences d'images, sont la Transformée en Cosinus Discrète (DCT), la Transformée en Ondelettes Discrète (DWT) ou la décomposition Pyramidale.

Dans un second temps, les coefficients obtenus après la transformation sont quantifiés (tronqués). La phase de quantification introduit l'erreur dans le système de codage.

La dernière étape consiste à coder les coefficients quantifiés par le codage entropique.

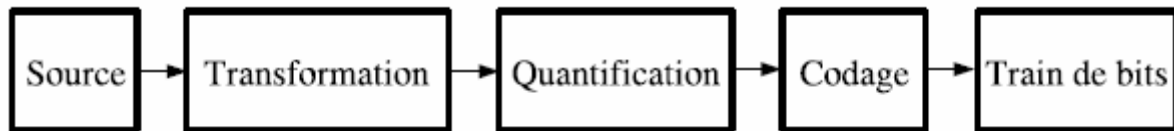


Figure 1.7 : Schéma classique d'un système de compression avec perte

3.6.1. Transformation

La transformation des données d'entrée est faite afin de mieux compacter l'énergie de la transformée d'image sur un nombre faible de coefficients. La transformation a pour objet de décorrélérer les pixels d'image. On opère la transformation sur un bloc ou directement sur l'image entière. On trouve principalement trois types de transformée :

- Transformée de Fourier Discrète DFT

La DFT permet de passer du domaine spatial au domaine fréquentiel. La DFT d'un signal discret x_n s'exprime par :

$$X_k = \sum_{n=0}^{N-1} x_n \exp \left(-i \frac{2\pi nk}{N} \right)$$

La fonction inverse permettant de remonter au signal original x_n connaissant sa transformée X_k est :

$$x_n = \sum_{k=0}^{N-1} X_k \exp \left(+i \frac{2\pi nk}{N} \right)$$

La DFT présente un coût de calcul très élevé. Ainsi, une variante de la transformée de Fourier dite rapide (FFT) a été développée pour assouplir ce coût.

- Transformée en Cosinus Discrète DCT

La DCT est une variante de la DFT et il en existe plusieurs. Dans le cadre du signal mono-dimensionnel, la plus connue est celle-ci :

$$X_k = \sum_{n=0}^{N-1} x_n \cos \left(\frac{\pi}{N} \left(n + \frac{1}{2} \right) k \right)$$

La DCT est très utilisée dans la compression d'image fixe, notamment dans la norme JPEG. Dans cette application la DCT s'effectue sur un bloc de pixels 8×8, et qui s'exprime par :

$$C(u, v) = \alpha(u)\alpha(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \cos \left[\frac{(2x+1)u\pi}{2N} \right] \cos \left[\frac{(2y+1)v\pi}{2N} \right]$$

$$\alpha(u) = \begin{cases} \sqrt{\frac{1}{2}} & \text{pour } u = 0 \\ 1 & \text{pour } u = 1, \dots, N-1 \end{cases}$$

Avec :

- Transformée en Ondelettes Discrètes DWT

Contrairement à la transformée de Fourier, la transformée en ondelettes permet de déterminer les différentes composantes fréquentielles d'un signal donné, ainsi que leur localisation spatiale ou temporelle. Par définition, les ondelettes sont des fonctions gérées à partir d'une ondelette mère Ψ , par dilatations et translations.

3.6.2. Quantification

La quantification est l'une des sources de perte d'information dans le système de compression. Son rôle est en effet de réduire le nombre de bits nécessaire à la représentation de l'information. Elle est réalisée avec la prise en compte de l'aspect psycho-visuel (l'oeil humain), ce qui permet de déterminer la distorsion tolérable à apporter au signal à coder.

On distingue deux sortes de quantification : la quantification scalaire (réalisée indépendamment pour chaque élément) et la quantification vectorielle (effectuée sur un groupe d'éléments de la source et nécessite l'élaboration d'un dictionnaire suite à un processus d'apprentissage).

3.7. Codage d'images fixes : standards JPEG & PNG

Pour notre méthode de compression, nous avons deux contraintes à respecter : la simplicité de mise en œuvre et la qualité. D'autre part, et en tenant compte de la particularité du flux des diapositives, nous avons décidé de coder ce flux par un codec d'image et non pas de vidéo. Ainsi, nous avons commencé par l'étude du JPEG, puisqu'il est le plus utilisé. Mais, suite à quelques expérimentations nous avons constaté que l'utilisation d'un autre codec sans perte

s'avère encore bénéfique pour certains cas. Ce choix sera détaillé dans le dernier chapitre. Pour cette raison, nous allons détailler, dans la suite, ces deux formes de codage.

3.7.1. JPEG

Norme de compression d'image fixe, établie en 1991, JPEG permet la compression sans perte (JPEG-LS) et avec perte d'information. C'est une norme qui a eu beaucoup de succès depuis plus de dix ans et que nous utilisons toujours aujourd'hui.

Elle s'impose surtout dans le domaine d'archivage d'images naturelles.



Figure 1.8 : Schéma-bloc de l'encodeur JPEG

3.7.1.1. Principe

L'encodage par JPEG est réalisé en quatre étapes successivement : transformée en cosinus discrète DCT, phase de quantification, réorganisation en zig-zag, suivie de codage RLC et codage de Huffman (Fig. 1.8) [WAL91].

3.7.1.2. DCT

La première étape consiste à découper l'image originale en blocs de taille 8×8.

Ensuite pour chaque bloc, on applique la DCT. On a donc une matrice dont les composantes sont les coefficients de la transformée DCT. Ainsi, on obtient deux sortes de coefficients, DC et AC. Le coefficient DC représente la moyenne des pixels appartenant au bloc courant (premier élément de la matrice transformée), les éléments restants sont des coefficients AC.

$$\begin{pmatrix} 150 & 170 & 132 & 185 & 147 & 190 & 215 & 220 \\ 165 & 185 & 130 & 190 & 175 & 196 & 223 & 199 \\ 155 & 163 & 180 & 220 & 202 & 173 & 197 & 170 \\ 143 & 154 & 160 & 170 & 211 & 185 & 190 & 166 \\ 130 & 140 & 172 & 190 & 193 & 150 & 180 & 140 \\ 135 & 164 & 198 & 180 & 177 & 141 & 172 & 135 \\ 170 & 190 & 163 & 140 & 165 & 132 & 160 & 140 \\ 160 & 200 & 145 & 135 & 170 & 199 & 190 & 129 \end{pmatrix} \quad \begin{pmatrix} 338 & -49 & -39 & 10 & -23 & 12 & -64 & 5 \\ 62 & -78 & 16 & -13 & 35 & 24 & -6 & -42 \\ 8 & -7 & 76 & 27 & -27 & -13 & -10 & -44 \\ -31 & 1 & 8 & -40 & 6 & 12 & 4 & 3 \\ -9 & -34 & 0 & 24 & -10 & -7 & 7 & -8 \\ -9 & 10 & 14 & -10 & 10 & -13 & 10 & 11 \\ 11 & 3 & -29 & -20 & -7 & 14 & 9 & 0 \\ -8 & 14 & -8 & -17 & 16 & 13 & -2 & 2 \end{pmatrix}$$

bloc de taille 8x8

Après une transformation par DCT

3.7.1.3. Quantification

C'est la deuxième étape. Cette phase consiste à donner les valeurs approximatives de la matrice précédente. Pour cela, une matrice de quantification est construite. Elle doit prendre en compte l'aspect psychovisuel, par exemple : l'œil humain est un filtre passe-bas.

On applique alors la quantification qui revient à diviser chacun des coefficients par la matrice du JPEG par exemple :

$$\begin{pmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{pmatrix}$$

On obtient :

$$\begin{pmatrix} 21 & -4 & -3 & 0 & 0 & 0 & -1 & 0 \\ 5 & -6 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 4 & 1 & 0 & 0 & 0 & 0 \\ -2 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

3.7.1.4. Réorganisation en zig-zag

L'objet de cette troisième étape est d'obtenir le maximum de suites de zéros.

En effet, la norme JPEG traite les valeurs zéro (coefficients haute-fréquence) de la matrice quantifiée en raison de leur nombre important. Ceci concerne seulement les coefficients AC du bloc. Le coefficient DC, quant à lui, est codé par le codage différentiel. Ce dernier consiste à coder la différence entre les deux coefficients DC successifs courant et précédent.

3.7.1.5. Codage entropique

La suite des coefficients sera ensuite codée en RLE. La dernière étape est le codage de Huffman des coefficients obtenus précédemment.

Ainsi, on a une image compressée en JPEG.

3.7.1.6. Limite du standard JPEG

La norme JPEG ne permet pas un taux de compression très élevé. En effet, au delà, des artefacts très gênants sont brusquement présents dans l'image codée (effets de blocs dus à la DCT et la quantification, voir Fig. 1.9).



Figure 1.9 : Dégradation d'image JPEG à un taux de compression élevé

3.7.2. PNG

Mis au point en 1995, le format PNG, qui est une abréviation de Portable Network Graphics, présente un standard de codage d'images en "champs de bits", portable, sans pertes (car aucune quantification n'est appliquée pour la création forcée d'une palette), bien compressé, et exhaustivement spécifié. Bien que la motivation initiale ayant conduit au développement du format PNG fut de remplacer le format GIF, ce nouveau format donne l'occasion d'introduire de nouvelles fonctions, absentes du codage GIF, et pour un coût minime en termes de développement.

Les caractéristiques GIF conservées dans le format PNG incluent :

- Images en couleurs indexées jusqu'à 256 couleurs.
- Affichage progressif : un fichier image correctement encodé peut être affiché au fur et à mesure qu'il est reçu à travers une liaison de communication, affichant très rapidement une version en très basse résolution suivie d'une amélioration progressive des détails.
- Transparence : des portions de l'image peuvent être spécifiées comme transparentes, créant un effet d'image non rectangulaire.

- Informations annexes : des commentaires textuels et d'autres données auxiliaires peuvent être enregistrées dans le fichier image.
- Indépendance totale vis à vis du matériel et des environnements.
- Compression sans pertes (5 à 25% meilleure que la compression GIF.), elle utilise un algorithme dérivé du LZ77 (de même que zip, gzip...).

Les données compressées sont construites à l'aide de la méthode deflate, qui applique sur les données en niveau de gris, RGB ou sur les index de palette une compression de type dictionnaire basée sur l'algorithme LZ77. Pour améliorer les performances de compression, les codes produits par la méthode LZ77 sont encodés par un algorithme de Huffman.

Aucune perte n'est introduite dans tout le processus. Une modélisation optionnelle par prédiction peut être opérée préalablement à la compression. Différents filtres sont proposés : le pixel peut être remplacé par une différence entre lui et

- son voisin à gauche,
 - son voisin sur la ligne précédente,
 - une moyenne arrondie des teintes du pixel de gauche et du pixel supérieur, ou
 - sa prédiction par une méthode proposée par Paeth. Elle se fait en prenant en compte le pixel de gauche et les pixels de la ligne supérieure sur la même colonne et une colonne antérieure.
- Ces filtrages permettent d'améliorer la compression, mais ils visent des natures très spécifiques d'image.

PNG préconise donc de ne pas risquer un mauvais choix de prédiction et de ne pas utiliser ces options [SIB03].

3.7.2.1. Caractéristiques du format PNG

Les caractéristiques nouvelles importantes du format PNG, absentes du format GIF, incluent :

- Images en vraie couleur jusqu'à 48 bits par pixel.
- Images en niveaux de gris jusqu'à 16 bits par pixel.
- Canal alpha complet (masque général de transparence).
- Informations gamma de l'image, qui permet la correction automatique d'affichage d'images avec une luminosité et un contraste correct quelle que soient les machines où ont été créées l'image, et celles où elles sont diffusées.
- Détection fiable et rapide d'erreurs dans le fichier.
- Premier affichage plus rapide dans le mode progressif.

PNG se veut [BOU96]:

- Simple et portable : les développeurs devront pouvoir implémenter PNG facilement.

- Libre de droits : selon le voeu et à la connaissance des auteurs du format PNG, aucun algorithme du domaine privé n'est utilisé.
- Efficacement compressé : tant les images en couleurs indexées qu'en vraies couleurs sont compressées avec la même efficacité que dans la plupart des autres formats d'images "sans pertes" couramment utilisés, et dans bien des cas avec un gain encore supérieur.
- Interchangeable : tout décodeur conforme au PNG doit pouvoir lire tous les fichiers codés selon PNG.
- Souple : ce format est ouvert pour des extensions futures, voire des implémentations privées, sans compromettre le principe d'interchangeabilité ci-avant.
- Robuste : son design permet d'effectuer un test d'intégrité complet du fichier, ainsi que la détection en temps réel d'erreurs simples de transmission.

3.7.2.2. Structure d'un fichier PNG

Ce format de fichier décrit les dimensions et les options de compression dans une en-tête. La partie des données compressées est ensuite incluse, exprimée dans le format zlib.

Un fichier est constitué donc d'une signature, permettant de signaler qu'il s'agit d'un fichier PNG, puis d'une série d'éléments appelés chunks (le terme "segments" sera utilisé par la suite). La signature d'un fichier PNG (en notation décimale) est la suivante :

137 80 78 71 13 10 26 10

La même signature en notation hexadécimale est :

89 50 4E 47 0D 0A 1A 0A

Chaque segment (chunk) est composé de 4 parties :

- La taille, un entier non signé de 4 octets, décrivant la taille du segment.
- Le type de segment (chunk type) : un code de 4 caractères (4 octets) composés de caractères ASCII alphanumériques (A-Z, a-z, 65 à 90 et 97 à 122) permettant de qualifier la nature du segment.
- Les données du segment (chunk data).
- Le CRC (cyclic redundancy check), un code correcteur de 4 octets permettant de vérifier l'intégrité du segment.

Les segments peuvent être présents dans n'importe quel ordre si ce n'est qu'ils doivent commencer par le segment d'en-tête (IHDR chunk) et finir par le segment de fin (IEND chunk).

3.7.3. JPEG vs PNG

Une première chose : Le PNG n'est pas le choix ultime dans tous les cas. C'est le meilleur choix lorsqu'il s'agit de sauver :

- Un screenshot classique de programme Windows ou d'un ancien jeu en 256 couleurs (ou moins)
- Logos
- Images ne contenant que du texte
- Images aux bords bien nets et/ou surfaces unies
- Documents scannés en 2 couleurs (noir & blanc)

Ce n'est par contre pas le meilleur choix lorsqu'il s'agit de sauver :

- Photos ou images sans bords bien nets
- Images de jeux en 16 millions de couleurs
- Documents scannés en niveau de gris ou en 16 millions de couleurs

Pour ces trois cas, le JPEG est souvent un meilleur choix pour ce qui est de la taille du fichier. Par contre on a une perte plus ou moins importante de la qualité, selon le taux de compression.

Souvent le PNG donne un fichier plus petit, pour une qualité d'image non altérée, contrairement au JPEG qui dégrade et au GIF qui oblige un passage en 256 couleurs maximum [ANT07].

Dans une comparaison faite par Adam Costello (membre du PNG-Group) un encodeur JPEG a été testé par rapport à l'encodeur PNG en se basant sur huit images couleurs standards de Waterloo BragZone's ColorSet Tab.1.6 [ROE99].

Table 1.6 : Comparaison entre PNG et JPEG

Classification	Nom	Nb Tot de pixels	Taille en JPEG	Taille en PNG	Différence relative
« Naturel »	lena	262.144	445.799	475.430	+6.6%
	Monarch	393.216	555.012	615.260	+10.9%
	Peppers	262.144	385.047	425.560	+10.5%
	Sail	393.216	767.374	808.606	+5.4%
	Tulips	393.216	616.536	680.881	+10.4%
« Artistique »	Clegg	716.320	653.299	484.589	-25.8%
	Frymire	1.235.390	935.285	251.865	-73.1%
	Serrano	499.426	293.532	106.765	-63.6%

Les résultats étaient fascinants. Dans les cinq images de l'épreuve catégorisées par l'Université de Waterloo sous le nom « naturel » JPEG a battu PNG par entre 5% et 11%, pas une différence énorme, mais certainement considérable. Cependant, dans les trois images marquées « artistique », PNG a prouvé une grande supériorité, avec une image plus de trois fois plus petite que la version JPEG correspondante.

3.7.4. Choisir le format le plus approprié

Si on dispose des images photographiques et leur qualité comme JPEG est acceptable, on doit donc utiliser le JPEG. JPEG sera plus petit que PNG presque toujours, surtout pour les images de la couleur. Inversement, si on a des images avec seulement quelques couleurs et/ou arêtes vives (tel que texte et graphique simple), JPEG n'est jamais la solution correcte; il faut utiliser au lieu PNG ou GIF. Pour les animations, on peut utiliser MNG ou GIF.

Le tableau 1.7 ci-dessous résume les tâches pour lesquelles les formats PNG, GIF, JPEG, TIFF ont tendance à être mieux adaptés [ROE99]:

Table 1.7 : Comparaison des usages typiques entre quatre formats d'images

	PNG	GIF	JPEG	TIFF
Edition, image en palette, sauvegarde rapide	X	X		X
Edition, image en TrueColor, sauvegarde rapide	X			X
Edition finale, compression optimale	X			
Web, image en TrueColor, pas de transparence			X	
Web, image en palette de couleur, pas de transparence	X	X		
Web, image, avec/sans transparence	X	X		
Web, avec transparence partielle	X			
Web, cross-platform color consistency	X			
Web, animation		X		
Web, portabilité max en navigateurs	discutable	X	X	
Web, possibilité de petites images	X		X	

A travers ce tableau on peut déduire que le format TIFF n'est pas adéquat au web vu, simplement, qu'il n'est pas supporté par les navigateurs en majorité.

Le GIF a été traditionnellement le format le mieux supporté grâce à sa simplicité, mais plusieurs applications (shareware et freeware) ne l'adoptent plus car ils doivent payer une redevance à la société détentrice des droits, Unisys.

PNG est relativement nouveau. Il est maintenant supporté par la plupart des éditeurs d'image.

Le choix d'un format adéquat au web dépend totalement de quel aspect exigé dans l'image : la transparence écarte directement le JPEG, la transparence partielle élimine aussi le GIF.

Pour les images opaques et photographiques, JPEG est le choix raisonnable (sa compression ne peut pas être battue).

3.8. Normes de codage vidéo

Pour la vidéo, le problème de taille est qu'il s'agit d'une application très gourmande en débit. La compression et la décompression de la vidéo sont gérées par un module appelé codec. Les codecs sont fournis soit par du matériel (caméscopes numériques ou cartes d'acquisition), soit par des logiciels. Certains codecs ont un taux de compression fixe, et donc un débit fixe, tandis que d'autres peuvent compresser chaque image différemment selon son contenu, avec par conséquent un débit susceptible de varier dans le temps [ADO00].

De nombreux organismes internationaux s'occupent de la normalisation des systèmes de codage vidéo : l'Union Internationale de Télécommunications-Télécommunications (UIT-T), l'Organisation Internationale de Normalisation (ISO) et la Commission Electrotechnique Internationale (IEC).

Le tableau 1.8 ci-après présente différents types de codecs vidéo ainsi que leurs domaines d'application [ADO00].

Table 1.8 : Exemples de codeurs vidéo avec domaines d'application

Format	Résolution	Type de compression	Débit (Mo/s)	Applications
MJPEG	720 x 486	Intratrame	0.5 – 25	Générales
MPEG-1	352 x 240	Intratrame	0.01 – 0.06	CDRom, Internet
MPEG-2	720 x 480	Intratrame et Intertrame	3.5	DVD, TV, satellite
DV	720 x 486	Intratrame	0.5 - 25	Grand public, télédiffusion

3.8.1. Le MJPEG

La première idée qui vient à l'esprit après s'être intéressé à la compression d'images est d'appliquer ce type de méthode à une succession d'images numériques (animation ou vidéo).

Le principe du Motion JPEG (noté MJPEG ou M-JPEG) consiste à appliquer successivement l'algorithme de compression JPEG aux différentes images d'une séquence vidéo.

Etant donné que le M-JPEG code séparément chaque image de la séquence, il permet d'accéder aléatoirement à n'importe quelle partie d'une vidéo. Ainsi, son débit de 8 à 10 Mbps le rend utilisable dans les studios de montage numérique.

Avantages : MJPEG est déjà largement supporté au niveau matériel, y compris par des composants bon marché comme la Pinnacle DC10, la série des Matrox Marvel, et une TONNE d'autres. MJPEG atteint des taux de compression honorables et une bonne qualité pour des images naturelles comme de la vidéo filmée.

Inconvénients : Voilà en quoi MJPEG devient moins attrayant. D'abord, les animations se compressent mal en JPEG. En raison du fonctionnement de JPEG, l'algorithme ne s'en sort pas très bien avec les changements brusques de couleur et de luminosité (par exemple, de fines lignes noires séparant des blocs de couleur), ce qui veut dire que MJPEG souffre du même problème. JPEG a des difficultés incontournables avec les bords nets, en partie car la norme a été créée pour compresser des images naturelles comme des photographies scannées. Il y a aussi le problème de la recompression. Ainsi, quand vous compressez du MJPEG vers un autre format pour la diffusion, vous obtenez une image de moins bonne qualité que si vous aviez utilisé un codec sans perte car des données manquent qui, si elles ne sont pas importantes pour le cortex visuel, sont très importantes pour l'analyse mathématique d'éléments comme la couleur [A&E04].

3.8.2. Norme MPEG

Le groupe d'experts MPEG a pour objet de développer des normes permettant la compression, la décompression, le traitement et le codage des images animées et des données audio. Le principe de codage MPEG est basé sur la compensation de mouvement. Dans un premier temps, on convertit les composantes couleurs présentes dans l'image en luminance et chrominances (que l'on note YUV). Ainsi, les composantes UV contiennent deux fois moins d'information que la luminance Y. Dans un second temps, chaque composante est découpée en macroblocs (MB) de taille 16×16 pixels. Chaque macrobloc est ensuite découpé en blocs de 8×8 pixels. Le nombre de chrominances UV associé à chaque macrobloc est défini par le sous-échantillonnage choisi qui peut être de trois types : 4:2:2, 4:2:0 ou 4:4:4. Le format 4:2:2 signifie ceci : pour chaque pixel de Y, est échantillonné un pixel sur deux pour U et un pixel sur deux pour V; c'est le format le plus répandu dans la norme MPEG. Le choix du format est effectué en fonction de l'application visée.

La norme MPEG utilise deux modes de compression, spatiale et temporelle. Ces modes ont conduit à la définition de trois images appelées : image intra I (correspondant à la compression spatiale), image prédite P et image bidirectionnelle B (correspondant à la compression temporelle).

3.8.2.1. Principe de la norme MPEG

a) Image intra I

Appelée aussi image-clé, l'image intra est une image JPEG. C'est une image de référence à partir de laquelle les images P et B sont prédites. Dès lors l'image I est intercalée afin que le décodeur puisse décoder l'image en choisissant l'image I la plus proche dans le flux reçu. Grâce à cette fonctionnalité, le décodeur peut faire une lecture rapide (avant et retour). Comme c'est une image JPEG, la transmission d'une image I au récepteur demande beaucoup de bande passante. Elle est donc très coûteuse en termes de débit de transmission.

b) Image prédite P

L'image P est une image prédite par rapport à une image I ou à une image P codée précédemment. C'est une image codée avec la compensation de mouvement.

En effet, seule la différence entre le bloc en cours et le bloc d'image précédente est codée. On cherche dans l'image précédente un macrobloc identique ou semblable pour optimiser la compression. Ainsi la différence obtenue est codée spatialement, comme une image I. L'opération effectuée par la technique d'estimation de mouvement fournit les informations nécessaires au décodage. Ces informations qui sont le vecteur de mouvement et l'erreur de prédiction sont transmises au décodeur afin que celui-ci puisse reconstruire le macrobloc final en y ajoutant ces informations.

c) Image bidirectionnelle B

C'est une image prédite, interpolée par rapport à deux images : l'une correspondant à une image précédente (image I ou P) et l'autre correspondant à une image suivante (image P). Les images I et P peuvent servir de référence, les images B ne servent jamais de référence (pour éviter la répercussion des erreurs de prédiction). L'image B permet le meilleur taux de compression, mais en contre-partie elle comporte une certaine complexité ainsi qu'un coût de calcul assez important.

Le codage vidéo selon la norme MPEG constitue une succession d'images I, P et B (Fig. 1.10). L'insertion des images B dans le flux nécessite de changer l'ordre des images dans le flux vidéo. En effet, le décodeur ne peut pas décoder une image B sans avoir obtenu l'image

de référence correspondant si cette dernière se situe plus loin dans la séquence. Dans ce cas un retard est forcément introduit afin que le décodeur puisse décoder l'image B.

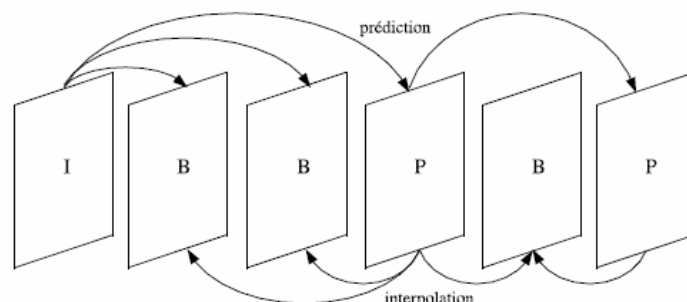


Figure 1.10 : Séquence d'images I, P, B de la MPEG

MPEG et MJPEG présentent les codecs les plus utilisés en compression vidéo. Cependant, aucun de ces deux méthodes ne garantit, au même temps, la forte compression (MPEG mieux que MJPEG) et la simplicité de l'algorithme (MJPEG plus simple que MPEG). Ainsi, un format adéquat à une application ne l'est pas pour une autre. Exemple, pour les applications de montage et de traitement vidéo le MJPEG est le plus utilisé, alors que pour une application de vidéo en demande (VoD) on utilise le MPEG pour gagner en capacité de stockage et en bande passante. Pour notre cas aussi, les deux normes montrent des avantages et des inconvénients (Tab.1.9).

Table 1.9 : Compression des transparents : MPEG vs MJPEG

Technique	Commentaire	Avantages	Inconvénients
MPEG	Traite le flux comme vidéo	Qualité moyenne, animation conservée	Complexité, temps de latence, perte d'informations
MJPEG	Traite le flux comme une suite d'images fixes	Débit faible, simplicité	Perte de l'animation, acquisition aléatoire

3.8.2.2. MPEG-1

La norme MPEG-1 (ISO/IEC-11172) créée en 1992, permet l'encodage de vidéo avec un débit de l'ordre de 1.5 Mbits/s (c'est-à-dire le CD-ROM 1x, le chiffre de 1,5 incluant l'audio) et une résolution de 352x240. La norme est destinée à l'archivage de données. Elle comporte plusieurs parties dont la partie vidéo et la partie audio. Cette dernière a donné naissance au format de compression audio MP3.

Avantages : MPEG-1 est le codec qui a permis la diffusion de la vidéo numérique. MPEG a pu être diffusé en flux, enregistré sur CD... C'est aujourd'hui encore un algorithme de compression parfaitement viable.

Inconvénients : MPEG-1 est vieux, ça ne fait pas de doute. Il est toujours très bon, et écrase tout le reste en termes de compatibilité, mais du point de vue de la qualité technique, on fait mieux. Pour résumer, il demande plus de bits/seconde que les autres codecs [A&E04].

3.8.2.3. MPEG-2

La MPEG-2 (ISO/IEC-13818), introduite en 1994, permet un débit de 20 Mbits/s pour la vidéo petit format, un débit de 3 à 6 Mbits/s pour la vidéo de qualité télévision (720×576). Ses résultats sont de meilleure qualité qu'avec son prédécesseur MPEG-1. La norme est utilisée pour le DVD, VCD et SVCD. Elle est également utilisée dans la diffusion de la télévision numérique.

La norme MPEG-2 permet d'avoir quatre résolutions telles que haute résolution (1920×1152), résolution normale (720×576) ainsi que basse résolution (352×288) ce qui n'est pas le cas de la norme MPEG-1 car elle ne définit qu'une seule résolution CIF (352×288). A un débit comparable, la qualité d'image de MPEG-2 est supérieure à celle de MPEG-1 en pleine résolution CCIR 601 (720×480).

En contre partie, MPEG-2 nécessite plus de puissance CPU que MPEG-1 pour la décompression, et n'offre presque aucun avantage par rapport à MPEG-1 dans les basses résolutions. Si vous faites de la diffusion sur le web, vous n'aurez aucun intérêt à compresser en MPEG-2 plutôt qu'en MPEG-1, puisque les avantages de MPEG-2 n'apparaissent qu'en pleine résolution télé, ce qui donne de gros fichiers [A&E04].

3.8.2.4. MPEG-4

La MPEG-4 (ISO/IEC-14496), introduite en 1998, est une norme de codage basée sur une approche orientée objet. Une scène devient alors une composition d'objets média hiérarchisés. Dans chaque hiérarchie, on trouve la description de la scène comportant l'arrière plan, les objets en mouvement séparés avec le fond ainsi que les objets audio.

La norme MPEG-4 a été définie pour recouvrir trois domaines : l'informatique, les télécommunications ainsi que la télévision. De ce fait, beaucoup de domaines d'application sont visés : par exemple, la télévision numérique, la production vidéo, le multimédia embarqué, le streaming vidéo. La norme MPEG-4 est optimisée pour trois gammes de débits : inférieur à 64 Kbits/s, de 64 à 384 Kbits/s ainsi que de 384 Kbits/s à 4 Mbits/s.

3.8.2.5. MPEG-7

La MPEG-7 (ISO/IEC-15938), introduite en 1997, est une norme de représentation du contenu des documents multimédias. La norme est une description standardisée des données permettant une interprétation informatique (métadonnées). MPEG-4 et MPEG-7 sont complémentaires. En effet, pour une séquence donnée, la MPEG-4 code la séquence tandis que MPEG-7 y ajoute une couche d'informations.

3.8.3. Normes UIT-T

La première norme élaborée par l'UIT-T est le H.261 en 1993. Cette norme vise les applications de visiophonie pour le réseau RNIS à des débits multiples de 64 Kbits/s [LOR03]. La procédure de codage est semblable à la norme MPEG, notamment le découpage en macroblocs, la DCT, la quantification, le codage entropique ainsi que la compensation de mouvement. L'estimation de mouvement d'un bloc se fait en direction horizontale et verticale. La norme accepte deux formats de vidéo : QCIF (176×144) et CIF (352×288).

Après l'apparition de la norme H.261, une nouvelle norme, H.263, a été élaborée en 1995. La norme est destinée à des applications à très bas débit. C'est une norme qui reprend le principe de base de la recommandation H.261 mais en y ajoutant quelques précisions telles que la compensation de mouvement.

La norme AVC (Advanced Video Coding)/H.264 ou MPEG-4 part-10 est une norme élaborée conjointement par le MPEG et l'UIT. La réalisation technique de la norme (appelée aussi H.264 tout court) a été effectuée au sein du Groupe JVT (Joint Video Team). La norme inclut plusieurs profils qui concernent les applications mobiles, E-streaming et radiodiffusion.

4. Mesures de qualité des codeurs

Pour une source donnée, la mesure permettant d'évaluer la capacité de codage est l'entropie [Sha48]. La définition de l'entropie H est donnée par l'expression :

$$H = - \sum_{n=0}^{M-1} \text{Pr}(x_n) \times \log_2(\text{Pr}(x_n))$$

où $\text{Pr}(x_n)$ est la probabilité de la valeur d'intensité x_n dans la source et M est la dynamique du signal (pour une image, typiquement $M = 256$).

Shannon a démontré qu'il est possible de diminuer le coût de codage d'une source donnée en regroupant les symboles à coder. Plusieurs techniques de codage sont inspirées par ce théorème, par exemple le codage de Huffman. On peut donc coder la source sans perte d'information (réversible) jusqu'à son entropie. En revanche, au delà de l'entropie une distorsion apparaît dans la source, codage irréversible (Fig. 1.11).

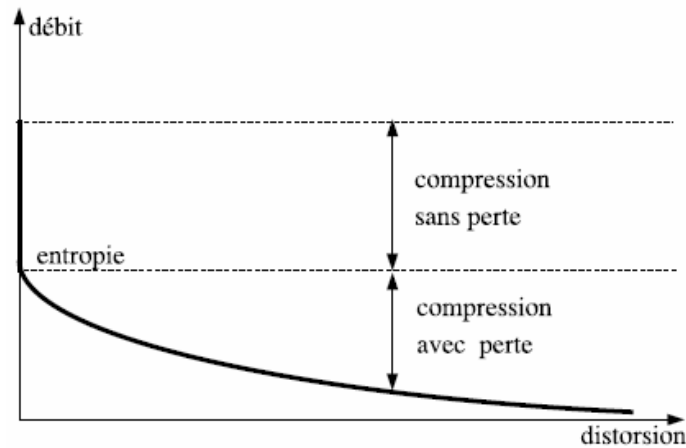


Figure 1.11 : Courbe de distorsion-débit

Deux techniques sont utilisées pour évaluer la distorsion.

1. Les méthodes subjectives, nécessitant des tests psycho-visuels de l'oeil humain. Les tests sont réalisés à plusieurs échelles avec des groupes de personnes, et doivent se dérouler selon la procédure des recommandations fournies par la CCIR.
2. Les méthodes objectives qui utilisent le rapport signal crête sur bruit PSNR entre la source initiale et celle distordue.

Soient x_n la valeur initiale du signal et \hat{x}_n sa valeur codée ou distordue, N le nombre d'éléments. L'erreur quadratique moyenne EQM est [TOT07] :

$$EQM = \frac{1}{N} \sum_{i=1}^N (sc_i - s_i)^2$$

avec sc_i , valeur décodée de l'échantillon s_i .

On utilise fréquemment une grandeur dérivée de l'EQM, plus physiologique : le rapport signal sur bruit crête (PSNR, Peak Signal Noise Ratio), défini (en dB) par :

$$PSNR = 10 \log \frac{(\max(s_i))^2}{EQM}$$

Par exemple, si s_i représente la valeur des niveaux de gris des pixels d'une image, codés sur 8 bits, alors $\max(s_i)=255$.

On considère que si le PSNR est inférieur à 30 dB, la compression conduit à une image décodée dont les différences avec l'originale sont visibles à l'oeil nu.

5. Conclusion

L'étude, élaborée dans ce chapitre, nous a révélé la diversité des techniques de compression développées jusqu'à maintenant dans le cas des images statiques ainsi que de la vidéo. Malgré cette variété, ces techniques sont toujours en progrès sans cesse en vue d'assurer une bonne réponse aux applications découvertes chaque jour. Désormais, on utilise des formats connus, mais de manière à ce que l'algorithme le mieux adapter à un segment de document particulier soit utilisé. Ceci est le point clé de notre proposition. Il s'agit d'adapter la technique la mieux adoptée au traitement de flux de données. Certainement, cette technique dépend du domaine d'application du traitement ainsi que du flux de données à traiter lui-même. C'est dans ce contexte que nous présenterons dans le prochain chapitre le domaine d'application retenu dans ce travail : la visioconférence. Nous développons en deuxième partie l'approche proposée.

Chapitre II. Visioconférence et le flux des diapositives : étude et spécifications

1. Introduction

Le système de visioconférence présente l'un des plus importants services multimédia diffusés sur IP (Internet Protocol) qui permet à des interlocuteurs distants d'échanger de l'audio et de la vidéo en temps réel tout en travaillant simultanément sur des documents communs.

Ce système a vécu de grandes évolutions en passant par l'intermédiaire de messages écrits et la transmission de la parole, jusqu'à l'ajout de l'image et de la vidéo.

Toutefois, une telle diversité de données provoque une problématique qui porte sur les méthodes et les techniques à utiliser pour assurer un bon traitement. Pour cela, il faut bien étudier les formats d'information présentes et de les différencier selon leurs caractéristiques. En effet, la tolérance de perte en qualité diffère généralement de la vidéo, du texte et du son : on peut accepter plus de perte en vidéo (représentant la salle de conférence, le conférencier, l'audience) que pour le son (discours du conférencier, questions proposées par les présents) ou le texte (figuré dans les diapositives).

Ainsi, dans ce chapitre, une vision globale sur les systèmes de visioconférence existants sera présentée. En particulier, on s'intéressera à un flux assez particulier dans ce système qui est le flux des diapositives ou transparents. Ce flux est assez spécifique et primordial dans n'importe quelle application de visioconférence, et la mise en place d'une technique adaptée

pour le traiter semble être bénéfique. De ce fait, nous proposerons, à la fin du chapitre, une nouvelle technique qui permet d'apporter une réponse dédiée à deux contextes opposés : une image fixe et une animation, tout en respectant les contraintes de complexité opératoire et de qualité.

2. Contexte général : la visioconférence

Notre travail s'intègre dans l'étude et la conception d'un système de visioconférence sur une plateforme à base de FPGA, permettant, en temps réel, le traitement et la diffusion IP de contenus VGA d'une présentation.

2.1. Positionnement par rapport au projet de l'équipe

Le système proposé (Fig.2.1) est un système de visioconférence complété par une unité de stockage assurant des services complémentaires (Web, moteur de recherche, VoD...). Il se résume en un système permettant, avec une connaissance technique minimale et un minimum d'intervention humaine, l'acquisition, la création et la télédiffusion de contenus audiovisuel tels que des contenus de cours, présentation d'un conférencier etc.

Les configurations du système sont multiples et permettant de différents usages (générique).

Le système peut être utilisé en mode automatique ou assisté.

Il permet de capturer, indexer et diffuser en mode direct ou différé, les sources vidéo et multimédia :

- Le signal vidéo (numérique ou analogique) et son,
- Fichiers textes, images, sources VGA liés à la présentation (transparents).

Le système permet de réaliser, gérer et diffuser automatiquement des pages HTML avec :

- Les fichiers son et vidéo streaming
- Les fichiers médias synchronisés
- Les fichiers texte, HTML, URL synchronisés
- Les chapitres d'accès aux séquences
- L'interface graphique interactive

Le système couvre tout le cycle de production et diffusion des contenus :

- Acquisition de différentes sources audiovisuelles (automatique ou assisté)
- Création traitement et indexation des contenus
- Archive des contenus (CD-ROM, intranet, Web)
- Gestion des contenus en ligne (SGBD)

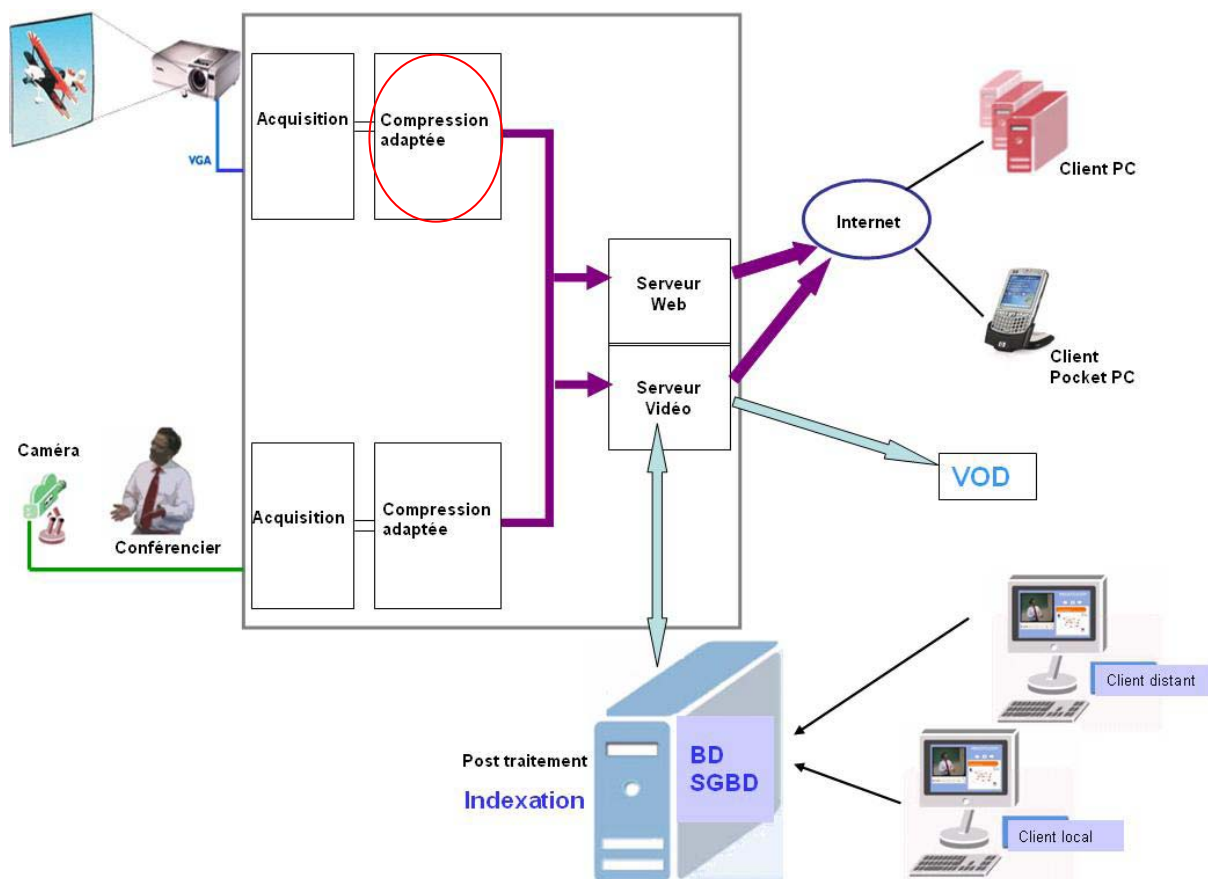


Figure 2.1 : Description globale du système

Pour notre cas, nous intervenons à travers un module de traitement du flux de la présentation du conférencier (bloc en rouge dans Fig.2.1). Cette intervention se manifeste par un ensemble de traitements (acquisition, filtrage, compression en vu de diffusion) qui exploitent les caractéristiques de ce flux. Ces caractéristiques imposent quelques contraintes sur toutes les étapes de spécification de l'algorithme et parfois même sur l'architecture du système global. En effet, une étude exhaustive sur les modules d'acquisition des données, pour une application de visioconférence, est déjà élaborée au sein de notre laboratoire CES. Cette étude a présenté les défaillances et les problèmes que posent les systèmes actuels et a proposé une solution adéquate qui consiste à mettre en place un module d'acquisition d'images [OUN04] permettant la récupération des diapositives directement depuis un signal vidéo analogique ou numérique provenant du PC du conférencier. Ce module se présente sous forme d'une carte à base de FPGA pouvant être intégrée directement dans un PC ou faire partie d'un dispositif électronique permettant la diffusion « on line » et/ou le stockage des données [OUN06].

En tant qu'une première étude de faisabilité de l'implémentation d'un tel algorithme dans un processus de traitement vidéo sur FPGA, ce travail a apporté une réponse satisfaisante à la majorité des usages visés. Cependant, il présente des insuffisances en terme de qualité : en effet, la technique d'acquisition utilisée néglige les effets d'animation, elle traite le flux des diapositives en tant qu'images fixes. Il a donc fallu mettre en question cette technique, et refaire l'étude et la spécification de ce module de traitement (fig.2.2) afin d'apporter une réponse adaptée à ce contexte.

Il convient encore de disposer d'algorithmiques intelligents qui permettent des opérations pertinentes (d'acquisition, de prétraitement, d'encodage, d'acheminement de données, de synchronisation, de gestion de mémoire...) et adaptées à ce contexte d'usage.

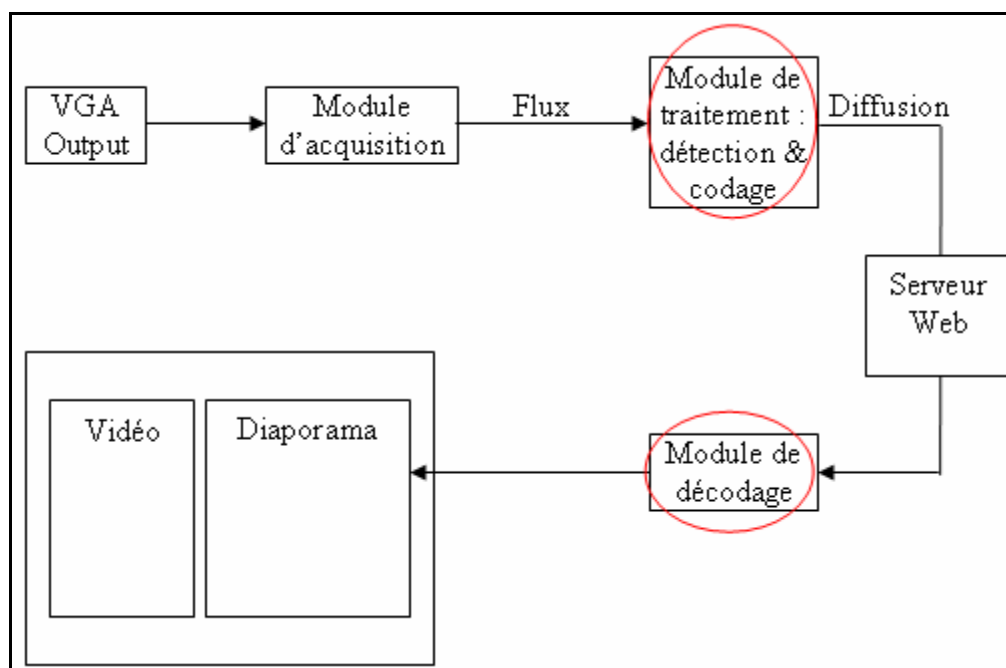


Figure 2.2 : Acquisition et traitement des diapositives

2.2. Traitement des transparents dans les systèmes de visioconférence

En général, un système de visioconférence englobe trois parties principales. La première étape concerne la détection et/ou capture des données présentées selon leurs formats. Un deuxième module est responsable du traitement (encodage/compression) des différents flux d'informations disponibles. La dernière partie s'engage de la collecte de ces flux et de les synchroniser pour les diffuser aux différents clients du système avec une qualité acceptable. Ces étapes dépendent du format de l'information à traiter. En effet, lors d'une visioconférence, plusieurs sources d'informations se présentent : vidéo, son, transparents et

autres. En particulier, la récupération et la diffusion des transparents du conférencier dans un bon état (qualité d'image, lisibilité) et avec un coût admissible (complexité de traitement, bande passante) semblent être très rentables pour les dispositifs de visioconférence.

On va présenter, dans la suite, les principaux travaux actuels dans le domaine de visioconférence, et on va essayer de montrer l'évolution de ces systèmes. Néanmoins, on va marquer, en ce qui concerne le flux des diapositives, l'absence d'une technique adaptée.

2.2.1. Acquisition

Depuis leur méthode d'acquisition du flux des diapositives, les procédures existantes souffrent de plusieurs problèmes. La méthode la plus simple était de récupérer les présentations PowerPoint et les envoyer à l'avance, par email, en vue d'effectuer à l'avance un certain nombre de préparations. En premier lieu, les images de synchronisation seront converties en format JPEG. Ensuite, elles seront redimensionnées pour s'intégrer dans des pages HTML. Enfin, il faut optimiser l'ensemble pour assurer sa cohérence avant sa diffusion à un maximum d'internautes [ERI00]. Ceci n'est pas toujours possible puisque dans la majorité des cas les présentations PowerPoint n'arrivent jamais à temps et on se rend compte qu'elles ont été modifiées, parfois le jour même de la présentation, ce qui rend le traitement de la synchronisation à l'avance inefficace [EMP07].

Pour remédier à ces problèmes, d'autres systèmes utilisent une caméra focalisée sur les transparents projetés par l'orateur. La qualité de l'image des transparents était mauvaise, elle dépend de l'éclairage de la salle de conférence et leur transmission consomme une bande passante importante. Une solution était d'utiliser un banc-titre. La qualité était un peu meilleure mais la bande passante consommée reste toujours importante.

D'autres systèmes utilisent des dispositifs de projection des documents vidéo munis d'entrées composites ou S-Video permettant la connexion d'équipements de lecture, magnétoscopes VHS par exemple. Le flux des transparents traité ici en tant qu'un flux vidéo continu occupe une bande passante énorme et peut souffrir d'une dégradation remarquable au niveau qualité d'image du à la compression.

Actuellement des entrées analogiques dans des formats « informatiques » courants sont également proposées pour le raccordement des micro-ordinateurs. Il s'agit de diffuser l'image écran d'un contenu informatique via une liaison directe de type VGA/XGA entre l'ordinateur et le dispositif de visioconférence qui prendra en charge la diffusion du contenu VGA/XGA (les transparents). Ceci présente l'avantage de réduire le bruit par rapport l'usage d'une

caméra vidéo. Cependant, le flux vidéo diffusé reste gourmand en bande passante et la qualité de la restitution dépendra souvent du type d'affichage utilisé sur le site distant. Avec un téléviseur ou un moniteur vidéo, elle pourra souffrir du transcodage (XGA vers H261/ H263) alors qu'avec un moniteur informatique elle conservera sa définition et sa qualité d'origine [GASS05].

Pour ce flux vidéo ordinaire, on peut se permettre de réduire la bande passante au détriment de la qualité de l'image (réduction de la résolution, forte compression) ce qui pose un problème pour le cas des transparents du conférencier qui doivent garder une qualité pertinente et une lisibilité sure.

2.2.2. Traitement : Image ou vidéo ?

Pour l'encodage vidéo sur IP, la tendance actuelle dans l'industrie s'oriente vers le MPEG qui donne une qualité de vidéo comparable à de la télévision pour un débit IP de l'ordre de 5 Mbit/s [PRE01]. Bien que cette méthode conserve bien les animations présentes dans une présentation, elle représente une solution non adéquate à un tel flux puisque les animations sont locales et non fréquentes (généralement une animation ne dépasse pas quelques secondes et on peut avoir plusieurs transparents qui ne présentent que des images fixes).

En outre, la mise en œuvre d'une telle technique est complexe et difficile à réaliser, exemple le MPEG2 est très encombrant (une heure = près d'un Goctet et cela pour uniquement une cinquantaine de transparents). D'autre part, le codeur vidéo lui-même présente une complexité opératoire énorme comparée par la nature de l'information traitée [PRE01]. Ce qui nous permet de confirmer qu'un bon compromis qualité/coût reste très difficile à atteindre avec les techniques classiques utilisées.

2.2.2.1. Séquence vidéo

Ces méthodes, qui traitent le flux des diapositives sous forme d'une séquence vidéo, font la récupération des diapositives à travers une ou plusieurs caméras. Après l'encodage, ils font recours à des étapes de post traitement pour améliorer la qualité et surtout pour garantir, lors de la diffusion., une bonne synchronisation entre la vidéo et le son du conférencier d'une part, et la présentation d'autre part.

a) Exemple 1

Une méthode présentée dans [MUK99] utilise deux cameras pour la capture vidéo qui sera par la suite numérisée et encodée en MPEG. La première caméra capture une vue d'ensemble

contenant le conférencier et l'écran de projection dans un même plan fixe (Figure 2.3). La deuxième caméra sert à repérer les mouvements du conférencier en capturant sa tête et ses épaules.

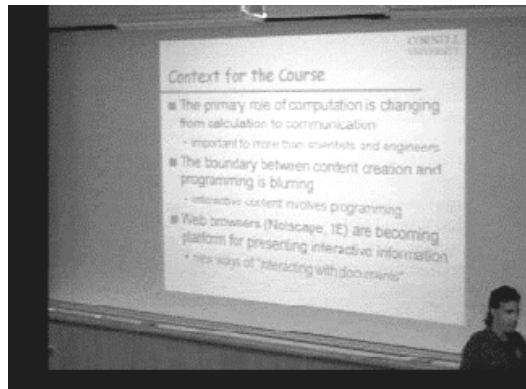


Figure 2.3 : capture de la projection

A la fin de la présentation, la séquence sera transmise à travers le réseau vers un serveur pour le traitement. Le conférencier doit envoyer aussi la présentation électronique vers le serveur.

Le rôle du serveur est de combiner la séquence et les diapositives pour créer une présentation qui peut être consultée par la suite sur les navigateurs.

Pour assurer la synchronisation, l'idée était de faire calculer une fonction $f(t): [0, d] \rightarrow S$, où d est la durée de la séquence vidéo $v(t)$ et $S = (S_1, S_2, \dots, S_n)$ est l'ensemble de n diapositives. Cette fonction est décomposée en deux étapes, comme suit :

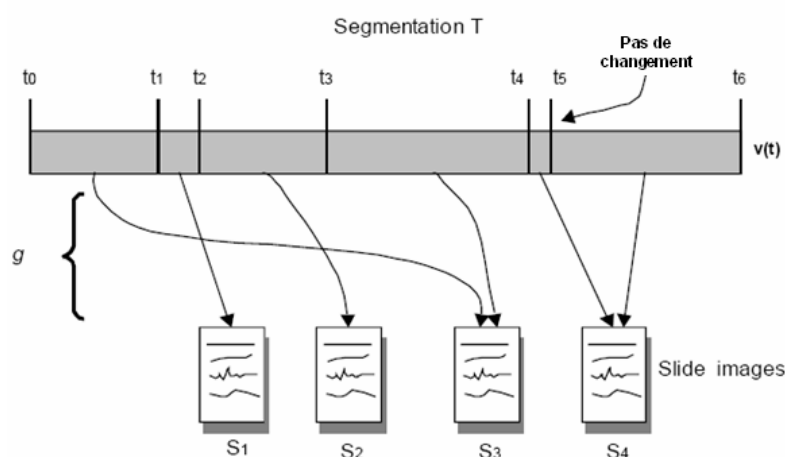


Figure 2.4 : Illustration de la segmentation et la fonction g

- Segmentation : compte tenu d'une séquence vidéo $v(t)$, il faut calculer un ensemble de valeurs de temps $T = (t_0 \dots t_i, \dots t_k)$, tels que les diapositives ne changent pas au cours de l'intervalle $[t(i), t(i+1)]$. Un changement de diapositive est détecté si la différence Δ

entre les deux images successives B1 et B2 dépasse un seuil prédéfini de façon à ce qu'aucun changement de diapositive ne passe inaperçu. Il a été constaté qu'environ un dixième des changements des diapositives signalés étaient faux. La plupart du temps, les fausses détections ont eu lieu lorsque le professeur s'approchait des projections de diapositives pour mieux expliquer par ses bras ou autres.

- Correspondance (matching) : calculer une fonction $g : T \rightarrow S$, tels que, pour tous les i dans $(0, 1 \dots, k-1)$, dans le segment de temps de $t(i)$ à $t(i+1)$, il y aura l'affichage de diapositives $g(t(i))$. La fonction est basée sur la distance de Hausdorff sur les images binaires suite à des phases de segmentation et de seuillage.

Cet algorithme est sensible à la capacité d'adapter le seuillage pour bien distinguer une zone texte de l'arrière-plan. Les paramètres par défaut de l'algorithme de seuillage donnent de bons résultats sur les diapositives qui ont des couleurs de fond très contrastées et avec une couleur d'arrière-plan claire. Pour les diapositives avec des couleurs de texte claires sur un fond sombre, et pour ceux ayant un faible contraste, les paramètres doivent être ajustés manuellement pour atteindre un niveau élevé de précision. Eventuellement cette méthode ne tient pas compte des animations des objets d'une diapositive [MUK99].

b) Exemple 2 :

Dans le but d'implémenter un système de détection de changement de diapositives, une étude très récente [REV08] était basée sur l'analyse de deux ressources d'information :

- Fichier vidéo de la présentation, récupérée directement depuis la sortie VGA du PC (elle ne présente donc ni l'orateur, ni le public)
- Images de la présentation, extraites directement à partir du document original à l'aide de l'API COM de Microsoft (problème de dépendance).

La première étape de ce système consiste à la détection d'un changement de diapositive : si un grand nombre de pixel change entre deux trames de la vidéo, alors un changement a peut-être eu lieu. Cette détection se base donc principalement sur un algorithme de comparaison trame par trame.

L'autre étape concerne la corrélation trame/diapositive et consiste à trouver quelle trame affichée à l'instant correspond à quelle diapositive du document original. Pour cela, l'auteur a présenté deux solutions. La première méthode, simple et répandue, était de comparer les histogrammes de couleurs des images respectives afin de déterminer leur niveau de ressemblance. Les résultats n'ont pas du tout été concluant pour la raison que le contenu de diapositive en général est beaucoup trop uniforme (souvent il n'y a que des changements de texte alors que tout le reste de l'image, le background, reste le même).

L'autre méthode était d'utiliser de la reconnaissance de caractère. Bien que cette méthode peut se révéler effective, elle reste très complexe, du fait qu'il faut effectuer une phase de prétraitement pour extraire les caractères, ce qui n'est pas une chose aisée dès le moment où l'on doit travailler avec des vidéos de basse résolution ou avec des fonds qui ne sont pas unis. En plus, cette méthode ne convient que pour les présentations qui ne contiennent que du texte.

2.2.2.2. Suite d'images fixes

Par ailleurs, d'autres solutions ont été développées pour traiter les transparents en tant qu'une suite discontinue d'images fixes, ceci réduit énormément la complexité de traitement. Le seul inconvénient c'est la perte de l'animation.

a) Exemple 1

Une solution software développée par l'association Aristote, appelée ARTS (Aristote Real Time Slides) [PRE07], est destinée à l'enregistrement et à la transmission en temps réel des transparents d'un cours, d'un séminaire, d'une conférence...

ARTS est capable de transmettre l'image des transparents, en temps réel, par le Web, en excellente qualité : il doit être alors associé à un serveur Web tel qu'Apache. ARTS est également capable d'enregistrer ces images, ainsi que les données annexes correspondantes, au format d'échanges commun de OTESA qui permet ensuite de générer automatiquement le script SMIL qui permet de rejouer le séminaire.

Cet outil est un exécutable JAVA, multiplateformes, qui fonctionne en tâche de fond dans le PC de l'orateur. A intervalles réguliers de N secondes (N paramétrable, habituellement 5 à 7 secondes), il fait une copie d'écran du PC. Si cette image est différente de la précédente (donc si le transparent a changé), il l'encode en JPEG (ou autre format) l'écrit sur disque, et met à jour une page Web de telle sorte qu'elle pointe sur cette nouvelle image. Si cette page est dans le domaine d'un serveur Web, un utilisateur distant peut donc bénéficier aussitôt de l'affichage de cette image, et la page Web se rafraîchit automatiquement grâce à un petit script Javascript qui, toutes les n secondes, regarde dans le serveur Web si la page a changé.

Une telle solution traite les transparents en tant qu'une suite discontinue d'images fixes, permet de réduire la complexité de traitement. Le seul inconvénient c'est la perte de l'animation.

b) Exemple 2

Le but du projet AVRiL est d'automatiser entièrement l'enregistrement vidéo des conférences de sorte qu'il n'y aura pas besoin de la présence d'une équipe de caméraman dans la salle ou d'une direction de l'équipage. En fait ceci est fait grâce à l'utilisation des caméras PTZ qui

visent intelligemment sur le plus important à ce moment-là que ce soit l'instructeur, la présentation, l'estrade ou les étudiants [AVR06].

En ce qui concerne les diapositives, le projet englobe un module appelé « Slide Transition Detection ». Ce module est responsable de la détection des transitions dans la présentation. Il consiste à enregistrer les transitions et faire des captures d'écran des diapositives à chaque transition. A l'entrée, il reçoit les diapositives en format PowerPoint et les événements qui déclenchent les transitions. En sortie, il fournit l'horodatage des transitions et des captures d'écran de diapositives à cet instant. Ceci grâce à un plug-in intégré avec Microsoft PowerPoint. Ce plug-in sert à sauvegarder le temps des transitions des diapositives.

Ce système englobe d'autres modules tels que :

- Module de mixage : intègre des règles pour la gestion de la diffusion des différents flux vidéo et PowerPoint afin de produire une vidéo de haute qualité.
- Encodeur : prend en entrée un flux vidéo brut et l'encode dans un format de sortie convenable.
- Module de présentation : prend la vidéo encodée et les diapositives et les synchronise en utilisant les informations d'horodatage.

2.3. Réflexions et synthèse

Suite à cette étude des systèmes existants, nous constatons qu'il y a deux tendances pour traiter ce flux : celle considérant un diaporama comme une suite d'images fixes, et celle le considérant comme une séquence animée.

Dans le premier cas, la technique de codage utilisée est le JPEG. Ces systèmes ont prouvé leurs profits à partir des avantages propres à ce standard tel que la simplicité de mise en oeuvre (ce qui peut réduire le coût de développement et d'implémentation) ainsi que la simplicité pour le montage et l'édition [SSI06]. Dans le deuxième cas, le format de compression est le MPEG. Le MPEG a démontré ses avantages envers le traitement des séquences vidéo vu qu'il permet de réduire la quantité de stockage nécessaire et la bande passante utilisée [SSI06].

Cependant, malgré les avantages de ces deux techniques, elles souffrent encore de problèmes vu qu'elles ne tiennent pas compte des spécificités du flux traité. En fait, la première méthode, malgré sa simplicité, élimine les animations trouvées dans les présentations. L'utilisation de MPEG, dans la deuxième méthode, fait augmenter la complexité du traitement pour un flux assez simple, et évoque d'autres problèmes pour une diffusion synchrone et automatique.

Le tableau suivant résume les différentes techniques utilisées :

Table 2.1 : Techniques de compression du flux des transparents

	Type de flux	Qualité des diapos	Sensibilité à la lumière	Bande passante	Automatique	Nombre de trames envoyées	Résolution
Systèmes basés sur l'acquisition à partir d'une caméra	vidéo	médiocre	OUI	Elevée	Sans intervention sur le PC	25	CIF 352 x288
Systèmes basés sur l'acquisition à partir d'une connexion XGA	vidéo	Bonne	NON	Elevée	Sans intervention sur le PC du conférencier	>70	QCIF
ARTS	Images (non animées)	Moyenne	NON	faible	Avec intervention	0.5	1600x1200
Solution proposée (CES)	Images (animées)	Bonne	NON	faible	Sans intervention	1 chaque 30 sec en moyenne	1280x1024

3. Etude et spécification

3.1. Caractéristiques d'une présentation multimédia

La notion de présentation multimédia est devenue de plus en plus répandue et utilisée dans les différents domaines, par exemple l'enseignement, la publicité, les présentations commerciales, etc. Un système de présentation multimédia est un système informatique intégré caractérisé par sa capacité à traiter des informations exprimées dans plusieurs média, comme le son, la vidéo, l'image, le graphique, le texte, etc. [SAB99].

Pour garantir une bonne mise en œuvre d'un tel système, il faut bien répondre aux exigences suivantes : une grande capacité de stockage, une grande rapidité de traitement des données multimédia ainsi qu'un réseau à haut débit qui rend possible l'échange de l'information multimédia entre des sites interconnectés, notamment sur le web.

Dans cette section, on s'intéressera aux principales propriétés qui caractérisent une présentation multimédia dans le cadre d'une visioconférence. Les caractéristiques identifiées ci-dessus doivent être prises en compte par le système de traitement.

3.1.1. Hétérogénéité des objets média

Les objets média qui font partie d'une présentation multimédia ont des caractéristiques hétérogènes. Celles-ci peuvent être classées selon un certain nombre de critères dont les plus importants sont [SAB99] :

a) Le mode de perception de l'information par l'utilisateur

Il spécifie la manière selon laquelle l'utilisateur perçoit les objets média. Ces derniers peuvent être audibles (comme un message audio ou un clip de musical) ou visibles (comme un texte, une image, une animation ou une vidéo).

b) Le comportement temporel

Il caractérise la façon selon laquelle les objets média se comportent temporellement pendant leur présentation. On peut identifier trois types de comportements temporels:

- Comportement discret, pour les objets n'ayant pas de notion de durée temporelle intrinsèque comme une image fixe ou un texte ;
- Comportement continu, pour les objets constitués d'un flot continu de données et caractérisés par une vitesse spécifique de restitution, comme une vidéo ou une audio ;
- Comportement indéterministe, pour les objets ayant des durées qui ne peuvent être déterminées qu'à la fin de leur présentation. C'est par exemple le cas d'un bouton d'interaction utilisateur, puisque l'on ne peut savoir à quel moment l'utilisateur va appuyer sur le bouton.

c) Le format de codage

Des différents types de format de codage ont été définis pour compresser le contenu d'objets média afin de limiter l'espace de stockage utilisé, comme les formats MPEG, AVI et QuickTime pour la vidéo, les formats AU, AIFF, MPEG et WAV pour l'audio, et les formats GIF, JPEG, PNG et Bitmap pour les images. Ces formats font associer aux objets média un ensemble d'attributs qui spécifient le style de présentation pour ces objets. Par exemple les formats WORD, LATEX et Postscript permettent de spécifier entre autres la couleur, la police et la taille de caractères d'un objet texte.

d) L'adaptabilité

Ce critère définit la capacité des objets média à répondre aux exigences de la qualité du service rendu. Les objets tels que les clips vidéo et les images peuvent être adaptables (selon le codage utilisé) puisqu'ils tolèrent des dégradations dans leur qualité de présentation (par exemple réduire le nombre de couleurs utilisées) en fonction des ressources système disponibles. D'un autre côté, les objets comme le texte et le son ne sont le plus souvent pas

adaptables aux exigences de la qualité de service puisqu'ils ne tolèrent aucune perte de leur contenu ou de leurs données.

e) Le volume

Les objets média sont caractérisés par leur taille volumineuse qui exige une grande capacité de stockage et de transmission. Pour résoudre ce problème, de nombreux travaux ont porté sur la définition d'algorithmes de compression pour différents types d'objets. Le taux de compression obtenu dépend de l'algorithme appliqué, du type des objets média (image, vidéo, audio, vidéo/audio, etc.), et de la relation entre les unités successives d'un objet média (par exemple la répétition des informations dans les images successives d'une vidéo ou dans les échantillons successifs d'une audio).

La nature variée des objets média que nous venons d'identifier doit être prise en compte au sein des systèmes multimédia, ce qui induit de nouvelles difficultés techniques.

3.1.2. Qu'est-ce qu'un diaporama ?

Un diaporama est une succession de diapositives accompagnant généralement une conférence. Les logiciels de présentation, comme l'exemple de Powerpoint de Microsoft, permettent de créer ces diaporamas ainsi que des documents associés.

Exemple de présentation : Les diapositives s'enchaînent pendant l'exposé de l'orateur

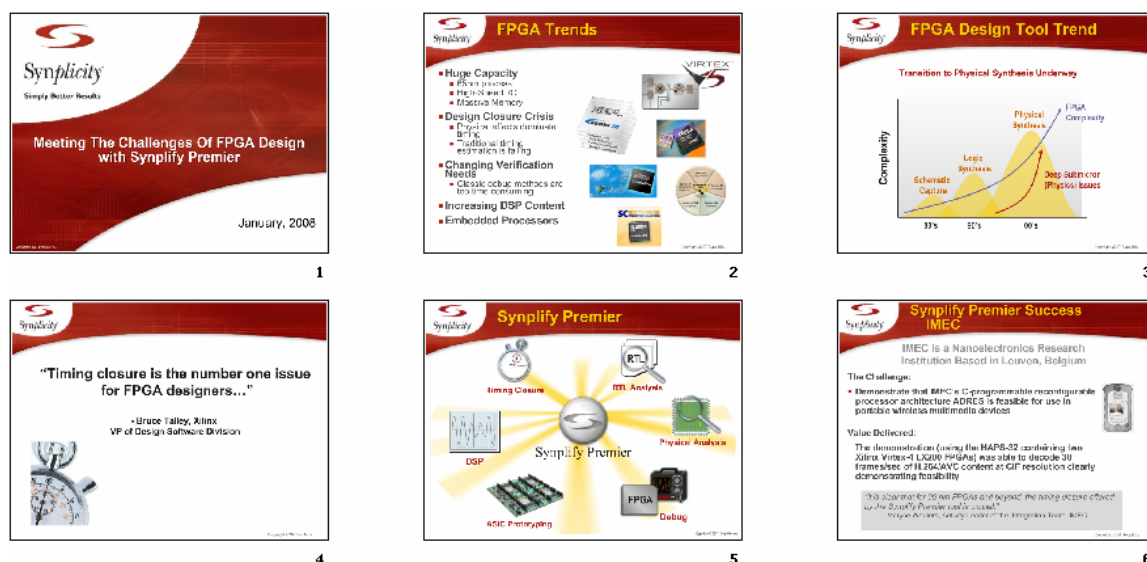


Figure 2.5 : Exemple de diapositives

Le diaporama peut être animé à deux niveaux :

- Les transitions qui caractérisent l'enchaînement des diapositives.
- Les animations qui s'appliquent aux différents objets de la diapositive.

On peut définir les caractéristiques d'un diaporama en fonction de certaines contraintes et des objectifs à satisfaire :

- Public concerné, support d'affichage (projecteur vidéo, transparent...),
- Contenu de l'exposé et message à faire passer,
- Formalisation d'une présentation (choix du nombre et du contenu des diapositives, mise en page...),
- Gestion du rythme de la présentation (durée, répartition des éléments entre les diapositives, choix des animations manuelles ou automatiques...).

3.2. Points d'intérêt

Le flux de transparents est formé par (figures 2.6) :

- Images fixes : suite de trames identiques représentant une image fixe ou une diapositive qui apparaissent à un temps spécifique et restent affichés pour une durée définie.
- Animations : séquence contenant des trames différentes formant une séquence animée. Les changements dus à l'animation peuvent être localisé dans quelques régions seulement.

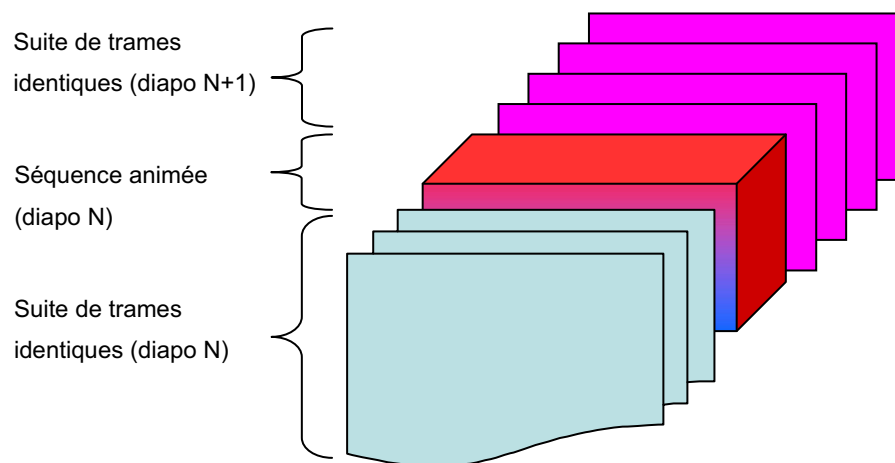


Figure 2.6 : Structure d'un flux de transparents

Pour avoir un bon traitement, ce flux présente plusieurs points d'intérêts à prendre en considération, dont on cite :

- Spécificité du flux des transparents : généralement un diaporama est une suite de trames représentant des images fixes ou diapositives (Figure 2.7). Une comparaison

entre trames permettra d'éliminer les trames identiques et gagner en traitement et en débit de transmission.



Figure 2.7 : Exemple de texte animé dans un PowerPoint

- Modifications au niveau textes : une comparaison pixel par pixel avec un seuil peut ne pas être efficace : un changement de texte n'affecte pas plusieurs pixels.
- Modifications au niveau images (des objets)
- Modification au niveau couleur : les composantes couleurs comptent aussi bien que la luminance contrairement à la vidéo, donc travailler dans l'espace de couleur RGB semble être plus intéressant que de travailler sur l'espace YUV utilisé généralement dans la vidéo.
- Bruit (acquisition en temps réel).
- Taux de modifications très variables : le choix du seuil de comparaison doit prendre en considération le style de modification produit (apparition de grand/petit objet, apparition de texte, animation lente/rapide, transition à une nouvelle diapositive...)
- Qualité de l'image (résolution, couleur, clarté des diapositives, lisibilité).
- Choix de la technique de compression (avec ou sans perte ou hybride).
- Choix de la fréquence de capture (à des intervalles réguliers ou selon un événement).

Ainsi, notre objectif visé est d'assurer une bonne détection des changements (éviter les fausses détections suite à un bruit), une meilleure compression adaptée aux formats d'informations que contient une diapositive et également garantir une bonne qualité visuelle à la réception (après décodage). Ce dernier point, nous a révélé l'importance du texte présenté. Ce texte doit donc être traité séparément et avec soin pour qu'il reste toujours lisible au récepteur.

De ce fait, on vient de définir trois points à étudier pour notre application : la détection de changement des diapositives, la détection et l'extraction des texte et la compression des données.

3.2.1. Méthodes de détection de changement de plan

Un des thèmes particuliers qui sera abordé dans ce projet est la comparaison d'image pouvant présenter de grandes différences en terme de résolution et de bruit. Les algorithmes utilisés dépendent beaucoup du type d'image à analyser ainsi que du but que l'on veut atteindre.

La plupart des méthodes de détection de changement de plan peuvent se regrouper en plusieurs catégories ([IDR97], [AHA96], [BRU99]) :

- a) méthodes de différence temporelle, consistent à localiser la région en mouvement par un seuillage d'une trame de différence qui contient les valeurs absolues des différences (pixel par pixel) entre deux ou trois trames successives. Ces méthodes se caractérisent par la simplicité de l'implémentation mais elles sont assez sensibles au bruit et aux mouvements importants dans la scène [LEF03]. Ceci peut être inadéquat pour le cas d'une séquence vidéo qui filme une scène ordinaire à partir d'une caméra. Mais pour notre cas, nous utilisons un flux acquis directement d'une sortie analogique et nous tenons bien compte des animations. Cependant, il faut prendre en compte du problème du bruit.
- b) Comparaison d'histogrammes. Cette méthode de détection est basée sur la comparaison de l'histogramme couleurs des images successives. Quand un changement important dans l'histogramme de couleur est détecté, c'est-à-dire une différence supérieure à un seuil préfixé, un changement de diapositives est signalé. L'inconvénient de ces méthodes est lié à la non-prise en compte de l'information spatiale contenue dans les images. En effet, dans une approche développée, ils ont utilisé l'histogramme couleurs ainsi que l'histogramme sur les niveaux de gris afin de détecter les changements de diapositives. L'histogramme des couleurs a prouvé un bon rendement pour des diapositives successives ayant différentes couleurs d'arrière-plan [MUK99]. Toutefois, réellement, la plupart des diapositives ont le même arrière-plan, correspondant généralement à un motif de conception. Dans ce cas, seul le texte et le contenu graphique de la présentation qui varient. En outre, les histogrammes sont invariants dans le cas des animations tel que les rotations, variation de l'angle de vue,

le zoom, etc [BOR96]. Ainsi, la technique d'histogramme n'est pas adaptée pour notre cas d'étude et semble moins sensible que la première méthode.

- c) Une autre technique qui est appelée double comparaison a été proposée par Zhang, Kankanhalli, Smoliar, et Tan (1993). Cette technique utilise deux seuils, un pour détecter les coupures et l'autre à détecter les cadres de départ pour la transition progressive. Malheureusement, cette technique fonctionne sur des données et non son inefficacité est le principal inconvénient.
- d) Estimation de mouvement. C'est une quantification du mouvement simple, par vecteurs de translation, orientés pixel, bloc ou objet, ou plus complexe, mettant en œuvre des méthodes de calcul de trajectoire dans des systèmes bouclés ou non. Les applications de l'E.M. sont surtout la réduction de la redondance temporelle pour la compression et l'analyse de scène. Quatre méthodes reviennent principalement dans l'E.M [BRA05] : la mise en correspondance de blocs, le filtrage spatio-temporel (avec ou sans compensation), la mesure de champ dense ou flux optique et la non transmission de l'information de mouvement. L'estimation de mouvement implique une complexité de calcul conséquente ce qui la rend excessivement coûteuse en temps de calcul. Cela peut prendre jusqu'à 80% des ressources matérielles de la machine. Une estimation de faible qualité est souvent source de problèmes importants comme par exemple des effets de blocs visuellement gênants. Ainsi, depuis plus de vingt ans, son étude reste un sujet de recherche important [GAR04].

On peut citer aussi d'autre méthodes trouvées dans des projets de recherche ou même des brevets :

- e) Méthode de Fribourg. Au lieu d'essayer de détecter les changements des diapositives, cette méthode est reposée sur la recherche de la stationnarité des diapositives, c'est-à-dire détecter les périodes « stables » et « instables » plutôt que les changements dans une séquence d'images à partir de la vidéo du diaporama. Chaque période stable correspond à une page du document projeté qui reste affichée tandis qu'une période instable représente soit une transition d'une page à une autre, soit une animation dans la présentation. L'hypothèse est qu'il n'y a pas de changement de diapositives pendant les 2 secondes qui suivent la détection d'un changement de diapositives. En effet, en cas réel d'une présentation, les diapositives qui sont visibles moins de 2 secondes ne doivent pas être considérées parce que les gens n'ont pas le temps de les lire [BEH04]. Ainsi, l'algorithme commence par subdiviser la séquence vidéo du diaporama dans plusieurs files d'attente de N trames. Ensuite, toutes les trames dans la file d'attente

sont converties en images binaires et sont traitées afin de déterminer si la file d'attente est stable ou non. La première image dans la file d'attente est ensuite comparée avec le reste des trames afin de calculer une valeur statistique, c'est-à-dire une combinaison entre la distance moyenne, la variance et l'écart-type. Si cette valeur dépasse un seuil préfixé, la file d'attente est considérée comme instable et elle est analysée afin de déterminer exactement la position de changement de la diapositive. Juste après un changement de diapositives, une transition d'apparition ou de disparition peut générer de chevauchement d'images dans les diapositives, c'est-à-dire la combinaison de deux images successives de diapositives. Cela est dû au débit relativement élevé du flux des trames de la vidéo acquise. En outre, après un changement de diapositives, et en raison du problème d'auto-orientation des webcams (autofocus: qui mettent environ 0.5 secondes pour capturer une image stable après un changement dans le document projeté), la différence de distance se stabilise progressivement. Pour avoir la position exacte du changement de diapositives, il faut chercher la trame pour laquelle la valeur de dissemblance soit la plus proche de la valeur moyenne, c'est-à-dire la moyenne du minimum et le maximum de la distance de dissemblance dans la file d'attente. De cette façon, on aboutit à une récupération exacte de la nouvelle image ou la première image de la file d'attente. Enfin, en raison de l'instabilité des images au cours de la période de transition, souvent l'image extraite pour l'identification ne correspond pas à la position exacte du changement de diapositives [BEH04].

- f) Un travail breveté [BOR07], semble être plus proche à notre contexte et à notre méthode, a proposé un algorithme de détection de diapositifs dans un flux vidéo par traitement de blocs. En effet, l'idée s'est basée sur le traitement de la trame de différence crée suite à une comparaison des pixels d'une première trame par rapport à une deuxième (pour le canal R, G et B respectivement). Cependant, à la différence des autres méthodes présentées à l'état de l'art, au lieu de traiter individuellement chaque pixel du trame, un bloc de déviation standard pour l'image de différence est calculé en utilisant par exemple des bloc de taille 3x3, 5x5 et 8x8. Les valeurs max et moyenne des valeurs de déviation standard de la trame en cours sont calculés pour avoir un facteur de seuil. Si la différence cumulative (différence entre la moyenne des valeurs max et la moyenne des valeurs moyennes) est supérieure à zéro, alors l'image est seuillée en utilisant la valeur de seuil. Les images binaires trouvées après seuillage (obtenues à partir de chaque composante de couleur séparément) seront combinées par

un ET logique. En fin, une région « heuristique » est performée pour extraire la forme exacte de l'objet.

Le Tab.2.2. [BOR07] et [DEL06] résume quelques méthodes utilisées :

Table 2.2 : Différentes méthodes de détection de changement de plan

Méthode de détection de changement de plan	Commentaires
Méthode de différence temporelle	(+) Pour les petits déplacements (+) Simple (-) Sensible au bruit (-) Sensible aux mouvements importants
Méthode de comparaison d'histogramme	(-) Non prise en compte d'info spatiales
Méthodes d'estimation de mouvement	(-) Couteuses en temps de calcul (80% des ressources) (-) Sensibles aux changements d'intensité
Méthodes d'extraction de BGND et FGND	utilise plusieurs techniques (tel que approche probabiliste...) (+) populaire pour le cas des arrières plan statiques (-) doit faire parcours à une période d'apprentissage (-) le changement brusque de la lumière durant le temps peut affecter le résultat

3.2.2. Méthodes de détection de texte

Dans une image ou une séquence vidéo, le texte joue un rôle important pour les applications d'indexation et d'analyse sémantique. Habituellement, les étapes fondamentaux pour une analyse basée sur le texte consiste à : la détection du texte, la localisation, la poursuite, la segmentation et la reconnaissance [QIA07].

Pour notre application, on fait recours à une détection du texte pour le localiser et le traiter (coder) séparément par un codage sans perte afin de pouvoir le restituer en bonne qualité. Etant donné que l'accès et l'exploration rapides des bibliothèques vidéo sur Internet est devenu de plus en plus requis dans plusieurs domaines tel que la visioconférence, l'E-learning la vidéo en demande...[QI00], on vise aussi à s'en servir de cette étape pour faire l'indexation des diapositives et des présentations.

Le paragraphe suivant sera consacré à une étude sur quelques méthodes les plus utilisées dans ce contexte, en se basant sur des caractéristiques de la présentation d'un texte.

a) Propriétés d'une région texte

- Contraste entre le texte et l'arrière-plan : Le contraste est très important pour rendre un texte lisible. Généralement une zone texte demande un contraste élevé et une luminosité stable. Le grand problème pour la localisation de texte est le fait d'avoir un contraste bas avec un arrière-plan complexe ce qui rend la détection presque impossible. Dans ce cas, il faut faire recours à quelques outils de pré-traitement [LEO05].
- Cohésion spatiale : Cette caractéristique dépend des aspects géométriques, tel que : la typographie (ou les polices de caractères utilisés), la taille, longueur des mots et des phrases ainsi que l'espacement, la compacité (relation entre les pixels appartenant à une lettre et ceux appartenant à l'arrière-plan), et la direction du texte (généralement horizontale).
- Texture : Considérant le texte comme un objet, il a assez de traits pour être détecté comme une texture. Un problème peut apparaître quand la texture de l'image et les caractéristiques du texte sont semblables, comme le cas des feuilles d'un arbre.
- Homogénéité de la couleur : Généralement les caractères sont monochromes. Quelques méthodes prennent l'homogénéité de la couleur comme le caractère principal, parce que la segmentation selon la couleur conserve les contours mieux que, par exemple, la segmentation selon le contraste qui peut brouiller les bords. Les caractères polychromes peuvent aussi être détectés, mais ils sont en rapport avec les aspects artistiques plus qu'au but instructif, donc les auteurs ont tendance à abandonner ce genre de caractères.
- Redondance : Les gens ont besoin de temps pour lire une phrase. Cela veut dire que, dans une seconde et pour 25 trames, le texte sera reproduit tant de fois pour rendre la phrase lisible. Les recherches dans le domaine de la vision ont montrées que l'être humain a besoin de deux à trois secondes pour pouvoir comprendre ou traiter une image. L'uniformité temporelle n'affecte pas seulement le temps de visualisation du texte mais aussi la variation de sa taille et ses mouvements dans la séquence, qui ne doivent pas changer brusquement d'une trame à une autre [LEO05].
- Mouvement du texte : ceci décrit les comportements du texte les plus rencontrés : texte statique (les caractères ne présentent aucune mouvement), texte défilé (mouvement linéaire) et le texte volant (peu rencontré, le texte se déplace dans tout l'espace ce qui le rend plutôt attractif qu'informatif).

b) Classification des méthodes

Les algorithmes peuvent être classés dans deux catégories, ceux qui travaillent sur le domaine compressé et ce qui travaillent sur le domaine spatial. Chaque méthode prend en considération

des caractéristiques différentes selon le type de séquence (par exemple événement de sport ou journal) et sa qualité [LEO05].

- Domaine de la vidéo compressé :

Dans ce cas, l'algorithme ne traite pas directement les blocs de l'image mais leurs coefficients après la DCT. Les coefficients DCT représentent la périodicité spatiale et directionnelle. Les coefficients AC des harmoniques horizontales montrent l'intensité des variations horizontales, et donc, ils seront plus grands dans le cas d'une ligne de texte. D'autre part, Les coefficients AC des harmoniques verticales montrent l'intensité des variations verticales, ce qui informe, dans le cas où ils sont plus grands, la présence de plusieurs lignes de texte.

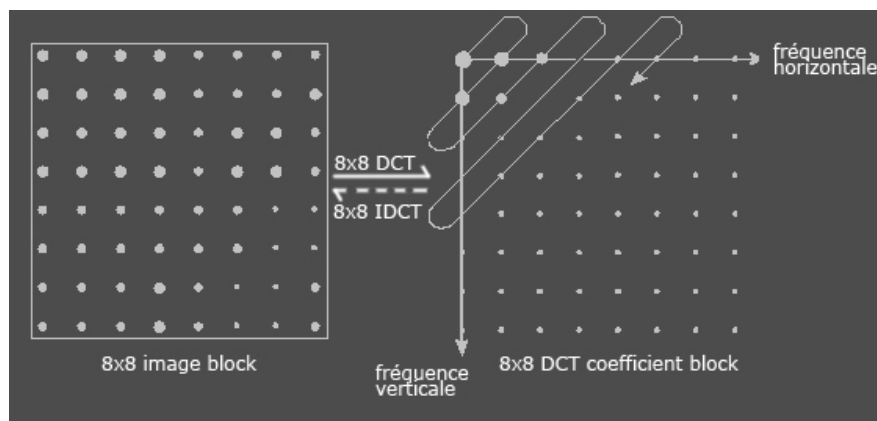


Figure 2.8 : Les Coefficients DCT

Dans ce cas, on peut citer les méthodes d'apprentissage tel que réseau de neurones [CHE04], la machine à vecteurs de supports (support vector machine) ou un perceptron multicouches [CHE01] qui demandent beaucoup de calcul vue la variabilité de la taille et des valeurs de niveaux de gris des caractères. Il y a des travaux basés sur la logique floue [TAN02]. On trouve aussi la méthode multi-frame adoptée pour réduire l'influence des arrière-plans complexes sur la détection des régions texte [WAN04], mais elle représente une faiblesse envers les textes animés. Dans [NGO05], on trouve une autre méthode, basée sur la complexité des arrière-plans, qui fait recours à plusieurs opérations de changement et de lissage pour enlever progressivement les bords des régions non-texte.

Bien que les coefficients DCT des blocs 8 x 8 aient quelques limitations dans le cas des textes de grandes tailles, ils prouvent leur efficacité dans la détection des textes utilisés actuellement dans les vidéos [YE07].

- Domaine spatial :

Ces méthodes exploitent les valeurs et les positions des pixels et sont classées ainsi selon les caractéristiques de l'image :

- Méthodes basées sur les composants du texte. Elles supposent que les régions de texte sont homogènes et ont des couleurs constantes et satisfont certaine dimension, forme, et contraintes de l'alignement spatiales. Cependant, ces méthodes ne sont pas efficaces quand le texte a des couleurs semblables avec ceux de l'arrière plan.
- Méthodes basées sur la détection des contours : Les régions de texte sont détectées sous la supposition que les contours de l'arrière plan et des objets graphiques sont plus clairsemés que ceux des régions de texte. Cependant, ce genre d'approches n'est pas très efficace pour détecter des textes avec grande taille de caractères.
- Méthodes basées sur la détection des textures : Elles utilisent la propriété que le texte dans les images a des caractéristiques texturales spéciales qui lui distinguent de l'arrière-plan. Bien que ces méthodes soient comparativement moins sensibles aux couleurs de fond, ils ne peuvent pas différencier les zones textes des arrières plans qui ont des caractéristiques semblables à un texte.
- Méthodes basées sur la corrélation : Ces méthodes utilisent différentes manières de calcul de corrélation pour décider si un pixel appartient à un caractère ou non.

4. Approche proposée

La spécification de notre technique s'est basée sur quelques hypothèses qui sont en relation avec le flux des diapositives qu'il traite. En fait, la compression d'un tel flux particulier à l'aide d'une technique de compression existante semble être inefficace et inadéquate.

Notre démarche était de partir d'une analyse des spécificités du flux des diapositives pour aboutir à un traitement simple et efficace. Ce flux demande un traitement particulier vu qu'il englobe des données hétérogènes : image fixe et animation. Notre algorithme devra donc assurer une bonne compression des images fixes, l'élimination des redondances et la détection des animations.

Notre système contient différents blocs de traitement : la phase d'initialisation (permettant la construction de la première image de référence), la comparaison entre images pour signaler la présence d'un changement, la différenciation entre les régions changées (ceux appartenant à une zone texte et les autres), l'encodage sélectif des blocs, leur transmission et la mise à jour d'une image de référence. Durant cette section, nous détaillons le processus logique de notre approche donné par la figure suivante :

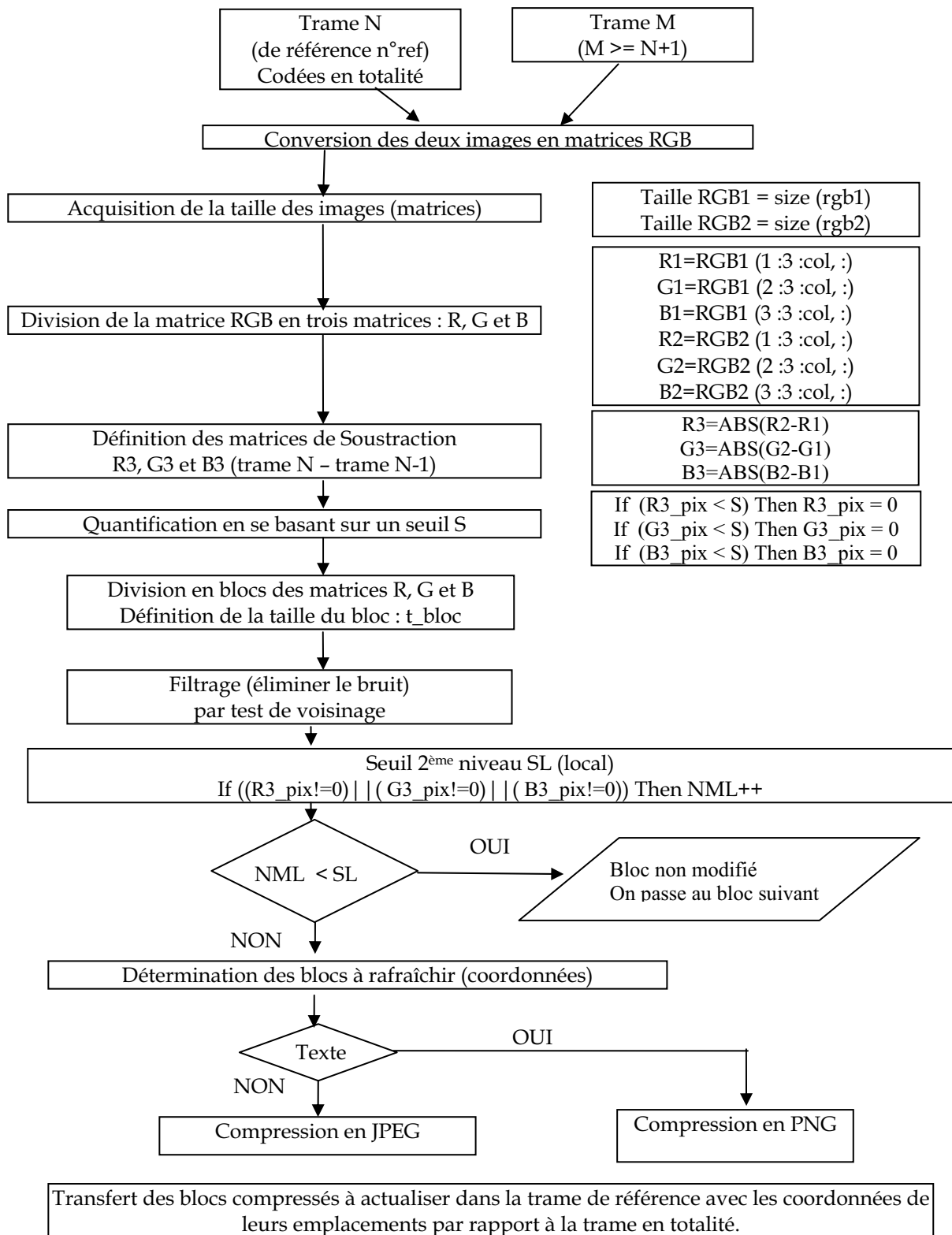
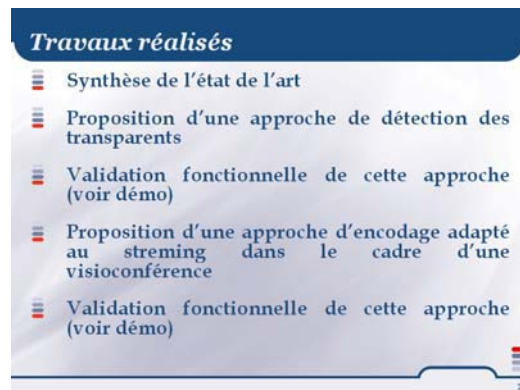


Figure 2.9 : Diagramme synoptique de l'algorithme

4.1. Phase d'initialisation

La première étape consiste à faire l'acquisition de la première trame, qui doit être compressé et envoyée en totalité à la destination. Cette première trame sera utilisée comme référence pour les étapes de décodage.

La trame de référence correspond à la première image du début du traitement ou à une nouvelle image de référence (après un nombre de changement).



Trame N-1 de taille M x N: (référence)

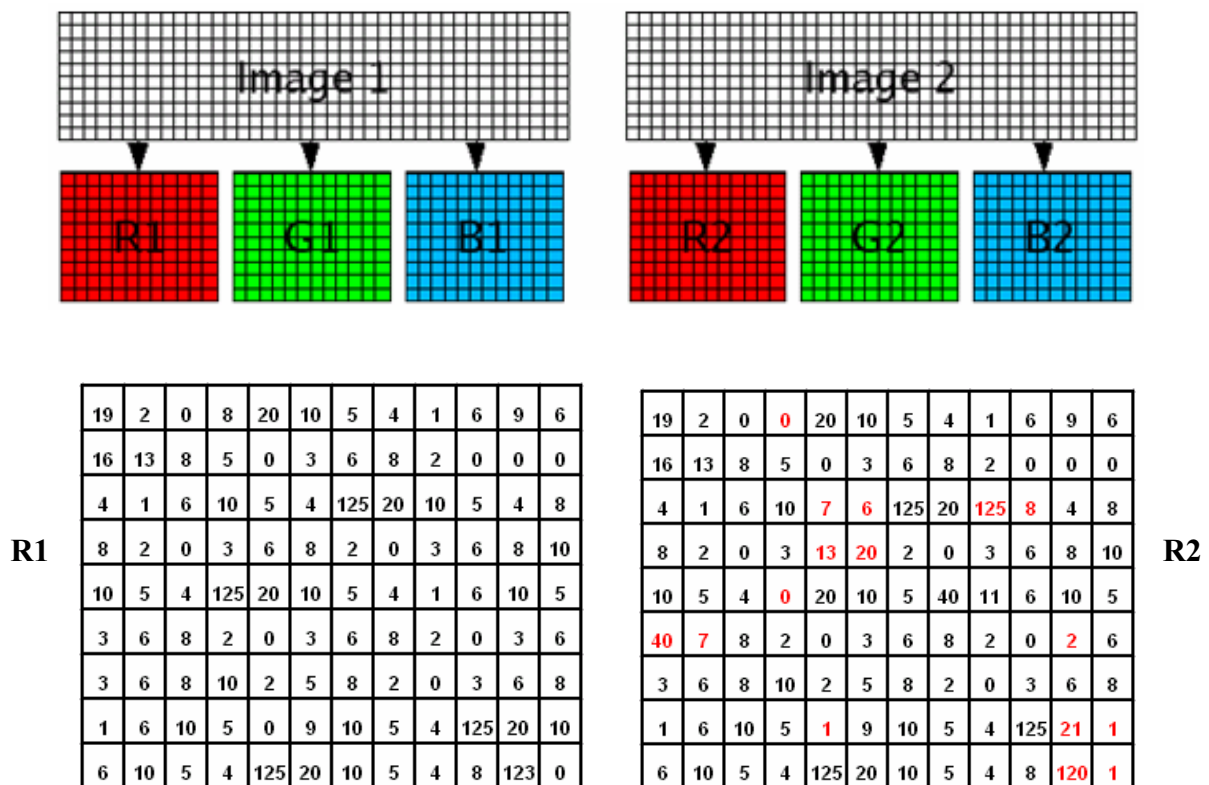
Suite à une détection de changement dans les signaux acquis, on fait une deuxième acquisition de la trame N.



Trame N (en cours) de taille M x N

- Extraction des composantes couleurs dans l'espace RGB.

Généralement, l'œil humain est plus sensible à la composante luminance qu'aux composants de chrominances. Mais dans le cas particulier des diapositives, le jeu des couleurs est très important, et les différences des couleurs seront plus perceptibles. Pour cette raison et par souci de garder le plus de clarté aux diapositives, nous avons choisi de travailler avec l'espace de couleurs RVB (privilégier les composantes couleurs).



A la fin de chaque cycle de traitement, on revient à la phase d'initialisation où on sauvegarde la dernière trame détectée changée et on l'utilise comme référence pour la comparer ensuite avec la nouvelle trame acquise. Cette idée a pour objectif de garantir la détection de changement même pour les lentes animations et pour les animations des objets de petite taille.

4.2. Module de détection et localisation du changement

Pour la détection de changement de diapositives, nous avons choisi la méthode de différence temporelle vu sa simplicité et son adaptation pour le cas des petits déplacements. Ainsi, après acquisition des deux trames N et N-1, on fait créer une trame de différence. Les éléments de chaque matrice de différence seront arrondis (quantifiés) en se basant sur un seuil S. Autrement dit, si la différence ne dépasse pas un seuil préalablement fixé, les images seront jugées identiques ; les faibles différences ne sont pas perceptibles par l'œil.

Par la suite, on fait compter le nombre de pixels modifiés (NM) et le comparer par rapport à un seuil (SG appelé seuil global, SG = 0,1% du nombre total des pixels).

0	0	0	8	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	2	2	0	0	115	3	0	0
0	0	0	0	7	12	0	0	0	0	0	0
0	0	0	125	0	0	0	36	10	0	0	0
37	1	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	1	9
0	0	0	0	0	0	0	0	0	0	3	1

Matrice de différence

(M x N)

$$R = \text{abs}(R1-R2)$$

0	0	0	8	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	2	2	0	0	115	3	0	0
0	0	0	0	7	12	0	0	0	0	0	0
0	0	0	125	0	0	0	36	10	0	0	0
37	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	9
0	0	0	0	0	0	0	0	0	0	3	0

R quantifiée

(exp : seuil S=1)

Cette première comparaison globale permet de court-circuiter les étapes suivantes inutiles en cas d'égalité parfaite ou quasi égalité entre la trame en cours et la trame de référence.

Cependant, dans certains cas, il se peut que NM (nombre de pixels modifiés) soit supérieur à SG (seuil global) alors que nous n'arrivons pas à détecter une différence pertinente entre les deux trames N et N-1, et les différences des valeurs de pixels détectées sont dues aux bruits ou aux erreurs d'acquisition.

Exemple : SG = 2

Dans la première matrice de différence, NM > SG, mais les deux trames comparées sont considérées identiques, et les valeurs non nuls détectées dans cette matrice ne représentent pas une différence pertinente.

0	0	0	0	0	0	0	0	0	0	0	0
50	0	0	0	0	0	0	0	75	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	75	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

Matrice 1

Dans ce cas, les différences détectées seront éparpillées n'importe où dans la matrice de différence.

Dans le cas contraire, ces valeurs seront groupées (matrice 2). On calcule $NM > SG$ et on détecte ainsi une région localement modifiée et les deux trames comparées seront considérées différentes.

0	0	0	0	0	0	0	0
0	0	0	0	0	0	75	50
0	0	0	0	0	0	75	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Matrice 2

De ce fait, nous avons modifié cette méthode en effectuant une comparaison, dite locale, où on fait calculer le nombre de pixels modifiés dans chaque bloc. En effet, cette méthode d'échantillonnage par bloc des images est utilisée en vue d'améliorer la qualité de détection des changements. Elle permet un traitement qui est intermédiaire, entre le niveau local, comme le cas des méthodes basées sur pixels, et du niveau global comme les méthodes à base d'histogramme. Le principal avantage des méthodes basées sur les blocs est leur manque de sensibilité au bruit. Cette méthode nous a permis aussi de localiser les changements aperçus et donc de ne traiter que les blocs jugés modifiés, créant ainsi une indexation pertinente qui permet de détecter la portion précédente dans le cas d'une image statique ou d'une animation. Ce point est déterminant car il devrait permettre d'apporter une réponse fiable à notre problématique [EMP07] :

- discriminer quelque soit le contexte une image fixe d'une animation,
- résoudre les problèmes de bruit inhérents au signal vidéo.
- gagner encore en traitement (moins d'appel aux modules de compression) ainsi qu'en quantité d'information à envoyer.

La division en blocs peut être dynamique ou statique. Dynamique veut dire que la taille du bloc est variable selon les modifications dans les trames, statique veut dire que la taille du bloc est fixée préalablement.

0	0	0	8	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	2	2	0	0	115	3	0	0
0	0	0	0	7	12	0	0	0	0	0	0
0	0	0	125	0	0	0	36	10	0	0	0
37	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	9
0	0	0	0	0	0	0	0	0	0	3	0

Division en blocs

Ainsi, à ce niveau, on prend chaque bloc tous seul et on compare le nombre de valeurs non nulles par rapport à un seuil prédéfini SL (seuil local) afin de déterminer les blocs modifiés (régions localement modifiées).

Ce processus nous permet de déterminer des blocs à actualiser, c'est-à-dire, les blocs de la trame en cours qui seront envoyés afin d'actualiser l'image de référence à la réception.

0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	2	2	0	0	115	3	0	0
0	0	0	0	7	12	0	0	0	0	0	0
0	0	0	125	0	0	0	36	10	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	9
0	0	0	0	0	0	0	0	0	0	3	0

Seuillage Local (>1)

Chaque bloc sera envoyé avec son numéro dans l'image en totalité et par la suite on fait extraire ses coordonnées pour pouvoir le restitué à son emplacement après décodage.

Détermination des blocs à actualiser

**Filtrage
(Élimination du
bruit par comparaison
de voisinage)**

4.4. Module de codage sélectif

Cette étape permet de détecter les zones textes modifiées pour les conserver en haute qualité par rapport aux autres blocs qui seront codés avec un codage avec perte et avec un coefficient de quantification assez élevé (figure 2.10).

Pour cela, nous avons fait recours à deux méthodes : une qui concerne le domaine spatial et l'autre dans le domaine compressé.

La première idée était basée sur la comparaison d'histogramme de couleurs sous la supposition qu'un texte présente des couleurs constantes. Ainsi, pour chaque bloc jugé modifié, on calcul son histogramme et si le nombre de couleurs dépasse un seuil prédéterminer on affirme que ce bloc présente du texte. L'expérience a donné de bons résultats pour des cas, mais pas pour d'autres. Dans un deuxième lieu, nous avons passé aux méthodes appartenant au domaine de la vidéo compressée qui se basent sur la caractéristique de la transformée en cosinus discrète. Pour chaque bloc on fait vérifier, après passage par la DCT et la quantification, la présence d'un coefficient AC. Ainsi, cette méthode fait détecter non seulement les régions texte mais aussi les blocs contenant du contour. Ces zones sensibles seront donc codées sans perte. Au contraire de la première méthode, l'expérience a donné des résultats satisfaisants pour les différents cas.

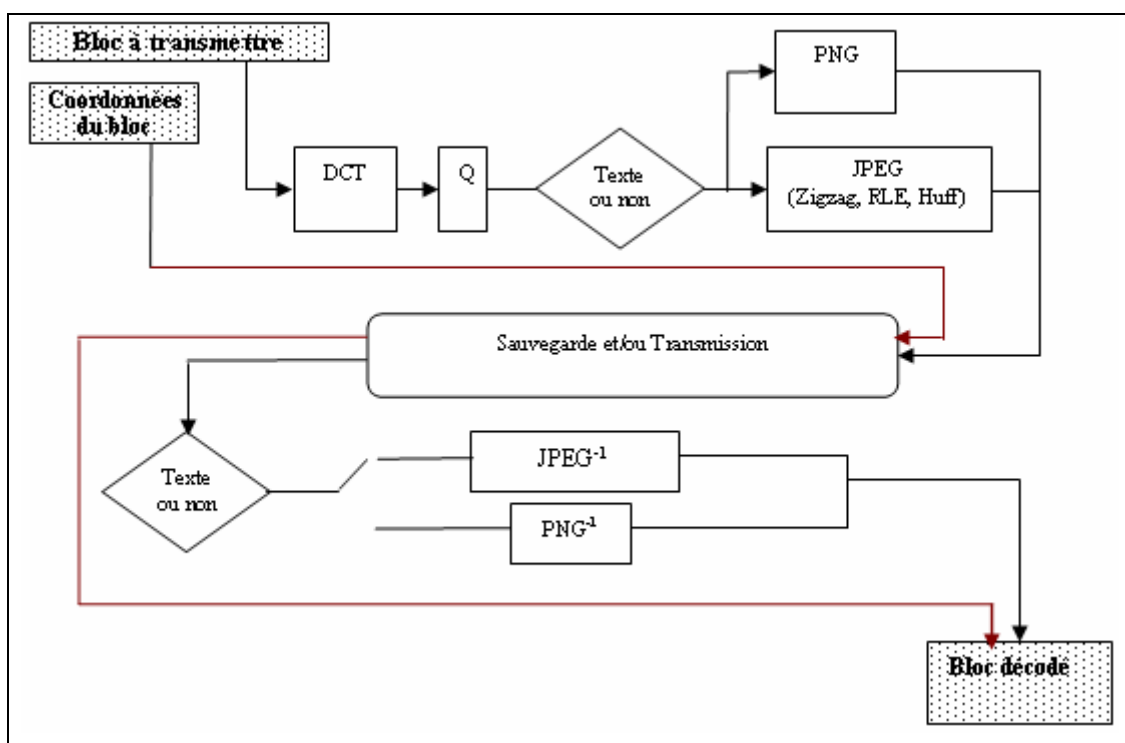


Figure 2.10 : Module de codage/décodage

Chaque bloc (de différence) sera encodé et transféré. Cet encodage dépend de la nature du bloc. Pour choisir un format de codage bien adapté aux spécificités de notre flux, nous avons fais recours à quelques expérimentations (Figure 2.11) qui touchent les mesures de taille et de qualité pour les formats les plus utilisés en codage d'image : JPEG, PNG, GIF et TIFF.

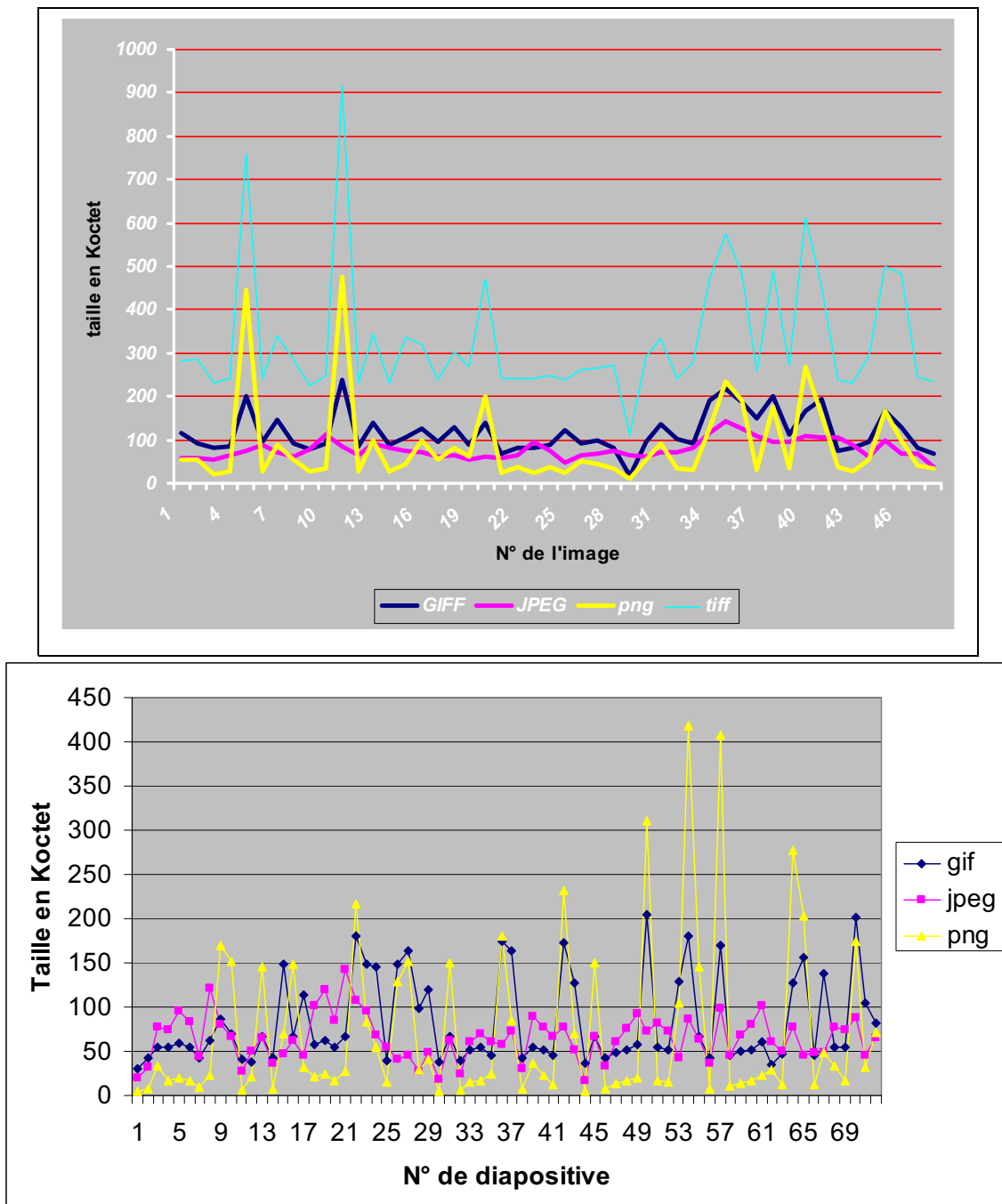


Figure 2.11 : Comparaison de différents formats de fichiers images

Une première comparaison, élimine les formats TIFF et GIF qui donnent, presque toujours, une faible compression par rapport aux autres formats. Pour les deux autres formats, nous constatons, qu'en moyenne, le JPEG fournit une meilleure compression par rapport au PNG qui, dans des cas, semble mieux que le JPEG, alors que dans d'autres cas, il présente des pics. Une analyse de ces deux cas (figure 2.12) nous a montré que les pics, présentés par le PNG, sont pour des cas où les images figurent des photos ou ont plusieurs couleurs. Il faut bien noter aussi que ces deux formats sont totalement différents, vu que l'un est avec perte et l'autre sans perte.

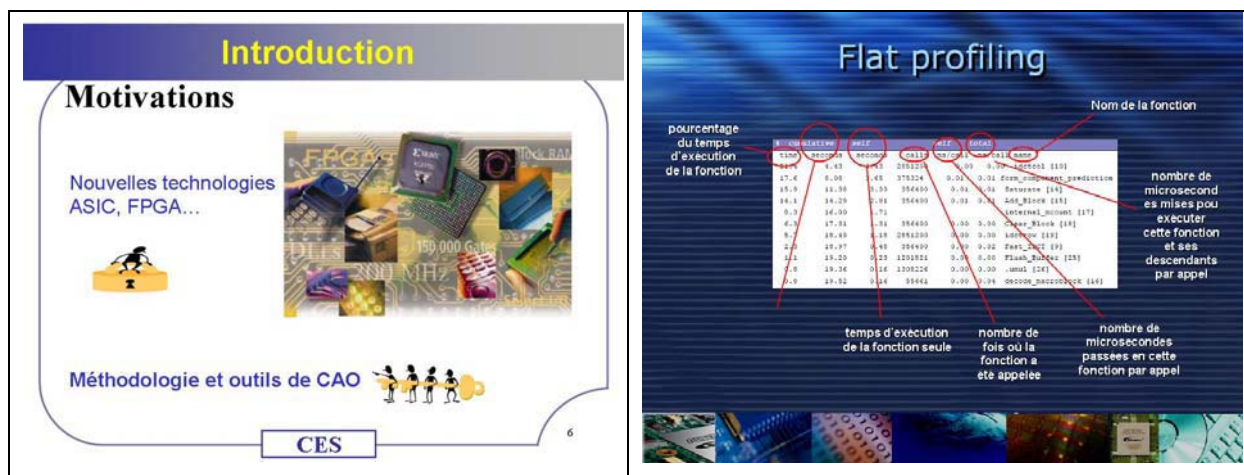


Figure 2.12 : Images présentant des pics en PNG par rapport au JPEG

Ainsi, le choix des deux formats JPEG et PNG doit être basé sur les caractéristiques de ces deux encodeurs (voir §I.2.7.4). En effet, la plupart des objets utilisés dans les diapositives sont des textes ou des objets graphiques, et suite à ces expérimentations sur des diapositives, nous avons constaté que JPEG est plus adapté pour les images contenant plusieurs couleurs, des photos et des zones graphiques. Alors que le PNG prouve sa qualité de compression pour les images contenant moins de zones texturées et plus de texte. En plus, le PNG garantit une compression sans perte, une chose qui est très bénéfique pour les diapositives qui contiennent en général, des informations assez fragiles et des détails assez fins tels que les textes, des contours assez fins... qui doivent être traités délicatement. La moindre perte dans ces informations peut rendre inutile ces supports : texte illisible, distorsion...

Chaque bloc compressé sera transmis avec sa position dans la trame. Pour pouvoir distinguer entre les blocs codés en PNG et ceux en JPEG, on ajoute un bit de différenciation au début du flux transmis de chaque bloc.

5. Conclusion

L'étude menée dans ce chapitre a été consacrée à la présentation du domaine d'application visé, où on a insisté sur les particularités du flux des diapositives : objet de notre travail. Notamment, cette étude a mis l'accent sur les méthodes de traitements usuelles et leur inadéquation avec la spécificité de ce flux. Comme solution, nous avons proposé des points d'intérêt à prendre en considération lors de la mise en œuvre d'une technique adaptée. Ces points sont portés par l'analyse des besoins des utilisateurs, ainsi que les spécificités et la nature des informations à traiter. En se basant sur ces études et en tenant compte de ces points d'intérêt, nous avons pu définir notre approche dédiée pour le traitement du flux des diapositives. Notre démarche expérimentale ainsi que les résultats obtenus seront l'objectif du chapitre suivant.

Chapitre III. Expérimentation & résultats

1. Introduction

L'objectif principal de ce projet consiste à étudier, concevoir et valider un algorithme adapté aux spécificités du flux des diapositives dans un contexte de visioconférence. Cette étape nécessitera le développement d'un environnement de test ou de validation. Cet environnement s'agit d'une application software qui permettra la validation virtuelle du prototype. Pour cela, nous avons implémenté les différents modules et fonctions à l'aide de l'environnement de programmation Microsoft Visual Studio 2005 en utilisant le langage C.

Nous présentons dans ce chapitre les résultats expérimentaux de notre système ainsi que quelques interprétations des résultats obtenus avec des exemples illustratifs.

2. Approche conceptuelle

Notre spécification constitue la première étape d'un flot de conception HW/SW basé sur l'approche de partitionnement dite « COSYMA » qui consiste à [KOU03] :

1. Spécification des besoins.
2. Description fonctionnelle du système.
3. Définir les différentes tâches ainsi que les fonctions à réaliser.
4. Implémenter toutes les fonctions en logiciel.
5. Migrer vers le matériel jusqu'au respect des contraintes de performance.

Une telle spécification dépend fortement du coefficient de pondération de la fonction de coût, des caractéristiques de l'architecture cible choisie. Souvent l'architecture cible est composée d'un seul processeur logiciel couplé à un ou plusieurs FPGA ou ASIC et éventuellement une mémoire commune. Peu de techniques de partitionnement ciblent vers une architecture hétérogène composée d'un ensemble de processeurs logiciels (microprocesseur, DSP, ASIP) et matérielle (FPGA, ASIC).

L'architecture générique monoprocesseur est constitué d'un ensemble de FPGA correspond peu à la réalité industrielle. Les composants programmables ont des performances plus faibles (surface de silicium occupé, fréquence max de fonctionnement) et un coût plus élevé (production en grande série) que les circuits non programmables. De plus même les « system on chip » ne sont pas monoprocesseur car ils disposent de plus en plus souvent d'un cœur de DSP et d'un cœur de microcontrôleur (MCU). Généralement, ces techniques de partitionnement automatiques ne sont donc pas intéressantes que pour le prototypage rapide sur une carte constituer d'un microprocesseur, d'un ensemble d'un FPGA et éventuellement d'une mémoire commune. Mais pour ce type d'application, la rapidité d'obtention d'une implémentation est un critère aussi important que la qualité du partitionnement obtenu : il s'agit avant tous de faire une vérification fonctionnelle et non une analyse réelle des performance du futur produit industriel.

En plus, les techniques de partitionnement automatique souffrent généralement du fait qu'elles ne prennent pas en compte l'expérience et le bon sens des concepteurs. Dans la réalité industrielle, le partitionnement d'un système ne pose problème que pour une petite partie du système. Le concepteur peut facilement faire un partitionnement grossier du système avant de se concentrer sur les parties délicates. Or les techniques de partitionnement automatiques considèrent le système dans son ensemble et n'utilise pas le fait que le concepteur peut fournir un partitionnement initial proche de la solution. Elles vont donc balayer un ensemble d'alternatives inutiles et dans certains cas sont moins efficaces qu'un partitionnement interactif. Inclure le concepteur dans la boucle de recherche de la solution optimale d'un partitionnement, offre aussi un avantage plus subtile : cela permet d'éliminer la suspicion des

concepteur face aux résultats obtenus par un partitionnement automatique. De plus, le concepteur a aussi la maîtrise complète de la solution retenue avec toutes ses justifications. Pour ces raisons, nous avons opté pour un partitionnement manuel, qui prend en considération les contraintes globales du système. Par la suite, nous allons focaliser nos travaux sur la spécification et la validation fonctionnelle d'un module de traitement spécifique permettant la compression des diapositives provenant d'une source VGA/XGA de l'ordinateur du conférencier en vue de leur diffusion sur IP. L'algorithme devrait donc assurer l'extraction des transparents ainsi que leur compression.

3. Environnement de développement

Comme support de tests pour nos expérimentations, nous avons utilisé des présentations de cours disponibles sur Internet et quelques présentations de soutenances des projets des membres de notre laboratoire qui représentent toutes sortes de styles de présentation et ceci afin de couvrir les diverses caractéristiques comme le nombre de diapositives, couleur de fond, couleur / taille de police et la variabilité de fond, des graphiques, etc. Cela nous permet d'étudier les différents cas de figure de notre contexte applicatif et donc à valider des choix et des paramètres de l'algorithme présenté précédemment. Pour contrôler nos résultats, nous avons fait recours à des mesures de qualité à l'aide de critères objectif (PSNR) et subjectif.

Notre environnement de développement retenu est le Microsoft Visual Studio 2005. Le langage de programmation est le C. Ce choix est lié à notre démarche d'implémentation où on vise ici un « prototypage virtuel » ou une validation fonctionnelle pour passer ensuite au « prototypage réel » avec l'implémentation sur carte FPGA. En fait, La méthodologie actuelle de design du système consiste, après spécification, à partitionner en modules hardware (langage de description hardware HDL, tel que Verilog ou VHDL) et software (langage C) les fonctions qui sont développées. Une telle conception appelée mixte est un très bon moyen pour optimiser la performance globale du composant programmable FPGA tout en profitant de la réutilisation de quelques méga-fonctions, du processeur embarqué Nios et des performances de l'outil de développement Quartus.

Pour la récupération du flux des diapositives, nous avons utilisé un composants, intégré dans le MVS, qui permet de faire des captures d'écran (figure 3.1).

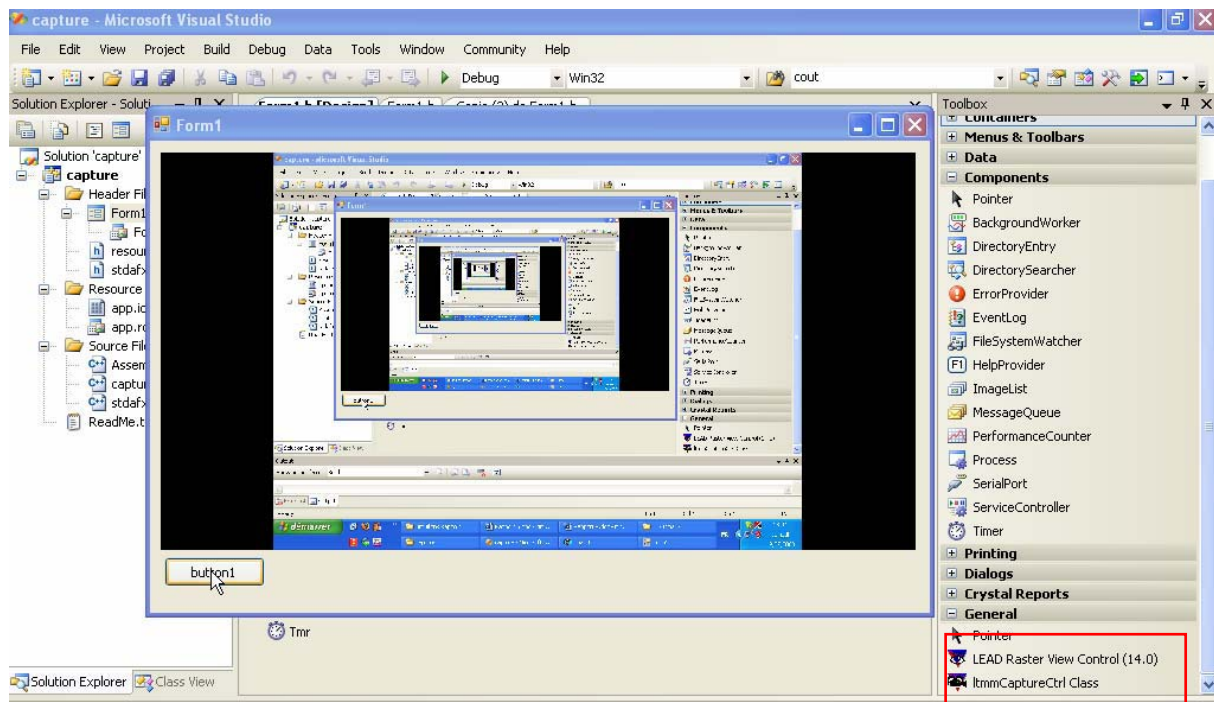


Figure 3.1 : Module de capture d'écran

Ces captures d'écran seront sauvegardées en fichiers pour faire le traitement et l'encodage selon notre approche. Ensuite, on sauvegarde les données codées dans un fichier qui peut être par la suite sauvegardé ou diffusé en réseau via le protocole FTP à un autre ordinateur. A la réception, la machine fait le décodage et restitue les images.

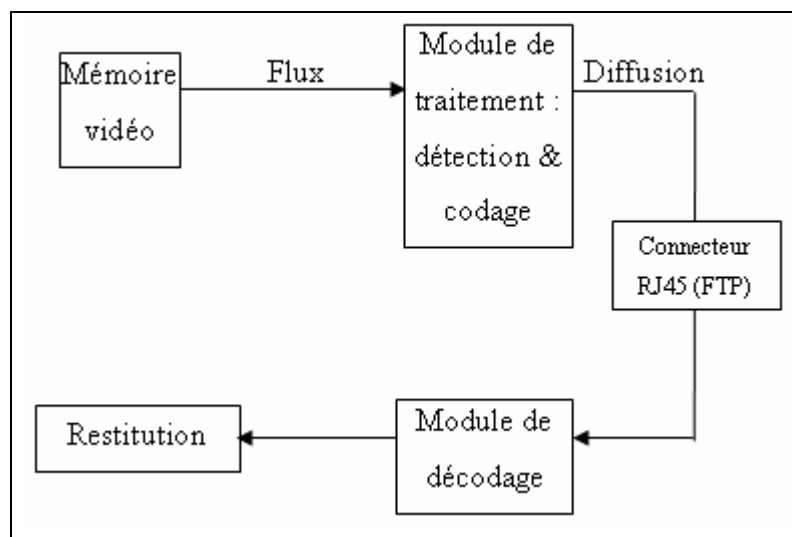


Figure 3.2 : Etapes d'expérimentation

4. Différentes étapes d'implémentation

Comme on a cité dans le deuxième chapitre, notre technique englobe plusieurs fonctions qui peuvent être présentées sous quatre modules : module de détection de changement, module de détection des zones texte, module de codage adapté et le module de décodage pour la restitution des images.

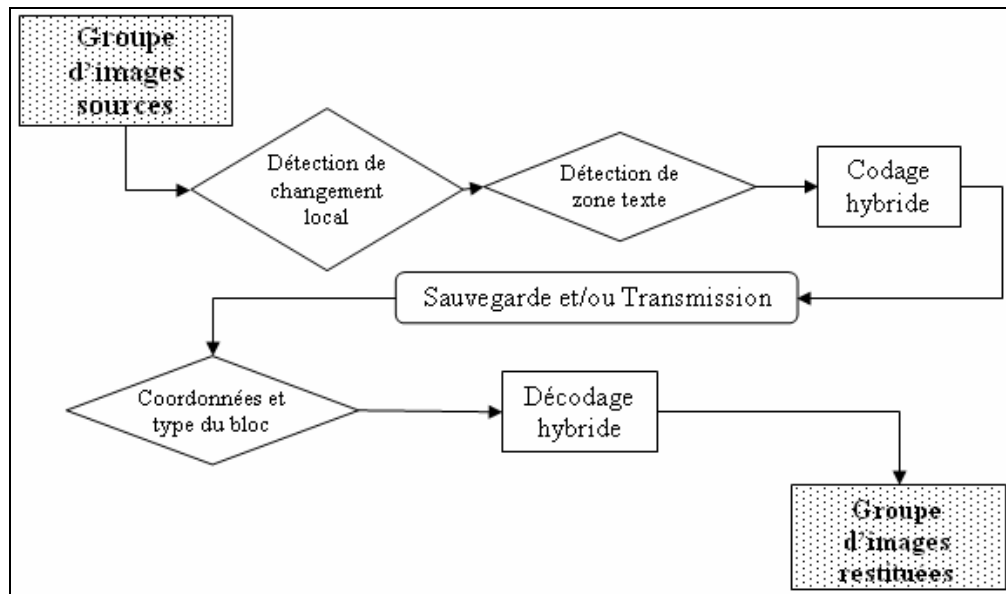


Figure 3.3 : Schéma blocs des modules de la technique proposée

4.1. Module de détection de changement

Comme nous l'avons mentionné précédemment, la détection de changement est signalée suite à une comparaison pixel par pixel appliquée sur les blocs de deux trames de façon à avoir, dans le cas positif, une image de différence ne contenant que les blocs détectés modifiés (Figure 3.4).

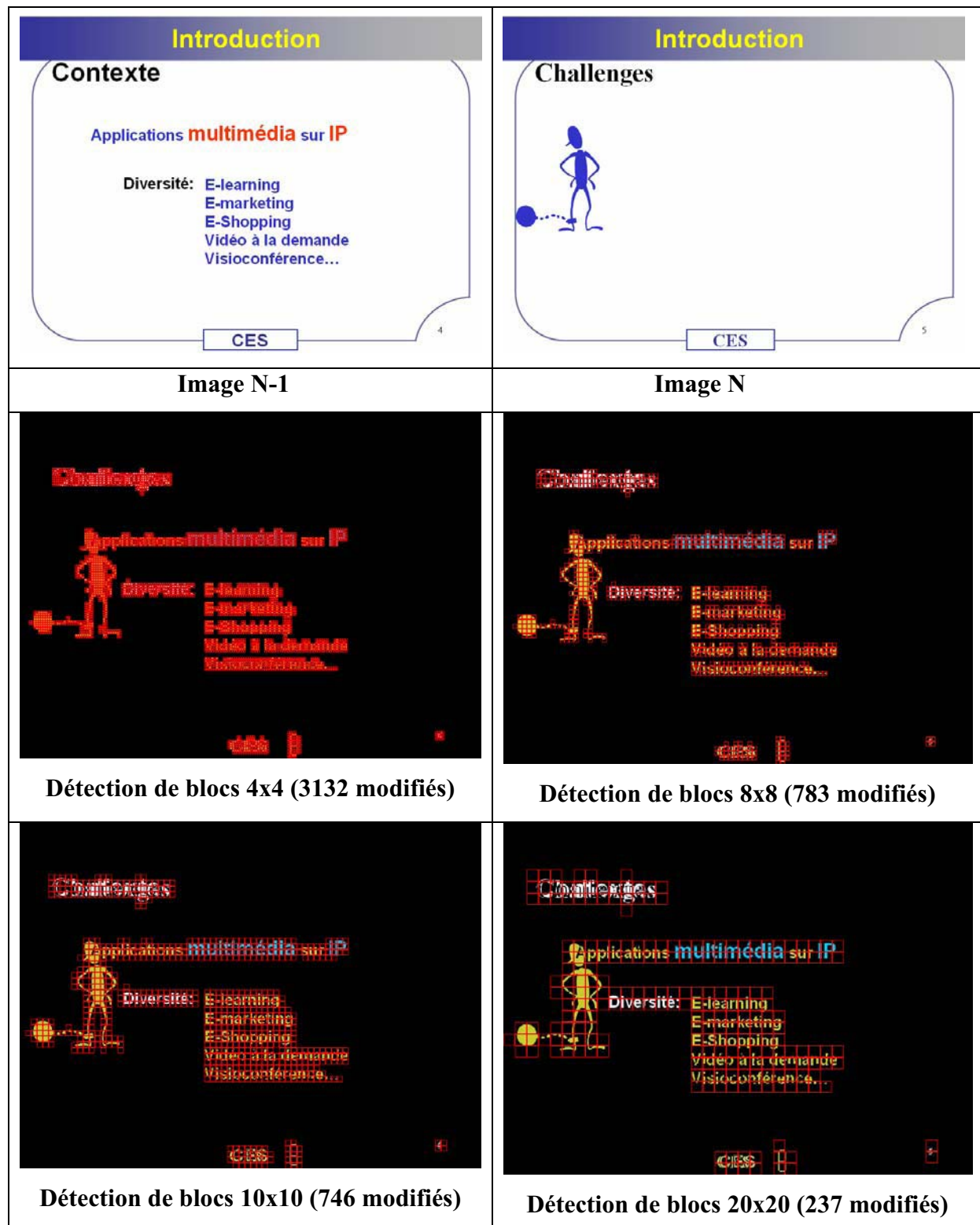


Figure 3.4 : Module de détection des blocs modifiés

Cette méthode nous garantit un gain considérable en quantité d'information à traiter et à transférer. Ce taux dépend directement du choix de la taille des blocs et des seuils déterminés. Pour un bloc de taille 20x20 on aura 237 blocs modifiés ce qui veut dire 237 appel aux

fonctions de codage, mais présente aussi une détection moins fine. Au contraire, une détection par bloc de taille 4x4 demande plus de traitement et moins de zones noires (non modifiées). Dans le tableau suivant on peut voir un exemple testé sur quelques images.

Table 3.1 : Taux de transfert

Taille du bloc	Nombre de blocs pour une résolution (800x600)	Nombre de blocs à transmettre	Seuil local	Taille du vecteur position	Taux de transfert
8x8	7500	321	10	13	$(321 \times 64 \times 24 + 321 \times 13) / (7500 \times 64 \times 24) = 4,31 \%$
20x20	1200	85	10	11	$(85 \times 400 \times 24 + 85 \times 11) / (1200 \times 400 \times 24) = 7,1 \%$
20x20	1200	93	0	11	$(85 \times 400 \times 24 + 93 \times 11) / (1200 \times 400 \times 24) = 7,1 \%$

Suite à quelques testes, nous avons choisi un seuil local égale à dix pixels. Nous avons utilisé des blocs de taille 8x8 car ça donne une détection fine et c'est déjà la taille utilisée pour l'encodage en JPEG.

Nous avons fait un teste sur un exemple de présentation du cours « Codesign ». Cette présentation contient 45 diapositives, elle dure 20 min, de résolution 800x600 et la majorité de ses diapositives contient différents effets d'animation et de transition. Nous avons récupéré les résultats suivants :

- Nombre total d'images (pour une vidéo de 25 img/s) : 30000
- Nombre total d'images jugées différentes : 895 ➔ 2.98 %,
- Nombre total des blocs codés : 1 594 051 blocs sur 6 712 500 blocs des images jugées différentes, soit 23.74%.

A la fin de chaque cycle de traitement, on revient à la phase d'initialisation où on sauvegarde la dernière trame détectée changée et on l'utilise comme référence pour la comparer ensuite avec la nouvelle trame acquise. Cette idée a pour objectif de garantir la détection de changement même pour les lentes animations et pour les animations des objets de petite taille. En fait, si on procède à une comparaison de trames successives, on risque, pour un seuil relativement grand, de ne détecter l'animation qu'en trois trames, par exemple, alors qu'elle couvre plusieurs (voir figures 3.5 et 3.6) :

Diversité: E-learning	Diversité: E-learning E-learning	Diversité: E-learning E-marketing
-----------------------	-------------------------------------	--------------------------------------

Figure 3.5 : Détection par comparaisons de trames successives

Diversité: E-learning	Diversité: E-learning	Diversité: E-learning
Diversité: E-learning	Diversité: E-learning	Diversité: E-learning
Diversité: E-learning	Diversité: E-learning E-learning	Diversité: E-learning E-learning

Figure 3.6 : Détection par comparaisons de trames non-successives (pour même seuil)

4.2. Détection de texte

Pour cette étape, nous avons essayé, dans un premier temps, une méthode basée sur le calcul des histogrammes des blocs. En effet, un bloc modifié sera jugé appartenant à une région texte s'il contient un nombre de couleur supérieur à un seuil. Cette méthode a donné de bonnes résultats dans des cas mais pas dans d'autres et les testes n'ont pas pu donner un seuil adéquat (voir figure 3.7).

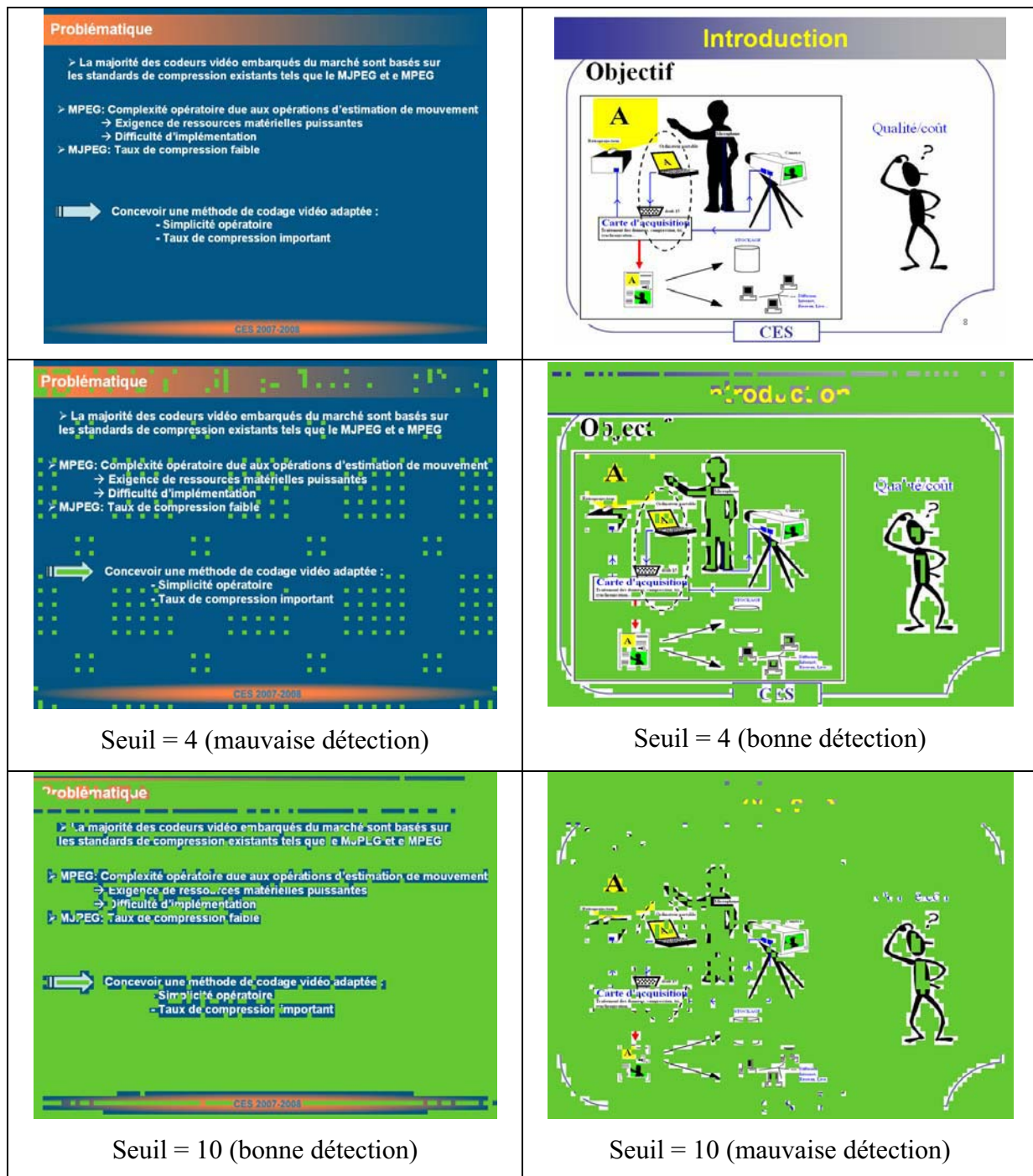


Figure 3.7 : Détection de texte par calcul d'histogramme

Ainsi, nous avons passé à une autre méthode sur domaine compressé : un bloc est jugé texturé si, après quantification, il ne contient que la composante DC et une composante AC. La figure 3.8 montre des exemples des images de références codées puis restituées par notre méthode. Les blocs jugés « non texte » sont colorés en vert et ceux contenant du texte sont visibles.

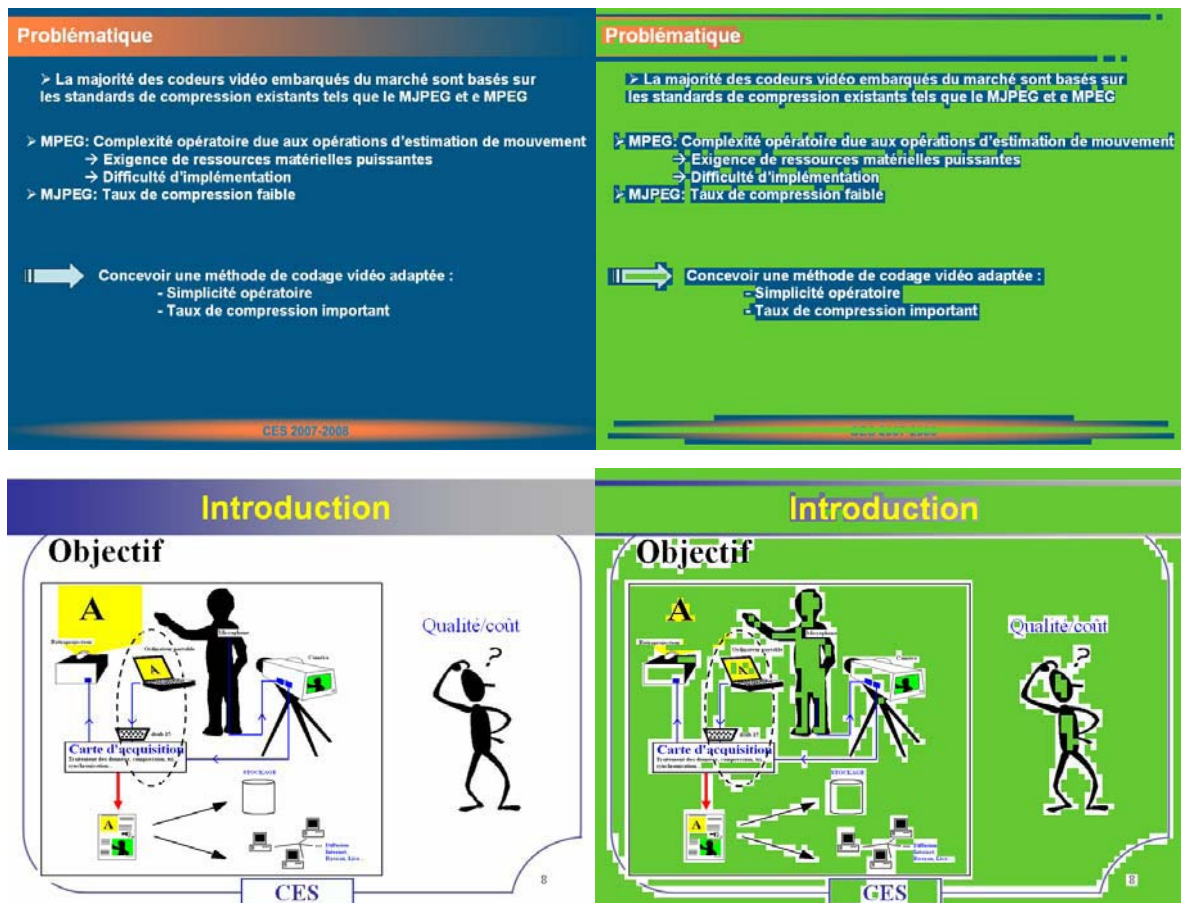


Figure 3.8 : Détection de texte dans le domaine compressé

4.3. Module de codage

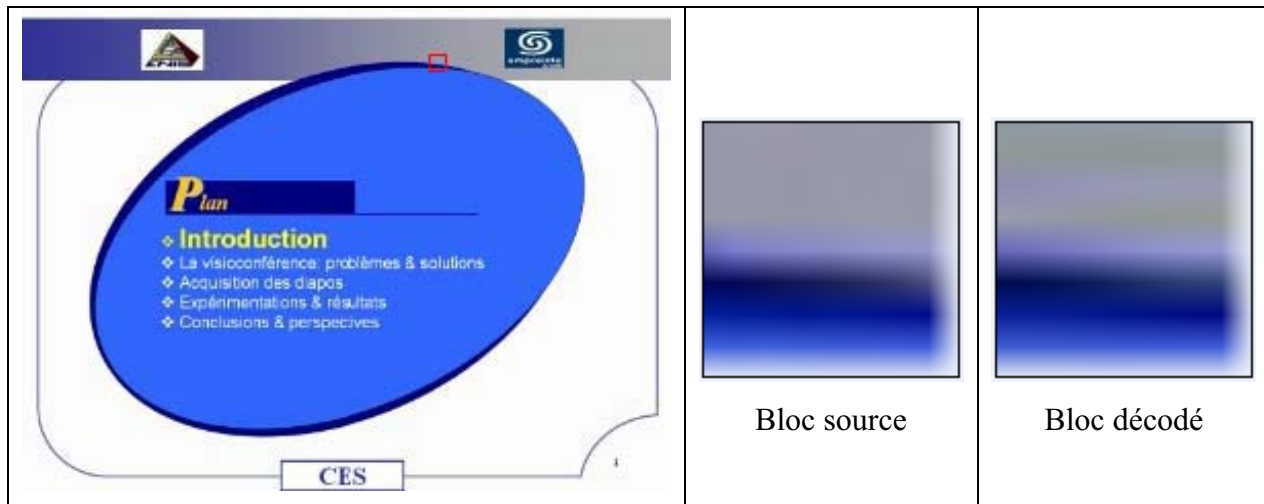
Pour la compression avec perte des zones qui ne contiennent pas de texte, nous avons implémenté un module de codage et de décodage en JPEG.

Ce module comporte les sous-modules suivants :

- Calcul de la DCT.
- La quantification
- Le parcours en ZigZag.
- Le codage RLE.
- Le codage Huffman.
- La conversion décimal \leftrightarrow binaire.

Notamment, chaque sous-module lui correspond son inverse.

La figure 3.9 représente les différentes étapes du codage et de décodage JPEG réalisés par le module développé.

**Bloc Source**

154	154	153	153	111	13	36	102
155	154	153	155	134	14	33	99
156	153	153	153	139	21	32	101
154	153	152	153	143	28	28	99
153	154	155	154	146	36	30	98
155	154	154	157	152	46	28	98
154	154	155	155	155	58	24	97
155	157	153	154	156	83	21	95

Fonction DCT

461.688	159.625	-30.269	-69.757	87.687	-41.604	-6.606	18.331
-14.581	5.715	16.557	-19.377	3.654	12.453	-12.601	5.846
1.041	-0.796	1.044	2.542	-4.781	0.812	4.346	-6.286
-2.347	0.723	2.863	-4.721	-0.460	3.695	-2.137	0.203
-0.937	-0.711	1.432	0.803	-3.437	0.345	2.315	-3.548
-1.841	0.021	1.854	-1.674	0.291	0.808	0.237	-0.214
0.718	-0.664	1.221	0.665	-0.928	0.672	0.955	-1.614
-0.247	-0.273	1.050	-1.016	-0.927	0.646	0.204	0.447

Quantification

96	48	-10	-14	12	-3	0	1
-4	1	4	-3	0	1	-1	0
0	0	0	0	0	0	0	-1
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Parcours en ZigZag

96, 48, -4, 0, 1, -10, -14, 4, 0, 0, 0, 0, 0, -3, 12, -3, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 1, 0, 1, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

Codage RLE

0	0	0	1	0	0	0	5	0	0	10	1	0	13	19
96	48	-4	1	-10	-14	4	-3	12	-3	1	1	-1	-1	0

Codage Huffman

11110110000011110001100001000111100110110101101100011001001111111011100
 1011110001001111110101110010001111111100001010

Décodage RLE

0	0	0	1	0	0	0	5	0	0	10	1	0	13	20
96	48	-4	1	-10	-14	4	-3	12	-3	1	1	-1	-1	0

Parcours en ZigZag

96, 48, -4, 0, 1, -10, -14, 4, 0, 0, 0, 0, -3, 12, -3, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 1, 0, 1, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

Quantification⁻¹

96	48	-10	-14	12	-3	0	1
-4	1	4	-3	0	1	-1	0
0	0	0	0	0	0	0	-1
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Fonction DCT⁻¹

460	158	-31	-68	86	-36	0	18
-15	3	16	-18	0	17	-18	0
0	0	0	0	0	0	0	-5
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Bloc Source

152	151	147	159	115	14	39	97
152	150	149	157	120	17	38	97
154	149	152	154	128	23	36	97
155	148	155	152	137	32	32	96
156	147	158	151	145	43	28	96
156	148	158	151	152	55	23	97
157	149	158	152	156	64	19	97
157	149	158	153	158	69	16	97

Figure 3.9 : Module du CODEC JPEG

Pour les blocs de texte, on fait le codage en PNG grâce à la librairie *libpng* et *zlib* disponible sur [PNG] et [ZLI].

4.4. Phase de réception et décodage

Le décodeur reçoit les blocs objets (flux binaire) via le canal de transmission, vérifie leur catégorie et procède au décodage. Si l'image reçue est une image de référence, le décodeur la mémorise comme telle (phase d'initialisation). Dans le cas contraire, les nouveaux blocs reçus sont utilisés pour construire la nouvelle image en faisant une combinaison avec la dernière image présente chez l'utilisateur. Cette construction se fait par une soustraction pixel par pixel pour chaque bloc en se basant sur ses coordonnées.

La partie concernant la décompression de l'image est maintenue complètement standard (JPEG et/ou PNG) ce qui donne encore un avantage à notre méthode.

Suite à quelques expérimentations, nous avons remarqué que, après décodage, il y a eu des erreurs dans les images restituées (Figure 3.10) :



Figure 3.10 : Décodage en JPEG : cumulation des erreurs

Ces artefacts sont dus essentiellement à la compression temporelle. En effet, le JPEG est un codage qui présente de perte. Cette perte, due essentiellement à la quantification, s'accumule d'une image à une autre. Une première solution est d'agir sur le coefficient de quantification, mais ça nous fait perdre en taux de compression. Pour cela, nous avons proposé une autre hypothèse de rafraîchir l'image de référence après un certain nombre de détection de changement. Autrement dit, après chaque N trames détectées changées, la N^{ème} sera codée en totalité et envoyée.

Ainsi, dans notre système, deux images de références co-existent : celle du système local et celle du système distant (utilisateur). La mise à jour d'une image de référence au décodeur doit être effectuée aussi régulièrement que possible afin de maintenir la cohérence de l'image finale. Celle de l'encodeur est mise à jour après chaque trame détectée différente.

5.5. Mesure de qualité

Nous utilisons une stratégie pour évaluer les résultats basés sur des critères objectifs. Ce critère consiste à utiliser le PSNR pour mesurer la distorsion entre l'image initiale et l'image reconstruite.

Les Fig. 3.11 et 3.12 présentent les valeurs du PSNR correspondants aux images reconstruites pour l'exemple de la présentation appelée « Codesign ».

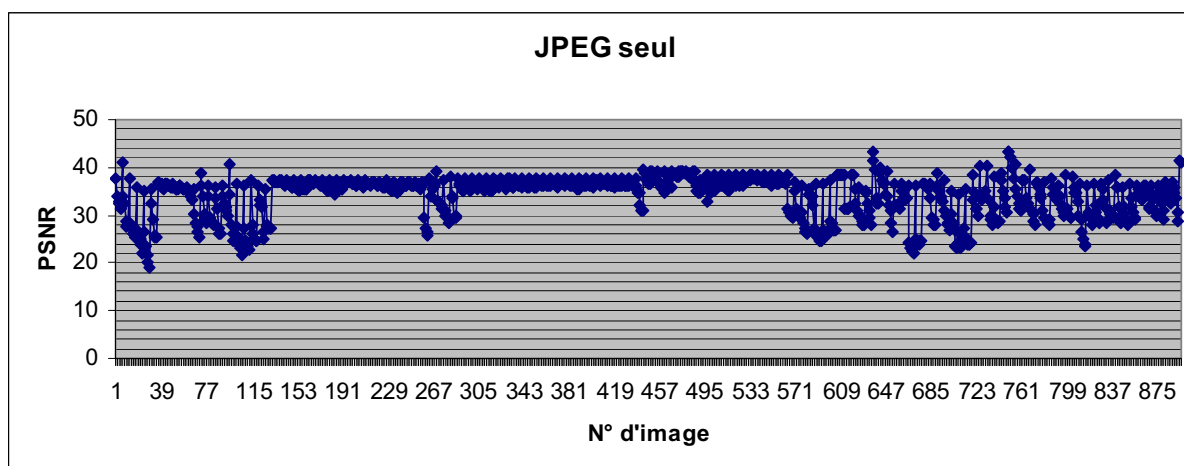


Figure 3.11 : Mesure de PSNR pour le flux codé en JPEG seul

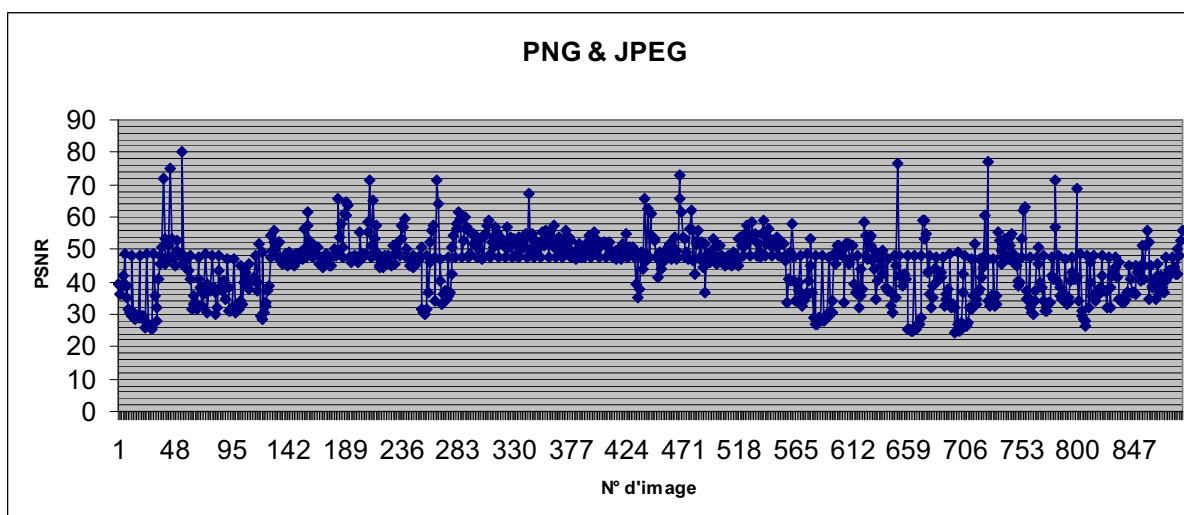


Figure 3.12 : Mesure de PSNR pour le flux codé en JPEG et PNG

La première figure est faite pour le cas où nous avons codé tous les blocs détectés modifiés en JPEG. La moyenne du PSNR est égale à 33 dB, qui est relativement bonne surtout que le coefficient de quantification Q qu'on a choisit est faible ($Q=0,4$). Pour la deuxième figure nous avons fais les mesure pour les images codées en PNG et JPEG. On remarque bien que la moyenne du PSNR est élevée à 42 dB. De même on remarque bien la différence, en ce qui concerne la qualité de visibilité du texte, entre notre méthode et le JPEG (figure 3.13).

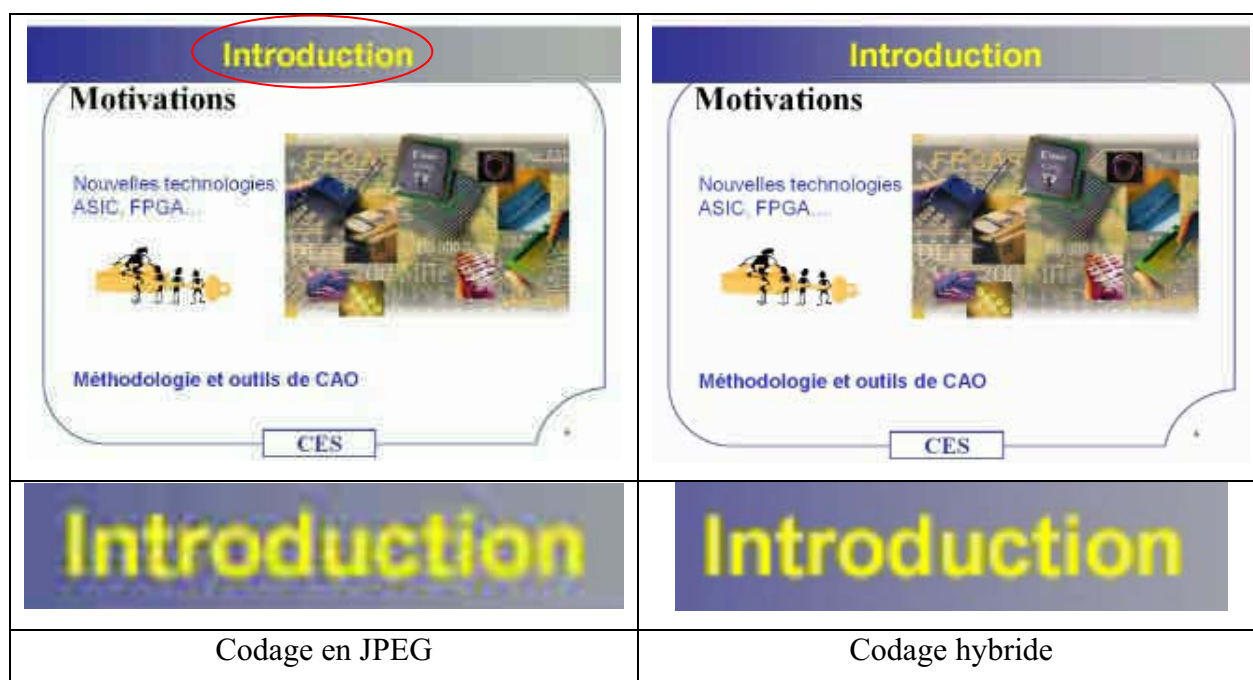


Figure 3.13 : Qualité du diapositives entre le JPEG et notre méthode

5.6. Synthèse

Ce travail nous a permis de mettre en œuvre plusieurs méthodes et de convoquer plusieurs domaines. Pour la détection de changement (autrement dit la présence d'animation), nous avons fais recours à une méthode de comparaison pixel par pixel en se basant sur les blocs de différence entre deux trames. Cette phase nous a permis de gagner en débit puisque on ne transmet qu'en cas de détection de changements pertinents et que les blocs jugés modifiés.

Nous avons fais recours aussi à une phase de prétraitement qui a pour but d'éliminer le bruit accompagnant la récupération des signaux provenant d'une sortie XGA. Il s'agit d'une méthode de filtrage assez simple et satisfaisante basée sur un teste les voisinages qui élimine les pixels isolés (qui représentent généralement du bruit d'acquisition).

Pour garantir un bon compromis entre la qualité du rendu visuel et le taux de compression, nous avons déterminé une méthode de codage hybride. En effet, nous avons fait recours à une phase de détection et localisation de texte qui exploite les caractéristiques de la transformé en cosinus discrète. Cette étape est suivie de l'encodage en JPEG ou en PNG. Les résultats de la mesure de qualité objective (PSNR) et subjective (lisibilité du texte) montrent bien l'apport de cette approche.

5- Conclusion

Nous avons présenté dans ce chapitre les résultats obtenus suites aux expérimentations et tests faits sur notre technique proposée pour le traitement et la détection du flux des diapositives lors d'une présentation dans une conférence. Cette technique contient différents blocs de traitement : la phase d'initialisation (permettant la construction de la première image de référence), la comparaison entre images pour signaler la présence d'un changement, la différenciation entre les régions changées (ceux appartenant à une zone texte et les autres), l'encodage sélectif des blocs ainsi que leur restitution et la mise à jour d'une image de référence. Cette approche apporte plusieurs avantages du fait qu'elle exploite des techniques et des outils existants, elle apporte une réponse dédiée à chaque type de données traitées et a prouvé de bons résultats en taux de compression et qualité obtenue.

Conclusion générale & perspectives

Très efficace pour une conversation à distance entre deux personnes, voire un petit nombre de personnes, la visioconférence facilite les échanges pour les préparations de réunions, de cours, de conférences, de colloques, pour la mise en commun de résultats de recherches...

Cependant, un tel service multimédia déployé sur IP évoque une problématique qui porte sur l'importance du volume des données à traiter et à transmettre. Ce volume, qui demande un débit élevé et un traitement rapide, pose de sérieux problèmes de bande passante et de temps de réponse. Dans ce contexte se situ notre travail. Il s'agit de concevoir et valider un algorithme de traitement vidéo qui devra permettre, automatiquement, l'acquisition, le traitement et la diffusion du contenu VGA/XGA de la présentation d'un conférencier.

Afin de répondre à cet objectif, différents points ont été étudiés.

Le premier point concerne le type de données à envoyer : un diaporama n'est ni une séquence d'images fixes ni une vidéo mais comprend les deux aspects ensembles. Pour cela nous avons étudié les différentes normes de codage vidéo et les standards de codage d'images fixes pour présenter une solution adéquate aux spécificités de notre flux. Cette étude nous a conduit à élaborer un système de codage (codec) servant des avantages des standards existants, et apportant un codage adapté qui assure un bon compromis qualité/débit.

Le deuxième point concerne la détection des animations et des transitions. L'idée était basée sur des comparaisons locales des trames reçues pour repérer et localiser les changements produits. Cette étape nous a permis de conserver les animations qui se produisent durant toute la présentation ainsi que de gagner en quantité de traitement et en débits de transmission. De même, cette méthode est caractérisée par une simplicité de calculs (allocation des ressources matériels de traitement) par rapport à d'autres méthodes.

Le dernier point touche la qualité des données. Dans un diaporama on cherche toujours à présenter tous les points clés mais avec le minimum d'écriture et avec un beau design. Ceci invoque l'importance de chaque information présente dans les diapositives. Pour cela, l'idée était de chercher une adéquation entre les propriétés des informations présentes et des

standards de codage. On a donc essayé de diviser les blocs jugés modifiés en deux régions : le texte et les régions graphiques. Les blocs appartenant aux régions texte sont ensuite encodés par le PNG (codage sans perte ce qui garde une bonne lisibilité), et les autres seront en JPEG (on peut même tolérer un taux de compression élevé).

Ainsi, nous avons pu atteindre notre objectif d'assurer la diffusion automatique du flux des diapositives avec un bas débit et le minimum de calcul tout en garantissant, à la réception, un rendu visuel des images acceptable par les utilisateurs.

Comme perspective, nous visons à faire une analyse détaillée du performance de différentes fonctions (complexité opératoire de traitement) afin de se préparer à l'étape suivante de conception. Il s'agit du partitionnement HW/SW qui doit tenir compte de la complexité opératoire de traitement de chaque fonction (temps d'exécution). On peut aussi envisager le taux d'utilisation ou d'appel à la fonction (distinguer entre les fonctions partagées entre les différents modules et les fonctions privés). Par ailleurs, on envisage à intégrer ce travail dans le projet global et mettre en place un démonstrateur.

Bibliographie

- [A&E04] « *Compression vidéo avancée* », A&E's Technical Guides to All Things Audio and Video 2004
- [ADO00] « *Initiation à la vidéo numérique* », par le groupe Adobe Dynamic Media, juin 2000
- [AHA96] G. Ahanger and T.D.C. Little. « A survey of technologies for parsing and indexing digital video », *Journal of Visual Communication and Image Representation*, 7(1) :28–43, Mars 1996
- [ANT07] « PNG vs GIF vs JPEG », www.antp.be/PNG/tweakpng, 2007-06-27
- [AVR06] Dr. Sohaib Khan and Dr. Umar Saif, AVRiL (Automated Video Recording of Lectures), « End-of-Quarter Engineering Report version 1.0 » 13-Dec 2006, www.avril.sproj.com
- [BEH04] Behera, A., Lalanne, D. and Ingold, R. « Looking at projected documents: Event detection & document identification », *Intl. Conf. on Multimedia Expo (ICME '04)*, 2004
- [BER03] Manuscrit de cours en Couleur : Université Claude Bernard Lyon
- [BOR07] Lokesh R.Boregowda et al., « Video motion detection using block processing », *US Patent Application Publication*, N°US 2007/0058837 A1, Mar.15, 2007
- [BOR96] J.S. Boreczky, L.A. Rowe, « Comparison of Video Shot Boundary Detection Techniques ». Storage and Retrieval for Image and Video Databases IV, Proc. of IS&T/SPIE 1996 Int'l Symposium. on *Electronic Imaging: Science and Technology*, San Jose, CA, February 1996
- [BOU96] Thomas Boutell, « *Portable Networks Graphic 1.0 Specification* », 1996
- [BRA05] Patrice BRAULT, "Estimation de mouvement et segmentation d'image", PhD thesis le 29 novembre 2005, UNIVERSITE PARIS-SUD XI Faculté des Sciences d'Orsay

- [BRU99] R. Brunelli, O. Mich, and C.M. Modena. « A survey on the automatic indexing of video data », *Journal of Visual Communication and Image Representation*, 10(2) :78–112, Juin 1999
- [CHE05] Pierre Checa & Emilien Taque, « Les modes de représentation des couleurs », rapport de projet, SILR2, 8 juin 2005
- [CHE04] D. Chen, J. Odobez, J. Thiran, « A localization/verification scheme for finding text in images and video frames based on contrast independent features and machine learning methods », *Signal Process.: Image Commun.* 19 (2004) 205–217
- [CHE03] Datong CHEN, « Text detection and recognition in images and video sequences », Ph.D. thesis, École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland, August 2003.
- [CHE01] D. Chen, H. Bourlard, J.P. Thiran, « Text identification in complex background using SVM », *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, Prague, Czech, 2001, pp. 621–626
- [DEL06] Bertrand DELHAY, « Estimation spatio-temporelle de mouvement et suivi de structures déformables. Application à l'imagerie dynamique du coeur et du thorax », PhD thesis, INSA Lyon, Decembre 2006.
- [DUN05] « Imagerie numérique », Centre de Formation et de Ressources Spécifiques de Dunkerque, 13 novembre 2005
- [EMP] <http://www.empreinte.com>
- [EMP07] « Projet de recherche d'algorithme de traitement vidéo : demande d'aide arritt transfert », EMPREINTE MULTIMEDIA SA, 29/05/2007
- [EMP04] « Etude de développement FPGA-PCI », EMPREINTE MULTIMEDIA SA, Etude de faisabilité 2004, Aide à l'innovation ANVAR
- [ERI00] ERICSON, « Automated Conversion of a Visual Presentation Into Digital Data Format », International Application Published Under the Patent Cooperation Treaty (PCT) 2000
- [GAR04] Vincent Garcia, « Estimation de mouvement subpixélique par blocs adaptée à la couleur avec modèle de mouvement », Rapport de DEA en Image Vision (septembre 2004), Laboratoire I3S Université de Nice - Sophia Antipolis
- [GEM00] GEMME, « La visioconférence : usages, stratégies, moyens. Pour le développement de l'usage de la visioconférence dans les établissements

- d'enseignement supérieur », Rapport pour le Ministère de l'Education nationale, de la recherche et de la technologie, juin 2000, 70 p.
- [GIP07] « La vidéo, Etat de l'art sur la vidéo », GIPSA-Lab & LISTIC-Lab, 31 octobre 2007
- [IDR97] F. Idris and S. Panchanathan. « Review of image and video indexing techniques ». *Journal of Visual, Communication and Image Representation*, 8(2) :146–166, Juin 1997
- [KOU03] M.Koudil, « Méthode de conception conjointe des systèmes embarqués », cours 02/03 de l'Institut National de Formation en Informatique, Algérie
- [KUN93] M. Kunt, G. Grunland, et M. Kocher. « *Traitement de l'information : Traitement numérique des images* », Volume 2. Presses polytechniques et universitaires romandes, 1993
- [LEF05] S. Lefèvre, N. Vincent, « Extraction d'indices spatiaux et temporels dans des séquences vidéo couleur », Atelier Fouille de données complexes dans un processus d'extraction des connaissances, *Congrès Extraction et Gestion des Connaissances*, Paris, 2005
- [LEF03] S. Lefèvre, J. Holler, N. Vincent : « A Review of Real-time Segmentation of Uncompressed Video Sequences for Content-Based Search and Retrieval » *Journal Real Time Processing*, 2003, Vol. 9, N°1, pp. 73-98.
- [LEF02] Sébastien LEFEVRE, « Détection d'événements dans une séquence vidéo », Thèse en Informatique, Ecole Doctorale : Santé, Sciences et Technologies, Décembre 2002
- [LEO05] Miriam Leon, Antoni Gasull, « Text detection in images and video sequences », *Image processing group, Department of Signal Theory and Communications*, Technical University of Catalonia, Spain, 2005
- [LOR03] F. Loras and J. Fournier, « H.264/MPEG-4 AVC, un nouveau standard de compression vidéo ». In *CORESA '03*, Lyon, 16-17 Janvier 2003
- [LUK07] Rastislav Lukac and Konstantinos N. Plataniotis, « *Color Image Processing, Methods and Applications* », CRC Press © 2007 by Taylor & Francis Group.
- [MEN06] Ida Mengyi Pu, « *Fundamental Data Compression* », ISBN: 0-7506-6310-3, ELSEVIER 2006
- [MIA99] John Miano, « *Compressed Image File Formats, JPEG, PNG, GIF, XBM, BMP* », Addison Wesley Publishing - September 1999 ISBN: 0201604434

- [MUK99] Mukhopadhyay, S., and Smith, B. (1999), « Passive capture and structuring of lectures », *proceedings of the seventh ACM international conference on Multimedia* (Part 1), Orlando, Florida, ACM Press, 1999, pp. 477–487].
- [NGO05] C.-W. Ngo, C.-K. Chan, « Video text detection and segmentation for optical character recognition », *Multimedia Syst.* 10 (3) (2005) 261–272
- [OUN06] Tarek OUNI, « Contribution aux systèmes de visioconférence: Conception d'un module d'acquisition de diapositives », rapport de maîtrise soutenu Novembre 2006 à l'ENIS
- [OUN04] « Conception haut niveau d'une carte d'acquisition d'image », Tarek OUNI, rapport de PFE à l'ENIS 2004
- [PNG] www.libpng.org
- [PRE07] Jacques PREVOST et al., « ARTS Aristote Real Time Slides ARTemis Slides », Mars 2007
- [PRE01] Jacques Prévost, « La vidéo numérique sur IP et la communauté Renater », *JRES 2001*
- [QI00] W. Qi, L. Gu, H. Jiang, X.-R. Chen, H.-J. Zhang, « Integrating visual, audio and text analysis for news video », *Proceedings of the International Conference on Image Processing*, Vancouver, Canada, vol. 3, 2000, pp. 520
- [QIA07] Xueming Qian, et al., « Text detection, localization, and tracking in compressed video », *Signal Process. Image Commun.* (ELSEVIER 2007), doi:10.1016/j.image.2007/06/05
- [REN04] « Représentation et codage des images », Manuscrit de cours de l'Institut d'électronique et de télécommunications de Rennes.
- [REV08] Jean Revertera, « Système de détection de changement de slides », Master 2008, École d'Ingénieurs et d'Architectes de Fribourg
- [ROE99] Greg Roelofs, « *PNG: The Definitive Guide* », Published by O'Reilly & Associates, USA 1999.
- [SAB99] L. Sabry « Schéma d'exécution pour les documents multimédia distribués », Thèse à l'Université Joseph Fourier Grenoble, France. 25 janvier 1999.
- [SCH01] Karl SCHWERDT, « Compression vidéo fondée sur l'apparence », Thèse en Imagerie, Vision et Robotique, Institut Polytechnique de Grenoble, Mai 2001
- [SHA48] Claude E. Shannon. « A Mathematical Theory of Communication ». *The Bell System Technical Journal*, 27 :379–423,623–656, 1948

- [SIB03] Cédric Sibade, « Compression de Données pour les Systèmes de Traitement de Document Grand Format », Thèse Doctorat de l'Université de Marne-la-Vallée, le 15 décembre 2003
- [SSI06] OnSSI: On-Net Surveillance Systems Inc., « MJPEG vs MPEG4: Understanding the differences, advantages and disadvantages of each compression technique », White Paper 2006
- [STE06] Doston Stéphanie, « L'encodage des images informatique », Dossier d'évaluation TICE 1, 2006
- [TAN02] X.-O. Tang, X.-B. Gao, J.-Z. Liu, H.-J. Zhang, « A spatialtemporal approach for video caption detection and recognition », *IEEE Trans. Neural Networks* 13 (4) (2002) 961–971
- [TCO02] « Compression de données », Cours de L'Institut TCOM 2002
- [TOT07] Théodore TOTOZAFINY, « Compression d'image couleur pour application à la télésurveillance routière par transmission vidéo à très bas débit », thèse à l'université de Pau et des Pays de l'Adour, Ecole doctorale des sciences exactes et de leurs applications, le 3 juillet 2007
- [WAL91] Gregory K. Wallace, « The JPEG Still Picture Compression Standard », Submitted in December 1991 for publication in *IEEE Transactions on Consumer Electronics*
- [WAN04] R. Wang, W. Jin, L. Wu, « A novel video caption detection approach using multi-frame integration », in: *Proceedings of the International Conference on Pattern Recognition*, Cambridge, UK, August 2004, pp. 449–452
- [WIL08] Dr. Willard R. Daggett, « Color in an Optimum Learning Environment », *International Center for Leadership in Education*, USA, Mars 2008
- [YAN89] S.D. Yanowitz and A.M. Bruckstein, « A New Method for Image Segmentation », *Computer Vision, Graphics and Image Processing*, vol. 46, no. 1, pp. 82-95, Apr. 1989.
- [YE07] Q. Ye et al., « Text detection and restoration in natural scene images », *J. Vis. Commun.* (2007), doi:10.1016/j.jvcir.2007.07.003
- [ZLI] www.zlib.net





Techniques de traitement d'image dédiées aux flux des diapositives lors d'une visioconférence

Ismail KTATA

الخلاصة: إن نشر خدمات الوسائط المتعددة عبر شبكة الانترنت قد أثار مشكلة تتعلق بالكمية الكبيرة من البيانات التي سيتم تحليلها وبعثها عبر الشبكة. و في هذا السياق، يتمثل عملنا في تصميم تقنية تسمح باستقبال و تحليل المعلومات المتضمنة لعرض الأستاذ المحاضر. لذلك، قمنا بتقديم حل يلاءم خصوصيات هذا النوع من المعلومات و ذلك انطلاقاً من التقنيات المستخدمة حالياً. و قد تمكن هذا الحل من تلبية أهدافنا من خلال حسن كشف و تحديد التغيرات و الحركية الطارئة، خفض كمية المعلومات المبعوثة و ضمان صور ذات نوعية مقبولة و مقروءة عند الاستقبال.

Résumé : Les services multimédia déployés sur IP évoquent une problématique qui porte sur l'importance du volume des données à traiter et à transmettre. Dans ce contexte se situ notre travail. Il s'agit de concevoir et valider un algorithme (codec) qui devra permettre, l'acquisition et le traitement du flux VGA/XGA contenant la présentation d'un conférencier lors d'une application de visioconférence. Pour cela nous avons présenté une solution adéquate aux spécificités de notre flux en s'inspirant des systèmes présents actuellement. Cette solution a pu répondre à notre objectif d'assurer la diffusion du flux des diapositives avec un débit relativement faible et un minimum de calcul. Bien entendu, cette technique permet aussi de garantir, à la réception, un rendu visuel des images acceptable par les utilisateurs.

Abstract: The multimedia services deployed over IP evoke a problem that concerns the large volume of data to be processed and transmitted. In this context, our work consists to design and validate a processing algorithm that will allow the acquisition, processing and dissemination of the VGA / XGA flow that contents the presentation of a lecturer in the case of a visioconference application. For that, we have presented an adequate solution to the specifics of our flow based on codecs currently used. This solution has been able to meet our goals of disseminating the flow of slides with a low speed and minimum calculation while guaranteeing in the reception an acceptable images quality.

المفاتيح: فيزيوكنفرنس، ضغط الصورة، وكشف تغير المشهد، الكشف عن النص، الترميز، جايباخ، بينج

Mots clés: Visioconférence, compression d'image, changement de plan, extraction de texte, CODEC, JPEG, PNG

Key-words: Visioconference, image compression, scene and slide change detection, text detection, CODEC, JPEG, PNG