

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Saad Dahlab de Blida



Faculté des sciences
Département informatique
Mémoire pour l'obtention d'un diplôme de Master
En informatique
Option : génie logiciel
Sujet :

DÉTECTION DE LA COPIE DE LA VIDÉO BASÉE SUR LE CONTENU

Présenté par : Meraga Mohamed Fateh Promotrice : M^{me} Ait Saadi Karima
Hadj Ahmed Saleh Edine Co-promoteur : M. Zair Mustapha
Organisme d'accueil : Centre de Développement des Technologies Avancées
CDTA

Soutenu le : 25/10/2013...devant le jury composé de :

Président (e) *Bannouan*

Examineur *Baouia*

Examineur

Promotion 2012-2013

MA-004-167-1

REMERCIEMENTS

TOUT D'ABORD NOUS REMERCIONS ALLAH DE NOUS AVOIR DONNÉ LA SANTÉ ET LA VOLONTÉ POUR MENER À BIEN CE MODESTE TRAVAIL.

NOUS REMERCIONS VIVEMENT NOTRE PROMOTRICE M^{ME} AIT SAADI KARIMA D'AVOIR ACCEPTÉ DE NOUS ENCADRÉ ET PROPOSÉ CE SUJET, NOUS EXPRIMONS TOUTS NOTRE GRATITUDE ET NOS S'INSÈRE REMERCIEMENTS POUR TOUT SON EFFORT, COOPÉRATION, AIDE, PATIENCE, ENCOURAGEMENT ET VAILLANCE POUR LA RÉALISATION DE CE TRAVAILLE, PRIENT DIEU DE LEUR DONNÉ TOUTE LA FORCE POUR CONTINUER LEUR NOBLE MISSION.

NOUS REMERCIONS MONSIEUR ZAIR MUSTAPHA POUR SES CONSEILLES ET SON AIDE.

NOUS TENONS À REMERCIER LES MEMBRES DE JURY DE NOUS AVOIR FAIT L'HONNEUR D'ÊTRE EXAMINATEURS TOUTS EN APPORTANT LEURS REMARQUES ET LEURS CONTRIBUTIONS À L'ENRICHISSEMENT DE CE MÉMOIRE.

NOUS REMERCIONS ENCORE TOUS LES ENSEIGNANTS DU DÉPARTEMENT D'INFORMATIQUE, POUR NOUS AVOIR FORMÉ TOUT AU LONG DE NOTRE CURSUS.

Mahamed Fateh et Saleh edine

Dédicaces

Je dédié ce modeste mémoire à:

Ma chère mère **Nacira** et mon cher père **Fodile** à qui je dois mes sucées, que dieu le tout puissant protège ma mère et que je puisse lui rendre un petit bout de son sacrifice

Mes chères sœurs **Monia, Karima, Lamia**

Mon frère **Kamel**

Je les remercie pour leurs soutiens tout au long de mes études.

Mes neuves et nièces surtout **Rania, Khadidja, Chaima.**

Toute ma grande famille

Mon binôme et très chère ami **SALEH EDINE**

Mes meilleurs amis **Nour et Nacer**

La personne qui ma toujours inspiré **HNSNA**

Mes amis **Abd el wahab, Hamza, ZinEdine, Houcine, Youcef, Hamza** avec qui j'ai passé des moments formidables durant mes années d'études

Tout personne qui m'a aidé dans ma vie de prés ou de loin

Mohamed Fateh

Dédicaces

Je dédié ce modeste mémoire :

A mon père **Houcine** ,à ma mère **Louisa** ,à mon chère frère
Tarek ,à toute ma grande famille.

A mon binôme et très chère ami **Mohamed Fateh**

A mon meilleurs ami **Nacer**.

A mes amis **Abd el wahab** , **Hamza** , **ZinEdine** , **Houcine** , **Mohamed**
avec qui j'ai passé des moments formidables durent mes années
d'études

A toute personne qui m'a aidé dans ma vie de prés ou de loin

Saleh edine

Résumé

L'apparition et le développement des nouvelles technologies numériques dans les appareils d'acquisition numériques et le multimédia a conduit à la création et au partage de grandes quantités de données vidéo. Ce qui a conduit aussi à la création des fraudes et des copies multiples d'un même contenu vidéo. Il faut dire que plus que 50% des vidéos sur Youtube sont de la musique et des divertissements, et ils sont stockés et indexés sous le même nom. La différence de la durée d'une vidéo par quelques secondes, la qualité de la vidéo et la source à partir de laquelle est obtenue la vidéo sont les quelques différences entre les vidéos avec un contenu similaire. En conséquence, Il ya un besoin croissant pour la protection des droits de propriété intellectuelle de contenu multimédia numérique. Dans la littérature, on trouve un certain nombre de mécanismes qui peuvent être utilisés pour protéger le contenu multimédia. Ces derniers comprennent des méthodes telles que le cryptage, le hachage cryptographique, le tatouage numérique ou la détection de copy basée sur le contenu (CBCD : Content Based Copy Detection).

La problématique abordée dans ce mémoire s'insère dans ce contexte, il s'agit de développer un système efficace de la détection de copy basée sur le contenu afin de détecter la réplique visuelle de la vidéo.

Mots clés : Traitement d'images, tatouage numérique, protection du contenu multimédia.

Abstract

The appearance and development of new digital technologies in the digital acquisition devices and multimedia has led to the creation and sharing of large amounts of video data. This has also led to the creation of fraud and multiple copies of the same video content. I must say that more than 50% of videos on YouTube are music and entertainment, and they are stored and indexed in the same name. The difference in the duration of a video by a few seconds, the video quality and the source from which is obtained the video are some differences between videos with similar content. As a result, there is an increasing need to protect intellectual property rights of digital media content. In the literature, there are a number of mechanisms that can be used to protect the media. These include methods such as encryption, cryptographic hash, digital watermarking or CBCD (Content Based Copy Detection).

The problem addressed in this thesis is to develop a simple and efficient system for detection of copy-based content to detect visual replica of the video

Sommaire

Introduction générale	01
Chapitre I : Généralités	
1. Introduction	03
2. Généralités sur la détection de copie basée sur le contenu	04
2.1. Définition de la détection de copie	04
2.2. Applications de détection de copie basée sur le contenu	05
3. État de l'art de la détection de copie basée sur le contenu.....	05
3.1. Les méthodes basées les caractéristiques Spatiale	06
3.1.1. Les méthodes basées sur les mesures ordinales	06
3.1.2. Les méthodes basées sur l'histogramme de couleur	08
3.2. Les méthodes basées sur les caractéristiques temporelles	08
3.3. Les méthodes basées sur les caractéristiques Spatiotemporelles.....	09
4. Conclusion	09
Chapitre II : L'opérateur VLBP	
1. Introduction	10
2. LBP de base.....	10
3. Volume Locale Patterns Binaires.....	11
3.1. VLBP de base.....	12
3.2. VLBP de rotation.....	14
3.3. VLBP de rotation uniforme.....	16
4. Conclusion	17
Chapitre III : La norme H.264/AVC	
1. Introduction	18
2. Structure du codeur H.264	19
2.1. Terminologie	19
2.2. Le fonctionnement du codec H.264	20
2.3. Prédiction intra-image	22
2.3.1. Prédiction des blocs 4x4 de luminance	22
2.3.2. Prédiction des blocs 16x16 de luminance	23
3. Etat de l'art de la détection de copie basée sur le contenu dans la norme H.264...24	
3.1. Génération de la signature	24

3.2. Le processus de comparaison des signatures.....	26
4. Conclusion	27

Chapitre IV : Conception et réalisation

1. Introduction.....	28
2. Système proposé.....	28
2.1. Extraction des caractéristiques	29
2.1.1. Extraction des caractéristiques à partir de tous le contenu des images...29	
2.1.1.1. Segmentation de la vidéo en images.....	30
2.1.1.2. Application de l'opérateur VLBP.....	30
2.1.1.3. Construction des histogrammes	31
2.1.2. Extraction des caractéristiques à partir des zones textures des images...32	
2.1.2.1. Découper en GOP.....	34
2.1.2.2. Application de l'opérateur VLBP.....	35
2.2. Comparaison des histogrammes	35
2.2.1. Normalisation des histogrammes	35
2.2.2. Calculer la distance entre les histogrammes	36
2.2.3. La décision	36
3. Présentation du logiciel.....	36
3.1. Le bouton « Ouvrir ».....	37
3.2. Le bouton « Compresser »	38
3.3. Le Bouton « Identifier »	39
3.4. Le Bouton « Visualiser ».....	40
3.5. Le Bouton « Enregistre ».....	41
3.6. Le Bouton « Fermer ».....	42
4. conclusion	42

Chapitre V : Tests et résultats

1. Introduction.....	43
2. Les attaque	44
2.1. Modification du contenu d'une image de la séquence	44
2.2. Insertion de motif	45
2.3. Perte d'image	45

2.4. Recadrage.....	45
2.5. Décalage (shift)	45
2.6. Retournement (Flip)	46
2.7. Zoom.....	46
2.8. Contraste.....	46
2.9. Luminance.....	47
3. résultats des tests	47
3.1. Détection basée sur tous le contenu de la vidéo	48
3.2. Détection basée sur les zones texturées de la vidéo	49
4. Conclusion	51
Conclusion Générale	52

Bibliographie

Liste des Figures

Figure I.1 Architecture commune pour les systèmes de détection de copie vidéo....	4
Figure I.2 Classification des méthodes CBCD existantes.....	6
Figure I.3 Principe de la méthode de Mohan.....	7
Figure II.1 Exemple de calcul du code LBP.....	11
Figure II.2 Principe du <i>VLBP</i> de base : (a) Volume de texture dynamique (volume à droite $L = 1$, le volume à gauche $L = 2$) ; (b) ensembles des pixels voisins symétriques circulaires ($R = 1$ et $P = 8$) et (c) les points voisins sur la surface du cylindre ($P = 4$).....	12
Figure II.3 exemple de calcul $VLBP_{1,4,1}$	14
Figure III.1 Historique des différents standards vidéo ISO et ITU.....	18
Figure III.2 Schéma du codeur H.264.....	20
Figure III.3 Schéma du décodeur H.264.....	21
Figure III.4 Génération de la signature dans [6].....	25
Figure III.5 Extractions des signatures (a) vidéo r requête et (b) la vidéo cible.....	25
Figure III.6 processus de comparaison.....	26
Figure IV.1 Les étapes de notre système de détection de copie.....	28
Figure IV.2 Les étapes de l'extraction des caractéristiques à partir des images.....	30
Figure IV.3 Histogramme des valeurs <i>VLBP</i> de base.....	31
Figure IV.4 Des images de la vidéo concaténée.....	31
Figure IV.5 Histogramme des valeurs <i>VLBP</i> de rotation uniforme.....	32
Figure IV.6 Exemple de sélection des MBs de types Intra_4x4 et Inter_4x4 (les zones texturées) la vidéo akiyo.....	33
Figure IV.7 Les zones texturées dans la première image (I-frame) et les zone correspondent dans les deux autres images de la sous séquence.....	33
Figure IV.8 Les étapes de l'extraction de la signature dans les zones texturées.....	34
Figure IV.9 Structure d'un GOP.....	34
Figure IV.10 Processus de comparaison des histogrammes.....	35
Figure IV.11 vue générale du système de détection de copie basée sur le contenu dans la norme H.264/AVC.....	37
Figure IV.12 Le bouton Ouvrir.....	37

Figure IV.13 paramètres d'affichage de la séquence vidéo.....	38
Figure IV.14 Le bouton « Compresser ».....	38
Figure IV.15 La rubrique « Compression ».....	39
Figure IV.16 Le bouton « Identifier ».....	39
Figure IV.17 Comparaison des vidéos.....	40
Figure IV.18 Le bouton « Visualiser ».....	40
Figure IV.19 Histogramme des caractéristiques de la vidéo requête.....	41
Figure IV.20 Le bouton « Enregistre ».....	41
Figure IV.21 Identification d'utilisateur.....	41
Figure IV.22 Le bouton «fermer».....	42
Figure V.1 La vidéo d'origine.....	44
Figure V.2 Des images de la vidéo requête (modification du contenu d'une image de la séquence).....	44
Figure V.3 Des images de la vidéo requête (insertion du texte).....	45
Figure V.4 Des images de la vidéo requête (recadrage).....	45
Figure V.5 Des images de la vidéo requête (décalage).....	45
Figure V.6 Des images de la vidéo requête (retournement).....	46
Figure V.7 Des images de la vidéo requête (zoom).....	46
Figure V.8 Des images de la vidéo requête (augmentation de contraste).....	47
Figure V.9 Des images de la vidéo requête (diminution de contraste).....	47
Figure V.10 Des images de la vidéo requête (augmentation de luminance).....	47
Figure V.11 Des images de la vidéo requête (diminution de luminance).....	47
Figure V.12 Les distances entre les histogrammes de la vidéo d'origine et les vidéos modifiées en utilisant tout le contenu pour l'extraction de la signature.....	49
Figure V.13 Les distances entre les histogrammes de la vidéo d'origine et les vidéos modifiées en utilisant tout les zones texturées pour l'extraction de la signature.....	51

Liste des Tableaux

Tableau III.1	Les modes de prédiction des blocs de taille 4x4 de luminance.....	23
Tableau III.2	Les modes de prédiction des blocs de taille 16x16 de luminance.....	23
Tableau V.1	Les paramètres utilisés dans la norme H.264/AVC.....	44
Tableau V.2	Les distance entre les histogrammes de la vidéo d'origine et les vidéos modifiées en utilisant tout le contenu pour l'extraction de la signature.....	48
Tableau V.3	Les distance entre les histogrammes de la vidéo d'origine et les vidéos modifiées en utilisant tout les zones texturées pour l'extraction de la signature.....	50

Introduction générale

Introduction générale

L'exploitation illégale des documents vidéo sur Internet est devenue un enjeu majeur pour l'industrie audiovisuelle en général et cinématographique en particulier. Pour lutter contre ce phénomène de piraterie, les ayants-droit ont besoin de nouveaux outils permettant d'assurer une veille sécuritaire sur leurs actifs numériques. De tels outils s'appuient sur des technologies permettant l'identification de copies vidéo illégalement partagées sur Internet. Ces technologies doivent être à la fois pertinentes et robustes aux différentes transformations que peuvent subir les copies vidéo par rapport à leurs originaux.

Le tatouage numérique (ou watermarking) est une approche qui a fortement émergé depuis une dizaine d'années. Le tatouage se propose de dissimuler au sein d'un document un message (marque, empreinte digitale), tout en laissant le document marqué exploitable. Le message est unique à la vidéo et dans la plupart de temps invisible. La détection de copie devient une question de recherche du contenu vidéo pour cette information cachée. Mais cette méthode présente plusieurs inconvénients. L'inconvénient major est que les vidéos anciennes qui ont été libérées sans empreintes ne peuvent pas bénéficier de ce processus. D'autre part beaucoup de transformations vidéo affectent l'empreinte digitale. Si la vidéo est copiée et ré-encodée d'une manière qui modifie l'empreinte digitale, alors ce dernier n'est plus utile à des fins de détection de copie [14].

Une autre approche consiste à extraire des caractéristiques distinctives du contenu. Si les mêmes caractéristiques peuvent être trouvées dans le contenu des deux vidéos, alors l'une des vidéos peut être une copie de l'autre. Ceci est connu comme la détection de copies basée sur le contenu (DCBC) [21][22].

Comme les séquences vidéo sont généralement compressées pour être transmises ou stockées, il existe plusieurs standards de compression vidéo tels que le MPEG1, MPEG2, H.263 et H.264. Par conséquent, la protection des données dans ces normes est indispensable.

La problématique que nous posons dans ce mémoire est de développer un système simple et efficace de la détection de copie basée sur le contenu afin de détecter la réplique visuelle des vidéos compressées par la norme H.264/AVC.

La structure de ce mémoire reflète la logique de nos objectifs. Il comprend cinq chapitres qui sont présentés comme suit :

- Dans le premier chapitre, nous présentons les concepts de base concernant la détection de copie vidéo basée sur le contenu, ainsi un bref état de l'art.
- Le second chapitre, comporte la méthode VLBP que nous utilisons pour l'extraction de la signature de la vidéo.
- Le troisième chapitre contient un le principe générale de fonctionnement de la norme H.264, et un état de l'art des systèmes de détection de copie video compressé par cette dernière.
- Une présentation détaillée de notre système de détection de copie basée sur le contenu pour les vidéos compressées par la norme H.264/AVC, ainsi une présentation de notre logiciel et les différents outils permettant la communication avec le logiciel fera l'objet de quatrième chapitre.
- Le cinquième chapitre, comporte les différents testes et expérimentations réalisés afin de valider nos résultats obtenus.
- Enfin, nous terminerons par une conclusion générale en énonçant quelques perspectives.

Chapitre I : Généralités

1.Introduction

Le but de la détection de la copie vidéo [3] est de localiser les segments dans une vidéo requête qui sont copiées ou modifiées à partir d'une archive de vidéo de référence. Habituellement, les segments copiés sont soumis à diverses transformations comme Cam enregistrement, image dans l'image, unere-compression, recadrage, le changement de contraste, ...etc. Certaines de ces transformations sont intrinsèques au processus de création du vidéo, par exemple, Les effets de blocs de l'encodage, le débit binaire et les changements de résolution, ...etc. d'autres sont introduites intentionnellement à des fins diverses. Toutes ces transformations rendent la tâche de détection plus difficile.

La détection de copie vidéo est essentielle pour de nombreuses applications, par exemple, la découverte de l'infraction de copyright de contenu multimédia, le suivi du temps d'antenne commercial, ...etc. Généralement, il ya deux approches complémentaires pour la détection de copie de la vidéo: le tatouage numérique de la vidéo et la détection de copies à base de contenu (DCBC)[3].

Le tatouage numérique a été proposé pour la protection du droit d'auteur et les images numériques [4]. L'idée de base consiste à insérer une information (signature, marque) dans le média (image, vidéo ou audio). Aujourd'hui, en ce qui concerne la vidéo, tous les films DVD, jeux vidéo, etc. cachent des informations prouvant la propriété intellectuelle du propriétaire[1].

L'inconvénient est que ces informations peuvent être fragiles aux transformations usuelles (ré-encodage, le changement de la résolution et débit binaire). Par exemple, les données cachées sur un film seront probablement perdus lorsque la vidéo est compressée et envoyée à un site web de partage de vidéos. Comme solution à cet inconvénient, les approches de détection de copie basée sur le contenu (DCBC) ont vu le jour [1][4][5]. Cette solution est en fait, un champ de recherche complémentaire pour l'approche basée sur le tatouage numérique.

2. Généralités sur la détection de copie basée sur le contenu

2.1. Définition de la détection de copie

L'idée principale de la détection de copie basée sur le contenu (CBCD) est que le média lui-même est la signature, autrement dit, le média qui est dans notre cas la vidéo contient des informations uniques qui peuvent être utilisées pour détecter les copies. L'avantage principal de la détection de copie vidéo basée sur le contenu est le fait que l'extraction de la signature peut être faite après que la vidéo soit distribuée.

Un système de détection de copie vidéo basée sur le contenu (figure I.1) extrait les signatures de la vidéo d'origine ainsi que celle de la vidéo de test. Par la suite, Il compare les deux signatures afin de déterminer si le media de test est une copie de l'origine [2].

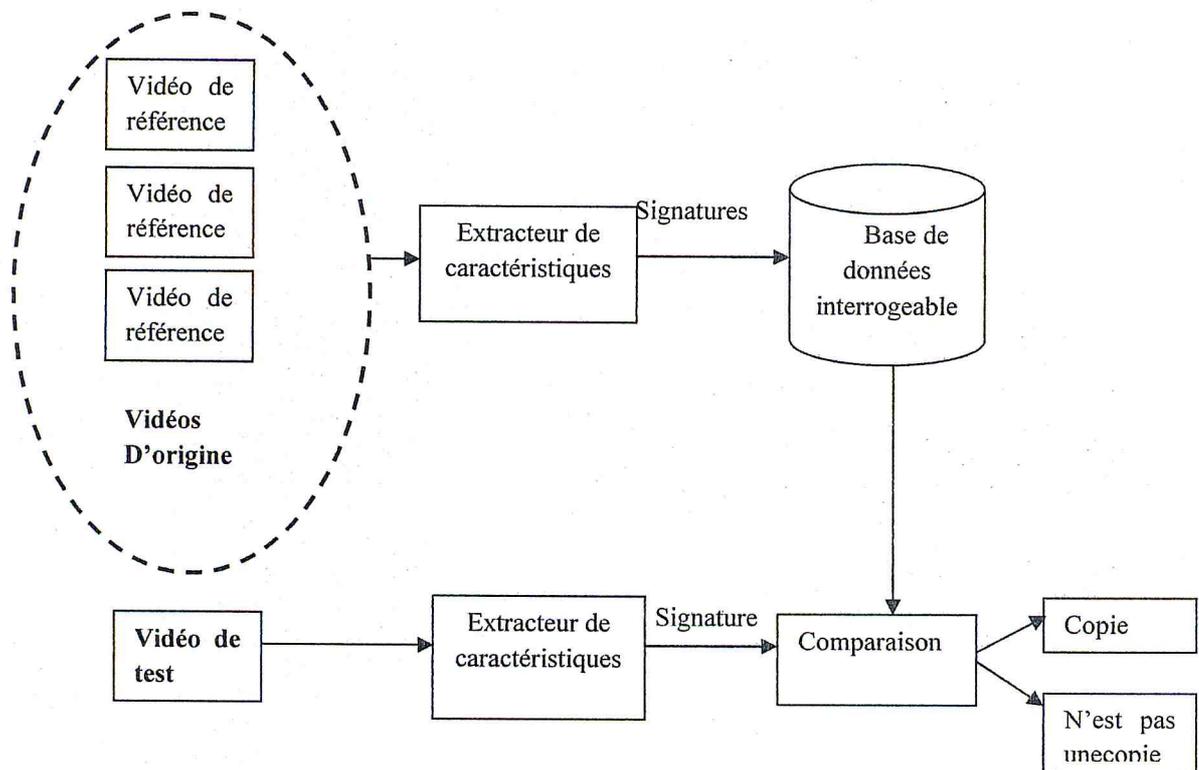


Figure I.1 Architecture commune pour les systèmes de détection de

2.2. Applications de détection de copie basée sur le contenu

Les systèmes de détection de copie des vidéos peuvent être utilisés à différentes usages [21] :

- Surveillance d'une chaîne de télévision pour détecter les copies. Cela permet à une entreprise qui a acheté des espaces publicitaires de s'assurer qu'elle obtient le temps de communication qu'elle a payé. Dans ce cas, on peut supposer que l'intégrité du contenu sera disponible et non transformé [sw03] ;
- Elimination des redondances dans une base de données ;
- Protection du contenu vidéo face aux copies.

3. État de l'art de la détection de copie basée sur le contenu

Les méthodes existantes pour CBCD extraient généralement un certain nombre de traits pertinents (appelés signatures ou empreintes digitales) à partir d'images ou d'un flux vidéo et cette signature peut être extraite à partir des caractéristiques spatiales comme les mesures basées sur la luminance (mesures ordinales), couleurs comme l'histogramme de couleurs, caractéristiques temporelles comme le vecteur de mouvement ou encore une combinaison de ces caractéristiques. D'où les différentes approches de détection de copie basées sur le contenu [6]. Ces approches peuvent être classifiées selon le schéma illustré sur la figure I.2.

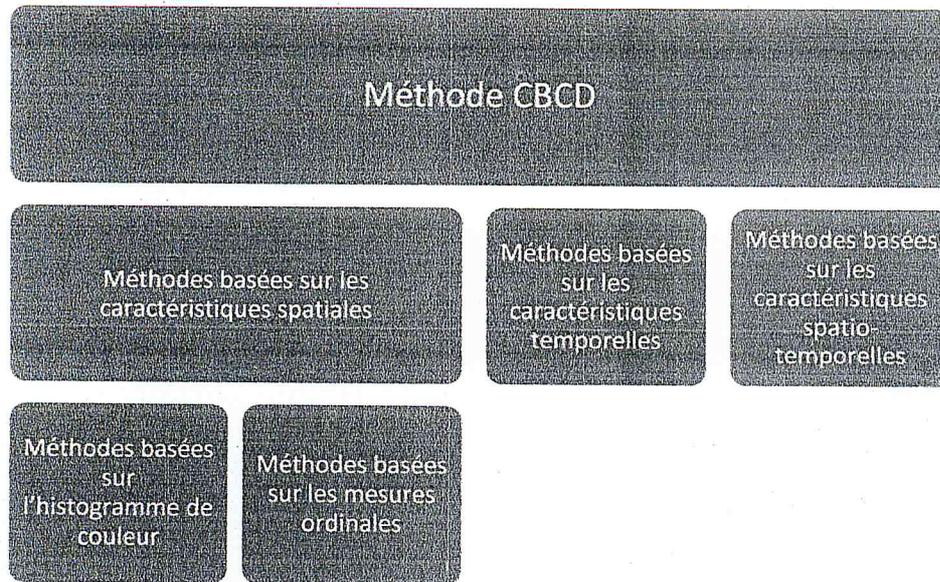


Figure I.2 Classification des méthodes CBCD existantes.

3.1. Les méthodes basées les caractéristiques Spatiales

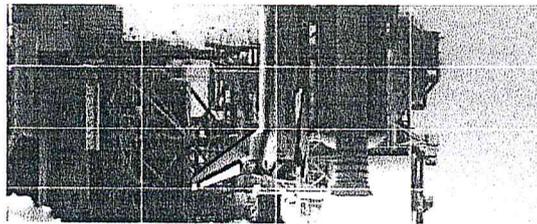
Les méthodes basées sur les caractéristiques spatiales peuvent être regroupées en 2 classes :

3.1.1. Les méthodes basées sur les mesures ordinales

L'une des premières méthodes CBCD basée sur les caractéristiques spatiales a été proposée par Bhat et al. [7] pour les images numériques, puis Mohan et al. [2] l'ont adapté aux séquences vidéo. La mesure ordinale de chaque image de la séquence qui représente la moyenne des intensités des niveaux de gris a été calculée, puis concaténée.

Le principe de l'approche de Mohan est comme suit (figure I.3) [2] : Subdivision de chaque image en N blocs de taille $M \times M$, Calcul les intensités moyennes R_i de chaque bloc, rangement des R_i dans l'ordre croissant ou décroissant pour générer une signature de l'image (1) :

$$S(t) = (R_1, R_2, \dots, R_n) \quad \text{I.1}$$



(a)

74	128	46	133	5	10	0	11
62	78	58	145	3	6	1	12
60	82	116	157	2	7	9	13
87	70	214	167	8	4	15	14

(b)

(c)

Figure 3 Principe de la méthode de Mohan : (a) l'image divisé en blocks ; (b) les intensités moyennes des blocs; (c) rangement des intensités moyennes.

Lee et al. [8] ont conçu une méthode de détection de copie dont la signature est basée sur le centroïde d'orientations du gradient. Cette méthode a été prétendue être robuste à la majorité des attaques de traitement d'images usuelles telles que le redimensionnement, compression, etc.

Joy et al. [1][4][13] ont proposé une méthode basée sur le point d'intérêt. Cette approche est souvent associée à l'utilisation d'images clés sur lesquels les points d'intérêt sont calculés.

Kim et al. [4] ont critiqué la taille de la partition utilisée dans [9], indiquant que les procédés antérieurs utilisant des partitions de taille plus de 2×2 ne considèrent pas les effets des changements asymétriques dans les valeurs de la partition. Pour cela, ils ont utilisé des mesures ordinales sur des blocs de taille 2×2 pour augmenter la robustesse de la méthode publiée dans [9].

3.1.2. Les méthodes basées sur l'histogramme de couleur

L'histogramme de couleur a été proposé à l'origine par Naphade et al. [2]. Ils ont proposé d'utiliser des histogrammes YUV comme signature de chaque image de la séquence et d'employer l'intersection des histogrammes comme une mesure de distance entre deux signatures. Un histogramme YUV est calculé pour chaque image de la vidéo. Ils utilisent 32 classes pour présenter la composante de luminance Y, et 16 classes pour chaque composante de chrominance U et V. Donc, la signature d'un clip vidéo est représentée par l'histogramme concaténé de 64 classes.

Lienhart et al. [9] ont proposé une méthode où un vecteur de cohérence de couleur a été utilisé pour caractériser les images clés d'un clip vidéo. Sanchez, Binefa et al. [6] ont proposé uniquement l'utilisation de la composante de couleurs pour la détection de copie.

Les signatures à base de couleurs sont très faibles. Un codage ou un ré-encodage peuvent souvent causer des variations globales de couleur pour les paramètres de quantification. Une autre faiblesse des signatures basées sur la couleur est que les couleurs similaires peuvent être présentes en très différents clips vidéo. Un exemple : deux clips vidéo avec des scènes différentes de l'océan. Les histogrammes résultants auraient une grande intersection parce que chaque histogramme serait principalement bleu, mais ils ne sont pas des copies. En général, les signatures de couleur ne sont pas robustes à des changements de couleurs communes et ils ne sont pas suffisamment preuves de discrimination [5].

3.2. Les méthodes basées sur les caractéristiques temporelles

Un certain nombre de techniques basées sur la nature temporelle des séquences vidéo ont été également proposées. Indyk et al. [10] ont proposé quelques approches de CBCD en exploitant les caractéristiques temporelles de la vidéo. Radhakrishnan et al. [11] ont utilisé une méthode d'extraction de signature vidéo qui repose sur des projections de différentes images entre les images consécutives d'une vidéo. Ils ont affirmé que leur méthode est robuste aux traitements du signal tels que les changements de la luminance, la compression, le changement de résolution.

Indyk et al. [10][13] ont proposé une méthode dont le principe est comme suit : tout d'abord, une séquence vidéo est divisée en images clés. Ensuite, la durée de chaque image clé est prise comme une signature temporelle, et la séquence des durées des images forment la signature de la vidéo.

Hampapuret al. [2] ont estimé les vecteurs de mouvement de chaque block, puis ils ont quantifié les quatre orientations pour former une signature de mouvement.

3.3. Les méthodes basées sur les caractéristiques Spatiotemporelles

Les techniques basées sur la correspondance de séquences spatio-temporelles sont robustes à de nombreuses distorsions causées par la numérisation et la compression.

Geert Willems et al. [12] ont proposé une nouvelle méthode de détection de copie de vidéo basée sur les caractéristiques spatio-temporelles. Où ils ont montré que l'utilisation de la combinaison des caractéristiques locales spatiales et temporelles augmente les performances d'un système de détection de copie.

Adarsh Nagaraja [19] a proposé une technique de détection de copie basée sur l'analyse des scènes et la texture. Il a utilisé l'opérateur local spatio-temporelle LBP-TOP (LBP : Linear Binary Pattern).

4. Conclusion

Dans ce chapitre nous avons présenté les concepts nécessaires pour se familiariser avec les notions de base de traitements d'images en général et le domaine de détection de copie en particulier. Nous avons cité quelques approches dédiées à la détection de copie vidéo basée sur le contenu qui permettent au propriétaire d'une vidéo d'assurer ces droits intellectuelles. Dans le prochain chapitre, nous allons introduire l'opérateur VLBP que nous avons adopté pour notre système de détection de copie basée sur le contenu.

Chapitre II : L'opérateur VLBP

1. Introduction

L'extraction de la signature à partir d'une vidéo est un grand problème pour les propriétaires des vidéos qui veulent protéger leurs vidéos du piratage et de modifications. Il est nécessaire que le système d'extraction de la signature soit robuste aux différentes attaques telles que l'insertion de logos dans la vidéo, changement de la luminance, renverser une image...etc.

Chaque méthode à ses points forts et ses inconvénients. Les méthodes basées sur l'histogramme de couleur leur inconvénient est que ces méthodes sont faibles aux changements globaux dans les couleurs de la vidéo telle que la compression affecte la qualité de la vidéo entière. Alors que les méthodes basées sur les caractéristiques temporelles ignorent l'information spatiale [2].

Il existe d'autres techniques qui permettent l'extraction des caractéristiques à partir d'un voisinage local telles que les méthodes basées sur l'opérateur LBP et LBP-TOP. L'opérateur LBP est généralement employé dans les applications telles que la cartographie de la texture, la reconnaissance de l'expression du visage, la détection d'objets dans les images [15].

Les principaux avantages de l'opérateur LBP est l'invariance aux changements de niveaux de gris monotones et l'efficacité de calcul. Ce qui le rend approprié pour les tâches de traitements d'images et la détection de copie [17].

2.1. LBP de base

L'opérateur LBP est défini comme une mesure de texture invariante, son application dans le traitement d'images a conduit à d'excellentes performances dans la classification des différents types de textures [17] [15]. La façon la plus simple pour calculer la valeur LBP est de prendre un bloc de 3x3 pixels, un code binaire est calculé par seuillage du niveau de gris de pixel du centre avec les P pixels de voisins. Tous les P voisins prennent une valeur 1 si leur valeur est supérieure ou égale au pixel du centre et 0 sinon. Après, multiplier le code binaire avec des poids LBP, puis calculer la somme du résultat de la multiplication pour obtenir la valeur LBP. La valeur de LBP est calculée selon l'équation suivante :

$$LBP(x_c, y_c) = \sum_{n=0}^{P-1} 2^n \text{sgn}(g_n - g_c) \quad (\text{II.1})$$

Où : (x_c, y_c) sont les coordonnées du pixel central

g_c : représente le niveau de gris du pixel central dans un voisinage P .

g_n : représente les niveaux de gris des P pixels voisins.

La fonction $\text{sgn}(x)$ est définie comme suit:

$$\text{sgn}(g_n - g_c) = \begin{cases} 1 & \text{si } (g_n - g_c) \geq 0 \\ 0 & \text{sinon} \end{cases} \quad (\text{II.2})$$

La figure II.1 illustre l'algorithme de calcul d'une étiquette par l'opérateur LBP de base [15].

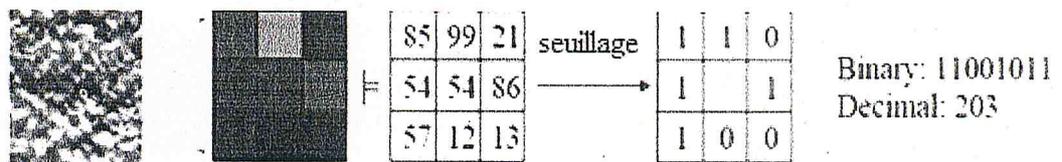


Figure II.1 Exemple de calcul du code LBP.

3. Volume Local Patterns Binaires

L'analyse des dynamiques textures à besoin de la combinaison de l'apparence et du mouvement ce qui n'est pas possible dans l'opérateur LBP. Zhao et al. [17] ont proposé d'utiliser l'opérateur VLBP (Volume Local Binary Patterns) pour résoudre les problèmes d'analyse des textures dynamiques.

3.1. VLBP de base

Pour extrapoler l'opérateur LBP à l'analyse des textures dynamiques (vidéo), il est nécessaire de définir la texture dynamique V dans un voisinage local d'une séquence de texture dynamique monochrome comme la distribution conjointe des niveaux de gris des $(3P+3)$ pixels d'image [17].

$$V = v(g_{t_{c-L},c}, g_{t_{c-L},0}, \dots, g_{t_{c-L},P-1}, g_{t_c,c}, g_{t_c,0}, \dots, g_{t_c,P-1}, g_{t_{c+L},0}, \dots, g_{t_{c+L},P-1}, g_{t_{c+L},c}) \quad (II.3)$$

Où $g_{t_c,c}$ correspond au niveau de gris du pixel central du volume local voisin. Les paramètres $g_{t_{c+L},c}$ et $g_{t_{c-L},c}$ correspondent aux niveaux de gris des pixels centraux dans les images voisines antérieures et postérieures à l'intervalle de temps L (figure II.2.a) et $g_{t,p}$ (avec $t=t_{c-L}, t_c, t_{c+L}$ et $p=0, \dots, P-1$) correspondent aux niveaux de gris des P pixels placés sur un cercle de rayon R ($R > 0$) dans l'image t (figure II.2.b) qui forment un ensemble de pixels voisins symétriques circulaires (figure II.2.c).

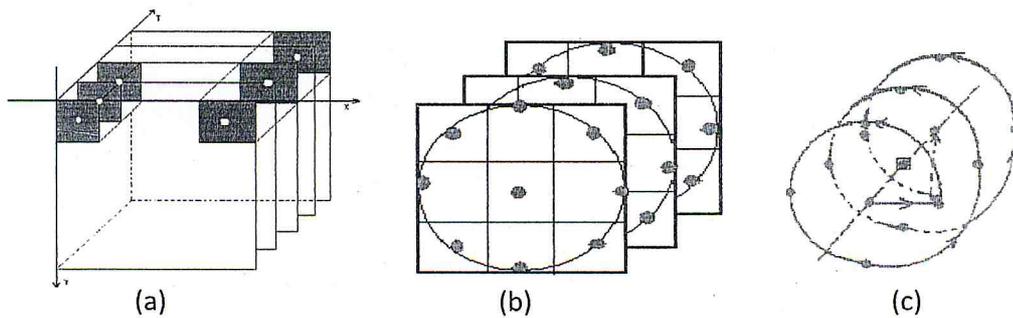


Figure II.2 Principe du VLBP de base : (a) Volume de texture dynamique (volume à droite $L = 1$, le volume à gauche $L = 2$) ; (b) ensembles des pixels voisins symétriques circulaires ($R = 1$ et $P = 8$) et (c) les points voisins sur la surface du cylindre ($P = 4$).

Si les coordonnées de $g_{t,c}$ sont (x_c, y_c, t_c) , les coordonnées de $g_{t,c,p}$ et ceux de $g_{t,c \pm L,p}$ sont calculées de la manière suivante :

$$((x_c - R \sin(2 \pi p / P), (y_c + R \cos(2 \pi p / P)), t_c). \quad (II.4)$$

et les coordonnées de $g_{t,c \pm L,p}$ sont :

$$((x_c - R \sin(2 \pi p / P), (y_c + R \cos(2 \pi p / P)), t_c \pm L). \quad (II.5)$$

La valeur du VLBP dans un voisinage local V est obtenue en soustrayant le niveau de gris du pixel du centre ($g_{t,c}$) aux niveaux de gris des pixels ($g_{t,p}$) appartenant à la région symétrique circulaire, la distribution devient [17]:

$$V \approx v(g_{t,c-L,c} - g_{t,c}, g_{t,c-L,0} - g_{t,c}, \dots, g_{t,c-L,P-1} - g_{t,c}, g_{t,c,0} - g_{t,c}, \dots, g_{t,c,P-1} - g_{t,c}, g_{t,c+L,0} - g_{t,c}, \dots, g_{t,c+L,P-1} - g_{t,c}, g_{t,c+L,c} - g_{t,c}). \quad (II.6)$$

Pour obtenir l'invariant par rapport à la mise à l'échelle de niveau de gris, seuls les signes des différences sont pris en considération au lieu des valeurs exactes de chaque pixel [17]:

$$V \approx v(\text{sgn}(g_{t,c-L,c} - g_{t,c}), \text{sgn}(g_{t,c-L,0} - g_{t,c}), \dots, \text{sgn}(g_{t,c-L,P-1} - g_{t,c}), \text{sgn}(g_{t,c,0} - g_{t,c}), \dots, \text{sgn}(g_{t,c,P-1} - g_{t,c}), \text{sgn}(g_{t,c+L,0} - g_{t,c}), \dots, \text{sgn}(g_{t,c+L,P-1} - g_{t,c}), \text{sgn}(g_{t,c+L,c} - g_{t,c})). \quad (II.7)$$

Où le paramètre sgn est la fonction signe définie dans l'équation (II.2).

Pour simplifier l'expression de V , l'utilisation de $V = v(v_0, \dots, v_q, \dots, v_{3P+1})$, et q correspond à l'index des valeurs dans V . Un coefficient binomial 2^q est attribué pour chaque signe $\text{sgn}(g_{t,p} - g_{t,c})$, après transformation de V en un numéro unique $VLBP_{L,P,R}$ qui caractérise la texture dynamique de volume local [17]:

$$VLBP_{L,P,R} = \sum_{q=0}^{3P+1} v_q 2^q \quad (II.8)$$

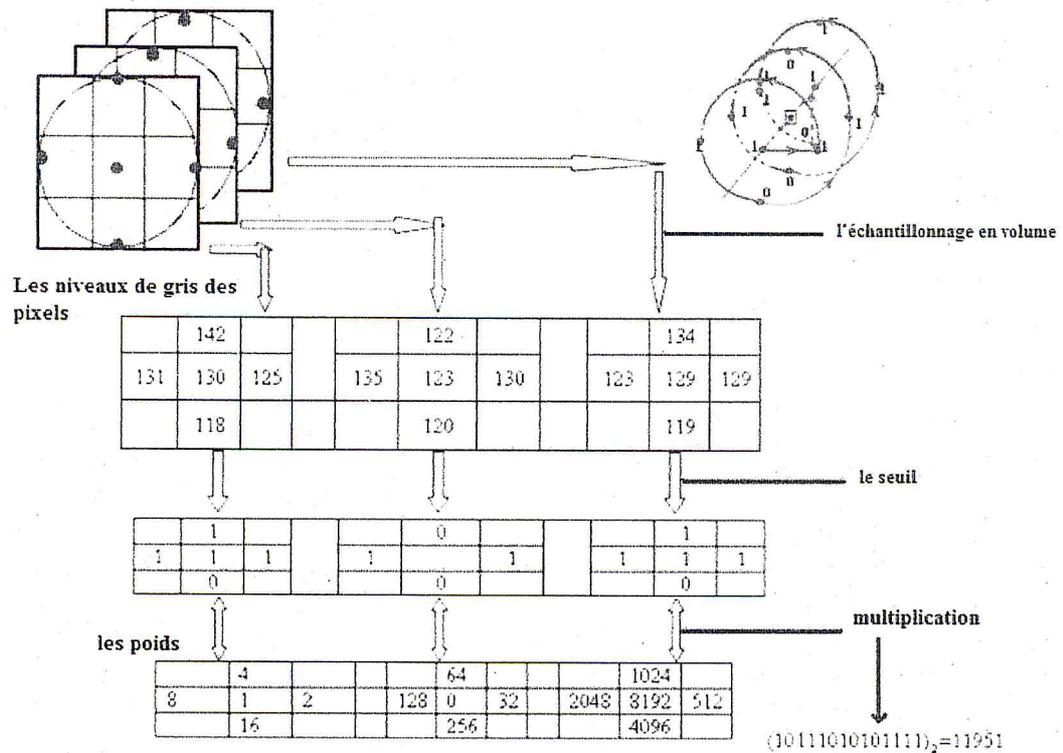


Figure II.3 exemple de calcul VLBP_{1,4,1}.

3.2. VLBP de rotation

Les textures dynamiques peuvent également être orientées arbitrairement et tournent souvent. La différence la plus importante entre la rotation sur une image fixe de texture et les textures dynamiques, est que toute la séquence de la texture dynamique tourne autour d'un axe ou multiaxes (si la caméra tourne pendant la capture) tandis que la texture fixe tourne autour d'un point [17].

Par conséquent, il est impossible de traiter l'ensemble du code VLBP pour obtenir la rotation du code comme le fait l'opérateur LBP [17] qui suppose une rotation autour du pixel central. Tout d'abord il faut diviser tout le code VLBP, la division donne :

$$V \approx v([sgn(g_{tc-L,c} - g_{tc,c}), [sgn(g_{tc-L,0} - g_{tc,c}), \dots, sgn(g_{tc-L,P-1} - g_{tc,c})], [sgn(g_{tc,0} - g_{tc,c}), \dots, sgn(g_{tc,P-1} - g_{tc,c})], [sgn(g_{tc+L,0} - g_{tc,c}), \dots, sgn(g_{tc+L,P-1} - g_{tc,c})], [sgn(g_{tc+L,c} - g_{tc,c})]) \quad (II.9)$$

La division donne $V_{preC}, V_{preN}, V_{curN}, V_{posN}, V_{posC}$, en ordre et V_{preN}, V_{curN} et V_{posN} représentent le code LBP de points de voisins dans l'image précédente, du centre et suivante, respectivement, tandis que V_{preC} et V_{posC} représentent les valeurs binaires des pixels centraux dans l'image précédente et suivante [17].

$$LBP_{t,P,R} = \sum_{p=0}^{P-1} sgn(g_{t,p} - g_{t,c}) 2^p \quad (II.10)$$

où $t = t_c - L, t_c, t_c + L$.

La formule précédente permet d'obtenir le code LBP $_{tc-L,P,R}$ (pour l'image précédente), $_{tc,P,R}$ (pour l'image centrale) et $_{tc+L,P,R}$ (pour l'image suivante). Et l'utilisation en premier lieu de la formule ci-dessous pour enlever l'effet de la rotation [17]:

$$LBP_{t,P,R}^{ri} = \min \{ROR(LBP_{t,P,R}, i) \mid i = 0, 1, \dots, P-1\}. \quad (II.11)$$

Où $ROR(x, i)$ effectue un décalage à droite au niveau des bits i fois [17]. En termes de pixels d'image, la formule précédente correspond simplement à tourner les pixels voisins dans le sens des aiguilles d'une montre autant de fois que le nombre maximum de bits les plus significatifs. Après avoir obtenu le code de rotation LBP respectivement $LBP_{tc-L,P,R}^{ri}$ (le code LBP pour l'image précédente), $LBP_{tc,P,R}^{ri}$ (le code LBP pour l'image centrale) et $LBP_{tc+L,P,R}^{ri}$ (le code LBP pour l'image suivante). La valeur VLBP de rotation est calculée par la formule suivante (II.12) [17]:

$$VLBP_{L,P,R}^{ri} = (VLBP_{L,P,R} \text{ et } (2^{3P+2} - 1)) + ROR(LBP_{tc+L,P,R}^{ri}, 2P+1) + ROR(LBP_{tc,P,R}^{ri}, P+1) + ROR(LBP_{tc-L,P,R}^{ri}, 1) + (VLBP_{L,P,R} \text{ et } 1) \quad (\text{II.12})$$

Par exemple, pour le code VLBP originale $(1,1010,1101,1100,1)_2$ ses codes après une rotation vers la gauche de 90, 180, 270 degrés sont respectivement $(1,0101,1110,0110,1)_2$, $(1,1010,0111,0011,1)_2$ et $(1,0101,1011,1001)_2$. Leur code de rotation doit être $(1,0101,0111,0011,1)_2$.

3.3. VLBP de rotation uniforme

Puisque les points voisins sont échantillonnés en volume, le nombre de modèles VLBP est grand. Si le nombre de points voisins dans une image est P , le nombre de modèles VLBP de base est 2^{3P+2} . Donc, pour réduire la longueur du vecteur de caractéristique, il existe un modèle «uniforme» pour calculer la valeur VLBP de rotation uniforme qui est notée $VLBP_{L,P,R}^{riu2}$ [17] (la formule II.13 et II.14).

$$VLBP_{L,P,R}^{riu2} = \begin{cases} \sum_{q=0}^{3P+1} v_q & \text{si } U(VLBP_{L,P,R}^{ri}) \leq 2 \\ 3P + 3 & \text{sinon} \end{cases} \quad (\text{II.13})$$

Et,

$$U(VLBP_{L,P,R}^{ri}) = |v_{3P+1} - v_0| + \sum_{q=1}^{3P+1} |v_q - v_{q-1}| \quad (\text{II.14})$$

et $V = (v_0, \dots, v_q, \dots, v_{3P+1})$ dans la formule II.14 exprime le code VLBP de rotation. "riu2" reflète l'utilisation de modèles de rotation uniformes, ayant une valeur U inférieure ou égale à 2. Ainsi, le nombre total de $VLBP_{L,P,R}^{riu2}$ est $3P + 4$ [17].

4. Conclusion

Dans ce chapitre nous avons donné le principe et les avantages de l'opérateur VLBP afin de l'utiliser dans l'application de détection de copie vidéo compressé par la norme H.264/AVC.

L'objectif de notre prochain chapitre est d'aborder en première lieu la norme H.264 afin de comprendre son fonctionnement pour pouvoir après développer la méthode de détection de copie dans la norme.

Chapitre III : La norme H.264/AVC

1. Introduction

La compression vidéo fait partie intégrante de nombreuses applications multimédias disponibles aujourd'hui. Pour certaines applications et certains équipements, par exemple les lecteurs DVD, la transmission de la télévision numérique, la vidéo conférence, la sécurité vidéo, et les caméras numériques, une bande de transmission limitée ou une capacité mémoire limitée contribuent à une demande de rapports de compression toujours plus élevés. Pour répondre à ces différents scénarios, plusieurs standards de codage vidéo tels que MPEG-1, MPEG-2, etc. ont vu le jour et se sont succédés au cours de ces dernières décennies (figure III.1).

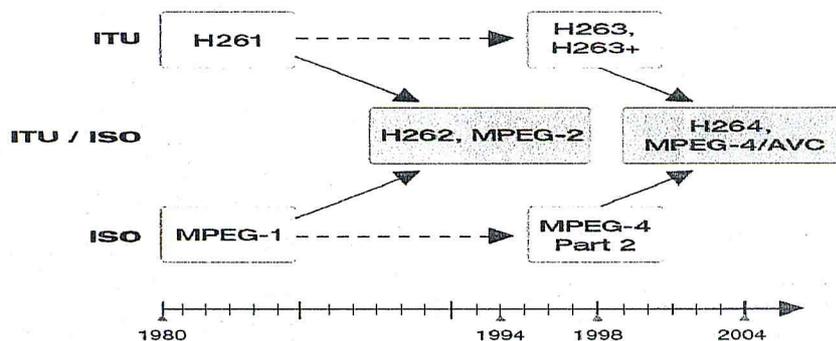


Figure III.1 Historique des différents standards vidéo ISO et ITU.

La norme vidéo du codage MPEG-2 qui a été développée, il y a environ 15 ans, est une technologie utilisée aujourd'hui dans le monde entier dans tous les systèmes de télévision numériques. Elle permet une transmission efficace des signaux de télévision par satellite (DVB-S), par câble (DVB-C) ou terrestre (DVB-T). De plus, elle permet aussi la transmission d'autres médias sur l'Internet haut débit (xDSL). Mais même sur le système DVB-T, la bande passante disponible n'est pas toujours suffisante. Par conséquent, le nombre de programmes devient limité, indiquant ainsi un besoin de compression vidéo à améliorer encore [20].

En 1998, Video Coding Experts Group (VCEG, ITU-T SG16 Q.6) a débuté un projet appelé H.26L ayant comme objectif de doubler l'efficacité de compression en

comparaison avec toutes les autres normes vidéo existantes. En décembre 2001, devant les premiers résultats de ce projet, le VCEG et le Moving Pictures Expert Group (MPEG. ISO/IEC JTC) ont formé une équipe de travail conjointe, le Joint Video Team (JVT) pour mener à terme le nouveau standard de compression vidéo H.264/AVC [20].

2. Structure du codeur H.264

Dans cette section, nous n'allons pas détailler tous les modules développés dans le codeur H.264/AVC, nous allons se contenter d'expliquer uniquement les parties utilisées dans le développement de l'application ciblée dans ce mémoire. Le fonctionnement détaillé de la norme est publié dans [23] [24].

2.1 Terminologie

La norme H.264 emploie une terminologie particulière que nous allons préciser [18][25]: une image (frame) d'une vidéo progressive est codée afin de produire une image codée. Un numéro lui est assigné (signalé dans le flux binaire), qui ne correspond pas forcément à l'ordre de décodage. Les images précédemment codées, les images de référence, sont utilisées pour la prédiction inter-image.

Une image codée est composée d'un certain nombre de macroblocs (MBs), par exemple pour le format 4:2:0, chacun d'eux contient 16x16 échantillons de luminance et deux ensembles de 8x8 échantillons, chaque ensemble étant associé à l'une des deux composantes de chrominance. Les MBs sont arrangés en tranches (slice en anglais), chaque tranche contenant un nombre de MBs compris entre un (un MB par tranche) et le nombre total de MBs d'une image (une tranche par image). H.264 utilise plusieurs types d'images : des images de type I (images codées en mode Intra-image), des images de type P (images prédites) et des images de type B (images bi-prédites).

- Les MBs de type I sont obtenus par prédiction à partir d'échantillons déjà décodés dans la tranche courante. Une prédiction est formée soit pour un MB

complet, soit pour chaque bloc 4x4 de luminance (et les blocs associés de chrominance) du MB.

- Les MBs de type P sont obtenus par prédiction inter-image à partir des images de référence.
- Les MBs de type B sont obtenus par prédiction inter-image à partir d'une ou plusieurs images de référence déjà codés et des images qui ne sont pas encore codés.

Un groupe d'images appelé aussi GOP (Group Of Picture) est un ensemble d'images consécutives constituant une séquence vidéo. Un GOP commence par une image de type I (ne contenant que des tranches I), suivie d'images de type P et/ou type B.

2.2. Le fonctionnement du codec H.264

Tout comme les autres standards de codage vidéo, H.264 ne définit pas un codec mais la syntaxe d'un flux binaire codé de vidéo ainsi que la procédure de décodage de ce flux. En pratique, un codeur/décodeur conforme à la norme inclut les fonctionnalités décrites sur les figures III.2 et III.3.

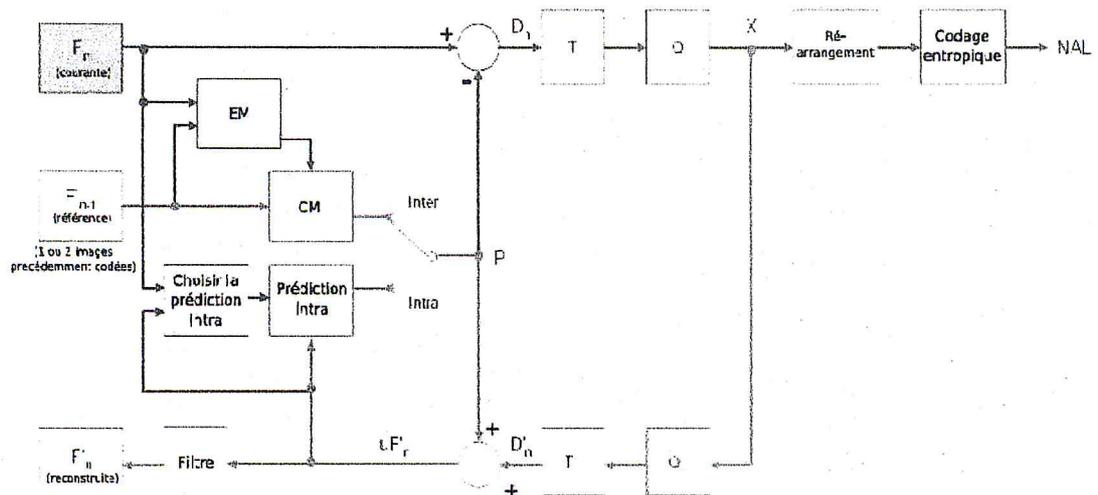


Figure III.2 Schéma du codeur H.264.

Le codeur (figure III.2) inclut deux chemins pour le flux de données, le chemin « avant » (de gauche à droite) et le chemin de reconstruction (de droite à gauche). Sur la figure III.3, le chemin de données du décodeur est présenté de droite à gauche, indiquant ainsi les similarités avec le codeur.

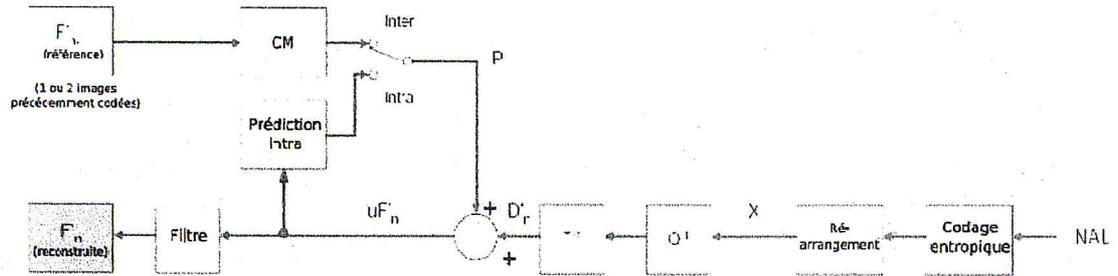


Figure III.3 Schéma du décodeur H.264.

D'après la figure III.2, sur le chemin avant, une trame ou une image F_n est partitionnée en MBs, une prédiction P est formée en fonction des échantillons reconstruits. Pour le mode Intra, P est obtenue à partir d'échantillons de la tranche courante ayant déjà été codés, décodés et reconstruits (uF'_n sur la figure III.2 ; notons que les échantillons utilisés ne sont pas filtrés). En mode Inter, P est obtenue par prédiction par compensation de mouvement à partir d'une image de référence sélectionnée dans la liste 0 (images précédemment codées, décodées et reconstruites F'_{n-1} ,) et/ou la liste 1 (images futures non codées de type B).

La prédiction P est ensuite soustraite au bloc courant, générant un bloc résiduel D_n . Celui-ci est alors transformé (T) par la DCT entière 4x4 et quantifié (Q) en appliquant la quantification scalaire afin de produire un ensemble de coefficients transformés et quantifiés X . Ces derniers, sont réarrangés puis transmis au codeur entropique. Les coefficients obtenus et les informations nécessaires au décodage (mode de prédiction, table de quantification, vecteurs de mouvement. . .) sont codés en un flux binaire qui est passé au NAL (Network Abstraction Layer) pour transmission ou codage adapté aux caractéristiques du canal de transmission.

En plus du codage et de la transmission, le codeur exécute également un décodage suivant le chemin de reconstruction. Cette étape est nécessaire pour fournir une référence pour les futures prédictions. Les coefficients X subissent un déquantification (Q^{-1}) et une transformation inverse (T^{-1}) pour produire un bloc des différences D'_n . La prédiction P est ajoutée à D'_n pour créer le bloc reconstruit uF'_n . Enfin un filtre est appliqué afin de réduire les effets de bloc. L'image de référence reconstruite est ainsi créée par une série de blocs F'_n .

Au niveau du décodeur H.264/AVC, le même procédé est appliqué. Les MBs reçus par le NAL sont réarrangés et décodés, pour obtenir les coefficients X . Les processus de quantification et transformation inverses sont appliqués pour obtenir D'_n . Grâce aux informations d'en-tête du flux binaire, le décodeur crée la même prédiction P que celle du codeur, ajoute D'_n pour produire uF'_n . Celle-ci est alors filtrée pour générer chaque bloc décodé F'_n [18].

2.3. Prédiction intra-image

La prédiction Intra n'utilise pas d'image de référence. Elle exploite la redondance spatiale d'une image, c'est-à-dire, le fait que les pixels dans une image sont spatialement corrélés. Le codeur H.264 a la particularité de travailler dans le domaine spatial, se référant aux échantillons voisins de blocs déjà codés et non dans le domaine transformé. La prédiction dépend de la taille de la partition considérée. Il existe donc plusieurs types de prédiction en fonction de la taille du bloc : H.264 permet la prédiction de type Intra des blocs 4x4 (Intra_4x4) ou des blocs 16x16 (Intra_16x16) de luminance et de blocs 8x8 de chrominance. Le codeur sélectionne le mode pour chaque bloc qui minimise la différence entre le bloc prédit P et le bloc courant.

2.3.1. Prédiction des blocs 4x4 de luminance

La prédiction de blocs 4x4 de luminance est une spécificité de H.264 par rapport aux autres standards qui utilisent des blocs de taille 8x8. Cette prédiction est plus adaptée pour les zones de détails, donc apporte une meilleure précision mais

nécessite plus de calculs. Neuf modes de prédiction sont possibles pour les blocs 4x4 selon la direction de prédiction [26].

2.3.2. Prédiction des blocs 16x16 de luminance

Le codeur H.264 permet également la prédiction de MBs entiers de luminance (16x16) pour les régions homogènes. Quatre modes sont possibles [18].

Les tableaux III.1 et III.2 récapitulent les modes de prédiction appliqués sur chaque région de l'image précise les calculs réalisés.

Tableau III.1 Les modes de prédiction des blocs de taille 4x4 de luminance.

Mode	Description
Mode 0 (vertical)	Les échantillons supérieurs A, B, C et D sont extrapolés verticalement.
Mode 1 (horizontal)	Les échantillons gauches I, J, K et L sont extrapolés horizontalement.
Mode 2 (DC)	Tous les échantillons [a-p] sont prédits par la moyenne de [A-D]et [I-L].
Mode 3 (diagonale bas-gauche)	Les échantillons sont interpolés avec un angle de 45°entre le bas-gauche et le haut-droit.
Mode 4 (diagonale bas-droite)	Les échantillons sont interpolés avec un angle de 45°vers le bas et vers la droite.
Mode 5 (vertical-droit)	Extrapolation avec un angle de 26,6°à gauche de la verticale (largeur=hauteur = 1=2).
Mode 6 (horizontal-bas)	Extrapolation avec un angle de 26,6°en-dessous de l'horizontale.
Mode 7 (vertical-gauche)	Extrapolation (ou interpolation) avec un angle de 26,6°à droite de la verticale.
Mode 8 (horizontal-haut)	Interpolation avec un angle de 26,6°au-dessus de l'horizontale.

Tableau III.2 Les modes de prédiction des blocs de taille 16x16 de luminance.

Mode	Description
Mode 0 (vertical)	Extrapolation à partir des échantillons supérieurs (H).
Mode 1 (horizontal)	Extrapolation à partir des échantillons gauches (V).
Mode 2 (DC)	Moyenne de H et V.
Mode 3 (plan)	Une fonction linéaire plane est ajustée à H et V. Cela fonctionne bien sur les zones de luminance variant doucement.

Dans ce qui précède nous avons abordé les différents modules jugés nécessaires pour décrire le fonctionnement de la norme H.264/AVC, ainsi les plus importantes caractéristiques qui ont trait au développement de système de détection de copie vidéo basée sur le contenu qu'on va décrire dans la section suivante.

3. Etat de l'art de la détection de copie basée sur le contenu dans la norme H.264

Il existe plusieurs approches de détection de copie vidéo compressé par les normes de compressions précédentes. Parmi celles-ci on trouve, l'approche proposée par Naphade [16] pour les normes MPEG-1/ 2. Dans celle-ci, la signature est générée à partir des histogrammes des coefficients DCT des blocs 8x8 des composantes YUV. Babu a décrit dans [20], une autre méthode de détection des copies vidéo dédiée pour les codecs MPEG. Elle est basée sur le mouvement global et les caractéristiques des objets locaux.

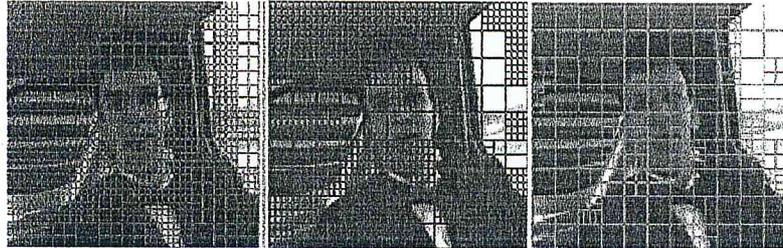
Comme la norme H.264/AVC est un standard nouveau, et sa structure diffère par rapport aux normes précédentes, nous n'avons pas trouvé qu'une seule méthode de détection de copies vidéo compressée par cette norme proposée par M. Athar et al. [6]. La signature est générée dans le domaine fréquentiel, au cours de la compression. Le principe de la méthode est décrit comme suit :

3.1. Génération de la signature

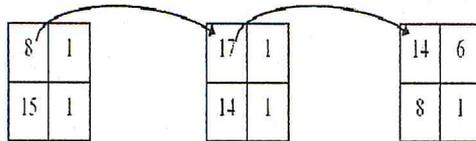
La signature de chaque GOP est calculée à partir des caractéristiques des trois premières images. Le processus d'extraction des caractéristiques de chaque image consiste à :

- subdiviser l'image en quatre régions (figure III.4.a) : HG (haut à gauche), HD (haut à droite), BG (bas à gauche) et BD (bas à droite).
- Calculer pour chaque région, le nombre des MBs de type Intra_4x4 au sein de l'image Intra, le nombre de MBs de type B_skip au sein de l'image B et le nombre de MBs P_skip au sein du frame P (figure III.4.b).

- Les nombres de MBs d'Intra_4x4, B_skip et P_skip sont classés dans un ordre décroissant et sont pris comme vecteur ordinal.
- Ces vecteurs sont ensuite combinés pour générer la matrice ordinale finale qui représente la signature du GOP (figure III.4. c).



(a)



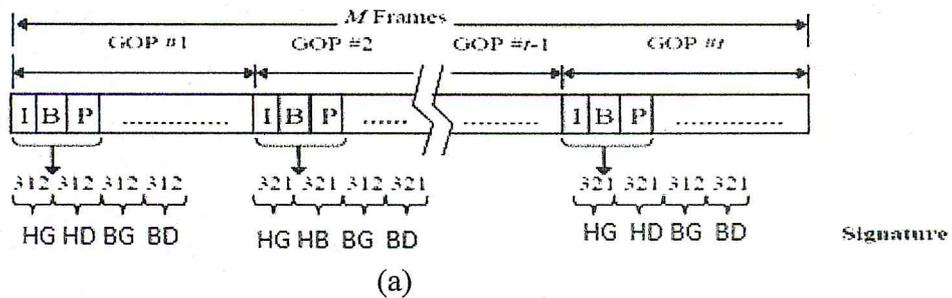
(b)

$$\begin{array}{l} \text{HG} = \\ \text{HD} = \\ \text{BG} = \\ \text{BD} = \end{array} \left| \begin{array}{ccc} 3 & 1 & 2 \\ 2 & 3 & 1 \\ 1 & 2 & 3 \\ 1 & 2 & 3 \end{array} \right|$$

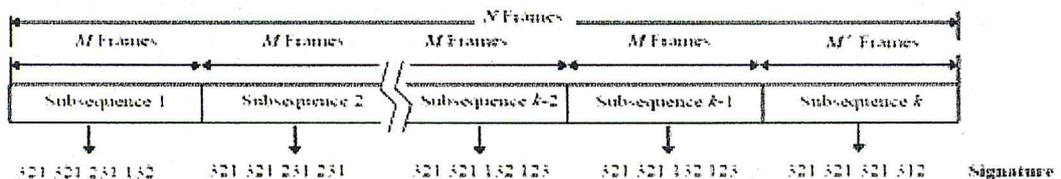
(c)

Figure III.4 Génération de la signature dans [6].

Le même traitement de génération de signatures est effectué sur la vidéo requête V_Q (figure III.5.a) et la vidéo cible V_T (figure III.5.b).



(a)



(b)

Figure III.5 Extractions des signatures (a) vidéo requête et (b) la vidéo cible.

3.2. Le processus de comparaison des signatures

Le processus de comparions de signatures entre V_Q et V_D est effectué selon le mécanisme illustré sur la figure III.6.

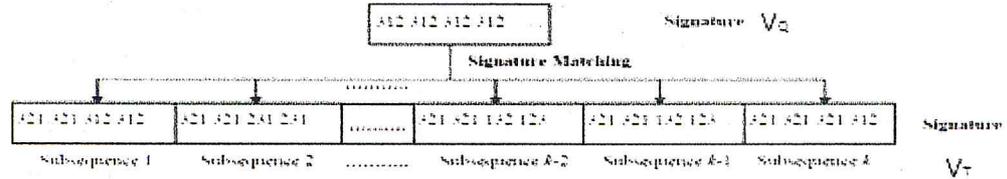


Figure III.6 processus de comparaison.

La détection de copie d'un segment vidéo V_T^r de la séquence cible V_T est déterminée en calculant la distance $D(V_Q, V_T^r)$:

$$D(V_Q, V_T^r) = \frac{1}{P} \sum_{p=1}^P d(\pi_{q,p}, \pi_{t,p}^r) \quad \text{III.1}$$

Où : $P = 4$, est le nombre total des partitions dans chaque frame et $\pi_{q,p}$ la matrice de rang pour une partition p .

La distance d est calculée comme suit:

$$d(\pi_{q,p}, \pi_{t,p}^r) = \frac{1}{C} \sum_{i=1}^M |\pi_{q,p}(i) - \pi_{t,p}^r(r + i - 1)| \quad \text{III.2}$$

Où : r et M sont les tailles du GOP et la vidéo respectivement.

Le paramètre C est un paramètre de normalisation définie comme suit :

$$C = \sum_{i=1}^M |M + 1 - (2 \times i)| \quad \text{III.3}$$

Si la valeur de $D(V_Q, V_T^r)$ est inférieure à un seuil ε prédéfini. Donc V_T est considérée comme vidéo d'origine sinon elle c'est une vidéo modifiée donc une vidéo copie.

4. Conclusion

Nous venons de présenter dans ce chapitre, le principe de fonctionnement en général de standard H.264/AVC et un bref état de l'art des méthodes de détection de copie basée sur le contenu pour les vidéos compressées par cette norme. Dans le prochain chapitre nous développerons en détails la méthode proposée pour la détection les copies des vidéos H.264/AVC.

Chapitre IV : Conception et réalisation

1. Introduction

Dans ce chapitre, nous allons détailler le concept du système développé pour la détection de copies vidéo compressées par la norme H.264/AVC. Deux approches basées sur les caractéristiques spatiaux-temporels de la vidéo sont proposées pour générer les signatures. Dans la première méthode, tout le contenu de l'image de la séquence vidéo est pris en considération pour extraire la signature, et comme amélioration de celle-ci en termes de temps de calcul, dans la deuxième méthode seulement les régions texturées sont utilisées.

2. Système proposé

La solution comporte deux processus. Le premier concerne l'extraction des caractéristiques des vidéos d'origines et leurs stockages dans la base de données. Le second est dédié à l'identification des copies vidéo. La figure IV.1 illustre le schéma général du système.

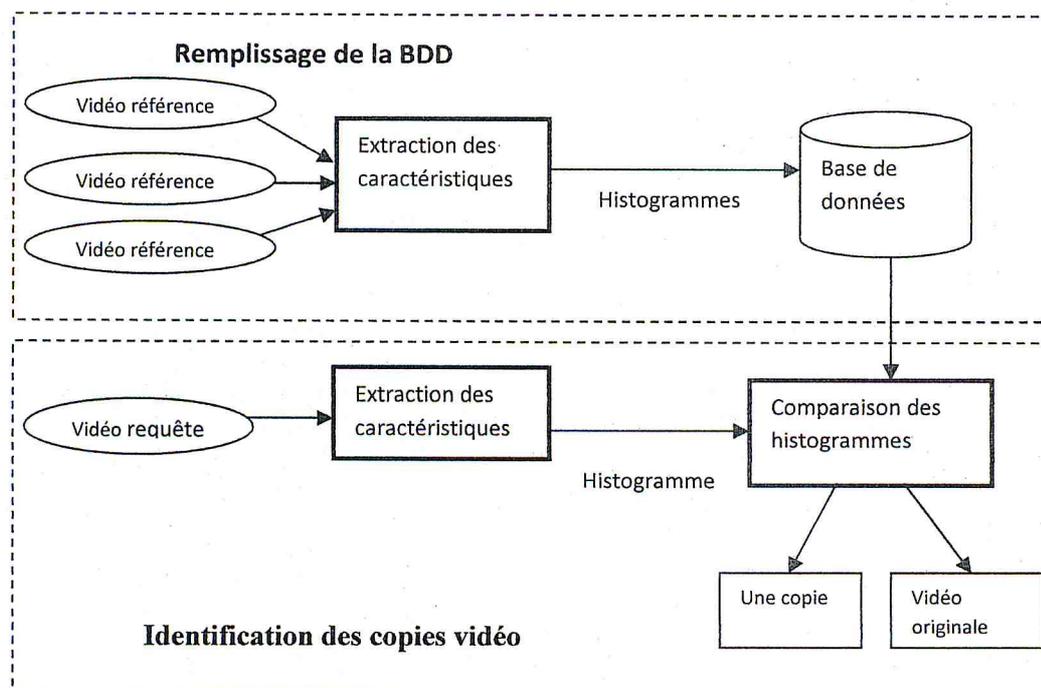


Figure IV.1 Les étapes de notre système de détection de copie.

La base de données comprend les histogrammes de caractéristiques correspondant aux vidéos d'origines qui sont par la suite comparés aux histogrammes de caractéristiques des vidéos requêtes pour l'identification

Pour l'extraction des caractéristiques, l'opérateur *VLBP* détaillée dans le chapitre II est employé. L'entrée de l'opérateur *VLBP* est une séquence de trois images. Dans un premier temps, Les caractéristiques sont extraites à partir de tous le contenu de des images, puis le domaine d'extraction est limité uniquement aux zones texturées.

2.1. Extraction des caractéristiques

Les images de la vidéo sont représentées par l'espace de colorimétrie YUV (Y : la luminance, U et V la chrominance). Dans notre système, seulement la luminance (Y) est utilisée car le système visuel humain (HVS : human visual system) est très sensible aux changements dans la composante de luminance que des chrominances [27].

2.1.1. Extraction des caractéristiques à partir de tous le contenu des images

La figure IV.2 représente les étapes de l'extraction.

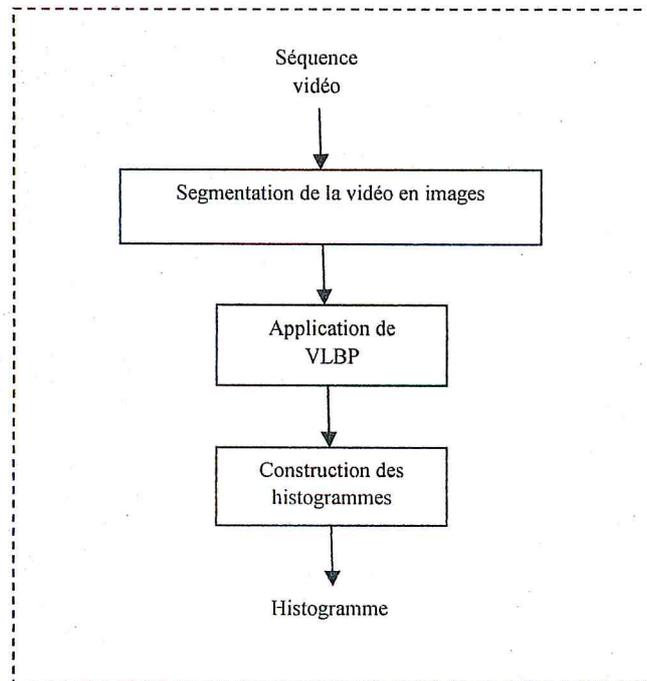


Figure IV.2 Les étapes de l'extraction des caractéristiques à partir des images.

2.1.1.1. Segmentation de la vidéo en images

Pour chaque image de la vidéo en entrée du codeur H.264/AVC, le numéro de l'image et ses valeurs de niveaux de gris sont sauvegardés.

2.1.1.2. Application de VLBP

L'opérateur VLBP de base et ses deux variantes sont développées dans notre système de détection de copies vidéo à savoir : VLBP de rotation et VLBP de rotation uniforme. Les étapes de l'opérateur *VLBP* de base sont définies comme suit (chapitre II) :

- La séquence d'images est découpée en sous séquences de même taille, selon le paramètre de facteur de temps L . Dans notre système L égale à 1 donc la vidéo est découpée en 3 images (l'image de centre, une image précédente et une image suivante).
- Après la segmentation de vidéo en 3 images, chaque image dans cette sous séquences est découpée en blocs, le paramètre R de la méthode *VLBP* qui définit le rayon de cercle est égale à 1. D'où les blocs sont de taille 3x3 pixels.

Chaque pixel de l'image est considéré comme un pixel central par rapport à ces voisins. Les pixels de l'extrémité ne sont pas considérés comme des pixels du centre car ils n'ont pas de voisins.

- Le paramètre P de l'opérateur $VLBP$ permet de définir le nombre de voisins autour de pixel central. Ce dernier est égal à 4 dans notre cas.

A partir des valeurs des VLBP, les histogrammes qui représentent la signature de la vidéo sont générés.

2.1.1.3. Construction des histogrammes

L'histogramme est construit à partir des valeurs VLBP et leurs redondances. L'histogramme illustré sur la figure IV.3 représente la signature de la vidéo obtenue à partir de certain nombre de séquence vidéo concaténée (figure IV.4). La taille de l'histogramme dépend de la méthode utilisée:

- Pour la méthode $VLBP$ de base, la taille de l'histogramme égale à 2^{3P+2} . Pour P égale à 4 la taille est de l'ordre de 2^{14} .

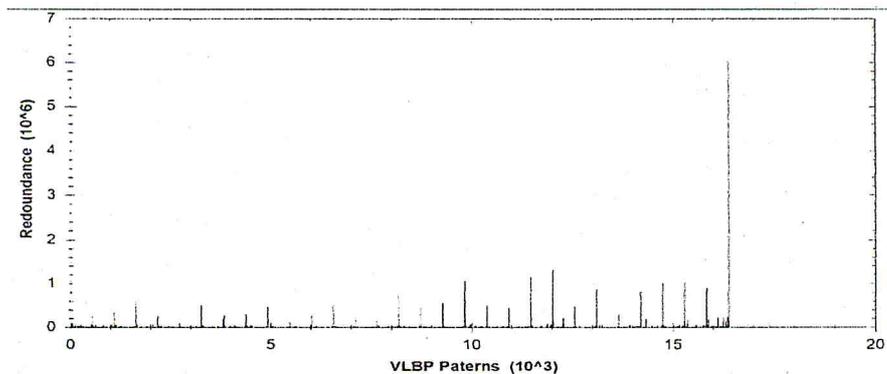


Figure IV.3 Histogramme des valeurs $VLBP$ de base.

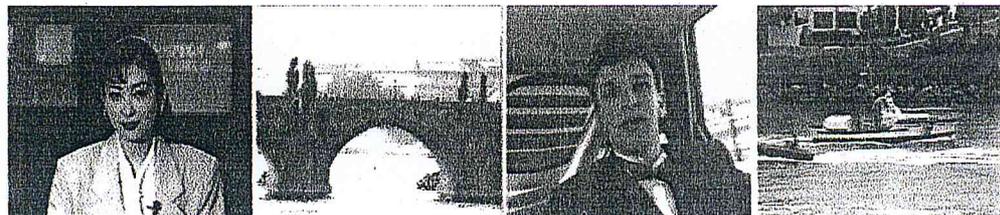


Figure IV.4 Des images de la vidéo concaténée.

- La taille de l'histogramme obtenu après l'application de la méthode *VLBP* de rotation est la même que celui obtenu en appliquant la méthode *VLBP* de base.
- Alors que la taille de l'histogramme obtenu après l'application de la méthode *VLBP* de rotation uniforme est de l'ordre de $(3P + 4)$. Pour P égale à 4, la taille égale à 16 (figure IV.5).

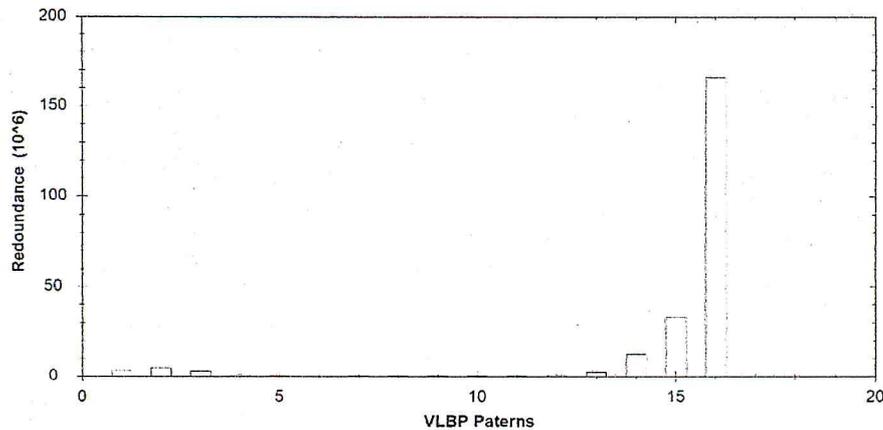


Figure IV.5 Histogramme des valeurs *VLBP* de rotation uniforme.

Le processus de calcul de l'opérateur *VLBP* de base et ses deux variantes (*VLBP* de rotation et de *VLBP* de rotation uniforme) est répété pour chaque bloc de 3x3 pixels des sous séquences de trois images. Ce qui nécessite un temps de calcul important.

Pour réduire le temps de calcul des valeurs des *VLBP* nécessaire pour la génération de la signature de la vidéo, nous proposons d'appliquer l'opérateur *VLBP* sur les régions texturées de la vidéo uniquement. Pour ce faire, nous prenons en considération les caractéristiques de la norme H.264/AVC.

2.1.2. Amélioration du temps de calcul de VLBP dans la norme H.264/AVC

Le codeur H.264/AVC a le privilège de différencier entre des zones texturées (les MBs de types Intra_4x4 et Inter_4x4) et les zones homogènes (les MBs de type Intra_16x16). Ces dernières ne contiennent pas beaucoup d'informations, il est donc possible de les ignorer lors de l'extraction des caractéristiques d'une vidéo. Par contre les MBs de types Intra_4x4 et Inter_4x4 sont des MBs texturées (figure IV.6),

elles contiennent des informations pertinentes pour caractériser une vidéo. Ce qui les rend importants pour la génération de la signature de la vidéo.

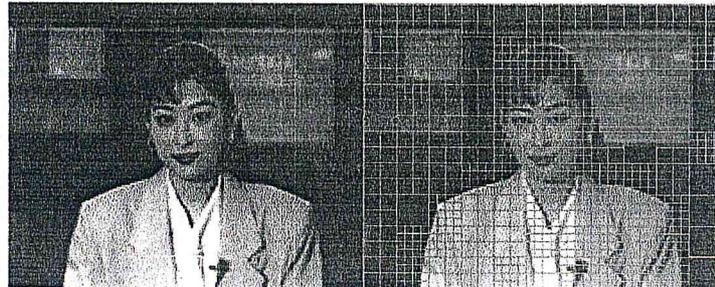


Figure IV.6 Exemple de sélection des MBs de types Intra_4x4 et Inter_4x4 (les zones texturées) la vidéo akiyo.

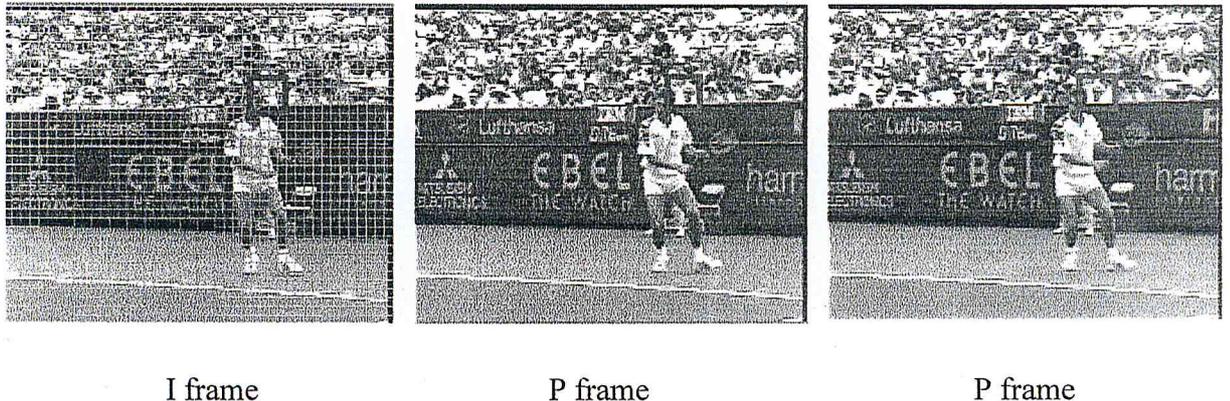


Figure IV.7 Les zones texturées dans la première image (I-frame) et les zone correspondent dans les deux autres images de la sous séquence.

Les MBs texturés sont sélectionnés pour chaque sous séquence de trois images d'un GOP de la vidéo en entrée. La figure IV.8 illustre les étapes de l'extraction la signature d'une vidéo compressée par H.264.

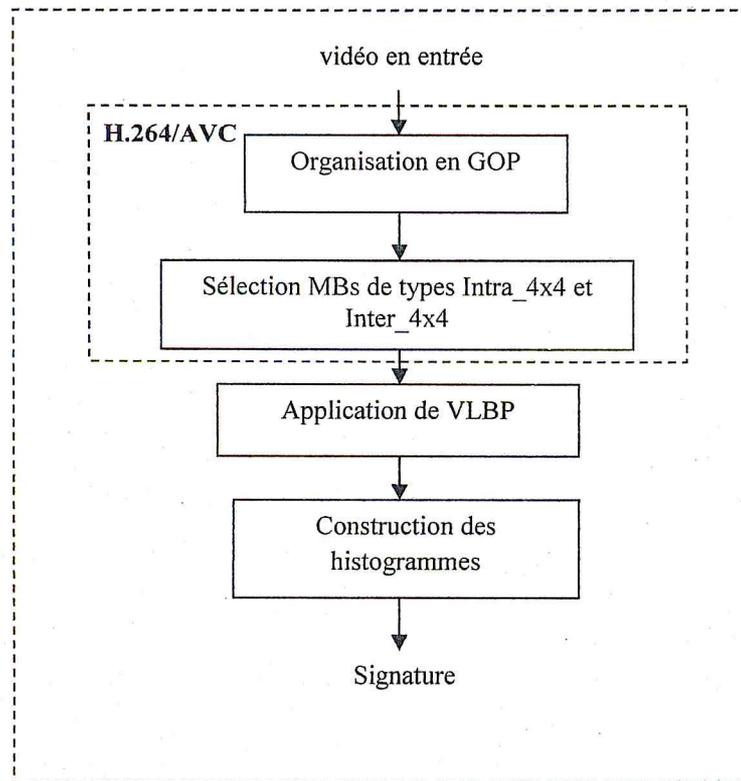


Figure IV.8 Les étapes de l'extraction de la signature dans les zones texturées.

2.1.2.1. Découpage en GOP

La structure et la taille d'un GOP ne sont pas spécifiées dans la norme H.264/AVC, elles sont définies selon l'application. Dans notre application, la vidéo est une suite de GOP. Chacun GOP comprend deux types d'images I et P. La structure de chaque GOP est de type I, P, P, P, ..., I (figure IV.9). Ce partitionnement en GOP est effectué dans le fichier "encodeur.cfg" spécifié dans la norme H.264.

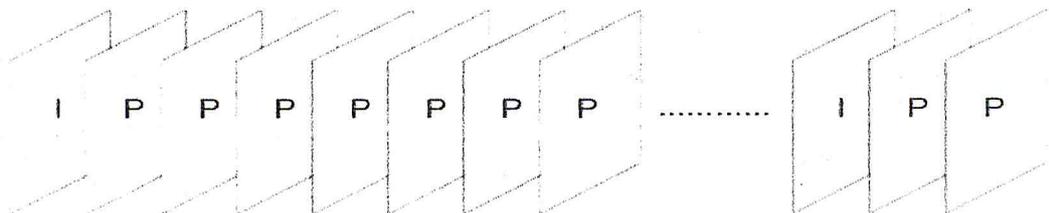


Figure IV.9 Structure d'un GOP.

2.1.2.2. Application de VLBP

Les paramètres de VLBP utilisés pour générer la signature à partir des régions texturées sont les même que celles employés dans l'application de VLBP sur tous le contenu de la vidéo. C'est-à-dire, le facteur de temps L , le rayon de cercle R et le nombre de voisins autours du pixel central P .

2.2. Comparaison des histogrammes

La comparaison des histogrammes est une tâche nécessaire pour prendre une décision si la vidéo est d'origine ou c'est une copie. Ce module nécessite l'histogramme de la vidéo d'origine qui est stocké dans la base de données et celui de la vidéo requête. L'histogramme de la vidéo requête est construit selon le même mécanisme utilisé pour construire l'histogramme de la vidéo d'origine. La figure IV.10 montre les étapes de la comparaison.

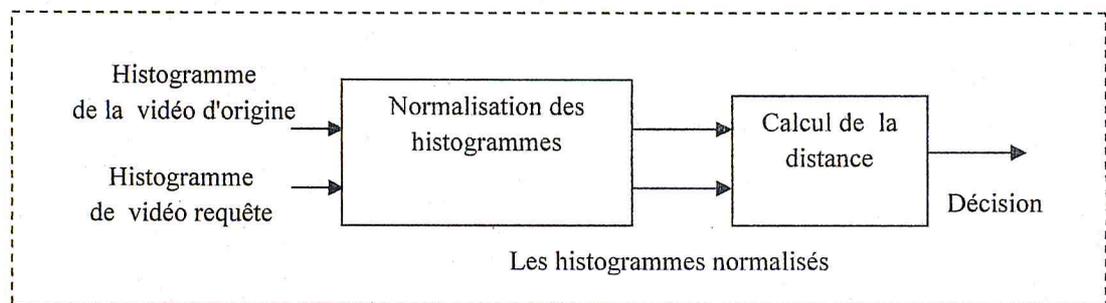


Figure IV.10 Processus de comparaison des histogrammes.

2.2.1. Normalisation des histogrammes

Après l'application de la méthode VLBP et ces variantes, des histogrammes sont construits. La normalisation des histogrammes est une tâche aussi nécessaire pour comparer les histogrammes obtenus [17]. Elle s'effectue selon la formule suivante:

$$H_{i \text{ (normalisé)}} = \frac{(H_i - \min)}{(max - \min)} \quad (IV.1)$$

Où H_i est l'effectif de classe courant et les paramètres max et min correspondent aux valeurs maximale et minimale entre les effectifs respectivement.

2.2.2. Calcul de la distance entre les histogrammes

La comparaison de l'histogramme de la vidéo requête avec celle de la vidéo d'origine stockée dans la base de données est basée sur la mesure de similarité ou dissimilarité. La distance employée est la khi deux (chi-distance), elle est donnée par la formule ci-dessous [4]:

$$D_{SqChi}(H_1, H_2) = \sum_{i=1}^N \frac{(H_1(i) - H_2(i))^2}{(H_1(i) + H_2(i))} \quad (IV.2)$$

Où H_1 est l'histogramme de la vidéo de référence, H_2 est l'histogramme de la vidéo requête, et N représente la taille de l'histogramme.

2.2.3. La décision

On compare la distance entre les histogrammes. Si la distance est supérieure à la valeur 0, la vidéo requête est considérée comme vidéo modifiée donc une copie, dans le cas contraire, la vidéo est d'origine.

Remarque : La base de données est construite à partir des vidéos d'origines et leurs histogrammes.

3. Présentation du logiciel

Nous allons présenter la fenêtre principale de l'application et les différents outils permettant la communication avec le logiciel. La figure IV.11 montre la structure de la fenêtre principale.

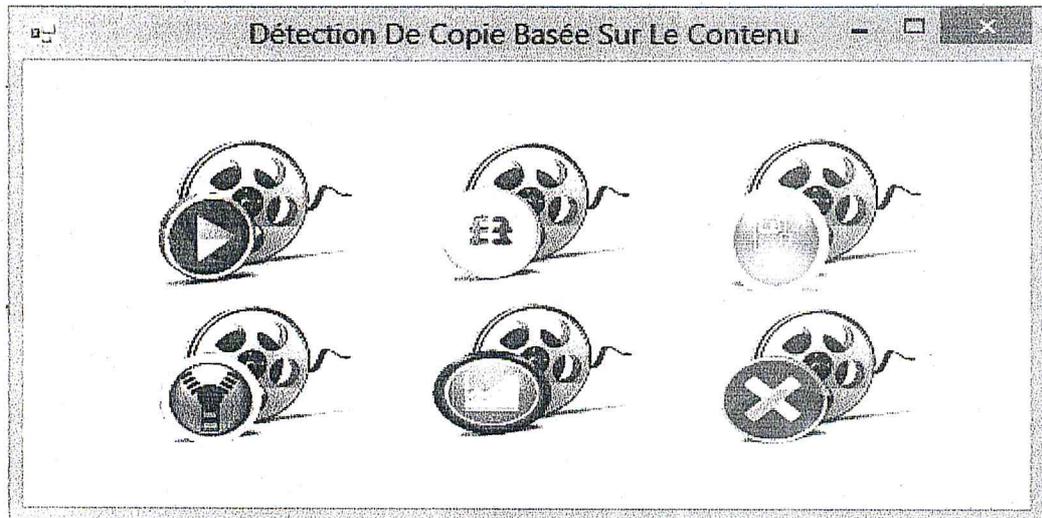


Figure IV.11 Vue générale du système de détection de copie basée sur le contenu dans la norme H.264/AVC.

3.1. Le bouton « Ouvrir »

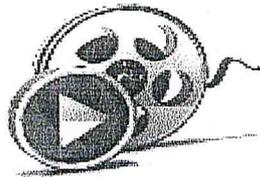


Figure IV.12 Le bouton Ouvrir.

Permet de visualiser une vidéo au format QCIF comme le montre la figure IV.12. la figure IV.13 montre le résultat de l'action sur le bouton « Ouvrir » et les paramètres d'affichage de la séquence vidéo

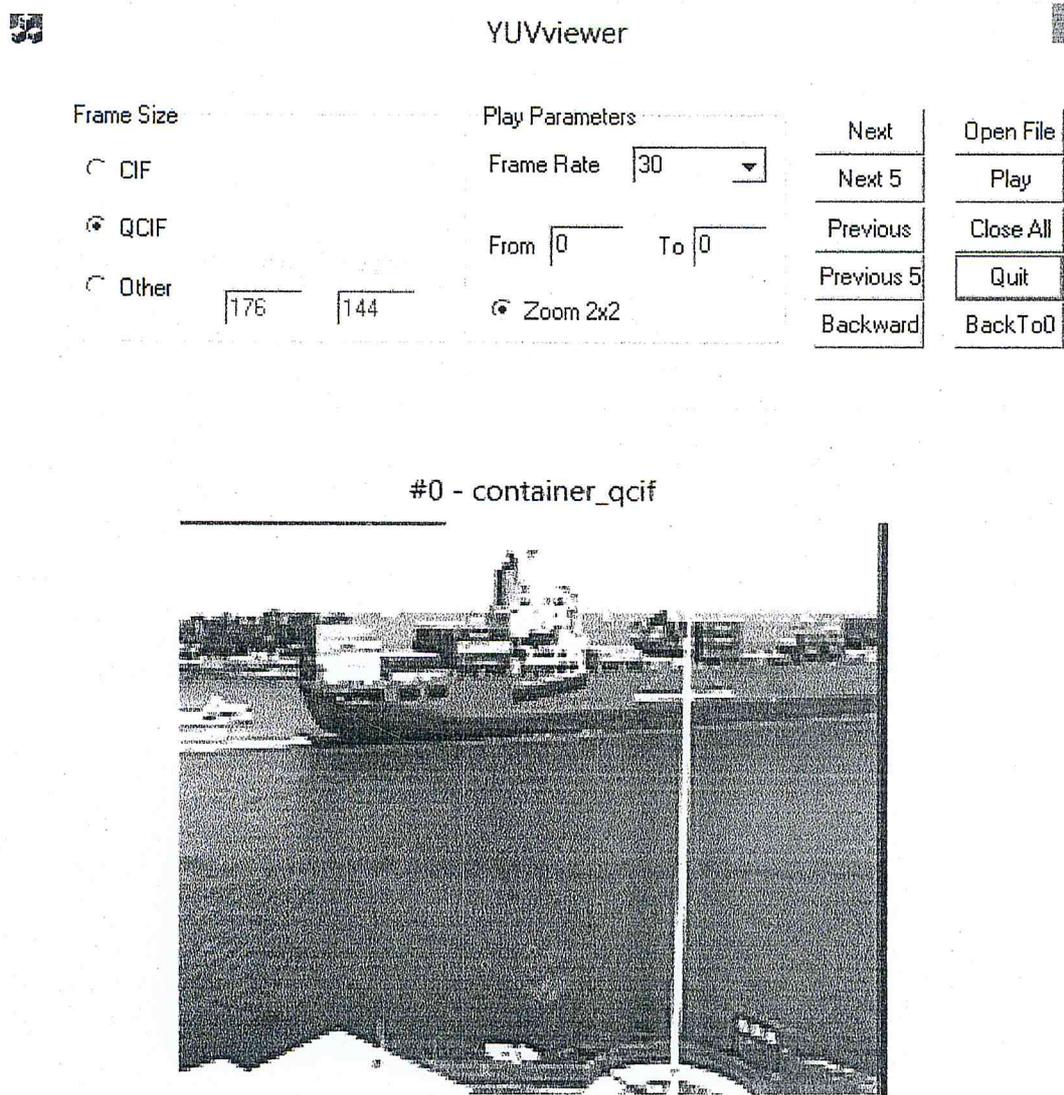


Figure IV.13 Paramètres d'affichage de la séquence vidéo.

3.2. Le bouton « Compresser »



Figure IV.14 Le bouton « Compresser ».

L'action sur ce bouton permet d'effectuer une compression vidéo en spécifiant les paramètres d'entrée suivants (figure IV.15) :

-Nom de la séquence : Le nom de la séquence vidéo a compressé

-Nombre d'image à coder : nombre des images de la séquence à compressé.

-Intra période : Le nombre d'image de type P à parcourir pour retrouver une image de type I, autrement dit c'est la taille de GOP.

-Mode de génération :

-Complete : Extraction des caractéristiques a partir de tout le contenu de l'image.

-Intra_4x4 : Extraction des caractéristiques a partir des régions texturée Intra_4x4.

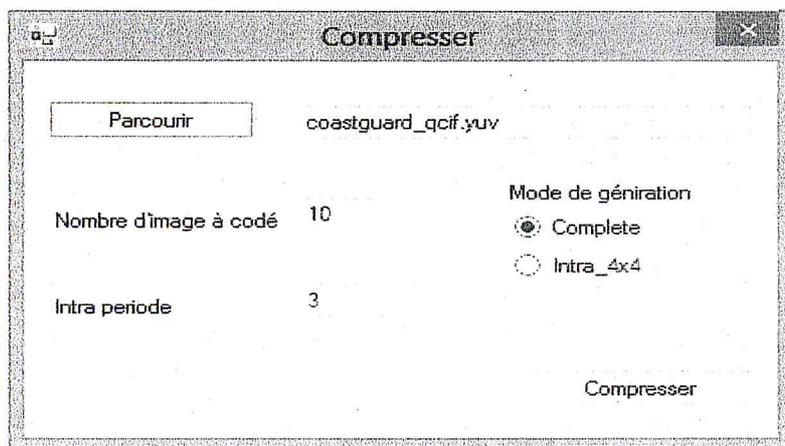


Figure IV.15 La rubrique « Compression ».

3.3. Le Bouton « Identifier »



Figure IV.16 Le bouton « Identifier ».

La figure IV.16 illustre le Bouton « Identifier ». L'action sur ce bouton permet d'afficher la fenêtre de la figure IV.17 qui englobe 3 boutons de comparaison :

-le bouton « VLBP De Base » : permet la comparaison selon l'opérateur VLBP de base.

- le bouton « VLBP Rotationnelle » : permet la comparaison selon l'opérateur VLBP de rotation.

-le bouton « VLBP Uniforme » : permet la comparaison selon l'opérateur VLBP de rotation uniforme.

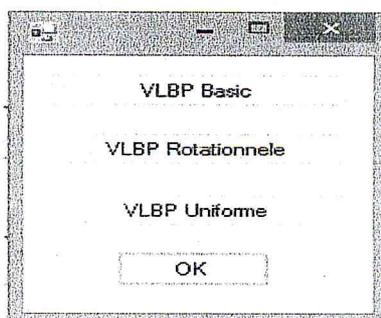


Figure IV.17 Comparaison des vidéos.

3.4. Le Bouton « Visualiser »



Figure IV.18 Le bouton « Visualiser ».

L'action sur ce bouton (Figure IV.18) permet d'afficher l'histogramme de la vidéo requête (Figure IV.19) que l'utilisateur veut identifier.

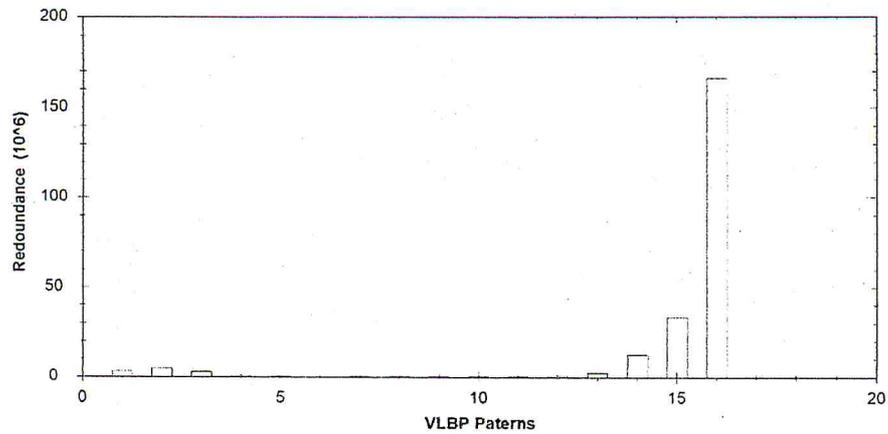


Figure IV.19 Histogramme des caractéristiques de la vidéo requête.

3.5. Le Bouton « Enregistre »

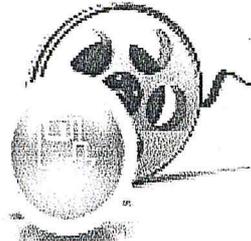


Figure IV.20 Le bouton « Enregistre ».

L'action sur ce bouton (Figure IV.20) permet aux utilisateurs authentifiés (figure IV.21) d'enregistrer des vidéos dans la base de données.

Nom D'utilisateur
CBCD

Mot De Passe

Valider

Figure IV.21 Identification d'utilisateur.

Si les informations de l'utilisateur sont correctes la figure IV.15 s'affiche. Ensuite l'utilisateur doit remplir les informations de la vidéo qu'il veut enregistrer.

3.6. Le Bouton « Fermer »



Figure IV.22 Le bouton fermer.

L'action sur ce bouton (figure IV.22) permet de quitter l'application.

4. conclusion

Dans ce chapitre nous avons présenté en détails, la solution développée pour la détection de copie vidéo compressée par le standard H.264/AVC. L'approche développée est basée sur le contenu, elle utilise l'opérateur VLBP et ces variantes.

Dans le chapitre suivant nous allons présenter les résultats obtenus lors d'une série de tests effectués sur des vidéos afin d'évaluer les performances de notre système.

Chapitre V : Testes et résultats

1. Introduction

Dans ce chapitre nous allons présenter les différents tests et expérimentations effectués afin de valider notre système de détection de copie. Pour cela nous avons testé notre procédure d'identification sur un ensemble de séquences.

Les tests présentes dans ce chapitre sont réalisées sur les séquences vidéo de format QCIF (résolution 176x144 pixels) à 30 images/seconde et en mode d'échantillonnage 4:2:0.

Il n'a pas été possible d'obtenir une importante base de données de fichiers YUV pour tester le système proposé, il y a seulement un nombre limité de séquences vidéo YUV QCIF disponibles en ligne. Un ensemble de 24 séquences vidéo différentes avec différentes longueurs ont cependant été obtenues [sw01]. Ces séquences vidéo couvrent différents types de vidéo : vidéo avec beaucoup de mouvements donc contiennent beaucoup de régions texturées telles que les vidéos Stefan, Forman et Flower, ainsi que des vidéos contenant peu de mouvements (beaucoup de zones homogènes) considérées comme vidéo statiques telle que Bridge-close et Akiyo.

Le langage utilisé pour la réalisation de notre logiciel est le logiciel de programmation Microsoft Visual Studio. Ainsi que la norme de compression vidéo H.264/AVC JM 10.0 [sw02] utilisée dans le travail a été téléchargée et compilée sous Visual C++. L'interface de l'application est réalisée sous Windows 7 en utilisant le langage de programmation C#. Les manipulations effectuées sur les vidéos pour les construire les vidéos requêtes sont faites en utilisant YUVTools version 3.0 [sw03].

Les plus importants paramètres de configurations du logiciel de référence H.264/AVC sont regroupés dans le tableau VI.1. Les autres paramètres sont maintenus à leurs valeurs par défaut.

Tableau V.1 Les paramètres utilisés dans la norme H.264/AVC.

Paramètres de codage	Description
InputFile	La séquence vidéo choisie
FrameToBeEncoded	Le nombre des images a codé
IntraPeriod	La période des images I

Dans nos expériences, le logiciel de référence H.264/AVC est configuré de telle façon à produire une image de type I à chaque 10 frames, ceci donne un nombre totale de frames dans le GOP égale 10.

2. Les attaques

Pour évoluer la performance de notre système de détection de copie basée sur le contenu, 11 attaques ont été appliquées sur une vidéo obtenue par concaténation de 18 différentes vidéos.

Les manipulations appliquées sur la vidéo d'origine (figure V.1) pour construire les vidéos requêtes sont :

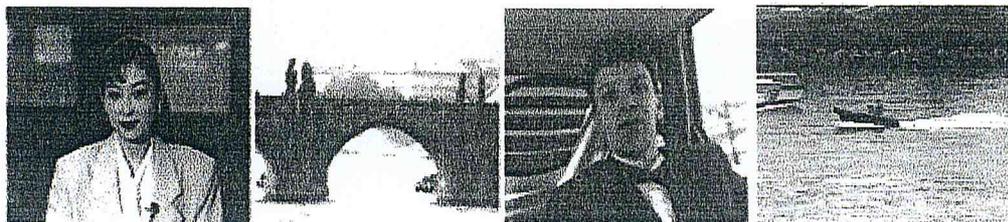


Figure V.1 La vidéo d'origine.

2.1. Modification du contenu d'une image de la séquence

Cette manipulation est faite en remplaçant certaines parties de l'image de la séquence par d'autres images (figure V.2).

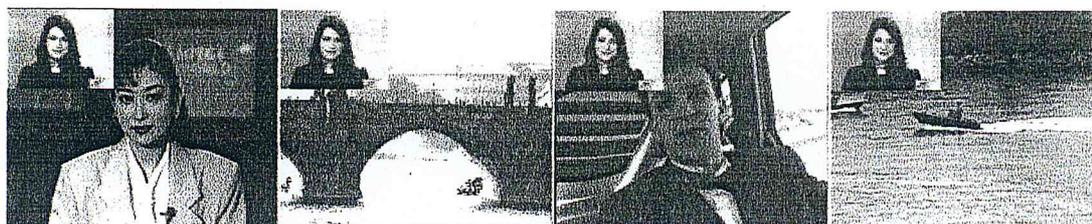


Figure V.2 Des images de la vidéo requête (modification du contenu d'une image de la séquence).

2.2. Insertion de motif

L'une des transformations vidéo les plus courants est l'insertion de logos ou textes (figure V.3).

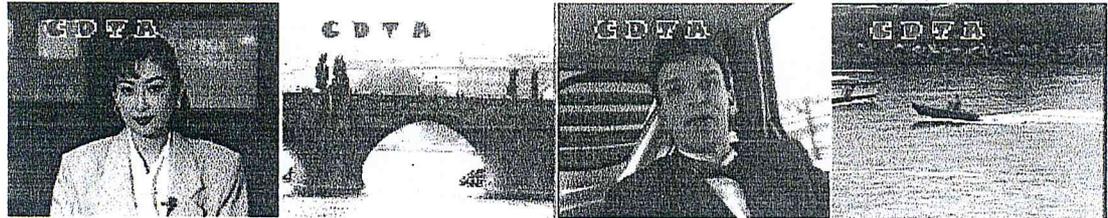


Figure V.3 Des images de la vidéo requête(insertion du texte).

2.3. Perte d'image

Des Images aléatoires peuvent être supprimées en faisant une copie soit intentionnellement pour éviter la détection ou dans le cadre du ré-encodage.

2.4. Recadrage

Lorsque nous recadrer une image, on définit une zone rectangulaire à l'intérieur d'un cadre d'image de la vidéo et supprime les pixels de l'image qui sont en dehors de cette région (figure V.4).

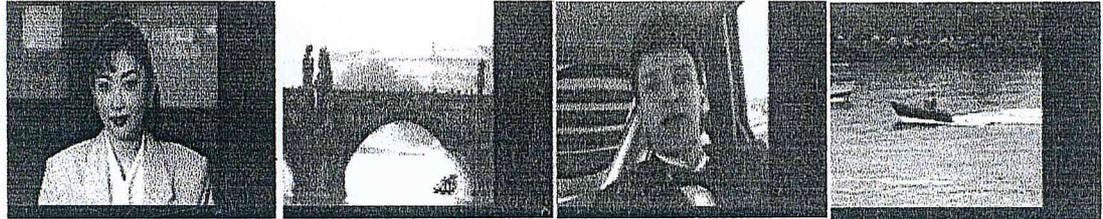


Figure V.4 Des images de la vidéo requête(recadrage).

2.5. Décalage (shift)

La copie de la vidéo a été traduite soit horizontalement, verticalement ou en deux (figure V.5).

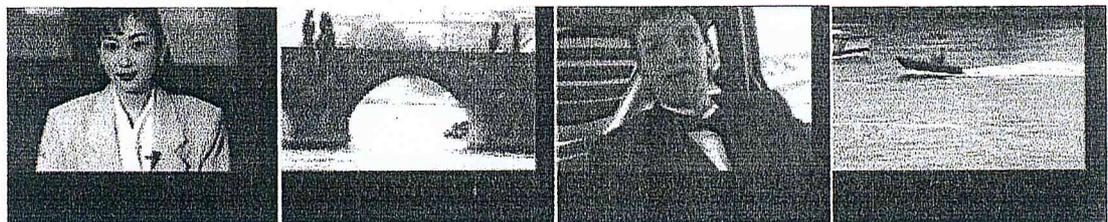


Figure V.5 Des images de la vidéo requête(décalage).

2.6. Retournement (flip)

Le clip est mis en rotation autour d'un axe vertical situé au centre du frame (figure V.6).

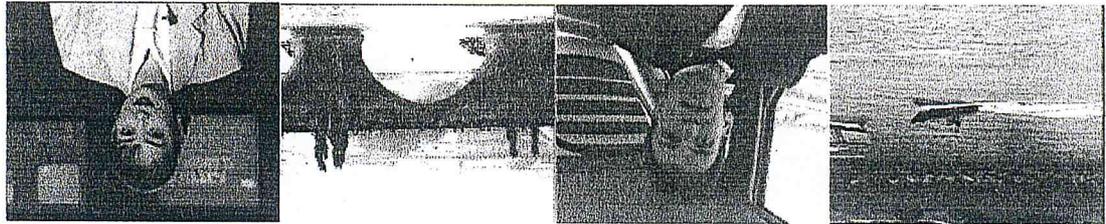


Figure V.6 Des images de la vidéo requête (retournement).

2.7. Zoom

Le zoom agrandit une partie de la vidéo afin qu'elle remplisse l'écran. Cet effet peut également être accompli par l'utilisation de recadrage et mise à l'échelle en combinaison (figure V.7).

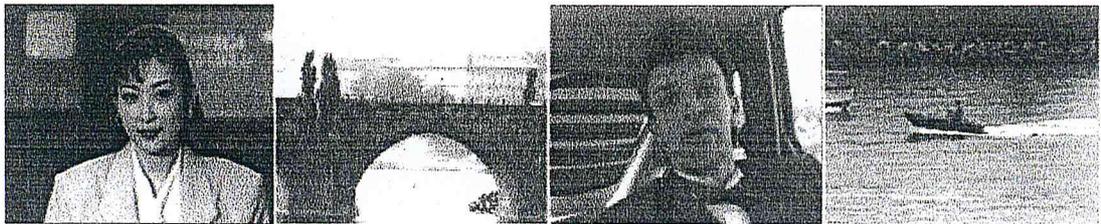


Figure V.7 Des images de la vidéo requête (zoom).

2.8. Contraste

Contraste est la propriété d'un objet qui permette de le distinguer des autres objets. Elle est affectée par la couleur et la luminosité de l'objet par rapport à d'autres objets dans le champ de vision. Plusieurs modèles mathématiques pour définir contraste existé. Dans l'espace de couleurs YCrCb, la composante de luminance (Y) est utilisée en tant que mesure de l'intensité lumineuse. Le contraste peut être calculé comme la différence de luminance sur la luminance moyenne. Ainsi les pixels qui sont proches de la luminance de pixels qui l'entourent auront un faible contraste et ne seront pas aussi distinguée (figure V.8 et figure V.9).

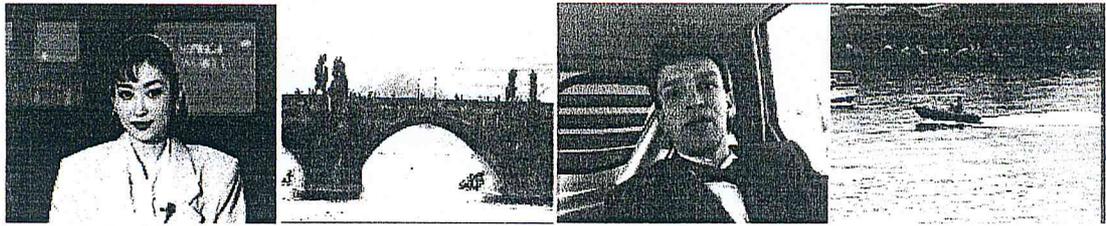


Figure V.8 Des images de la vidéo requête(augmentation de contraste).

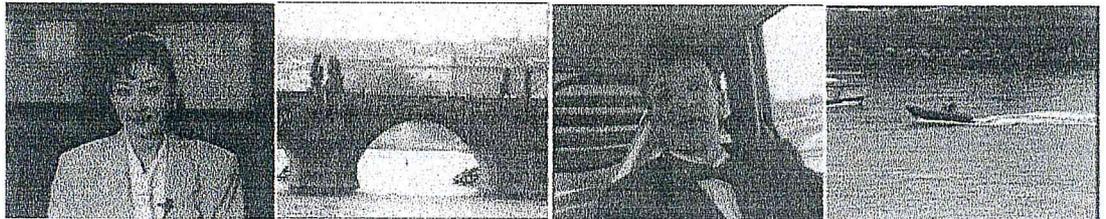


Figure V.9 Des images de la vidéo requête(diminution de contraste).

2.9. Luminance

Augmenter ou diminuer la luminance de la vidéo (figure V.10 et figure V.11).

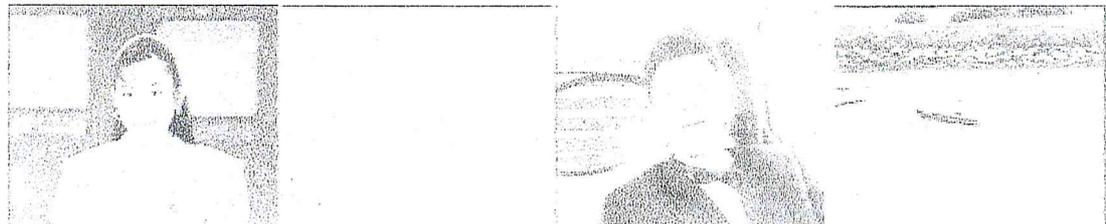


Figure V.10 Des images de la vidéo requête(augmentation de luminance).

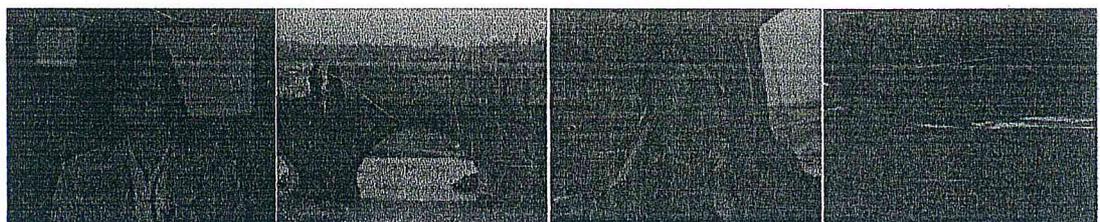


Figure V.11 Des images de la vidéo requête(diminution de luminance).

3. Les résultats des tests

Ce paragraphe comprend les résultats de détection en générant la signature à partir de tous le contenu de la séquence vidéo (contenu de chaque image vidéo) et le résultat de la détection en prenant en compte uniquement les zones texturées de la vidéo.

3.1.Détection basée sur tous le contenu de la vidéo

Les distances entre les histogrammes de la vidéo d'origine et celles des vidéos modifiées (requêtes) sont illustrées sur le tableau V.2 et Figure V.12

Tableau V.2 Les distance entre les histogrammes de la vidéo d'origine et les vidéos modifiées en utilisant tout le contenu pour l'extraction de la signature.

	VLBP Basic	VLBP de Rotation	VLBP de Rotation Uniforme
Modification du contenu d'une image de la séquence	1,059	1,028	0,0124
Insertion de motif (texte)	0,0155	0,014	0,00021
Perte d'image	0,1042	0,1045	0,003
Recadrage	2,857	2,922	0,224
Décalage	2,946	3,018	0,245
Retournement	0,213	0,00130	0
Zoom	0,914	0,938	0,0158
Augmentation de Contraste	3,611	3,672	0,428
Diminution de contraste	0,917	0,934	0,042
Augmentation de luminance	3,871	3,915	0,501
Diminution de luminance	0,917	0,934	0,042

Le tableau VI.1, ci-dessus montre les distances obtenues en appliquant la première approche pour les trois méthodes VLBP (VLBP basic, VLBP de rotation, VLBP uniforme) utilisées dans notre système. Les distances obtenues se différent d'une méthode à l'autre selon l'attaque appliquée. D'après le tableau VI.1 on trouve que la méthode VLBP de rotation a donnée de meilleurs résultats par rapport aux deux autres méthodes, seulement dans le cas de «Retournement» la

méthode *VLBP* basic a donnée un meilleur résultat par rapport au *VLBP* de rotation. Ceci est du au fait que *VLBP* de rotation élimine les effets de rotation.

Dans le cas de la méthode *VLBP* de rotation uniforme, les résultats obtenus sont très faibles dans tous les cas de modification. Nous déduisons que cette dernière n'est pas applicable dans les systèmes de détection de copies vidéo.

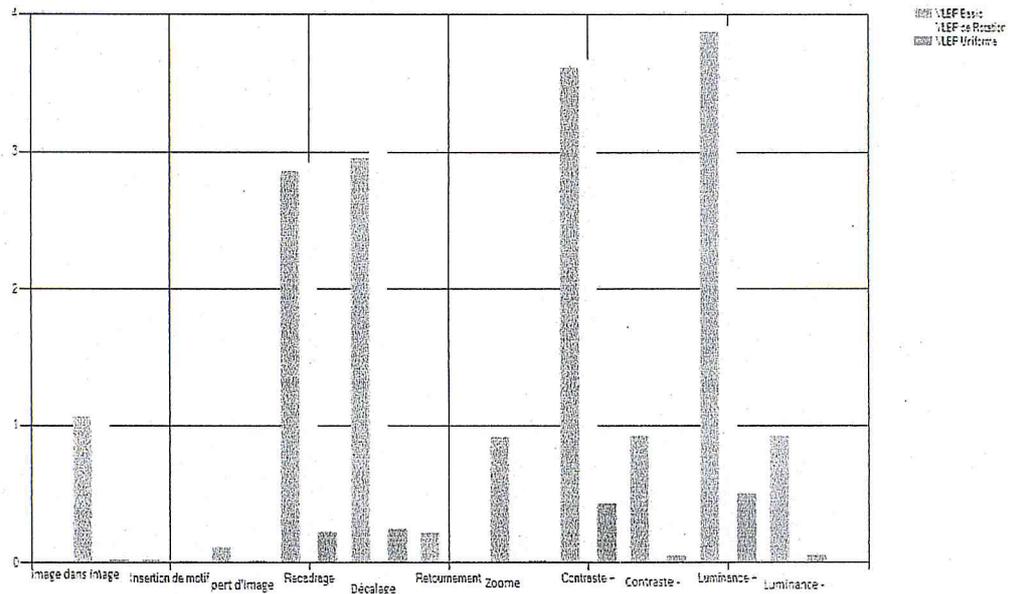


Figure V.12 Les distances entre les histogrammes de la vidéo d'origine et les vidéos modifiées en utilisant tout le contenu pour l'extraction de la signature.

3.2. Détection basée sur les zones texturées de la vidéo

Les distances entre les histogrammes de la vidéo d'origine et celles des vidéos modifiées (requêtes) en prenant en compte seulement les zones texturées de la vidéo sont illustrées sur le tableau V.3 et Figure V.13.

Tableau V.3 Les distance entre les histogrammes de la vidéo d'origine et les vidéos modifiées en utilisant tout les zones texturées pour l'extraction de la signature.

	VLBP Basic	VLBP deRotation	VLBP de Rotation Uniforme
Modification du contenu d'une image de la séquence	0,270	0,367	0,00029
Insertion de motif (texte)	0,426	0,425	0,0052
Perte d'image	0,0154	0,0134	0,00013
Recadrage	3,860	3,896	0,111
Décalage	4,446	4,517	0,150
Retournement	0,176	0,0061	0,000016
Zoom	1,236	1,129	0,0255
Augmentation de Contraste	5,265	5,327	0,231
Diminution de contraste	0,360	0,314	0,00688
Augmentation de luminance	4,804	4,834	0,186
Diminution de luminance	7,248	7,334	0,575

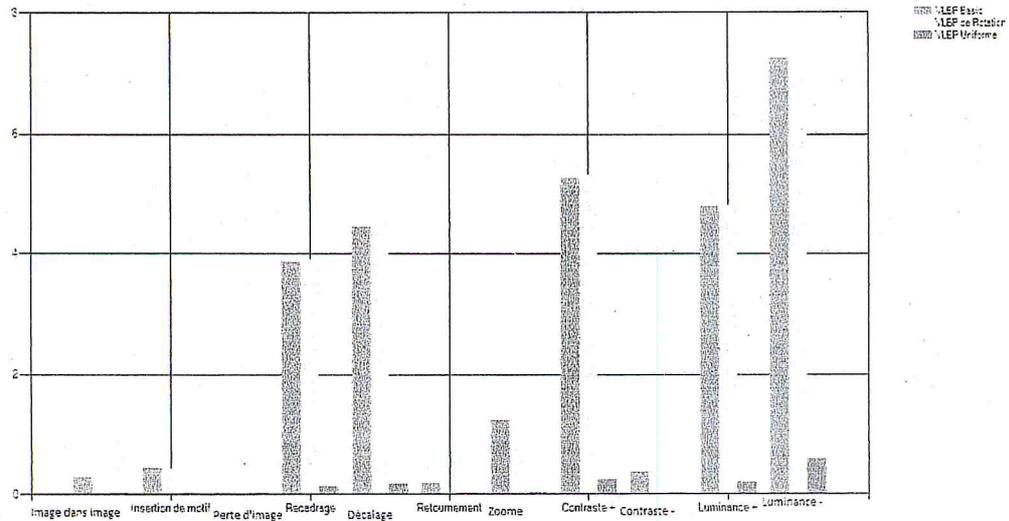


Figure V.13 Les distances entre les histogrammes de la vidéo d'origine et les vidéos modifiées en utilisant tout les zones texturées pour l'extraction de la signature.

D'après les résultats obtenus, nous constatons qu'en utilisant uniquement les zones texturées de la vidéo, le système de détection est plus performant que pour certains types d'attaques telles que la modification des niveaux de gris de la luminance et de contraste.

Au point de vu du temps de calcul, bien sure, la détection utilisant les signatures générées à partir des zones texturées est plus rapide. Par exemple pour une séquence de 30 images le temps de calcul est d'ordre de 1.8 secondes par contre, il est de 0.33 secondes en faisant des restrictions sur le contenu.

4. Conclusion

Dans ce chapitre nous avons exposé nos différents tests et expérimentations effectués ainsi que les résultats obtenus. Notre approche de détection de copies vidéo dans le standard de compression H.264/AVC a prouvé son efficacité en utilisant la méthode de *VLBP* de base et de rotation.

Conclusion Générale

Conclusion générale

L'objectif ciblé dans ce mémoire est la détection de la réplique visuelle de la vidéo. Pour ce faire la détection de copie vidéo basée sur contenu est utilisée.

Afin de bien mener le projet, une étude générale du domaine de détection de copie basée sur le contenu est faite. Elle comprend toutes les notions concernant la détection de copie vidéo. Ensuite un état de l'art des techniques de détection de copie vidéo basée sur le contenu est établi afin de s'inspirer de leurs avantages.

Une étude approfondie de la méthode *VLBP* de base, *VLBP* de rotation et *VLBP* de rotation uniforme est fait.

Aussi une étude de fonctionnement de la norme H.264/AVC est faite. Dans cette étude nous avons basée seulement sur les parties utilisées dans le développement de l'application ciblée dans ce mémoire. D'autre part un état de l'art des techniques de détection de copie vidéo basée sur le contenu pour les vidéos compressées par cette norme est incluse.

La solution développée contient deux approches pour l'extraction des caractéristiques. La première utilise tout le contenu de l'image pour l'extraction des caractéristiques alors que dans la deuxième seulement les régions texturées de l'image sont utilisées. Les caractéristiques sont extraites en utilisant la méthode *VLBP* pour les deux approches.

Afin d'évaluer les performances du système réalisé en termes de la robustesse face aux attaques, des tests ont été effectués sur un certain nombre de vidéos.

Tout au long de la réalisation de ce projet, nous avons approché un domaine qui nous semble très intéressant dans un futur proche qui est lié à la sécurité des vidéos compressées par la récente norme H.264/AVC.

En termes de perspectives de ce travail, nous proposons l'amélioration de la performance de système réalisé en utilisant d'autres caractéristiques de la vidéo telle que la chrominance et aussi en terme d'utilisation d'autres modules de la norme H.264/AVC telle que les coefficient DCT.

Bibliographies

- [1]. Muhammet Bas_tan , Ug̃ur Gudukbay , Ozgur Ulusoy , Onur Kucuk, “ Video copy detection using multiple visual cues and MPEG-7 descriptors “.
- [2]. Arun Hampapur , Ki-Ho Hyun ,Ruud Bolle, “ Comparison of Sequence Matching Techniques for Video Copy Detection “, IBM T.J Watson Research Center, 30 Saw Mill River Road, Hawthorne, NY 10532, USA School of Computer and Information Eng, YoungSan University,150,Junam-ri, Kyongman, Korea 626-840.
- [3]. Zhu Liu, Tao Liu, David Gibbon et Behzad Shahraray, “ Effective and Scalable Video Copy Detection “, AT&T Labs – Research, Middletown, NJ 07748, Polytechnic Institute of NYU, Brooklyn, NY 11201.
- [4]. Onur Kuçuktunç, “ CONTENT-BASED VIDEO COPY DETECTION USING MULTIMODAL ANALYSIS “ a thesis submitted to the department of computer engineering and the institute of engineering and science of bilkent university in partial fulfillment of the requirements for the degree of master of science, July, 2009 .
- [5]. R. Cameron Harvey, “ SPATIO-TEMPORAL VIDEO COPY DETECTION “ a Thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in the School of Computing Science , 2011.
- [6]. Mohammad Athar Ali, Eran A. Edirisinghe, “ Efficient Spatiotemporal Matching For Video Copy Detection in H.264/AVC Video “ Journal international des applications informatiques (0975 - 8887) Volume 41 - N ° 15, Mars 2012.
- [7]. Dinkar N. Bhat and Shree K. Nayar, “ Ordinal Measures for Image Correspondence “ Transactions d'IEEE sur L'analyse de structure ET intelligence artificielle, vol. 20, no. 4, avril 1998.
- [8]. Sunil Lee, Chang D, “ Robust Video Fingerprinting for Content-Based Video Identification “ Transactions d'IEEE sur les circuits et systèmes pour la technologie vidéo, Vol. 18, no. 7, juillet 2008.
- [9]. R. Lienhart, C. Kuhmünch et W. Effelsberg, “ On the Detection and Recognition of Television Commercials” Université de Mannheim ,1996.
- [10]. Indyk, P., Iyengar, G., and Shivakumar,“ Finding pirated video sequences on the Internet “.

- [11]. Regunathan Radhakrishnan et Claus Bauer, " Content-based Video Signatures based on Projections of Difference Images " , Regunathan Radhakrishnan and Claus Bauer, Dolby Laboratories Inc 100 Potrero Ave, San Francisco, CA.
- [12]. Geert Willems, Tinne Tuytelaars et Luc Van Gool, " Spatio-Temporal Features for Robust Content Based Video Copy Detection " .
- [13]. Jian Lu " Video fingerprinting for copy identification: from research to industry " , applications " janvier 2009, Society of Photo-Optical Instrumentation Engineers.
- [14]. Guekhakhma said " Authentification du contenu H.264/AVC le tatouage numérique et la signature numérique " mémoire pour l'obtention d'un diplôme d'ingénieur d'état en informatique 2008/2009.
- [15]. Timo Ojala, Matti Pietikäinen, Topi Mäenpää, " Multiresolution Gray Scale and Rotation Invariant Texture Classification with Local Binary Patterns " .
- [16]. M. Naphade, M. Yeung, B. Yeo, " A novel scheme for fast and efficient video sequence matching using compact signatures, 2000 " .
- [17]. Guoying Zhao, Matti Pietikäinen, " Dynamic Texture Recognition Using Volume Local Binary Patterns " Machine Vision Group.
- [18]. Olivier BROUARD, " Pré-analyse de la vidéo pour un codage adapté Application au codage TVHD en flux H.264 " Thèse de Doctorat de l'Université de Nantes, juillet 2010.
- [19]. Adarsh Nagaraja, " SpatioTemporal LBP based video copy detection " .
- [20]. R. Babu, K. Ramakrishnan, " Compressed domain video retrieval using object and global motion descriptors " , Outils et applications multimédias 32 (2007) 93-113.
- [21]. Julien Law-To, Li Chen, Alexis Joly, Ivan Laptev, Nozha Boujemaa, Fred Stentiford, Valerie Gouet-Brunet, Olivier Buisson, " Video Copy Detection: a Comparative Study " .
- [22]. Christian Käs and Henri Nicolas, " Efficient Video Copy Detection: Sequence Matching in the H.264 Compressed Domain " , Laboratoire Bordelais de Recherche en Informatique (LaBRI), University of Bordeaux 1, 33405 Talence, France.

- [23]. Référence H.264/AVC “ Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264 ISO/IEC 14496-10AVC) “ Tech. Rep., Joint Video Team (JVT), Doc. JVT-G050, March 2003.
- [24]. Iain E. Richardson, “ THE H.264 ADVANCED VIDEO COMPRESSION STANDARD “ deuxième édition 2010.
- [25]. AXIS “ La compression vidéo H.264. Nouvelles possibilités dans le secteur de la vidéosurveillance “ 2008.
- [26]. Iain E. G. Richardson, “ H.264 and MPEG-4 Video Compression Video Coding for Next-generation Multimedia “ The Robert Gordon University, Aberdeen, UK.
- [27]. Rita Cucchiara, Costantino Grana, Massimo Piccardi, Andrea Prati, Stefano Sirotti, “ Improving Shadow Suppression in Moving Object Detection with HSV Color Information”.

Les sites web

[sw01]. <http://trace.eas.asu.edu/yuv/>

[sw02]. H. Schwarz, M. Wien, and J. Vieron, "Joint scalable video model JSVM-5 software," JVT-R203 (Output Document from Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG), Jan. 2006. [Online]. Available:<http://ftp3.itu.ch/av-arch/jvt-site/>

[sw03]. <http://www.sunrayimage.com/>

