

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE ABDELHAMID IBN BADIS - MOSTAGANEM



**Faculté des Sciences Exactes et d'Informatique**  
**Département de Mathématiques et informatique**  
**Filière : Informatique**

MEMOIRE DE FIN D'ETUDES

Pour l'Obtention du Diplôme de Master en Informatique

Option : **Ingénierie des Systèmes d'Information**

Présenté par :

**Dja-daouadji abd elmadjid**

**Chaouki khaled**

THEME :

**Création de résumés vidéo à partir de  
descripteurs visuels**

Soutenu le :

Devant le jury composé de :

Nom et Prénom	Grade	Université de Mostaganem	Président
Nom et Prénom	Grade	Université de Mostaganem	Examinateur
Nom et Prénom	Grade	Université de Mostaganem	Encadreur

Année Universitaire 2020-2021

## **Dédicaces**

A

*Nos parents*

*Pour les sacrifices déployés à nos égards ; pour leur patience Leur amour et leur confiance en nous Ils ont tout fait pour notre bonheur et notre réussite. Qu'ils trouvent dans ce modeste travail, le témoignage de notre Profonde affection et de notre attachement indéfectible. Nulle dédicace ne puisse exprimer ce que nous leur devons, Que dieu leur réserve la bonne santé et une longue vie.*

A

*Nos amis surtout \*\*\*En témoignage de nos sincères reconnaissances pour les efforts Qu'ils ont consentis pour nous soutenir au cours de nos études. Que dieu nous garde toujours unis*

A

*Toute personne qui nous a aidé à faire notre projet*

*Djadsouadjjabdelmadjid*

*Phasoukikhaled*

## **Remerciements**

*Nous voulons exprimer par ces quelques lignes de remerciements notre gratitude envers tout d'abord notre encadreur, Mr. Hennikarime-abdelkader, pour ses conseils et son encadrement. Après à tous ceux en qui par leur présence, leur soutien, leur disponibilité et leurs conseils nous avons trouvé courage afin d'accomplir ce projet.*

*En fin, nous ne pouvons achever ce projet sans exprimer notre gratitude à tous les enseignants de l'Université Abel-Hamid Ibn Badis de Mostaganem, pour leur dévouement et leur assistance tout au long de cette année.*

## **Résumé**

De nos jours, la quantité des vidéos sur Internet et leurs domaines d'applications est impressionnante.

Grace à l'évolution technologique où on peut citer à titre d'exemple les vidéos des journaux télévisés, du web, du sport, ou dans le domaine des divertissements comme des films, des séries télévisées, des documentaires. Cette diversité et grande masse de vidéo disponibles sur internet nécessite de développer des outils qui permettent d'organiser ce type de données afin d'en faciliter l'accès et rendre leur localisation plus rapide et plus efficace.

Un résumé de vidéo est très important pour l'utilisateur pour comprendre plus rapidement le contenu de ce dernier et pour effectuer des recherches rapides.

### **Mots-clés:**

résumé vidéo, descripteur visuel, image clé, plan, cut.

## **Abstract**

Now a days, the number of videos of different types and domains has become impressive

Thanks to technological change, we can cite as examples videos of television news, the web, sport, or in the field of entertainment such as films, television series, documentaries

.this diversity and mass of video available on the Internet requires developing tools that allow to organize this type of data in order to facilitate access and make their location faster and more efficient.

A summary of a video is very important for the user to understand the user's Content more quickly and to perform quick searches.

### **Keywords:**

video summary, visual descriptor, key image, shot, cut.

## Liste des figures

Figure N°	Titre de la figure	Page
Figure 01	exemple d'un descripteur de texture	06
Figure 02	Exemples de textures ordonnées	06
Figure 03	Exemples de textures aléatoires	07
Figure 04	Exemples de textures hybrides	07
Figure 05	Base d'images de texture de Brodatz	07
Figure 06	Exemple d'histogramme d'une image couleur	10
Figure 07	Une image 5x5 avec 4 niveaux de gris et ses matrices de cooccurrence	11
Figure 08	exemple d'un descripteur de forme	13
Figure 09	Les premières formes du dictionnaire de Guzman.	14
Figure 10	exemple d'un descripteur CSS	15
Figure 11	exemple d'un descripteur de Fourier	15
Figure 12	exemple de Moments de Zernike de l'image	17
Figure 13	Déférentes invariances des moments de Hu: (a) Image Lena originale, (b) Image à échelle 50%, (c) Image avec rotation (45°).	18
Figure 14	Invariance des moments de Hu a la réflexion : (a)Image Lena originale, (b) Image Lena miroir.	19
Figure 15	Exemples de points d'intérêt détectés par les différentes approches	21
Figure 16	Principe du descripteur SIFT [09]	22
Figure 17	Etapes de la méthode SIFT	22
Figure 18	Etapes de la méthode SURF	23
Figure 19	Les mouvements courants de la caméra	24
Figure 20	Exemples de codecs Vidéo	27
Figure 21	Exemple sur le codage en pixels.	27
Figure 22	Instruction qui fait l'extraction des images .	32
Figure 23	Splash Screen de l'application.	33
Figure 24	Première interface de l'application.	33

<b>Figure 25</b>	interface de résumer et présentera les information de vidéo .	34
<b>Figure 26</b>	Enregistrement des images clés dans le dossier Save.	34
<b>Figure 27</b>	Enregistrement des histogramme dans le dossier S.	35
<b>Figure 28</b>	Interface d'échantionage de vidéo.	35
<b>Figure 29</b>	Affichage des histogramme après échetionage .	36
<b>Figure 30</b>	Tableau des informations sur les plants Génères.	36

## Liste des tableaux

Tableau N°	Titre du tableau	Page
Tableau 01	Moments des images Lena (a), (b), (c)	18

## Liste des abréviations

<b>Abréviation</b>	<b>Expression Complète</b>	<b>Page</b>
<b>RGB</b>	Rouge Vert Bleu	04
<b>XYZ</b>	un espace de couleur défini par la Commission internationale de l'éclairage en 1931	04
<b>HZV</b>	Hue Saturation Value	06
<b>YCbCr</b>	consiste à séparer la notion de couleur en deux concepts(Luminance- Chrominance)	06
<b>CSS</b>	Curvature Scale Space	14
<b>SIFT</b>	Scale Invariant Feature Transform	21
<b>SURF</b>	Speeded Up Robust Features	22
<b>MERL</b>	Mitsubishi Electric Research Laboratories	24
<b>MPEG-7</b>	Moving Picture Experts Group version 7	24

# Table des matières

Introduction Générale .....	4
Chapitre 1 Etat de l'art sur les méthodes de création de résumés vidéo à partir de descripteurs visuels .....	4
1.1 Introduction .....	5
1.2 Espace de couleur.....	4
1.2.1 Espace de RGB .....	4
1.2.2 Espace de XYZ .....	4
1.2.3 Espace de HZV .....	5
1.2.4 Espace de YCbCr .....	5
1.3 Video .....	5
1.4 La texture .....	5
1.5 Type de Texture .....	6
1.5.1 Texture périodique .....	6
1.5.2 Texture aléatoires.....	6
1.5.3 Texture hybrides .....	7
1.6 Les descripteur d'image .....	7
1.6.1.1 Descripteurs globaux / locaux.....	8
1.6.1.2 Descripteurs bas niveau / haut niveau.....	8
1.6.2 Descripteurs de couleurs .....	8
1.6.2.1 L'histogramme de couleur .....	9
1.6.3 Descripteurs de Texture .....	10
1.6.3.1 Matrice de cooccurrence .....	10
1.6.3.2 Transformée de Fourier.....	11
1.6.3.2.1 Formule de Gabor .....	12
1.6.3.3 Les ondlettes .....	13
1.6.4 Descripteurs de Forme .....	13
1.6.4.1 Polygones de Guzman.....	14

1.6.4.2	Descripteur CSS.....	15
1.6.4.3	Descripteurs de Fourier.....	15
1.6.4.4	Moment de Zernike .....	16
1.6.4.5	Les moments de Hu .....	17
1.6.4.5.1	Caractéristiques.....	17
1.6.5	Descripteurs de points d'intérêts.....	19
1.6.5.1	Descripteur SIFT (Scale Invariant Feature Transform).....	21
1.6.5.2	Descripteur SURF (Speeded Up Robust Features).....	22
1.6.6	Descripteurs de mouvement .....	23
1.6.6.1	Descripteur d'activité de mouvement.....	24
1.6.6.2	Le descripteur de mouvement paramétrique (Parametric Motion Descriptor) .....	24
1.7	Algorithme de recherche .....	25
1.8	La fréquence des images (FPS).....	25
1.9	Les codecs vidéo .....	25
1.10	Le codage en pixels .....	25
1.11	Les mesures de similarité.....	25
1.11.1	Les Distances.....	25
1.12	Conclusion.....	25
<b>Chapitre 2 Conception et Implémentation .....</b>		<b>4</b>
1.	Introduction .....	5
2.	Outil et environnement.....	4
2.1	Visual studio 2015/2019 .....	4
2.2	OpenCV.....	4
2.3	Mysql server.....	5
2.4	Python .....	4
2.5	C# .....	5
3.	Notre Approche .....	4
3.1	Fonction d'extraction .....	4
4.	Application.....	4
4.1	Interface utilisateur .....	4

5. Conclusion.....	4
Conclusion Générale.....	26
Bibliographie.....	407

# Introduction Générale

La quantité d'informations audiovisuelles s'est accrue de façon spectaculaire avec l'apparition de l'Internet à haut débit et de la télévision numérique et aussi le développement à grande échelle de la photographie numérique. Retrouver un document parmi cette masse d'information est devenu une tâche complexe et difficile.

Plusieurs systèmes d'indexation et de recherche de documents multimédia par le contenu ont vu le jour, la majorité d'entre eux concernent les images. Un système typique de recherche d'images par le contenu permet aux utilisateurs de formuler des requêtes en présentant un exemple du type de l'image recherchée. Le système identifie alors parmi la collection d'images celles qui correspondent le plus à l'image requête, et les affiche. La correspondance entre l'image requête et l'ensemble des images de la base de données se fait en comparant les caractéristiques de bas niveau des images qui sont des mesures mathématiques de la couleur, la texture et/ou de la forme, etc.

Malgré leur utilité, ce genre de systèmes ne répond pas à tous les besoins des utilisateurs. Elle est souvent confrontée à plusieurs problèmes parmi lesquels, nous pouvons citer : la nécessité de l'intervention d'un humain, leur rigidité, leur subjectivité et les contraintes linguistiques. De plus, l'annotation manuelle des images ne pourra jamais décrire le contenu d'une image d'une manière exhaustive.

Dans notre travail nous nous intéressons à la création de résumés de vidéos à partir de descripteurs visuels. Le chapitre on va présenter les descripteurs visuels de bas niveau et de haut niveau.

# Chapitre 1

## Notions élémentaires de vision par ordinateur et état de l'art sur descripteurs visuels des images

### 1.1 Introduction

Dans ce chapitre nous allons détailler quelques notions élémentaires de vision par ordinateur et un petit état de l'art sur les descripteurs visuels des images.

### 1.2 Espace de couleur :

On trouve dans la littérature différents espaces de couleur dont on présente ici les principaux.

#### 1.2.1 Espace RGB

C'est l'espace de couleur de base. La représentation des couleurs dans cet espace donne un cube appelé « *cube de Maxwell* ». Le système de couleur RGB est un système de couleur additif, c.-à-d. que les couleurs sont obtenues par le mélange des trois couleurs de base qui sont le *rouge*, le *bleu* et le *vert*.

#### 1.2.2 Espace XYZ

L'espace XYZ (CIE 1931) est défini à partir d'une transformation linéaire sur l'espace RGB telle que toutes les couleurs du spectre visible soient contenues dans le triangle xyz. Les coordonnées XYZ des couleurs naturelles ont ainsi toujours des valeurs positives.

#### 1.2.3 Espace HZV

C'est un espace dérivé de l'espace RGB, Il a été formellement décrit en 1978 par Alvy Ray Smith [26]. Les couleurs dans cet espace sont représentées selon des notions de teinte (*Hue*), de pureté (*Saturation*) et de luminosité (*Value*). La teinte caractérise la couleur. Sa valeur varie

entre 0 et 360°. La saturation caractérise la pureté de la couleur, sa valeur varie entre 0 et 100 %. La luminosité correspond à la brillance perçue de la couleur. Elle varie entre 0 et 100 %.

#### **1.2.4 Espace YCbCr**

L'espace YCbCr est souvent utilisé dans la compression des images. La luminance (appelée aussi Luma) représentée par la valeur Y de la couleur et calculée. Par une somme pondérée des composantes de R, G et B :

$$Y = 0,299 * R + 0,587 * G + 0,114 * B$$

Les deux autres composantes de cet espace représentent l'information chromatique et elles sont calculées à partir de Luma :

$$Cr = R - Y$$

$$Cb = B - Y$$

### **1.3 Vidéo :**

La vidéo regroupe l'ensemble des techniques permettant la constitution d'un signal électrique représentant des images animées et la restitution de celles-ci.

Le mot « *vidéo* » vient du latin « *video* » qui signifie : « je vois ». C'est l'apocope de vidéophonie ou vidéogramme.

### **1.4 La texture :**

La texture est la répétition d'éléments de base construits à partir des pixels qui respectent un certain ordre. Le sable, l'eau, l'herbe, la peau sont autant d'exemples de textures.

L'aléatoire joue un rôle particulier dans les textures.

Le traitement des textures se fait par une décomposition de la totalité en plusieurs élémentaires, pas mal de fonctions qui traitent cela, sous différents niveaux.

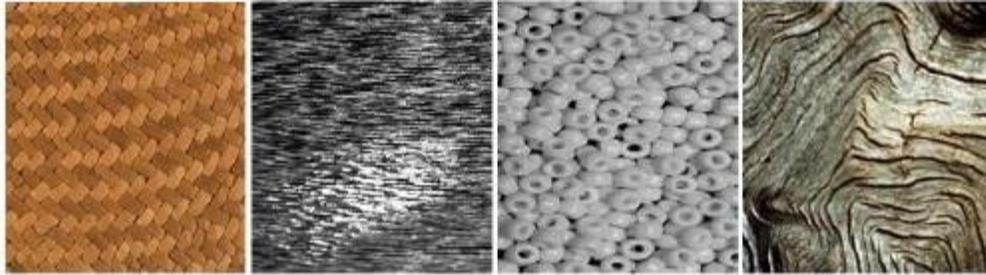


Figure 01 - exemple d'un descripteur de texture.

## 1.5 Type de texture

Selon l'aspect visuel, la texture peut être répartie en trois types:

- 1- les textures périodiques.
- 2- les textures aléatoires.
- 3- les textures hybrides.

### 1.5.1 Texture périodique

Une texture périodique est formée de primitives arrangées d'une façon spécifique. Le motif de base se répète de manière régulière. Il s'agit d'une texture ordonnée (Figure 02).



Figure 02 - Exemples de textures ordonnées

### 1.5.2 Texture aléatoires

Une texture aléatoire semble totalement désordonnée, où Il est impossible d'isoler un motif de base (Figure 03).



Figure 03 -Exemples de textures aléatoires

### 1.5.3 Texture hybrides

Une texture hybride possède simultanément le caractère déterministe et aléatoire stochastique (Figure 04).



Figure 04- Exemples de textures hybrides

Les spécialistes utilisent un catalogue d'images représentant différents types de textures (Figure 05). Ces images sont considérées comme une base de référence, pour valider les différentes méthodes d'analyse de texture [**Brod.66**].

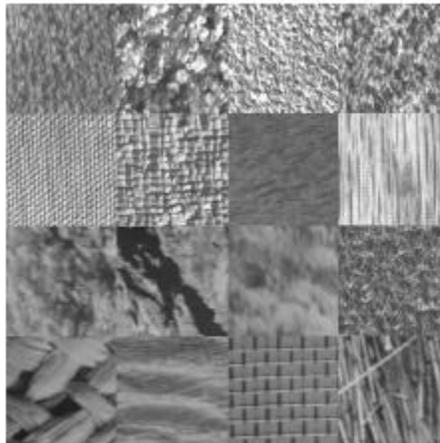


Figure 05 - Base d'images de texture de Brodatz

## **1.6 Le descripteur d'image**

Un descripteur d'images est une description numérique du contenu de l'image. Il doit être compact et précis, de plus une distance ou une mesure de similarité doit permettre de comparer les descripteurs et indirectement le contenu qu'ils représentent.

Nous nous intéressons dans ce manuscrit uniquement aux descripteurs visuels.

Dans les descripteurs visuels on peut distinguer : les descripteurs de couleur, de texture, de forme et de mouvement.

Ces descripteurs peuvent être catégoriser selon le niveau d'extraction .On peut catégoriser les descripteurs des images selon plusieurs critères :

### **1.6.1.1 Descripteurs globaux / locaux :**

- ❖ Descripteurs globaux : extraits à partir de l'image entière .
- ❖ Descripteurs locaux : extraits sous forme d'un ensemble de caractérisations pour chaque primitive de l'image, c'est à dire par zones d'intérêt (points, blocs, régions ,etc.). Par exemple, la couleur pour chaque région de l'image.

### **1.6.1.2 Descripteurs bas niveau / haut niveau :**

- ❖ Descripteurs de bas niveau : décrivent le contenu bas niveau de l'image, principalement en termes de couleurs, textures et formes. Ce sont les descripteurs les plus utilisés dans les systèmes actuels, car les plus simples à mettre en place.
- ❖ Descripteurs de haut niveau : ces descripteurs décrivent le contenu sémantique de l'image, et sont principalement des mots clés fournis par l'utilisateur lors de l'indexation.

## **1.6.2 Descripteurs de couleurs**

La couleur est l'information visuelle la plus utilisée dans les systèmes de recherche par le contenu. Ces valeurs tridimensionnelles font que son potentiel discriminatoire soit supérieur à

la valeur en niveaux de gris des images. Avant de sélectionner le descripteur de couleur approprié, la couleur doit être déterminée d'abord.

Dans ce qui suit nous présentons les descripteurs basés sur la couleur.

### 1.6.2.1 L'histogramme de couleur

L'histogramme d'une image couleur de résolution  $M * N$  est la représentation d'une fonction discrète  $v(C)$  qui, à chaque couleur  $C$  présente dans l'image, associe le nombre  $v(C)$  de pixels ayant cette couleur, encore appelé *population de couleur C*. Cette fonction est donc une application de l'ensemble  $R^3$  des couleurs dans l'ensemble  $N$  des entiers naturels :

$$v: \begin{cases} R^3 \rightarrow N \\ C \rightarrow V(C) \end{cases}$$

Le quotient de  $v(C)$  par le nombre total  $M*N$  de pixels de l'image peut être vu comme la densité de probabilité  $k(C)$  d'une variable aléatoire dont l'image constitue un ensemble de réalisations :

La couleur moyenne de l'image est donnée par :

$$K(C) = \frac{v(C)}{MN} \text{ avec } \sum_c K(C) = 1 \quad \bar{c} = \sum_c C \cdot K(C)$$

En règle générale, chaque composante colorimétrique est codée sur 8 bits. L'histogramme d'une image de résolution  $M * N$  est alors constitué par un tableau tridimensionnel comportant  $2^2$  cellules. Chaque cellule de ce tableau doit pouvoir contenir un nombre entier positif ou nul au maximum égal à  $MN$ , donc codé sur un nombre  $nb$  de bits tel que  $nb > \log_2 (MN)$ .

En effet,  $n$  bits ne permettent de compter que jusqu'à  $2^n - 1$ , et un nombre de bits est toujours entier ; de plus, le codage des nombres en informatique fait en général appel à des multiples de 8 bits. L'histogramme occupe alors  $224 \cdot nb$  bits.

L'inconvénient majeur des histogrammes de couleurs qui est le manque de l'information spatiale concernant la distribution de la couleur, des histogrammes locaux ont été mis en œuvre.

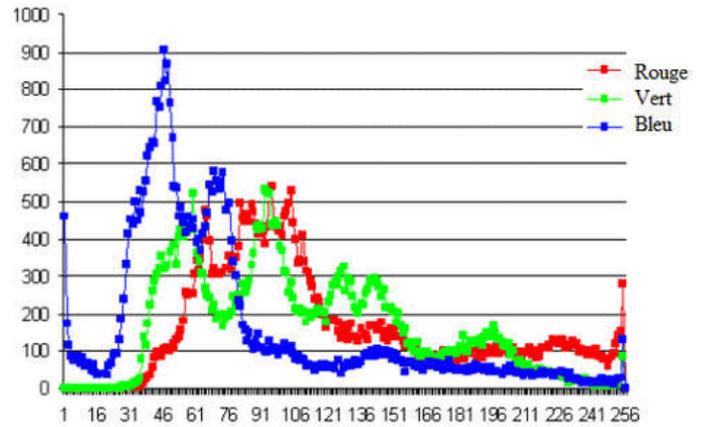


Figure 06-Exemple d'histogramme d'une image couleur.

### 1.6.3 Descripteurs de textures

#### 1.6.3.1 Matrice de cooccurrence :

Dans les années 70 Haralick et Al [04] ont proposé une des premières méthodes de caractérisation de texture basé sur la *Matrice de cooccurrence*.

La matrice de cooccurrence mesure la probabilité pour qu'un couple de niveaux de gris, vérifiant une loi spatiale donnée, apparaisse dans l'image, le niveau de gris d'un pixel d'une image dépend fortement des niveaux de gris des pixels voisins. La légitimité de la démarche provient d'une modélisation Markovienne des textures : la connaissance de toute l'image autour d'un pixel donné est pratiquement équivalente à la connaissance du seul voisinage autour de ce pixel. Haralick a proposé 14 indices à partir de cette matrice pour résumer les diverses propriétés de la texture, dans le but d'intégrer cette information dans une classification [05].

Parmi les 14 paramètres de caractéristiques de texture, nous citons:

a)-Moyenne  $= \sum_j \sum_i p(i, j)$

b)-Variance  $= \sum_j \sum_i (i - \text{Moyenne})^2 p(i, j)$

c)-Energie  $= \sum_j \sum_i (p(i, j))^2$

d)-Inertie  $= \sum_j \sum_i (i - j)^2 p(i, j)$

e)-Entropie  $= \sum_j \sum_i (i - p(i, j))^2 \log(p(i, j))$

$$\text{f)-Le contraste} = \sum_j \sum_i (i - j)^2 p(i, j)$$

$$\text{g)-L'homogénéité} = \sum_i \sum_j \frac{1}{1+(i-j)^2} p(i, j)$$

$$\text{h)-Le cluster shade} = \sum_i \sum_j (i + j - 2\mu)^3 p(i, j)$$

$$\text{i)-Le cluster prominence} = \sum_i \sum_j (i + j - 2\mu)^4 p(i, j)$$

Où :

- (i,j) : coordonnées dans la matrice.
- P(i,j) : valeurs normalisées de la matrice.

L'exemple ci-dessous (figure ...) présente une image à 4 niveaux de gris, et ses 4 matrices de cooccurrence dans quatre différentes directions (0°, 45°, 90°, 135°). La taille de la matrice de cooccurrence est égale à N<sup>2</sup>, où N correspond au nombre de niveaux de gns.

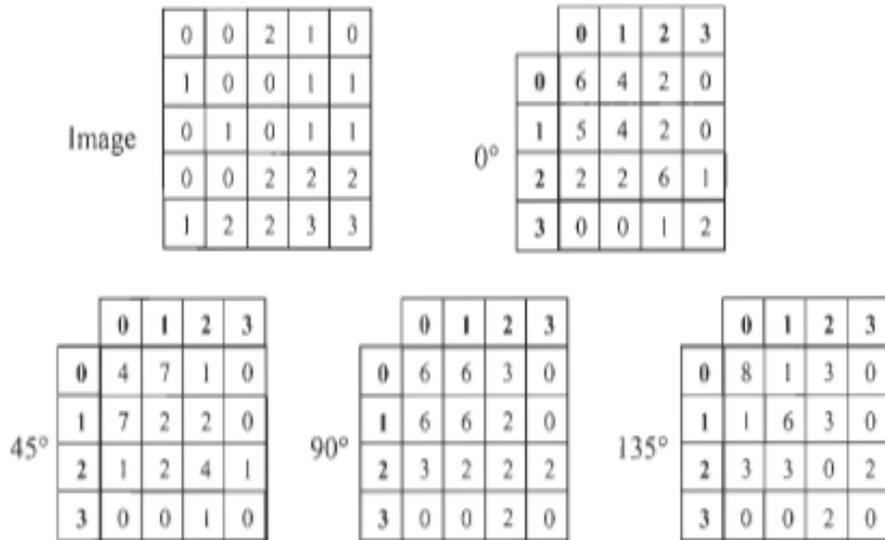


Figure 07 - Une image 5x5 avec 4 niveaux de gris et ses matrices de cooccurrence.

### 1.6.3.2 Transformée de Fourier

La transformée de Fourier F permet de convertir une image I de dimension N × M depuis son domaine spatial vers son domaine fréquentiel selon la formule suivante :

$$F(u, v) = \sum_{n=1}^N \sum_{m=1}^M I(n, m) \exp \left( -2\pi j \left( \frac{un}{N} + \frac{vm}{M} \right) \right) \quad (\text{e.q 01})$$

où

- F(u,v) est la transformée de Fourier de l'image I(n,m).

De manière similaire, [Maillard, 2003] applique une série de six filtres gaussiens sur des spectres radiaux estimés pour quatre orientations principales (0°, 45°, 90°, 135°) dans un voisinage local de l'image. Des attributs calculés à partir des spectres filtrés sont ensuite utilisés pour classifier des imagerie de texture représentant différentes classes (forêt, zones résidentielles, parcelles agricoles, etc.).

Ainsi, [Wassenaar et al., 2002] analysent la localisation des pics correspondant à des parcelles viticoles dans l'espace de Fourier pour estimer l'orientation des rangs et la distance inter-rang.

Cette contrainte a mené à l'utilisation d'autres techniques telles que les filtres de Gabor et les transformées en ondelettes fournissant à la fois une localisation fréquentielle et spatiale dans l'image. Ces techniques sont décrites dans ce qui suit.

### 1.6.3.2.1 Formule de Gabor :

Comme pour l'analyse de Fourier, la connaissance des  $Wf(\lambda, b)$  pour toutes les valeurs réelles de  $\lambda$  et de  $b$  déterminera complètement et de manière unique le signal  $f$ .

On a grâce à Gabor la formule inverse donnant  $f$  à partir des  $Wf(\lambda, b)$  dite formule de reconstruction ou de synthèse et la conservation de l'énergie.

**Théorème 3.1** Soit  $\omega \in L^1 \cap L^2$  centrée en 0, telle que  $|\hat{\omega}|$  soit une fonction paire et  $\|\omega\|_2 = 1$ . C'est ce qu'on appelle la fenêtre. On a pose  $\omega_{\lambda, b}(t) = \omega(t - b)e^{-2i\pi\lambda t}$ ,  $\lambda$  et  $b$  étant des réels, et on considère pour tout signal  $f \in L^2$  les

$$Wf(\lambda, b) = \int_{-\infty}^{+\infty} f(t) \overline{\omega(t - b)} e^{-2i\pi\lambda t} dt.$$

On a la conservation de l'énergie

$$\int_{\mathbb{R}} \int_{\mathbb{R}} |Wf(\lambda, b)|^2 d\lambda db = \int_{-\infty}^{+\infty} |f(t)|^2 dt,$$

et la formule de reconstruction

$$f(x) = \int_{\mathbb{R}} \int_{\mathbb{R}} Wf(\lambda, b) \omega_{\lambda, b}(x) d\lambda db,$$

au sens où si  $g_A(x) = \int_{\mathbb{R}} \int_{\mathbb{R}} |\lambda| \leq A Wf(\lambda, b) \omega_{\lambda, b}(x) d\lambda db$  alors  $g_A \rightarrow f$  dans  $L^2$ .

### 1.6.3.3 Les ondelettes :

La transformée en ondelettes consiste à extraire des caractéristiques de texture, et pour cela elle décompose un signal en utilisant une famille de fonctions  $\varphi_{m,n}(x)$  obtenues par la translation et la dilatation d'une fonction mère  $\varphi(x)$ .

Les ondelettes sont générées à partir de cette fonction  $\varphi(x)$

$$\varphi_{m,n}(t) = \frac{1}{\sqrt{m}} \varphi\left(\frac{t-n}{m}\right) \quad m > 0, n \in \mathbb{R} \text{ (e.q 02)}$$

Où :

- $m$  est le facteur d'échelle (dilatation).
- $n$  est le facteur de translation.
- $1/\sqrt{m}$  est un facteur de normalisation à travers les différentes échelles.

### 1.6.4 Descripteurs de Forme :

La forme est utilisée pour caractériser les objets dans les images. On distingue deux catégories de descripteurs de formes : les descripteurs basés régions et les descripteurs basés frontières.

Les premiers sont utilisés pour caractériser l'intégralité de la forme d'une région, Les seconds portent sur la caractérisation des contours de la forme.

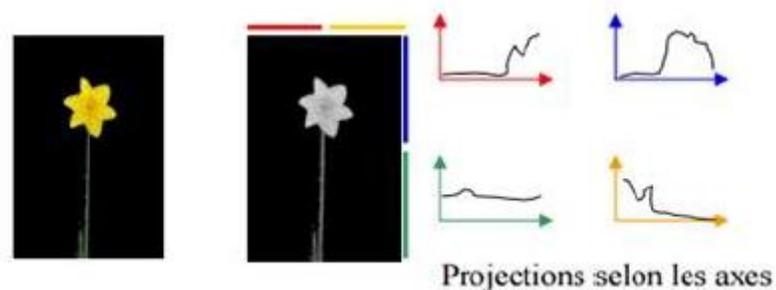


Figure 08 - exemple d'un descripteur de forme.

### 1.6.4.1 Polygones de Guzman :

C'est l'une des méthodes les plus anciennes [Guzman, 1968]. L'approche de Guzman consiste à envelopper l'objet à reconstruire dans des « boîtes » de formes de plus en plus précisément adaptées. Elle peut donc s'appliquer soit à partir d'une image d'étiquettes de la forme à analyser, soit à partir d'une liste des pixels de contour de la forme. Les formes prototypes aux quelles seront comparées les objets sont construites sur un maillage carré de façon systématique et se classent en niveaux en fonction de la longueur de leur périmètre.

Cette approche rencontre plusieurs limites :

- Pour les ordres grands, les formes deviennent très nombreuses et le dictionnaire trop grand. La recherche du représentant se faisant par comparaison au prototype devient très longue.
- Les distances d'une forme à un prototype ne s'imposent pas de façon unique, laissant place à des classes différentes selon le critère adopter.
- Les relations « verticales » entre niveaux sont complexes et ne permettent pas d'accélérer le parcours de l'arbre de façon commode.
- Certaines formes se retrouvent identiques dans des niveaux différents, à un facteur d'échelle près (par exemple le carré élémentaire se retrouve dans tous les niveaux impairs).

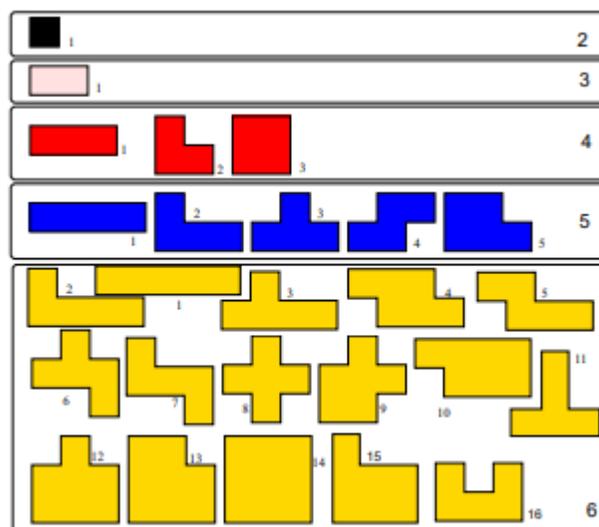


Figure 09 – Les premières formes du dictionnaire de Guzman.

### 1.6.4.2 Descripteur CSS :

Ce descripteur a été proposé initialement par Mokhtarian, Abassi et Kitler [11]. Le CSS est un descripteur multi-échelle, est basé sur la description des parties concaves d'un objet. Invariant à l'échelle, à la translation et à la rotation.

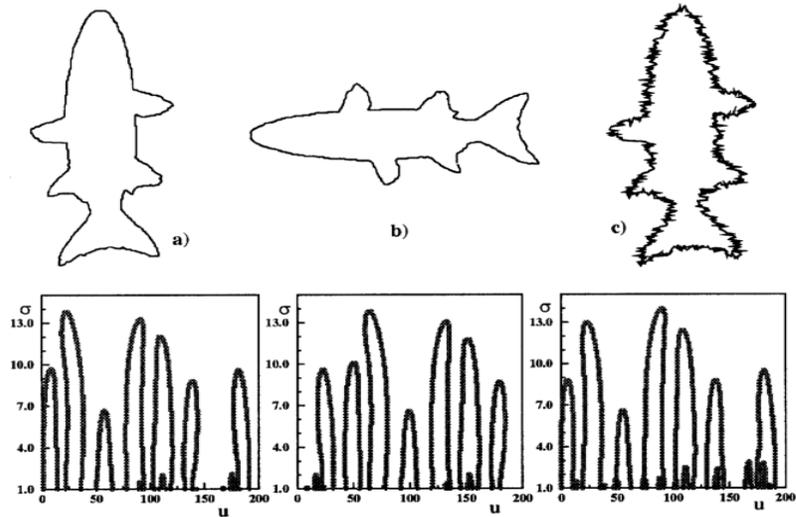


Figure 10 - exemple d'un descripteur CSS .

### 1.6.4.3 Descripteurs de Fourier :

Les descripteurs de Fourier ont pour but de quantifier les variations du contour dans le domaine fréquentiel. Ils permettent de distinguer les composantes principales du signal (basses fréquences) des variations rapides (bords d'échiquetés, bruit, etc : hautes fréquences). Suivant la forme à caractériser, on aura besoin d'harmoniques de plus ou moins haute fréquence.

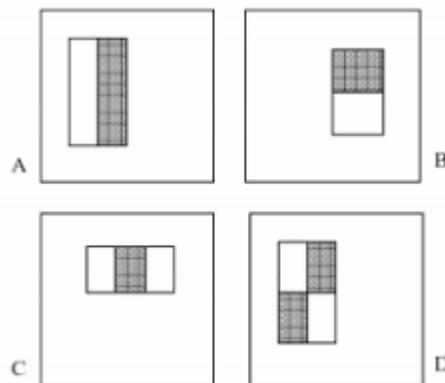


Figure 11 -exemple d'un descripteur de Fourier.

Les descripteurs de Fourier sont calculés à partir du contour des objets. Leur principe est de représenter le contour de l'objet par un signal 1D, puis de le décomposer en séries de Fourier. Les DFs sont généralement connus comme une famille de descripteurs car ils dépendent de la façon dont sont représentés les objets sous forme de signaux.

#### **1.6.4.4 Moments de Zernike:**

Les moments de Zernike d'une image sont construits par projection sur ces fonctions. Elles possèdent trois propriétés principales : l'orthogonalité, l'invariance en rotation et le compactage de l'information, et les basses de fréquences de l'image sont principalement codées dans les moments de bas ordres.

Ensuite, ces moments constituent un descripteur robuste à fort pouvoir descriptif. Cependant, la manière classique de comparer deux descripteurs ne prend pas en compte l'information de phase des moments complexes. Le nouveau comparateur renvoie une mesure de similarité plus précise ainsi qu'un angle optimal de rotation entre les deux images, tout en conservant la même complexité que l'approche classique. Les expériences démontrent que notre comparateur surpasse le comparateur classique en termes de mesure de similarité. En particulier, la robustesse envers le bruit et la déformation est grandement améliorée.

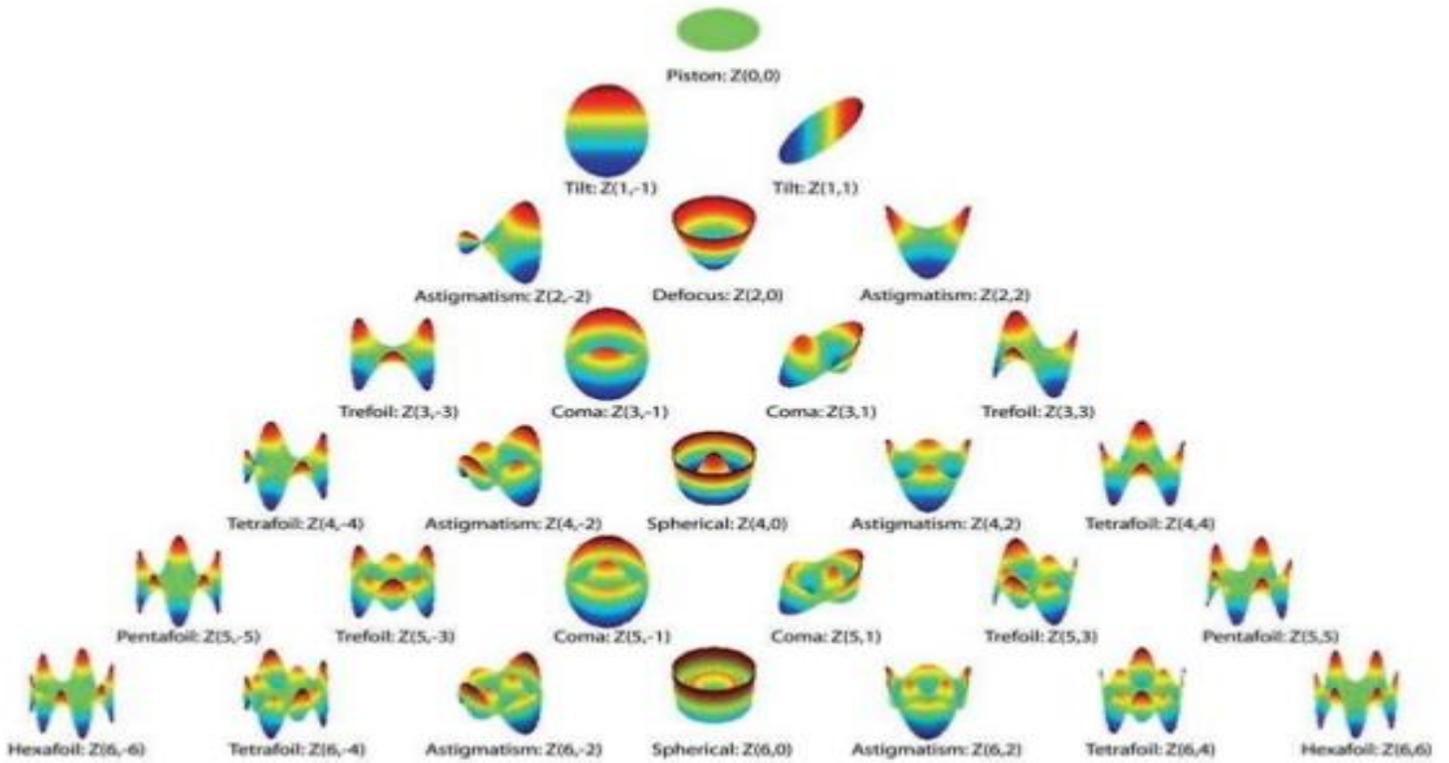


Figure 12 - exemple de Moments de Zernike de l'image

#### 1.6.4.5 Les moments de Hu:

Hu [12] a proposé un ensemble de sept moments invariants aux translations, rotations et changement d'échelle. Ils sont très utilisés dans la littérature pour la description de formes en vue d'une classification ou d'une indexation.

Hu a introduit un ensemble de moments invariants d'ordre 2 et 3 pour lesquels il a démontré qu'ils étaient invariants vis-à-vis les translations, les rotations et les changements d'échelle.

##### 1.6.4.5.1 Caractéristiques:

Les moments de Hu sont des descripteurs très efficaces grâce à leur discriminante et invariance.

Ils présentent plusieurs caractéristiques on peut les résumées comme suit :

- Les sept moments sont invariants : aux translations, aux changements d'échelle, aux rotations. Ainsi, les moments ont la capacité de décrire les images indépendamment de leur position, taille et rotation.



Figure 13 -Déférentes invariances des moments de Hu: (a) Image Lena originale, (b) Image à échelle 50%, (c) Image avec rotation (45°).

	Image originale	Image a échelle (50%)	Image avec rotation (45°)
Moments 1( $\theta_1$ )	0.00136231116696452	0.00136220379791500	0.00136227079570425
Moments 2( $\theta_2$ )	7.02880130654029e-09	7.01844258006804e-09	7.03714910093645e-09
Moments 3( $\theta_3$ )	1.25880893191976 e-12	1.25821022074247e-12	1.21047235313512e-12
Moments 4( $\theta_4$ )	1.21288689507751 e-11	1.21231175733490e-11	1.21327223424181e-11
Moments 5( $\theta_5$ )	-2.30096163064288 e-24	-2.28427250668171e-24	-4.04950793899630e -24
Moments 6( $\theta_6$ )	-9.43332703342113 e-16	-9.42232594345681e-16	-9.42571129386932e-16
Moments 7( $\theta_7$ )	4.73366911987338 e-23	4.72924778469978e-23	4.63192386152712e-23

Tableau 01 -Moments des images Lena (a), (b), (c)

- Les moments de Hu sont également invariant aux réflexions, l'invariance aux réflexions peut être problématique quand il s'agit de reconnaître des images "miroirs", c'est pourquoi Hu a ajouté le septième invariant. Ce dernier permet de distinguer l'image miroir.



Figure 14 - Invariance des moments de Hu a la réflexion :

(a)Image Lena originale, (b) Image Lena miroir.

- Les moments de Hu peuvent être utilisés comme des descripteurs de rectangulaire, ellipticité et circularité.

### 1.6.5 Descripteurs des points d'intérêts

Nous pouvons définir un point d'intérêt comme étant un point dans l'image où des changements significatifs se produisent. Des exemples de point d'intérêt sont les coins, les jonctions, les points noirs sur fond blanc ou tout autre point marqué par un changement important de la texture. En 1988, Harris [46] présente une extension du gradient 2D permettant de mettre en évidence les points d'intérêt spatiaux (noté SIP pour "Spatial Interest Points"). Les points d'intérêt spatiaux sont définis à partir de la matrice Hessienne H définie par (e.q 03) :

$$H(x, y) = \begin{pmatrix} \frac{\partial^2 I(x,y)}{\partial x^2} & \frac{\partial^2 I(x,y)}{\partial x \partial y} \\ \frac{\partial^2 I(x,y)}{\partial x \partial y} & \frac{\partial^2 I(x,y)}{\partial y^2} \end{pmatrix} \text{ (e.q 03)}$$

avec  $I(x, y)$  représentant l'intensité d'un pixel de coordonnées  $(x,y)$  dans l'image I. Pour introduire une notion de facteur d'échelle, l'image est spatialement lissée par un filtre gaussien de réponse impulsionnelle (e.q 04) :

$$G_s = \left( \frac{1}{\sqrt{2\pi\sigma_s}} \right)^2 \exp \left( -\frac{x^2+y^2}{2\sigma_s^2} \right) \text{ (e.q 04)}$$

Avec  $\sigma$  représentant l'écart-type de la gaussienne, définissant l'importance du lissage. La matrice Hessienne  $H$  devient alors la matrice  $M$  définie par (e.q 05) :

$$M(x, y) = I(x, y) XOR \begin{pmatrix} \frac{\partial^2 G_S(x,y)}{\partial x^2} & \frac{\partial^2 G_S(x,y)}{\partial x \partial y} \\ \frac{\partial^2 G_S(x,y)}{\partial x \partial y} & \frac{\partial^2 G_S(x,y)}{\partial y^2} \end{pmatrix} \text{ (e.q 05)}$$

Laptev propose une extension spatio-temporelle du détecteur de SIP de Harris et Stephens, pour détecter les points d'intérêt spatio-temporels, notés STIP ("Space-Time InterestPoints") dans la suite. La détection des STIP est réalisée en utilisant une matrice Hessienne  $H$  définie, pour un pixel  $(x, y)$  au temps  $t$  d'intensité  $I(x, y, t)$ , par (e.q 06) :

$$M(x, y, t) = I(x, y, t) XOR \begin{pmatrix} \frac{\partial^2 I(x,y,t)}{\partial x^2} & \frac{\partial^2 I(x,y,t)}{\partial x \partial y} & \frac{\partial^2 I(x,y,t)}{\partial x \partial y} \\ \frac{\partial^2 I(x,y,t)}{\partial x \partial y} & \frac{\partial^2 I(x,y,t)}{\partial y^2} & \frac{\partial^2 I(x,y,t)}{\partial x \partial y} \\ \frac{\partial^2 I(x,y,t)}{\partial x \partial t} & \frac{\partial^2 I(x,y,t)}{\partial x \partial t} & \frac{\partial^2 I(x,y,t)}{\partial t^2} \end{pmatrix} \text{ (e.q 06)}$$

La figure 15 présente quelques exemples de calcul des différents détecteurs. C'est un critère important pour les approches basées sur les statistiques spatiales des signatures locales dans les images.



(a) Détection de Harris : 323 points (b) Détection Har-Lap : 255 points



(c) Détection DoC : 558 points

Figure 15 - Exemples de points d'intérêt détectés par les différentes approches .

#### **1.6.5.1 Descripteur SIFT (Scale Invariant Feature Transform):**

SIFT est une méthode développée par David Lowe en 2004 , permettant de transformer une image en ensemble de vecteurs de caractéristiques qui sont invariants par transformations géométriques usuelles (homothétie, rotation) et de manière moins fiables aux transformations affines et à l'illumination. Le point fort de la méthode de Lowe est qu'elle est capable de mettre en correspondance des points distants avec des variations de caméra importantes.

Le descripteur est basé sur la distribution des orientations et positions du gradient dans la région. Le descripteur SIFT est obtenu en divisant la région en 4x4 parties, et en divisant chaque partie en 8 secteurs angulaires. On obtient donc un vecteur d'invariants de taille  $4 \times 4 \times 8 = 128$ .

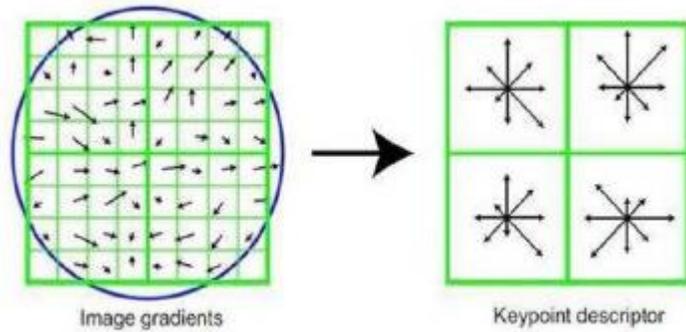


Figure 16 -

Principe du descripteur SIFT [09].

En effet, la mise en place de la méthode nécessite deux étapes principales.

- Premièrement, il est nécessaire d'extraire les caractéristiques d'un objet et de calculer ses descripteurs.
- Deuxièmement, il faut mettre en place une procédure de mise en correspondance (« matching »). C'est le but ultime de la méthode.

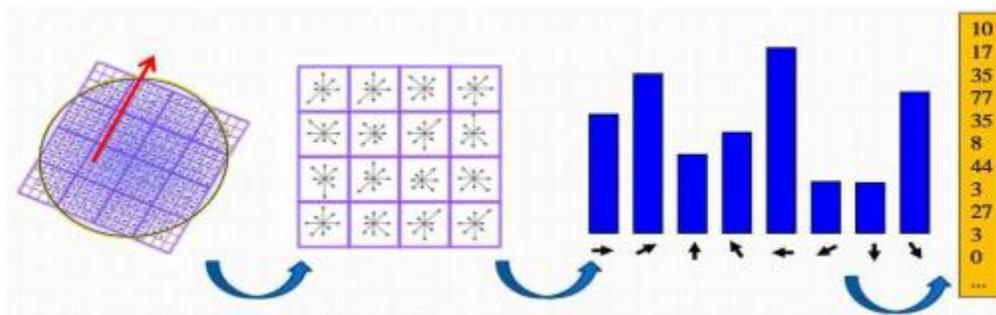


Figure 17 -Etapes de la méthode SIFT

### 1.6.5.2 Descripteur SURF (Speeded Up Robust Features):

SURF est un algorithme de détection de caractéristique et un descripteur, présenté par des chercheurs de l'Eth Zurich et de la Katholieke Universiteit Leuven pour la première fois en 2006, puis dans une version révisée en 2008. Il est utilisé dans le domaine de vision par ordinateur, pour des tâches de détection d'objet ou de reconstruction 3D.[08]

C'est un détecteur de points d'intérêts invariant aux changements d'échelle et aux rotations.

Afin de réduire les temps de traitement, il fait usage d'images intégrales pour le calcul de la convolution, et reprend les points forts des meilleurs détecteurs et descripteurs l'ayant précédé, en utilisant des mesures à base de matrices hessiennes rapides pour le détecteur et un

descripteur basé sur les distributions. Il décrit une distribution de réponses d'ondelettes de Haar dans le voisinage du point d'intérêt. Seulement 64 dimensions sont utilisées, ce qui permet de réduire le temps de calcul des caractéristiques et de recherche des correspondances. L'étape d'indexation est basée sur le signe du Laplacien, d'où l'accélération de la recherche de correspondances et l'amélioration de la robustesse du descripteur. [10]

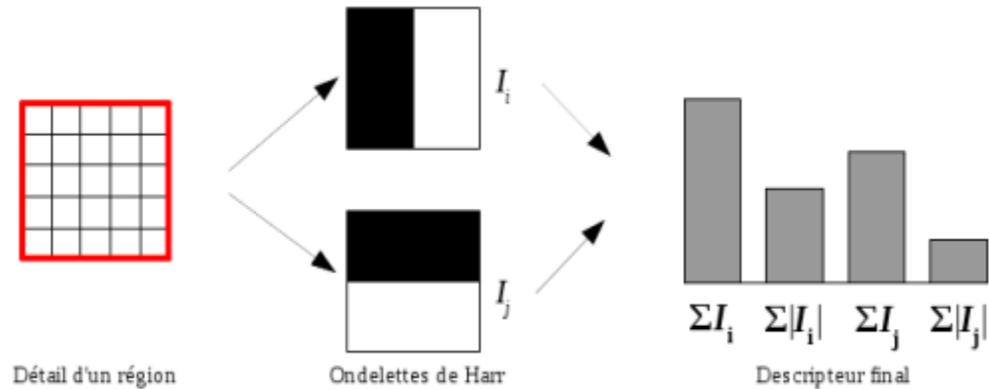


Figure 18- Etapes de la méthode SURF.

### 1.6.6 Descripteurs de mouvements

La vidéo est constituée d'une séquence d'images. Toutefois, l'extraction des caractéristiques visuelles que nous avons décrites dans les paragraphes précédents n'est pas suffisante pour indexer la vidéo. Cela est dû à la nécessité de décrire le contenu dynamique à l'intérieur de la vidéo. Ce dynamisme est le résultat des changements qui se produisent à travers le temps à l'intérieur du contenu de la vidéo. Ces changements sont le résultat du mouvement de la caméra, du mouvement des objets à l'intérieur de la vidéo, ou d'une combinaison des deux. Les mouvements de la caméra les plus courants sont: la translation, la rotation et le zoom. La figure 9 présente une illustration de ces différents types de mouvement de la caméra.

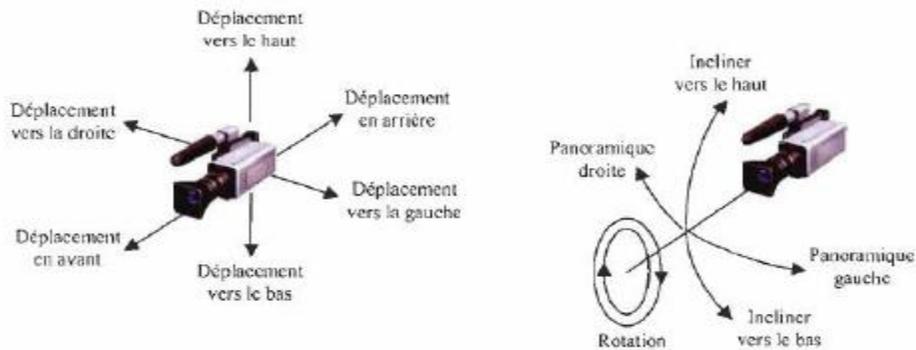


Figure 19 - Les mouvements courants de la caméra.

### 1.6.6.1 Descripteur d'activité de mouvement:

Ce descripteur (*Descripteur promu par Mitsubishi Electric Research Laboratories (MERL), Murray Hills (Etats-Unis).*) offre une mesure de la quantité du mouvement présent dans les séquences vidéos.

Il exploite les vecteurs de compensation de mouvement associés aux *macros blocs MPEG* (*blocs carrés de dimension 16× 16 pixels*), codés en mode prédictif. L'avantage réside ici dans le fait que ces vecteurs sont déjà inclus dans les flux MPEG-1, 2 et 4 et peuvent être déterminés sans nécessiter un décodage complet de la vidéo. Quant aux *macros bloc* codés en mode *intra*, qui n'ont pas de vecteur mouvement associé, leur vitesse est considérée comme nulle.

### 1.6.6.2 Le descripteur de mouvement paramétrique( Parametric Motion Descriptor ) :

Dans MPEG-7, le descripteur de mouvement paramétrique (*Descripteur promu par Institut National des Télécommunications (France), Université de Rochester (Etats-Unis) et Heinrich Hertz Institut (Allemagne).*) vient compléter les descriptions élémentaires de mouvement fournies par les trajectoires, qui se révèlent être souvent insuffisantes.

Le principe consiste à représenter le mouvement d'un objet entre deux trames successives par un modèle de transformation géométrique paramétrique, quadratique, projective, affine, affine simplifié ou constant.

Observons en outre, que ces modèles sont associés à des objets de forme arbitraire, définis en toute généralité comme des régions spatio-temporelles (ensemble de pixels de l'image, sur un intervalle de temps), ce qui offre un cadre unifié pour représenter les mouvements d'objet et les mouvements globaux.

## 1.7 Algorithme de Recherche [26]

1. Charger une vidéo.
2. Lire la vidéo //au choix d'utilisateur.
3. Découper la vidéo en images.
4. Stocker les images extraites dans un dossier spécifié.
5. Calculer le vecteur de description de chaque image.
6. Lire seuil.
7. image référence := la première image.
8. Calculer la distance entre l'image référence et les autres images.
9. Lorsque la distance dépasse le seuil, il y a un cut .
10. Regrouper les images similaire dans un dossier // plan.
11. Image référence := image de cut.
12. Répéter les étapes 8 jusqu'à atteindre le nombre d'images dans la vidéo.
13. Affiché les images clefs // les premières images de chaque plan.

## 1.8 La fréquence des images (FPS)

A l'époque des télévisions cathodiques, la fréquence du courant électrique définissait directement le nombre d'images maximum qu'on pouvait afficher. En France avec le courant EDF à 50 Hertz, les téléviseurs affichaient 25 images par secondes (50 divisé par 2). La norme **Pal** européenne est à 25 images par seconde. Aux Etats-Unis et au Japon où le courant électrique a une fréquence de 60 hertz, la norme est à 30 images par secondes (60 divisé par 2). La norme vidéo américaine **NTSC** est à 30 images par secondes. En vidéo vous retrouverez cette notion de nombre d'images par secondes sous l'appellation anglaise FPS (frame per second), mais avec le numérique et les écrans LCD on peut facilement s'affranchir des anciennes normes PAL ou NTSC. Les caméscopes numériques HD filmeront généralement en 30 images par seconde, mais vous pouvez très bien exporter ensuite vos montages vidéo en 25 images par secondes si vous le souhaitez. Certains caméscopes grand public vont jusqu'à 50

ou 60 images par secondes. Sachez que plus le nombre d'images est grand, plus l'image est fluide et agréable pour l'oeil, mais plus les fichiers sont lourds. Et à partir de 50 images par secondes vous pouvez faire de jolis ralentis qui ne seront pas saccadés.

## 1.9 Les codecs Vidéo

Plusieurs fichiers vidéo différents peuvent avoir la même extension (exemple *.avi* ou *.mts*) mais utiliser des codecs différents. C'est que le format est un conteneur à l'intérieur duquel on trouve un flux vidéo et un flux audio, chacun pouvant être codé avec de nombreux programmes de compression qui n'ont pas les mêmes qualités.

Pour tout souci de compatibilité et de qualité de rendu vous devez donc vous renseigner sur les codecs que vous utilisez dans vos vidéos: il en existe des propriétaires (peuvent poser des problèmes de compatibilité), et des libres, et ils peuvent être plus ou moins répandus et donc plus ou moins faciles à lire sur des lecteurs DVD, des baladeur numériques ou des ordinateurs qui ne savent pas forcément les décoder, à moins d'installer les codecs manquants, si l'appareil en question le permet.

Quelques codecs vidéo courants sont **H264**, **DivX**, **Xvid**, **MPEG-4**, **MPEG-2** etc.

Pour le son **WAV**, **MP3**, **MPEG-2** ...

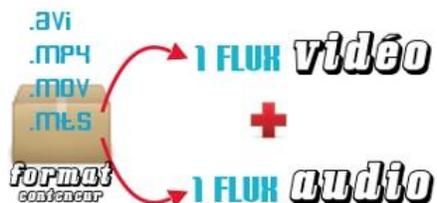


Figure 20 :Exemples de codecs Video

## 1.10 Le codage en pixels

Avec la vidéo, analogique à l'origine, le codage le plus courant des pixels est YUV, qui définit 2 paramètres pour afficher un pixel, la luminance et la chrominance. Luminance gère le niveau de lumière, et chrominance le niveau de couleur rouge et bleu, le vert étant calculé par déduction.

RGB est un autre format de codage des pixels qui utilise simplement les 3 couleurs primaires (RVB en français, Rouge Vert Bleu) mais correspond surtout à la visualisations d'images sur des ordinateurs ou des écrans CRT, LCD, plasma...

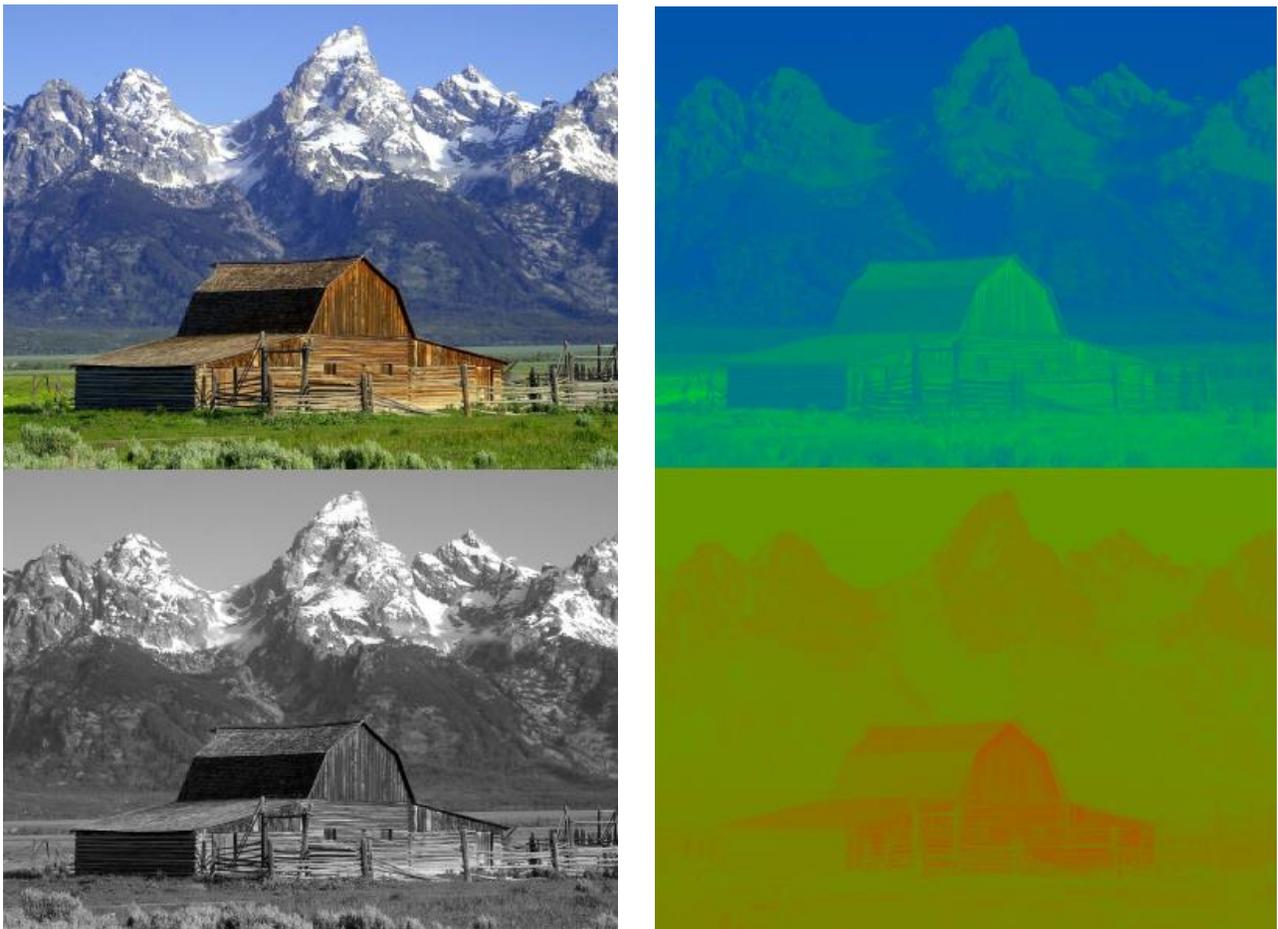


Figure 21 : Exemple sur le codage en pixels.

## 1.11 Les mesures de similarité

### 1.11.1 Définition

En mathématique et en informatique la similarité est un critère important pour l'identification de sous-groupe dans un groupe d'objets, de valeurs (numériques ou non), de données (connus ou reconnus) dans un « espace » ou système. La Mesure de similarité joue un rôle important dans la récupération. Une trame de la requête est accordée à un système qui récupère les vidéos similaires de la base de données. La distance (métrique) peut être qualifiée comme mesure de similarité, qui est la clef de composant dans le contenu en fonction de récupération de la vidéo. Dans l'extraction conventionnelle, les distances euclidiennes entre la base de données et la requête sont calculées et utilisées pour le classement. La trame de la requête est plus proche de la trame de la base de données, si la distance est plus petite. Si X et Y sont des vecteurs de caractéristique 2D de trame d'index de la base de données et la trame de requête, respectivement.

Pour comparer deux histogrammes, nous devons d'abord choisir une métrique ( $d(H_1, H_2)$ ).

### 1.11.2 Les Distances

- **Chi-squarre (Chi-carré)** : est l'une des mesures de distance qui peuvent être utilisés pour trouver la dissemblance entre deux histogrammes. Motivé par le fait que la discrimination de texture par le système de vision humain est basée sur les statistiques de second ordre.

$$d(H_1, H_2) = \sum \frac{(H_1(I) - H_2(I))^2}{H_{12}(I)}$$

- **Corrélation** : Corrélation est souvent utilisée comme un outil descriptif dans la recherche non expérimentale. On dit que deux mesures sont corrélées s'ils ont quelque chose en commun. L'intensité de la corrélation est exprimée par un nombre appelé coefficient de corrélation qui est presque toujours désigné par la lettre. Bien que généralement appelé le coefficient de Pearson de corrélation, il a d'abord été introduite par Galton (dans un article célèbre publié en 1886) et plus tard formalisée par Karl Pearson (1896), puis par Fisher (1935)

$$d(H_1, H_2) = \frac{\sum_I (H_1(I) - \bar{H}_1)(H_2(I) - \bar{H}_2)}{\sqrt{\sum_I (H_1(I) - \bar{H}_1)^2 (H_2(I) - \bar{H}_2)^2}}$$

OU :

$$\overline{H}_k = \frac{1}{N} \sum_J H_k(J)$$

N : nombre des éléments de l'histogramme

• **Bhattacharya** : est une méthode graphique pour la résolution d'une distribution en composantes gaussiennes.[23] ,elle a été développée par C.G. Bhattacharya en 1967 , Il était "un biologiste des pêches" de l'Inde ,Il a développé une technique pour la résolution de la distribution des «longueurs forkal de poissons (pagres)» en «groupes d'âge» ,Il est décrit que "Poissons groupes d'âge» - peuvent être estimées à partir de leur forkal longueur. La technique a depuis été appliquée par quelques individus dans le cadre d'un laboratoire clinique - pour la création d'intervalles de référence.

$$d(H_1, H_2) = \sqrt{1 - \frac{1}{\sqrt{H_1 H_2} N^2} \sum_I \sqrt{H_1(I) \cdot H_2(I)}}$$

**Le seuil** : Apres le calcul des distances, le résultat est une valeur comprise comme suit :

Si le résultat entre 1 et 0.8 : les deux images sont les mêmes

Sinon les deux images sont différentes.

## 1.12 Conclusion

Dans ce chapitre, Nous avons vu qu'il existe une multitude de descripteurs utilisés pour la représentation du contenu visuel de l'image qui sont regroupés en général suivant les catégories couleur, texture ou forme, points d'intérêt, ect. Le choix de ces attributs constitue la première étape de la recherche d'images par le contenu.

Ensuite, le choix de descripteur dépend fortement à la base d'images qu'utiliser et des connaissances à priori qu'on peut avoir sur la base.

# Chapitre 2

## Conception et implémentation

### 1. Introduction

Dans ce chapitre, nous arrivons à la phase la plus importante, c'est celle de l'implémentation et tests. Nous allons décrire successivement notre approche, notre algorithme, les outils de développement employé pour réaliser notre système, ainsi que les différentes interfaces de l'application.

### 2. Outil et environnement :

#### 2.1. Visual studio 2015/2019 :

Depuis les premières versions de Visual Studio, Microsoft tente de répondre aux différents besoins des professionnels de l'informatique en proposant différentes versions de l'IDE. Visual Studio n'est évidemment pas uniquement un éditeur de code destiné aux développeurs, c'est une plateforme où les différents intervenants (développeurs, chefs de projets, testeurs, etc.) en charge d'un projet logiciel vont pouvoir collaborer tout au long du développement d'un projet. [25]

#### 2.2. OpenCV :

OpenCV est une bibliothèque libre de vision par ordinateur. Cette bibliothèque est écrite en C et C++ et peut être utilisée sous Linux, Windows et Mac OS X. Des interfaces ont été développées pour Python, Ruby, Matlab et autre langage. Open CV est orienté vers des applications en temps réel. Un des buts d'OpenCV est d'aider les gens à construire rapidement des applications sophistiquées de vision à l'aide d'infrastructure simple de vision par ordinateur. La bibliothèque d'OpenCV contient près de 500 fonctions. Il est possible grâce à la « licence de code ouvert » de réaliser un produit commercial en utilisant tout ou partie d'OpenCV. Il n'est pas obligatoire de montrer le code du produit et les améliorations réalisées au domaine public.

### 2.3. MySQL server :

MySQL est un système de gestion de base de données (SGBD). Selon le type d'application, la licence est libre au propriétaire. Il fait partie des logiciels de gestion de base de données les plus utilisés au monde, autant par le grand public que par des professionnels, en concurrence avec *Oracle* et *Microsoft SQLServer*. Il stocke les données dans des tables. Les tables sont reliées par des relations définies, qui rendent possible la combinaison de données entre plusieurs tables dans une requête [2].

### 2.4. Python :

Python est un langage de programmation open source **créé par le programmeur Guido van Rossum en 1990**. Il s'agit d'un **langage de programmation interprété** multi-paradigme, Il favorise la programmation impérative structurée, et orientée objet. Il est doté d'un typage dynamique fort, d'une gestion automatique de la mémoire par ramasse-miettes et d'un système de gestion d'exceptions ; il est ainsi similaire à Perl, Ruby, Scheme, Smalltalk et Tcl.

### 2.5. C# :

C# (prononcé C sharp) est un langage de programmation aux nombreux paradigmes mais principalement considéré comme orienté objet. Le langage a été développé par Microsoft en 2002 (date de sortie de la première version). Il est fortement inspiré de langages comme Java ou C++. C# a été développé dans le but d'être une partie intégrante du framework .NET de Microsoft pour Windows. Il est donc aujourd'hui principalement utilisé pour développer des applications tournant sous Windows ou bien des applications web.

## 3. Notre Approche :

On découper la vidéo en des images selon échantillonnage sélectionné (par FPS ou par chaque second).

### 3.1 Algorithme d'extraire images:

1. Charger une vidéo
2. Lire (vidéo).
3. Lire second //chaque second or selon FPS de vidéo.
4. Découper la vidéo en images ( open-cv (vidéo , second)).
5. Stocker l'image(j) dans un fichier.
6. Stocker dans une base de données.

## 3.2. Fonction d'extraction :

```
def extracting_frames(video_path,save_path,skip_frames = second):
    __,file_name=os.path.split(video_path)
    file_name_without_ext = os.path.splitext(file_name)[0]

    length=length_of_video(video_path)
    if length == 0:
        #print("length is 0 , exiting extracting phase")
        return 0
    cap = cv2.VideoCapture(video_path)
    count =0
    random_string = rand_string(5)

    ret,frame = cap.read()
    test_file_path = os.path.join(
        save_path,'{}.jpg'.format(count))

    cv2.imwrite(test_file_path, frame)
    if os.path.isfile(test_file_path):
        #print("saving test frame was Successful,"+'Counting Extraction Phase')

    count = 0
    while ret:
        ret,frame = cap.read()
        if ret and count%skip_frames == 0:
            val='{}.jpg'.format(count)

            cv2.imwrite(os.path.join(
                save_path, val), frame)
```

Figure 22 : Instruction qui fait l'extraction des images.

## 4. Application :

### 4.1. Interface utilisateur:

Dans La première fenêtre de l'application, deux onglets sont disponible, **Fichier** et **Générer** (quelques informations utile), le premier et la deuxième étape est très important et nécessaires pour la bonne manipulation de l'application, successivement et dans l'ordre.

- Cette image présentée une Fenêtre de démarrage de démarrage mon application.



Figure 23 : Fenêtre de démarrage de l'application.

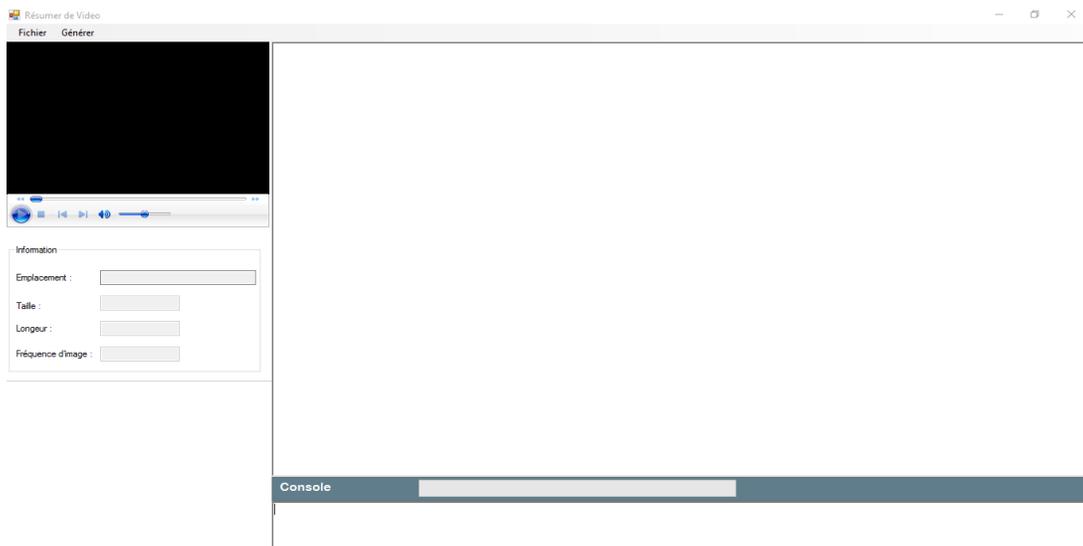


Figure 24 : Première interface de l'application.

- Dans cette interface présente de panel d'information :
  - Un panel présente lecture media et les informations de mon vidéo (Emplacement, taille, fréquence.....), ensuite affiche deux résultats de vidéo :
    1. résultats de découpage.
    2. résultats de résumé.

- Un deuxième panel présente les images après le résumer et le console de traitement.

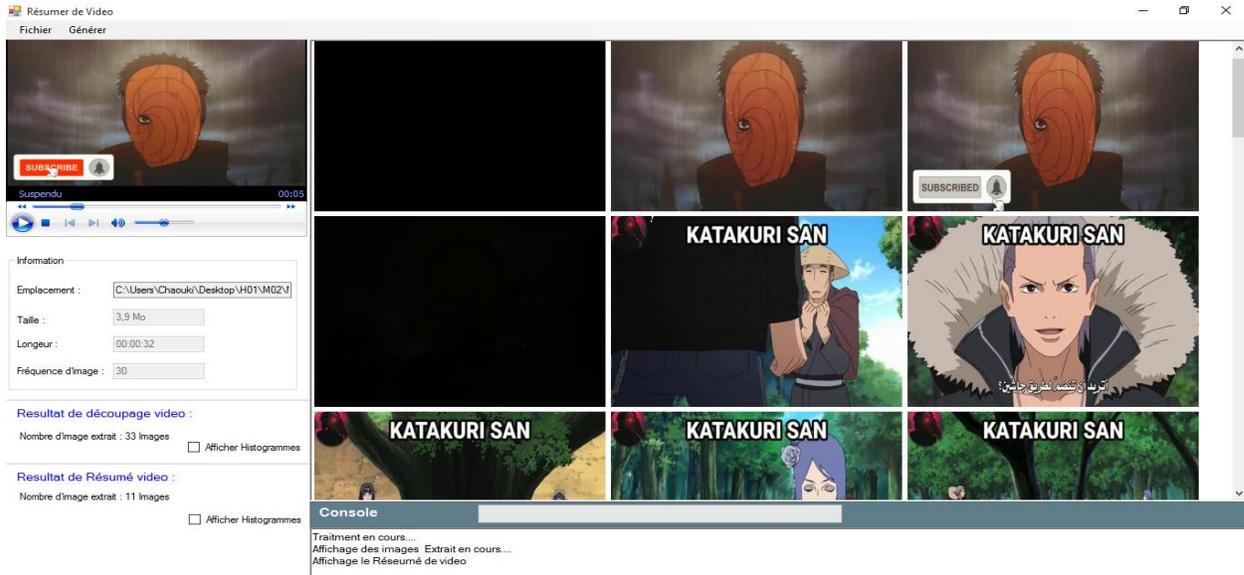


Figure 25 : interface de résumer et présentera les informations de vidéo.

- En même temps pendant le résumer, les images individuelles sont enregistrées dans le dossier Save.

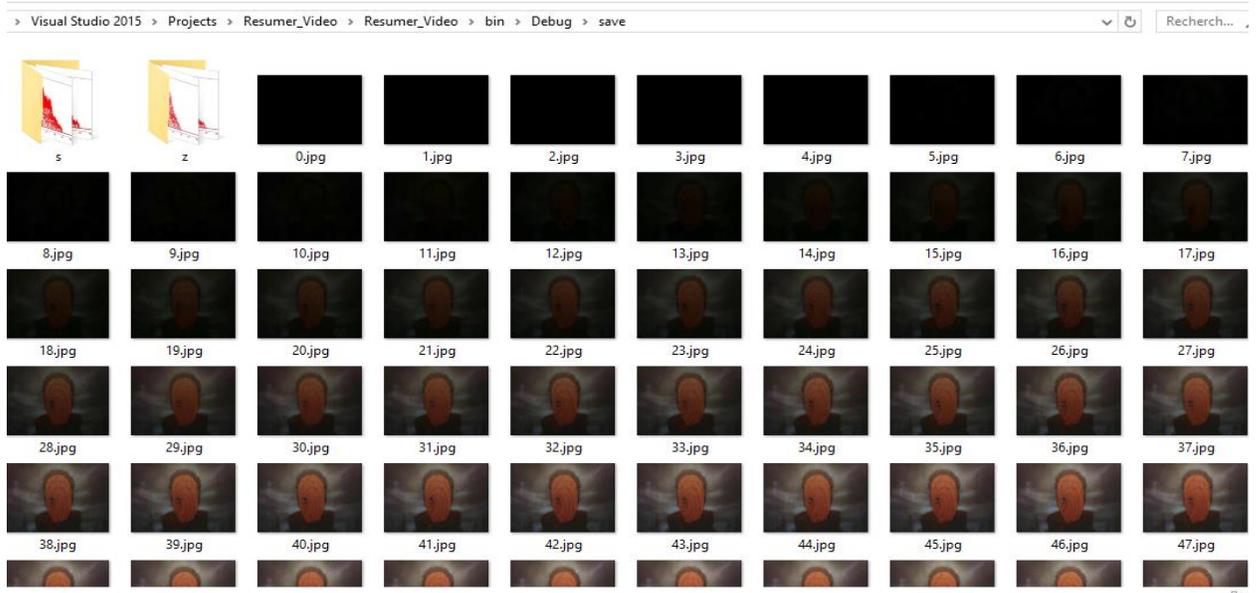


Figure 26 : Enregistrement des images clés dans le dossier Save.

- En même temps pendant l'enregistrement des images dans le dossier, les histogrammes des images vas créer automatique dans autre dossier S qu'on a créé.



Figure 27 : enregistrement des histogrammes dans le dossier S.

- Cette interface présente échantillonnage de vidéo et le résultat de découpage ensuite d'après la clique sur le carré on a affiché les histogrammes.

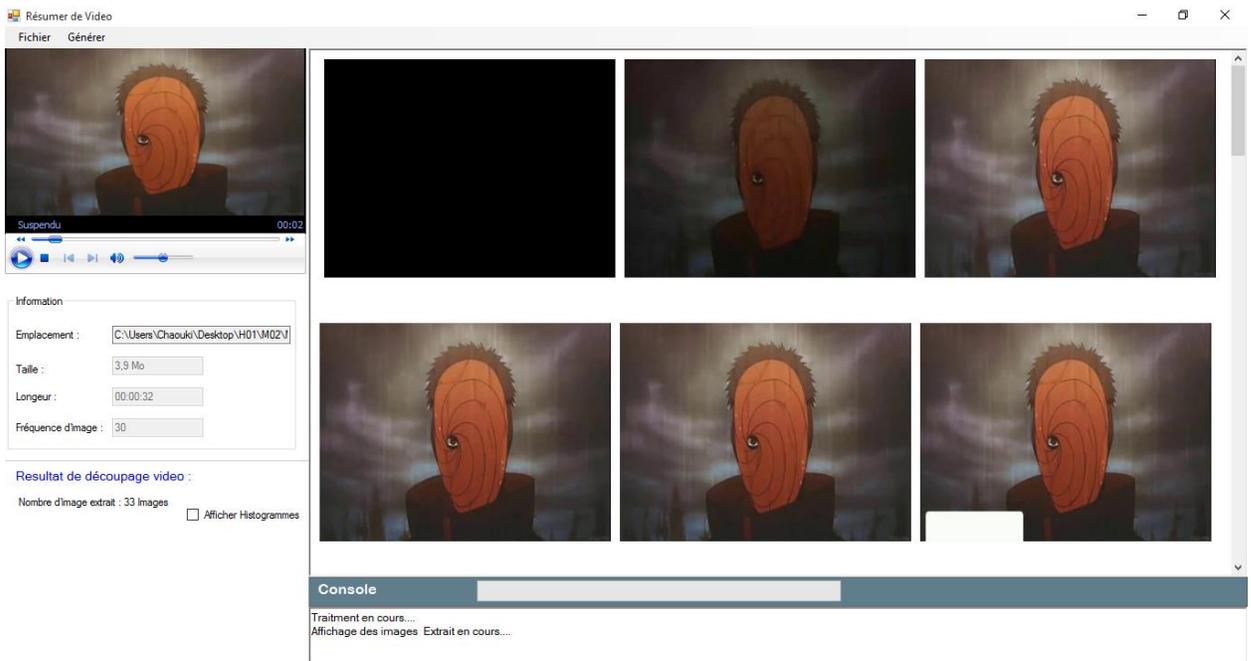


Figure 28 : interface d'échantillonnage de vidéo.

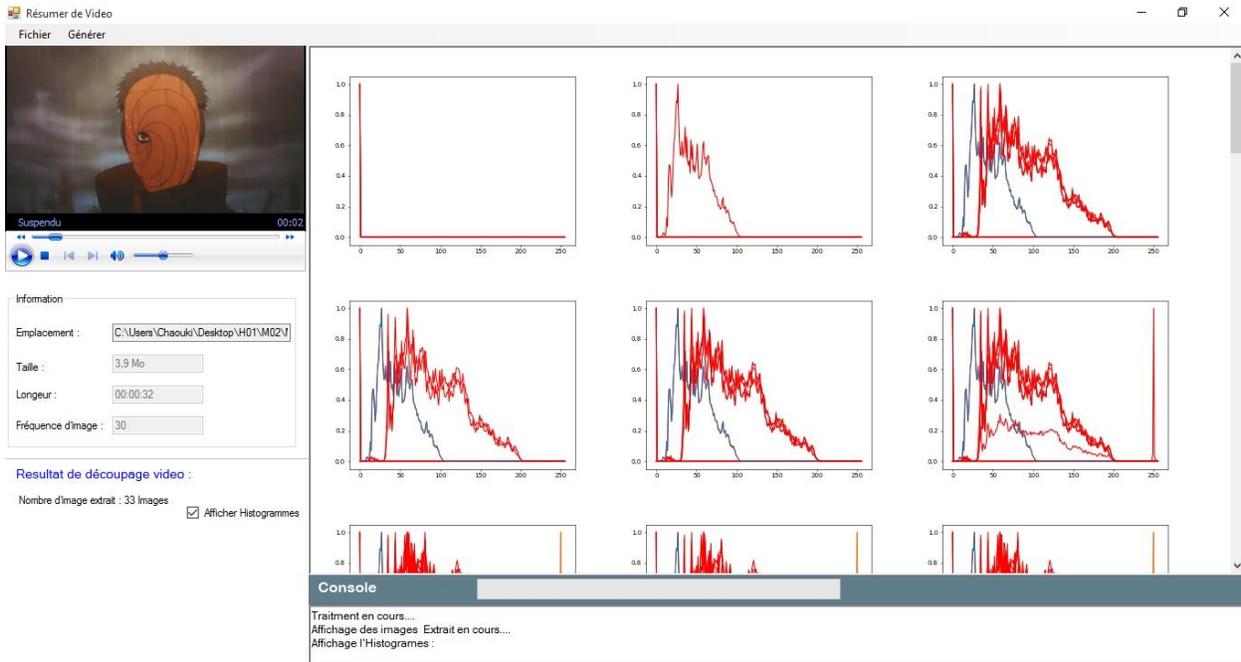


Figure 29 : Affichage des histogrammes après échantillonnage.

- Dernier interface présentera un tableau qui contient les résultats des informations sur les plans après l'opération de résumé .ce tableau présent l'image début et fin, le temps commence et le temps fin et la distance (Nombre d'image) entre l'image début et image fin.

The screenshot shows the 'Résumer de Vidéo' application window. The 'Information' section is the same as in Figure 29. Under 'Resultat de découpage video', it shows 'Nombre d'image extrait: 33 Images' and an unchecked 'Afficher Histogrammes' option. Under 'Resultat de Résumé video', it shows 'Nombre d'image extrait: 11 Images' and an unchecked 'Afficher Histogrammes' option. The main area displays a table titled 'Information sur les plants générés'. The table has columns: N°plan, Image de Début, Image de Fin, Commencer à, Terminer à, and Nombre d'Images. The console shows: 'Traitement en cours...', 'Affichage des images Extrait en cours...', and 'Affichage le Résumé de video'.

N°plan	Image de Début	Image de Fin	Commencer à	Terminer à	Nombre d'Images
0	0.jpg	60.jpg	0 s	60 s	60
1	60.jpg	180.jpg	60 s	180 s	120
2	180.jpg	330.jpg	180 s	330 s	150
3	330.jpg	390.jpg	330 s	390 s	60
4	390.jpg	450.jpg	390 s	450 s	60
5	450.jpg	510.jpg	450 s	510 s	60
6	510.jpg	660.jpg	510 s	660 s	150
7	660.jpg	780.jpg	660 s	780 s	120
8	780.jpg	870.jpg	780 s	870 s	90
9	870.jpg	960.jpg	870 s	960 s	90

Figure 30 : Tableau des informations sur les plants Génèrent.

## **5. Conclusion**

Au cours de ce chapitre nous avons détaillé notre application selon l'algorithme qu'on a défini, chaque interface à son propre rôle, Nous avons commencé par la description de l'environnement de développement. Puis nous avons présentera un exemple de code qui ont servi à réaliser les formules proposes.

Le but majeur de cette application est construire un système capable de faire un résumer de vidéo.

## Conclusion Générale

Durant cette thèse, nous nous sommes intéressés à la création de résumé de vidéo à partir de descripteur visuels d'images qui. L'objectif du résumé est d'extraire les images (ou les extraits) les plus représentatives du contenu de la vidéo afin de donner un aperçu rapide à l'utilisateur.

Le but de ce travail n'a pas été de proposer une nouvelle définition, néanmoins l'idée a été d'extraire, dans les images, une certaine information pertinente de la couleur, la texture et la forme, mouvement, point d'intérêt. Ces informations permettent une recherche efficace dans des bases d'images. Ensuite, nous avons étudié les différentes méthodes d'extractions Des caractéristiques visuelles.

Enfin nous nous sommes intéressés à l'intégration de nos méthodes proposée s'à recherche d'images. Ce système fonctionne avec des descripteurs visuels qui sont extraites d'une image telle que l'histogramme RGB, les ondelettes, la matrice de cooccurrence et les moments de Hu.

Nous avons détaillé ces descripteurs selon le type d'information manipulée en donnant des exemples de chaque type.

# Bibliographie

## Livre, monographie

[Broda.66] P.BRODATZ , « A photographic Album for arts and design», Dover publishing Co.,  
New York, USA, 1966.

[01] **Bertrand Collin, Jérôme Gilles, Emmanuel d'Angelo**, « traitement d'images » ,  
DGA/DET/CEP/ASC/GIP ,Jeudi 27 septembre 2007.

[02] **MASTER ACADEMIQUE**, recherche d'image par le contenu, mémoire de  
maîtrise.

[03] Recherche d'information dans les images, pdf.

[04] **R. M. Haralick, K. Shanmugan, I. Dinstein**.« Textural features for image classification  
», IEEE  
Trans. Syst. Man. Cybern.SMC-3(6): 610–621. 1973.

[05] [www.ummo.dz/IMG/pdf/AMROUN\\_Fazia.pdf](http://www.ummo.dz/IMG/pdf/AMROUN_Fazia.pdf)

[06] **G. Lu, J. Phillips**.« Using perceptually weighted histograms for color-based image  
retrieval».Proceedings of the 4th International Conference on Signal Processing.pp.1150–  
1153.

[07] **A. Divakaran, H. Ito, H. Sun, T. Poon**. « Scene change detection and feature extraction  
for indexing MPEG-2 and MPEG-4 sequences», IEEE Trans. Circuits Syst. VideoTechnology  
,October 1998.

[08] S.PIERARD:«Développement d'une méthode de représentation d'images sur base d'un  
vocabulaire visuel».

[09] David G. Lowe, : Distinctive Image Features from Scale-Invariant Keypoints , dans  
International Journal of Computer Vision, vol. 60, no 2, 2004, p. 91-110

[10] S.PIERARD:«Développement d'une méthode de représentation d'images sur base d'un  
vocabulaire visuel».

[11] **F. Mokhtarian, S. Abbasi, J. Kittler.** « Efficient and robust retrieval by shape content through curvature scale spac », Proc. International Workshop on Image Databases and MultiMedia Search, Amsterdam, The Netherlands, pp 35-42, 1996.

[12] **M. K. Hu.** « Visual Pattern Recognition by moment invariants ». IRE Transaction on Information Theory, Volume 8, n°2:179–187, 1962.

[**Maillard, 2003**]Comparing texture analysis methods through classification. Photogrammetric Engineering and Remote Sensing, vol. 69, no. 4, pp. 357-368.

[**Wassenaar et al., 2002**]WASSENAAR T., ROBBEZ-MASSON J.-M., ANDRIEUX P., 2002. Vineyard identification and description of spatial crop structure by per-field frequency analysis. International Journal of Remote Sensing, vol. 23, no. 17, pp. 3311-3325.

[4]M. Budnik, E.-L. Gutierrez-Gomez, B. Safadiet G. Quenot, Learned features versus engineered features for semantic video indexing, IEEE, 2015.

[11] A. Hamadi, P. Mulhemet G. Quenot, Extended conceptual feedback for semantic multimedia indexing, Multimed Tools Appl, 2014.

[12] J. Sanchez, F. Perronnin, T. Mensinket J. Verbeek, Image Classification with the Fisher Vector: Theory and Practice, International Journal of Computer Vision, 2013.

[16] :W.BOUACHIR :«*Recherche d'images et classification par mots visuels et descripteursflous*».Université de Moncton, avril 2010.

[17]: R.ArandjelovicetA.Zisserman: «*All about VLAD*».

[19]:H.Jegou et M.Douze et C.Schmid et P.Pérez:«*Aggregating local descriptorsinto a compact image representation*».

[26] Hal-Création automatique de résumés vidéo par programmation par contraintes-Haykel Boukadida