

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
MINISTERE D'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

وزارة التعليم العالي والبحث العلمي



SAAD DAHLEB UNIVERSITY – BLIDA

UNIVERSITE SAAD DAHLEB – BLIDA

جامعة سعد دحلب – البليدة

Faculté : Sciences  
Département : Informatique

MEMOIRE

Présenté en vue de l'obtention du diplôme de : MASTER

Intitulé

**Classification des images par combinaison de  
classifieur à base des caractéristiques locales**

Domaine : Maths-Informatique

Filière : Informatique

Spécialité : Génie des systèmes d'informatique

Réalisé par : BETTICHE Nadir

ZINE Omar El Farouk

Promotrice : Mlle Ben blidia nadja

Co-promotrice : Mlle Bacha Sihem

Membres de jury :

Mme S. Benstiti : Présidente

Mlle F.Z. Reguieg : Membre

Mlle L. Oukid : Membre

Promotion

2013/2014

MA-004-214-1

## *Remerciement*

*En préambule à ce mémoire, je souhaite adresser mes remerciements les plus sincères aux personnes qui m'ont apporté leur aide et qui ont contribué à l'élaboration de ce mémoire et à la réussite de cette formidable année universitaire.*

*Nous tenons à remercier sincèrement Mr. Ismain, qui s'est toujours montré à l'écoute et très disponible tout au long de la réalisation de ce mémoire, ainsi pour l'inspiration, l'aide et le temps qu'elle a bien voulu nous consacrer et sans qui ce mémoire n'aurait jamais vu le jour.*

*J'exprime ma gratitude à tous les consultants et internautes rencontrés lors des recherches effectuées et qui ont accepté de répondre à mes questions avec gentillesse en particulier Melle. Bacha Sihem.*

*Que les membres du jury trouvent ici le témoignage de notre reconnaissance pour avoir bien voulu juger notre travail.*

## *Dédicace*

*A mes très chers parents Abd El Karim et Djahida pour leur contribution, leur soutien et leur patience. Que dieu les protège.*

*A ma famille, qui m'a toujours encouragé et soutenu au cours de la réalisation du mémoire.*

*A mes chers sœurs.*

*A Mr Ismain.*

*A Mr Abd Kader.*

*A Mr Aymene.*

*A Melle salima et jawher.*

*A Mme Hayet.*

*A tous mes proches et amis.*

*Je dédie ce modeste travail*

*Zine Omar El Farouk*

## *Dédicace*

*A mes très chers parents Mohammed et Tournes pour leur contribution, leur soutien et leur patience. Que dieu les protège.*

*A ma famille, qui m'a toujours encouragé et soutenu au cours de la réalisation du mémoire.*

*A mes chères sœurs.*

*A Mr Ismain.*

*A tous mes proches et amis.*

*Je dédie ce modeste travail*

*Bettiche Nadir*

## Abstract

In this work we developed in Matlab, a complete system of classification of images based on local features.

This paper summarizes the design of our system of classification, which requires early feature extraction for each image in the database images by SIFT method, then extract the feature vectors. The feature vectors of the images in the database are used as input to our KNN and SVM classifiers, which will classify unknown images and compare them with the training images of the base to tag each image to its corresponding class in the end we present the recognition rate of our system.

Initially we evaluated simple classifiers (KNN) and complex (SVM) with local characteristics (application of SIFT) method for image classification and In a second step we evaluated and studied the impact of the combination of multiple KNN and SVM classifier for the same purpose.

### ***Keywords:***

Image classification, KNN classifier, SVM classifier, combination of KNN and SVM, Scale Invariant Feature Transform (SIFT), local characteristics.

### Résumé

Dans ce travail nous allons développer sous Matlab un système complet de Classification des images à base de caractéristiques locales.

Ce travail résume la conception de notre système de classification, qui nécessite au début l'extraction de caractéristique de chaque image dans la base des images utilisé par la méthode SIFT, puis extraire les vecteurs de caractéristiques. Les vecteurs de caractéristiques des images de la base servent comme entrée à notre classifieur KPPV ou SVM, qui va classer les images inconnues et les comparer avec les images d'apprentissage de la base pour étiqueter chaque image à sa classe correspondante, à la fin on présente le taux de reconnaissance de notre système.

Dans un premier temps nous allons évalué des classifieurs simple (KPPV) et complexe SVM avec des caractéristiques locales (application de la méthode SIFT) pour la classification des images, Dans un second temps nous allons évalué et étudié l'impact de la combinaison de plusieurs classifieur KPPV et SVM pour le même but.

#### *Mots clés :*

Classification des images, Classifieur KPPV, Classifieur SVM, combinaison de plusieurs Classifieur KPPV et SVM, Transformation de Caractéristiques Visuelles Invariante à l'échelle(SIFT), les caractéristiques locales.

<b>Introduction</b> .....	1
 <b>Chapitre I : Reconnaissance Des Formes</b>	
<b>I.1. Introduction</b> .....	3
<b>I.2. Définition de la RDF</b> .....	3
<b>I.3. L'objectif de la RDF</b> .....	4
<b>I.4. Processus de reconnaissance des formes</b> .....	4
<b>I.4.1. Codage</b> .....	5
<b>I.4.2. Prétraitement</b> .....	5
<b>I.4.3. L'extraction des caractéristiques</b> .....	5
<b>I.4.4. La reconnaissance</b> .....	6
<b>I.4.4.1. L'apprentissage</b> .....	6
<b>I.4.4.2. Décision</b> .....	7
<b>I.4.5. Le post-traitement</b> .....	7
<b>I.5. Classification</b> .....	7
<b>I.5.1. Définition</b> .....	7
<b>I.5.2. Les étapes d'une classification</b> .....	8
<b>I.5.3. Les méthodes de Classifications</b> .....	8
<b>I.5.4. Les approches de classifications</b> .....	9
<b>I.5.5. La classification supervisée</b> .....	10
<b>I.5.5.1. Méthode KPPV</b> .....	10
<b>I.5.5.2. Méthode SVM (Support Vector Machines)</b> .....	14
<b>I.5.6. La classification Non-supervisée</b> .....	19
<b>I.5.7. Les types du clustering</b> .....	21
<b>I.5.8. Les algorithmes de Clustering</b> .....	22
<b>I.5.9. Les méthodes hiérarchiques</b> .....	24
<b>I.5.10. Les limites de Clustering</b> .....	26
<b>I.6. Conclusion</b> .....	26

## Chapitre II : Traitement d'Images

II.1. Introduction .....	27
II.2. La chaîne de traitement d'image .....	27
II.2.1. Prétraitement .....	27
II.2.1.1. Le Filtrage .....	28
II.2.1.2. L'histogramme .....	31
II.2.1.3. Binarisation .....	32
II.2.2. Segmentation .....	32
II.2.2.1. Définition de la segmentation .....	33
II.2.2.2. Les différentes approches de segmentation .....	33
II.2.2.2.a. Approche contour .....	33
II.2.2.2.b. Approche Région .....	37
II.3. Conclusion .....	40

## Chapitre III : Extraction Des Caractéristiques

III.1. Introduction .....	41
III.2. Localisation des caractéristiques .....	41
III.2.1. Caractéristiques globales .....	41
III.2.2. Caractéristiques locales .....	44
III.2.2.1. Détection de zones d'intérêt .....	44
III.2.2.2. Terminologie .....	45
III.2.2.3. Typologie .....	45
III.2.3. Scale-Invariant Feature Transform (SIFT) .....	47
III.2.3.1. Qu'est-ce que SIFT? .....	47
III.2.4. Les étapes de la méthode SIFT .....	48
III.2.4.1. Détection des points-clés et calcul du descripteur SIFT .....	48
III.2.4.2. Détection d'extrema dans l'espace des échelles .....	49
III.2.4.3. Amélioration de la précision .....	50
III.2.4.4. Assignment d'orientation aux points d'intérêt .....	51





III.2.4.5. Les descripteurs des points d'intérêt.....	51
III.2.4.6. Utilisation pour la recherche d'objets dans des images.....	52
III.3. Conclusion .....	53
<b>Chapitre IV : Conception &amp; Expérimentation</b>	
IV.1. Introduction .....	54
IV.2. Interface.....	56
IV.3. Base des images utilisées.....	57
IV.4. Extraction des caractéristiques .....	57
IV.4.1. Détection d'extrema.....	57
IV.4.2. Assignation d'orientation aux points d'intérêts.....	58
IV.4.3. Descripteur de point-clé.....	58
IV.5. Classification des descripteurs .....	60
IV.6. Représentation de chaque image par histogramme.....	62
IV.7. Classification.....	62
IV.7.1. Application de KPPV.....	62
IV.7.2. Application du SVM .....	64
IV.7.2.1. SVM linéaire.....	64
IV.7.2.2. SVM non linéaire.....	66
IV.8. La combinaison de plusieurs Classifieurs .....	67
IV.8.1. Fusion des scores.....	67
IV.8.2. La combinaison (fusion).....	68
IV.9. Conclusion.....	69
Conclusion générale.....	70
Bibliographie.....	71

<b>Figure 1.1</b> : Schéma général d'un système de reconnaissance des formes.....	4
<b>Figure 1.2</b> : Les méthodes de classification.....	9
<b>Figure 1.3</b> : Les différents modèles de classification supervisée.....	10
<b>Figure 1.4</b> : Illustration des k-ppv d'un point.....	12
<b>Figure 1.5</b> : Hyperplan séparateur $\langle w, x \rangle + b = 0$ .....	15
<b>Figure 1.6</b> : Hyperplans canoniques et marge maximale.....	17
<b>Figure 1.7</b> : Illustration de regroupement en clusters.....	19
<b>Figure 1.8</b> : Les différents types de clustering.....	21
<b>Figure 1.9</b> : Exemple d'un problème de discrimination à deux classes, avec un séparateur linéaire : la droite d'équation $y=x$ . Le problème est linéairement séparable.....	21
<b>Figure 1.10</b> : K- means.....	23
<b>Figure 1.11</b> : Classification hiérarchique descendante.....	25
<b>Figure 2.1</b> : Quelques prétraitements.....	28
<b>Figure 2.2</b> : Image filtrée (passe-bas).....	29
<b>Figure 2.3</b> : Exemple de filtre.....	29
<b>Figure 2.4</b> : Exemple de masque gaussien.....	30
<b>Figure 2.5</b> : Image filtrée (passe-haut).....	30
<b>Figure 2.6</b> : Matrice filtrage médian.....	31
<b>Figure 2.7</b> : Histogramme.....	31
<b>Figure 2.8</b> : Exemple d'histogramme.....	32
<b>Figure 2.9</b> : Contour et ses dérivées .....	34
<b>Figure 2.10</b> : Segmentation par contour actif .....	36
<b>Figure 2.11</b> : Relation entre pixels.....	37
<b>Figure 2.12</b> : Segmentation par graphe d'adjacence. ....	38
<b>Figure 2.8</b> : Méthode de pyramidage par Quad-tree.....	39

<b>Figure 3.1</b> : Histogramme d'une image couleur sur 64 bins.....	42
<b>Figure 3.2</b> : Illustrations de DOG.....	49
<b>Figure 3.3</b> : Le calcul des points clés.....	40
<b>Figure 3.4</b> : Précision des points clés.....	51
<b>Figure 3.5</b> : Illustration des descripteurs des points d'intérêt.....	52
<b>Figure 4.1</b> : Architecture du système proposé.....	55
<b>Figure 4.2</b> : montre l'interface du programme utilisé.....	56
<b>Figure 4.3</b> : Application du filtre gaussien avec $\sigma=3$ et calcul de la différence de gaussien entre l'image originale et l'image de gaussien.....	58
<b>Figure 4.4</b> : Détermination des orientations à partir des points d'intérêt de l'image dans le voisinage des points d'intérêt.....	58
<b>Figure 4.5</b> : Figure illustre les valeurs des descripteurs d'une image de la base.....	59
<b>Figure 4.6</b> : Figure illustre l'ensemble des fichiers SIFT qui représente les descripteurs calculées à partir des images d'apprentissage et de teste de la base.....	60
<b>Figure 4.7</b> : Figure illustre le résultat de l'application de l'algorithme de K-means sur chaque fichier descripteur.....	61
<b>Figure 4.8</b> : Schématise la distance minimale entre chaque point d'intérêt et cluster d'une image.....	62
<b>Figure 4.9</b> : Un dessin montre que le choix du vecteur vert est dû à la vaste marge entre les deux classes.....	65
<b>Figure 4.10</b> : Montre la classification non linéaire d'un classifieur SVM.....	66
<b>Figure 4.11</b> : Un schéma qui montre le principe de la méthode BORDA.....	68

<b>Tableau 4.1</b> : Tableau représentant le résultat de l'application du classifieur KPPV avec différentes valeurs de K.....	60
<b>Tableau 4.2</b> : Tableau représentant le résultat de l'application du classifieur SVM avec différentes valeurs de K.....	61
<b>Tableau 4.3</b> : Tableau représentant les distances (euclidiennes et khi-2) d'une image avec ces voisins après l'application du KPPV.....	63
<b>Tableau 4.4</b> : Tableau représentant les classes des voisins de l'image de test.....	64
<b>Tableau 4.5</b> : Tableau représentant les degrés d'appartenance d'une image à des différentes classes.....	64
<b>Tableau 4.6</b> : Tableau représentant les degrés d'appartenance d'une image de test à des différentes classes par l'application du SVM linéaire.....	65
<b>Tableau 4.7</b> : Tableau représentant les degrés d'appartenance d'une image de test à des différentes classes par l'application du SVM non linéaire.....	67
<b>Tableau 4.8</b> : Tableau qui représente fusion des scores entre plusieurs classifieurs.....	67
<b>Tableau 4.9</b> : Tableau qui représente la combinaison entre plusieurs classifieurs.....	68

## Introduction générale

Le domaine de l'image numérique est un domaine en pleine expansion. Depuis quelques années, avec l'explosion d'Internet et aussi le développement à grande échelle de la photographie numérique, il n'est pas rare d'avoir des bases d'images numériques contenant plusieurs milliers et même plusieurs dizaines de milliers d'images, que ce soit des bases ciblées pour un domaine d'activité professionnelle (journalisme, tourisme, éducation, musées, ...) ou tout simplement pour les particuliers qui accumulent d'immenses bases de photographies numériques (souvenirs, voyages, famille, événements, ...).

Pour gérer et utiliser efficacement ces bases d'images, un système de classification et de recherche d'images est nécessaire. C'est pourquoi le sujet de la recherche d'images devient un sujet très actif dans la communauté internationale depuis plus d'une dizaine d'années.

Les nouveaux supports informatiques permettent de nos jours de manipuler des quantités très importantes d'informations. Parmi ces informations, les images prennent une part de plus en plus grande. De nombreuses applications manipulent de manière quotidienne des images (multimédia, infographie, médicale, etc.). Il apparaît donc aujourd'hui essentiel de concevoir des systèmes de classification automatique, capables de mener à terme la recherche d'information au sein de gigantesques collections d'images numériques. Or, la quantité des données disponibles augmente exponentiellement et leur gestion manuelle devient impossible.

Le but des systèmes de classification automatique est de permettre à un utilisateur de trouver, dans des bases d'images, toutes celles qui sont semblables à une image qui l'intéresse. Un programme de classification se conçoit comme un système qui prend en entrée une image de référence et qui retourne un critère de similarité entre l'image de référence et toutes les images de la base. Ceci permet de les trier de la plus similaire à la moins similaire.

*Chapitre 1*

Reconnaissance

Des

Formes

## I.1. Introduction

La reconnaissance des formes consiste en une automatisation des tâches de perception artificielle réalisées usuellement par le cerveau et le système sensoriel humain. Nous pouvons reconnaître facilement un visage, comprendre les mots et les phrases parlées, l'écriture ainsi que d'autres tâches de reconnaissance.

Depuis l'existence de l'être humain, son système cognitif a évolué de manière à effectuer ces tâches cruciales. Il était donc, nécessaire à consacrer des études pour inventer des machines qui peuvent effectuer le même genre de tâches par exemple la lecture des montants des chèques postaux pour saisir en ordinateur, des adresses des lettres pour leur classification selon la ville, la recherche dans une base énorme d'empreintes digitales pour juger un suspect, la détermination d'identité d'un locuteur à partir de son voix. Pour cela, la reconnaissance automatique des formes revient non pas pour remplacer l'homme par une machine mais pour le soulager des tâches critiques et fastidieuses.

La reconnaissance des formes (RDF) émergea comme une discipline autonome et très prometteuse. Elle fut née au milieu des années cinquante à partir des mathématiques (calcul des probabilités, algèbre linéaire, géométrie intégrale,...) de l'informatique et des sciences pour l'ingénieur. Ultérieurement, son développement a été lié à celui de l'Intelligence artificielle qui a été considéré pendant longtemps comme une composante.

Dans ce chapitre, on va présenter les différents processus de la reconnaissance des formes, après sa définition et objectif. Ensuite, les domaines pouvant utiliser la reconnaissance de formes et.

## I.2. Définition de la RDF

La reconnaissance des formes peut être définie comme l'ensemble des techniques informatiques de représentation et de décision permettant aux machines d'associer une étiquette à une donnée qui peut se présenter sous forme d'une image ou d'un signal. Des données différentes peuvent recevoir la même étiquette, ces données sont les réalisations ou les exemplaires de la classe identifiée par l'étiquette. Elle s'intéresse à la conception et la réalisation de systèmes (matériels ou logiciels) capables de percevoir et d'interpréter des signaux captés dans le monde physique. (Belaid, 1992)

### I.3. L'objectif de la RDF

Le problème que cherche à résoudre la reconnaissance des formes et d'extraire automatiquement des informations des données sensibles pour :

- ✓ Caractériser les classes de formes (apprentissage).
- ✓ Assigner automatiquement des données à ces classes (reconnaissance). (Dubuisson, 1990)

### I.4. Processus de reconnaissance des formes

Le processus classique de la reconnaissance des formes peut être représenté selon le schéma suivant :

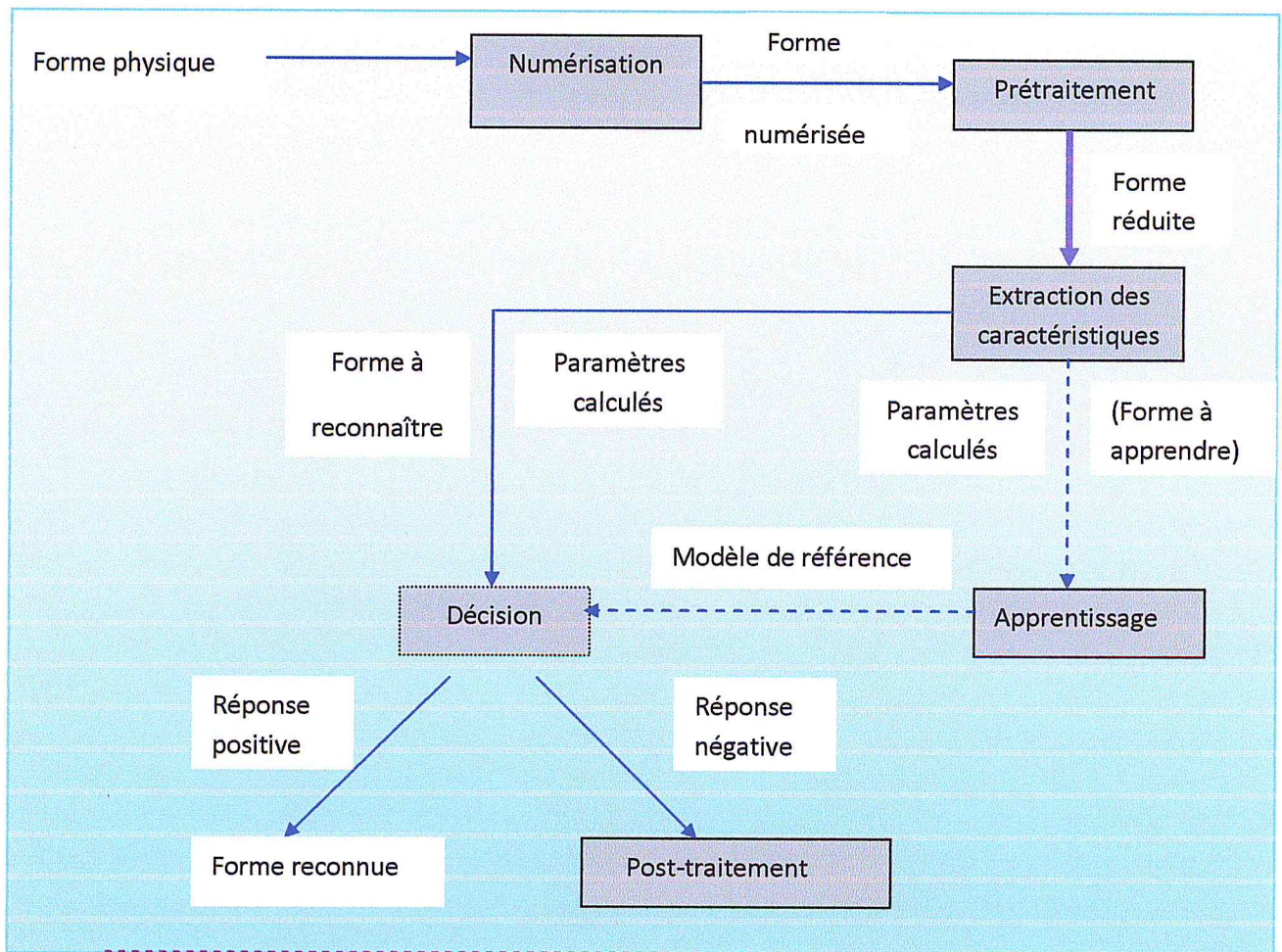


Figure 1.1 : Schéma général d'un système de reconnaissance des formes. (Dubuisson, 1990)



### I.4.1. Codage

Le processus de reconnaissance part du monde physique qui est un espace analogique de dimension infinie appelé espace des formes. Les objets dans cet espace sont décrits de différentes façons avec une multitude de propriétés difficiles à prendre en compte dans leur ensemble lors d'une reconnaissance. (Hong L, Oct 1999)

Le codage ou numérisation est une opération de transformation du monde physique avec des données analogiques vers un monde numérique discret avec des données numériques manipulable en machine. Cet espace est appelle espace de représentation de dimension très importante même si elle est finie. Cette transformation se fasse de la manière la plus fidele possible sans perte d'informations pertinentes et garder les propriétés essentielles de l'objet physique.

Le codage ou la transformation numérique des objets physiques se fait par l'association d'un capteur et récepteur, dont le capteur assure la transformation de la grandeur physique en signal adapté au système de traitement et le récepteur convertit ce signal en données numériques en effectuant un **échantillonnage** et une **quantification**.

- **L'échantillonnage** : procède à un découpage dans le temps du signal continu.
- **La quantification** : consiste à discrétiser les échantillons analogiques en les rapportant à une unité convenable  $v$ .

### I.4.2. Prétraitement

Les prétraitements sont utiles pour la normalisation et l'amélioration des contrastes et aussi pour éliminer des bruits qui peuvent être dus au capteur ou à des interférences avec d'autres sources de signaux (la parole au milieu sonore, l'encre du verso qui traverse le papier et dont la trace est visible sur la feuille du manuscrit ; les fonds images des cheque ...).

### I.4.3. L'extraction des caractéristiques

L'étape d'extraction de caractéristiques appelée aussi étape d'analyse est l'une des étapes les plus importantes et délicates dans la construction d'un système de reconnaissance des formes. Son but est de concentrer l'information présente dans la forme en un ensemble réduit de données, stable et représentatif. Le résultat de cette étape est un ensemble de paramètres caractérisant la forme.

Les caractéristiques extraites, doivent répondre à des critères bien précis, pour garantir l'efficacité de la décision sur la forme à classer. Ainsi, les caractéristiques doivent avoir les propriétés suivantes :

- ✓ Le processus de détection des caractéristiques doit être simple et rapide: les caractéristiques retenues ne doivent pas nécessiter des calculs trop longs et complexes.
- ✓ Le pouvoir discriminant des caractéristiques : les caractéristiques retenues doivent pouvoir décrire de façon non équivoque la totalité des formes appartenant à une classe tout en les différenciant des autres classes.
- ✓ La complémentarité des caractéristiques, ou, au moins la plus faible corrélation possible entre caractéristiques : les familles de caractéristiques retenues doivent mettre en évidence des propriétés différentes de la forme qui soient complémentaires autant que possible.

#### **I.4.4. La reconnaissance**

La reconnaissance regroupe deux tâches importantes dans tous système de reconnaissance des formes se sont : d'**apprentissage** et de **décision**. (Tou & Gonzalez, 1997)

##### **I.4.4.1. L'apprentissage**

L'apprentissage se charge d'acquérir des connaissances sur les formes pour tenter de définir des modèles de référence ou de caractériser des classes de décision.

L'apprentissage dépend de la méthode choisie pour la reconnaissance ainsi que le type du problème à traiter, mais peut être effectué selon deux façons :

➤ **Apprentissage supervisé** : L'apprentissage est dit supervisé si les différentes familles des formes sont connues a priori et si la tâche d'apprentissage est guidée par un superviseur ou professeur. On choisit un sous-ensemble de formes, chacune est analysée puis le professeur indique la classe dans laquelle il souhaite la voir rangée. L'apprentissage consiste alors à analyser les ressemblances entre formes de la même famille et les dissemblances entre formes de familles différentes pour en déduire les classes avec les meilleures séparatrices possibles. Les paramètres décrivant cette partition sont stockés dans une base d'apprentissage.

➤ **Apprentissage non supervisé** : Il s'agit de construire automatiquement les classes, sans intervention de professeur, à partir d'échantillons de référence et de règles de regroupement. Ce mode nécessite un nombre élevé d'échantillons et des règles de construction précises et non contradictoire, mais n'assure pas toujours une classification correspondant à la réalité de l'utilisateur.

En fait le rôle du module d'apprentissage consiste à caractériser chaque classe par exemple par des relations entre les paramètres définissant la forme.

L'apprentissage est un processus complexe et des boucles de contrôle peuvent éventuellement être prévues de manière à valider un ensemble d'échantillons d'apprentissage.

#### **I.4.4.2. Décision**

Son rôle est d'identifier la forme test à partir de l'apprentissage réalisé. On distingue plusieurs approches de décision qui peuvent être répertoriées de différentes manières.

#### **I.4.5. Le post-traitement**

Dans le cas où le résultat obtenu n'est pas satisfaisant on peut effectuer un post-traitement qui consiste à revenir aux étapes qui le précèdent à l'exception de la reconnaissance. Les buts de cette étape sont :

- ✓ Minimiser le taux d'erreur.
- ✓ Minimiser le risque (coût).
- ✓ Augmenter la fiabilité du système de RDF.

### **I.5. Classification**

#### **I.5.1. Définition**

La classification est une discipline reliée de près ou de loin à plusieurs domaines, elle est connue aussi sous noms variés (classification, clustering, segmentation, . . .) selon les objets qu'elle traite et les objectifs qu'elle vise à atteindre.

Pour attribuer une définition au terme « classification », il faudrait d'abord définir ses racines, ça vient du verbe "classer" qui désigne plus une action qu'un domaine, ou plutôt une série de méthodes qu'une théorie unifiée.

En mathématique, On appelle classification, la catégorisation algorithmique d'objets. Elle consiste à attribuer une classe ou catégorie à chaque objet (ou individu) à classer, en se basant sur des données statistiques. Elle fait couramment appel aux méthodes d'apprentissage et largement utilisée en reconnaissance de formes.

Il est important de noter qu'il ne faut pas confondre entre ces deux termes : « Classification » et « classement », au fait le mot classification en anglais signifie une chose, alors que le même mot en français ait une autre signification (utilité).

Dans un classement, on affecte les objets à des groupes préétablis, c'est le but de l'analyse discriminante que de fixer des règles pour déterminer la classe des objets. La classification est donc, en quelque sorte, le travail préliminaire au classement, savoir la recherche des classes "naturelles" dans le domaine étudié, en anglais « Cluster Analysis ».

### 1.5.2. Les étapes d'une classification

On peut résumer la classification en quatre étapes :

1. Choix des données.
2. Calcul des similarités entre les n individus à partir des données initiales.
3. Choix d'un algorithme de classification et exécution.
4. L'interprétation des résultats :
  - ✓ évaluation de la qualité de la classification.
  - ✓ description des classes obtenues.

### 1.5.3. Les méthodes de Classifications

On peut grouper les méthodes classificatoires en deux grandes familles, cette fois-ci, on prend en considération l'intervention ou non d'un « attribut classe » au fur et à mesure du processus de la classification, ces deux types sont :

- ✓ Classification non supervisée (Classification, Clustering) appelée aussi classification sans apprentissage qui vise à séparer automatiquement l'image en clusters naturels, c'est -à-dire sans aucune connaissance préalable des classes.
- ✓ la classification supervisée (Classement), qui s'opère à partir de la connaissance de chacune des classes définies.

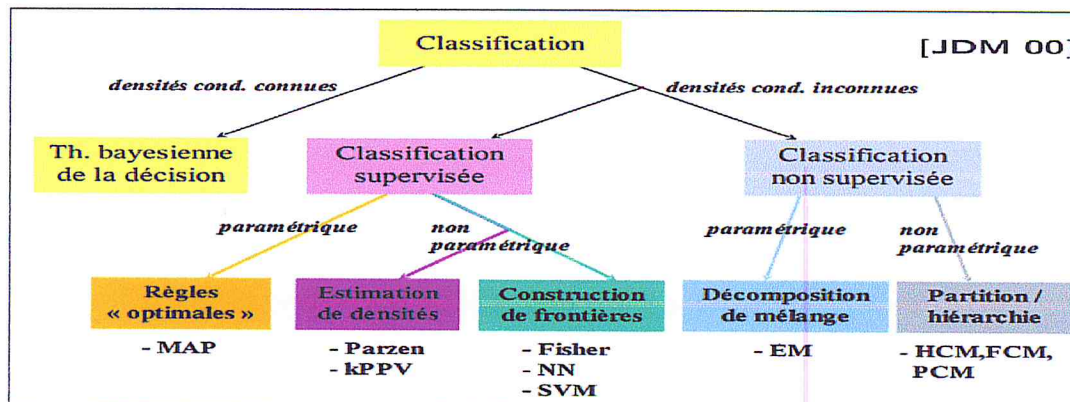


Figure 1.2 : Les méthodes de classification. (Berrani & all, 2002)

#### I.5.4. Les approches de classifications

On distingue d'une façon générale deux grands types de approches:

##### i. Approche non paramétrique :

Les approches dites non paramétriques (classification hiérarchique, méthode des centres mobiles) basée sur l'hypothèse : plus deux individus sont proches, plus ils ont de chances de faire partie de la même classe, en plus ce que distingue cette approche est qu'on ne fait aucune hypothèses sur le modèle que suivent les données, C'est le cas des plus proches voisins (k-PPV), donc il suffit de trouver les propriétés de convergence quand le nombre de données est grand.

##### ii. Approche paramétrique : « Probabilistes »

La seconde grande famille des méthodes de classification, ce sont les approches probabilistes, utilisent une hypothèse sur la distribution des individus à classier, c'est-à-dire, on suppose que l'on connaît la forme du modèle qui a généré les données.

Par exemple, on peut considérer que les individus de chacune des classes suivent une loi normale. Le problème qui se pose, est de savoir déterminer ou estimer les paramètres des lois (moyenne, variance) et à quelle classe les individus ont le plus de chances d'appartenir à partir de l'ensemble d'apprentissage. Les paramètres d'une loi peuvent être déterminés de

maintes façons, C'est le cas par exemple des classifications bayésiennes ou encore l'algorithme espérance maximisation. (Berrani & all, 2002)

Parmi les classificateurs de l'approche paramétrique on trouve : le classificateur euclidien, quadratique, gaussien. On va présenter le classificateur euclidien.

### 1.5.5. La classification supervisée

L'objectif de la classification supervisée est d'apprendre, à l'aide d'un ensemble d'entraînement, une procédure de classification qui permet de prédire l'appartenance d'un nouvel exemple à une classe. En d'autre terme, l'objectif est d'identifier les classes auxquelles appartiennent des objets à partir de leurs variables descriptifs. (Chu, 2005)

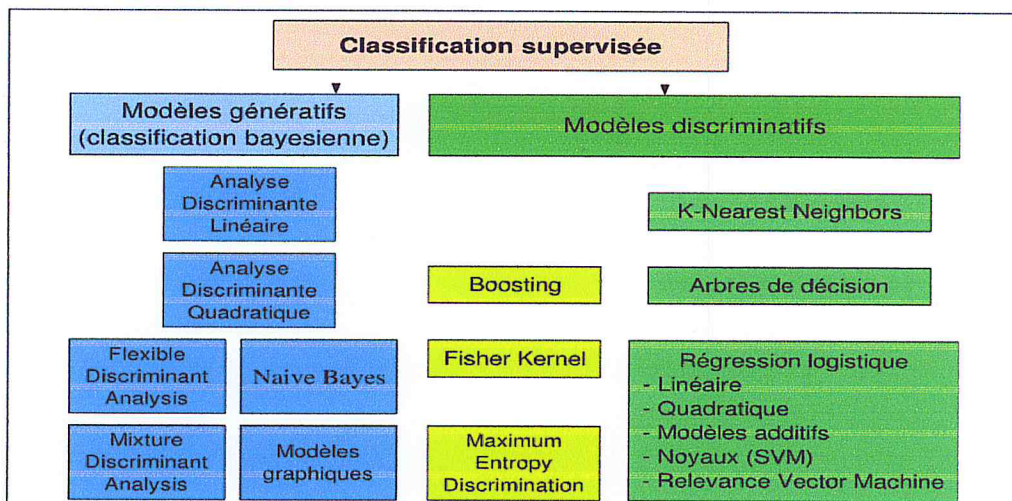


Figure 1.3: Les différents modèles de classification supervisée. (Chu, 2005)

#### 1.5.5.1. Méthode KPPV

##### i. Les k plus proches voisins Kppv

Plus connus en anglais sous le nom K-nearest neighbor (K-NN), ou encore Memory Based Reasoning. Cette méthode diffère des traditionnelles méthodes d'apprentissage car aucun modèle n'est induit à partir des exemples. Les données restent telles quelles : elles sont simplement stockées en mémoire.

Pour prédire la classe d'un nouveau cas (où ranger un nouveau document par exemple), l'algorithme cherche les K plus proches voisins de ce nouveau cas et prédit (s'il faut choisir) la réponse la plus fréquente de ces K plus proches voisins.

La méthode utilise donc deux paramètres : le nombre K et la fonction de similarité pour comparer le nouveau cas aux cas déjà classés.

$$d(x_i, x_j) \equiv \sqrt{\sum_{r=1}^n (a_r(x_i) - a_r(x_j))^2} \quad \dots\dots (4)$$

Cette méthode supervisée et non-paramétrique est souvent performante. De plus, son apprentissage est assez simple, car il est de type apprentissage par cœur (on garde tous les exemples d'apprentissage). Cependant, le temps de prédiction est très long, car il nécessite le calcul de la distance avec tous les exemples, mais il existe des heuristiques pour réduire le nombre d'exemples à prendre en compte. (Berrani & all, 2002)

## ii. Algorithme de KPPV

On résume l'algorithme de kppv en deux étapes essentielles :

1. initialisation, choix de :

✓ Nombre de classes, Valeur de k, exemples initiaux, mesure de similarité.

2. pour chaque vecteur d'objet à classer :

✓ mesurer la distance du vecteur avec tous les autres déjà classés.

✓ déterminer la liste des k vecteurs les plus proches de lui (k-ppv).

✓ déterminer la classe la plus représentée dans la liste des k-ppv et affecter notre vecteur à cette classe.

## iii. Sélection de la classe

La méthode la plus simple est de rechercher le cas le plus proche et de prendre la même décision. C'est la méthode 1-PPV (1-NN) du plus proche voisin. Si cette méthode peut fournir de bons résultats sur des problèmes simples pour lesquels les points sont bien répartis en groupes denses de même classe, en règle générale, il faut considérer un nombre de voisins plus important pour obtenir de bons résultats. (WIKIPEDIA, 2014)

Une première façon de combiner les  $k$  classes des  $k$  voisins les plus proches est le vote majoritaire. Elle consiste simplement à prendre la classe majoritaire. Une seconde façon est le vote majoritaire pondéré. Chaque vote, c'est-à-dire chaque classe d'un des  $k$  voisins sélectionnés, est pondéré. Le poids est inversement proportionnel à la distance entre le cas à classer et les autres  $k$  plus proches voisin classes (Une telle pondération s'appelle un noyau).

Dans les deux cas précédents, il est possible de définir une confiance dans la classe attribuée égale au rapport entre les votes gagnants et le total des votes.

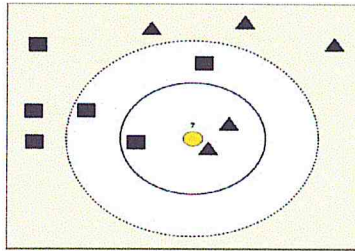


Figure 1.4 : Illustration des  $k$ -ppv d'un point.

Ce qu'on peut remarquer sur cette méthode, c'est le cout de calcul qu'elle impose au fur et à mesure de ce processus de classification, car ce cout augmente avec chaque vecteur qu'on vient de classifié, plus on ajoute des nouveaux vecteurs déjà classés, plus que ce cout augmente ce qui explique le temps d'exécution qu'elle prend pour classifier.

En plus de la sensibilité de cet algorithme à l'initialisation des paramètres d'entrées (le choix de  $k$ , la distance utilisée ..) alors il faut que lors de la sélection des paramètres d'entrées que ces derniers respectent certaines contraintes (comme que  $k$  ne soit pas un multiple du nombre de classes pour éviter une surreprésentation d'une classe par rapport à une autre malgré ces points,  $k$ -ppv reste une des méthodes les plus utilisées grâce à sa simplicité et robustesse et son caractère de généralisation à partir d'un nombre éminent de données d'apprentissage.

Ainsi certains travaux ont utilisé les paramètres du classificateur euclidien pour diminuer la contrainte du temps surtout lorsque le nombre d'apprentissage sera trop élevée, au lieu de calculer la distance entre le vecteur de la forme à classer avec les vecteurs d'apprentissage la distance sera calculée que par rapport à la moyenne de chaque classe de données d'apprentissage. (Berrani & all, 2002)



➤ Distance du  $\chi^2$  (chi-2)

Mesurons à présent quelque chose de très différent. Il ne s'agit plus de distances entre des points et une droite ou un barycentre mais entre des valeurs observées et des valeurs théoriques.

Cette différence entre observé et théorique est élevée au carré puis rapportée à cette même valeur théorique.

Supposons qu'on compare des effectifs observés  $O$  aux effectifs  $T$  qu'on observerait si la population épousait parfaitement une fonction de densité donnée, par exemple le poids des habitants d'un village par tranches de 5 kg. La distance retenue, dite du  $\chi^2$ , sera la suivante ( $i$  étant chaque tranche de poids).

$$d^2 = \sum_{i=1}^n \frac{(O_i - T_i)^2}{T_i} \dots\dots(5)$$

iv. Les k plus proches voisins flouent

Le principe de la méthode KPPV floue est similaire à la règle des KPPV classiques. Il se base sur une étape de recherche des k points voisins les plus proches du prototype à étudier. On examine parmi les K voisins retournés, le taux de mélange des classes. Le KPPV flou permet en plus, d'attribuer suivants les distances aux classes des k prototypes voisins, des degrés d'appartenance à ces classes.

Le point à classer  $x$  se voit attribuer un coefficient d'appartenance  $\mu_{ji}$  à chaque classe  $j$ . Celui-ci est fonction des distances et des coefficients d'appartenance de ses k plus proches voisins. Ces coefficients doivent vérifier l'appartenance à l'intervalle  $[0,1]$  de  $\mu_{ji}$ , pour tous les  $i$  et tous les  $j$ .

$$\sum_{j=1}^c \mu_{ji} = 1 \quad \text{pour tous les } i, \quad 0 < \sum_{i=1}^N \mu_{ji} \leq N \quad \text{pour tous les } j \dots\dots(6)$$

Les coefficients d'appartenance d'un nouveau point  $X_i$  à la classe  $j$  est donnée par :

$$u_{ji} = \frac{\sum_{l=1}^k \mu_{jl} (|x_l - x_i|)^{\frac{m-1}{2}}}{\sum_{l=1}^k \left( \frac{1}{(|x_l - x_i|)^{\frac{m-1}{2}}} \right)} \dots\dots(7)$$

Où  $\mu_{ji}$  est le coefficient d'appartenance à la classe  $C_j$ ,  $\mu_{ji} = 1$  si  $x_i$  appartient à la classe  $C_j$ ,  $\mu_{ji} = 0$  si non, de lième observation, parmi les  $k$  plus proches voisins de  $x_i$ .

La variable  $m$  quant à elle, détermine l'importance de la contribution de la distance dans le calcul de la fonction d'appartenance (contrôle l'efficacité de l'ampleur de la distance). C'est le paramètre de fuzzification : si  $m$  croît, la contribution des voisins est davantage pondérée et la notion de distance perd de son importance, si  $m$  tend vers l'unité, la contribution des voisins les plus proches sera favorisée, ainsi la notion de distance prend de l'importance, si  $m$  vaut 2, la contribution de chaque voisin est pondérée par l'inverse de la distance respective, au carré, qui sépare une observation de l'observation à classer. (Keller & all, 1999)

### I.5.5.2. Méthode SVM (Support Vector Machines)

#### i. Définition

Les Support Vector Machines souvent traduit par l'appellation de Séparateur à Vaste Marge (SVM) sont une classe d'algorithmes d'apprentissage initialement définis pour la discrimination c'est-à-dire la prévision d'une variable qualitative binaire. Ils ont été ensuite généralisés à la prévision d'une variable quantitative. Dans le cas de la discrimination d'une variable dichotomique, ils sont basés sur la recherche de l'hyperplan de marge optimale qui, lorsque c'est possible, classe ou sépare correctement les données tout en étant le plus éloigné possible de toutes les observations. Le principe est donc de trouver un classifieur, ou une fonction de discrimination, dont la capacité de généralisation (qualité de prévision) est la plus grande possible. (Vladimir, 1992)

#### ii. Les SVMs linéaires

##### a. Classifieur linéaire

Un Classifieur est dit linéaire lorsqu'il est possible d'exprimer sa fonction de décision par une fonction linéaire en. On peut exprimer une telle fonction par:

$$h(x) = \langle w, x \rangle + b = \sum_{i=1}^n w_i x_i + b \quad \dots\dots\dots(8)$$

Où  $(w \in \mathbb{R}^n)$  est le vecteur de poids et  $(b \in \mathbb{R}^0)$  le biais, alors que  $x$  est la variable du problème.  $X$  est l'espace d'entrée et qui correspond à  $\mathbb{R}^n$ , où  $n$  est le nombre de composantes des vecteurs contenant les données. Notons que l'opérateur  $\langle \rangle$  désigne le produit scalaire usuel dans  $\mathbb{R}^n$  sont  $w$  et  $b$  les paramètres à estimer de la fonction de décision  $h(x)$ .

Pour décider à quelle catégorie un exemple estimé  $x'$  appartient, il suffit de prendre le signe de la fonction de décision :  $y = \text{sign}(h(x'))$ . La fonction  $\text{sign}(\ )$  est appelée classifieur. Géométriquement (voir figure 2.18), cela revient à considérer un hyperplan qui est le lieu des points  $x$  satisfaisant  $\langle w, x \rangle + b = 0$ . En orientant l'hyperplan, la règle de décision correspond à

observer de quel côté de l'hyperplan se trouve l'exemple  $x'$ . On voit que le vecteur  $w$  définit la pente de l'hyperplan ( $w$  est perpendiculaire à l'hyperplan). Le terme  $b$  quant à lui permet de translater l'hyperplan parallèlement à lui-même.

L'objectif de la discrimination linéaire est de trouver la bonne fonction de décision  $h(x)$ . La classe de tous les hyperplans qui en découle sera notée  $H$ .

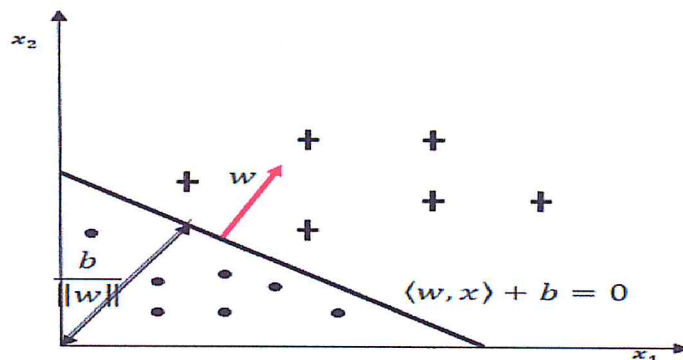


Figure 1.5 : Hyperplan séparateur  $\langle w, x \rangle + b = 0$ .

➤ **Algorithme SVM linéaire**

- ✓ Trouver un hyperplan.
- ✓  $(w, b) \in \mathbb{R}^{d+1}$  qui classe aussi bien que possible les données (points).
- ✓ Progressivement : un point à la fois, en modifiant les poids si nécessaire.

**b. Marge maximale de l'hyperplan**

La marge est la distance entre la frontière de séparation et les échantillons les plus proches. Dans les SVM, la frontière de séparation est choisie comme celle qui maximise la marge.

La marge géométrique représente la distance euclidienne prise perpendiculairement entre l'hyperplan et l'exemple  $x_i$ . En prenant un point quelconque  $x_p$  se trouvant sur l'hyperplan, la marge géométrique peut s'exprimer par :

$$\frac{w}{\|w\|} \cdot (x_i - x_p)$$

L'hyperplan à marge maximale est le modèle le plus utilisé dans les machines à vecteurs supports. L'estimation des paramètres  $(w^*, b^*)$  de l'hyperplan qui maximise la marge se fait en résolvant le problème d'optimisation suivant:

$$(w^*, b^*) = \operatorname{argmax}_{(w,b)} \{ \min_i (y_i (wx_i + b)), \|w\| = 1 \} \dots\dots(9)$$

Dire que les deux classes de l'échantillon d'apprentissage  $S$  sont linéairement séparables est équivalent à dire qu'il existe des paramètres  $(w^*, b^*)$  tels que l'on a pour tout  $i(=1,2,\dots,n)$ :

$$\begin{aligned} w^*x_i + b^* &> 0 \text{ si } y_i = 1 \\ w^*x_i + b^* &< 0 \text{ si } y_i = -1 \end{aligned} \dots\dots(10)$$

ce qui est équivalent à :

$$y_i(w^*x_i + b^*) > 0 ; \quad \forall i = 1, 2, \dots, n \quad \dots\dots(11)$$

La définition consiste à dire qu'il doit exister un hyperplan laissant d'un côté toutes les données positives et de l'autre, toutes les données négatives. Dès lors, on peut définir deux plans se trouvant de part et d'autre de l'hyperplan et parallèles à celui-ci, sur lesquels reposent les exemples les plus proches. La figure 2.19 illustre cette situation.

Dans notre définition de l'hyperplan, il est possible que différentes équations correspondent au même plan géométrique :

$$a(\langle w, x \rangle + b) = 0 \quad \dots\dots(12)$$

$a$  : est une constante quelconque.

Il est donc possible de redimensionner  $(w^*, b^*)$  de telle sorte que les deux plans parallèles aient respectivement pour équations :

$$\begin{aligned} (w^*x_i + b^*) &= 1 \\ (w^*x_i + b^*) &= -1 \end{aligned} \dots\dots(13)$$

Ces deux hyperplans sont appelés hyperplans canoniques. Ainsi la marge  $\gamma$  entre ces deux plans est égale à :

$$\gamma = \frac{2}{\|w^*\|} \quad \dots\dots(14)$$

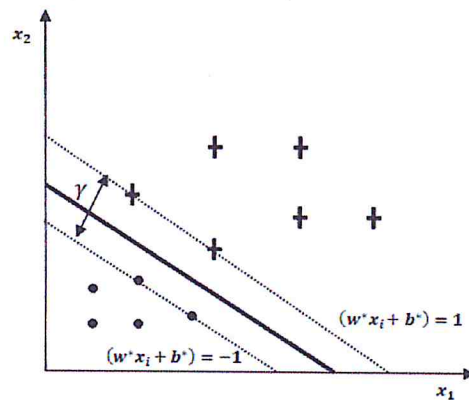


Figure 1.6 : Hyperplans canoniques et marge maximale.

### iii. SVM non-linéaires

Le paragraphe précédent décrit le principe des SVM dans le cas où les données sont linéairement séparables. Cependant, dans la plupart des problèmes réels, ce n'est pas toujours le cas et il est donc nécessaire de contourner ce problème (difficile de séparer n'importe quel jeu de données par un simple hyperplan). Si par exemple les données des deux classes se chevauchent sévèrement, aucun hyperplan séparateur ne sera satisfaisant.

L'idée est de projeter les points d'apprentissage  $x_i$  dans un espace  $T$  de dimension  $q$ , plus élevée que  $n$  grâce à une fonction non-linéaire  $\phi$  qu'on appelle fonction noyau, choisie a priori et d'appliquer la même méthode d'optimisation de la marge dans l'espace  $T$ . L'espace  $T$  ainsi obtenu est appelé espace des caractéristiques ou aussi espace transformé.

Tout ce qu'il nous reste à faire c'est de résoudre le problème (2.18) dans l'espace  $T$ , en remplaçant  $\langle x_i, x_j \rangle$  par  $\langle \phi(x_i), \phi(x_j) \rangle$ . L'hyperplan séparateur obtenu dans l'espace  $T$  est appelé hyperplan optimal généralisé. Le produit scalaire peut se calculer facilement à l'aide d'une fonction symétrique, dite noyau, définie par :

$$K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle \quad \dots\dots(15)$$

Le noyau  $K(x,y)$  est une fonction de deux variables, symétrique et positive. Dans ce cas, la frontière de décision devient :

$$h(x) = \text{sign} \left( \sum_{i=1}^n y_i \alpha_i^* K(x_i, x_j) + b^* \right) \quad \dots\dots(16)$$

Dans la pratique on choisit un noyau  $K$  qui satisfait les conditions de Mercer afin de garantir la décomposition (2.24). Une famille de ces fonctions noyaux qui sont très appropriées aux besoins des SVM peut être définie, en voici les plus utilisés:

**a. Noyau polynomial d'ordre  $p$ :**

$$K(x_i, x_j) = (\langle x_i, x_j \rangle + 1)^d \dots\dots(17)$$

La dimension de l'espace transformé induit par un noyau polynomial est de l'ordre  $\frac{(p+d)!}{p!d!}$ , où  $p$  est la dimension de l'espace de départ.

**b. Noyau linéaire**

$$K(x_i, x_j) = x_i \cdot x_j \dots\dots(18)$$

**c. Noyau gaussien de largeur de bande  $\sigma$**

$$K(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma}\right)$$

Le paramètre  $\sigma$  permet de régler la largeur de la gaussienne. En prenant un  $\sigma$  grand, la similarité d'un exemple par rapport à ceux qui l'entourent sera assez élevée, alors qu'on prenant un  $\sigma$  tendant vers 0, l'exemple ne sera similaire à aucun autre.

**iv. Domaines d'application de SVM**

Vu leur composition comme des techniques d'apprentissage, les SVMs sont utilisés dans les domaines de :

- ✓ Reconnaissance de formes/Classification.
- ✓ Vision Machine: Identification de visage, reconnaissance d'expression faciale : Surpasse les approches alternatives (1.5% taux d'erreur).
- ✓ Reconnaissance des chiffres manuscrits: les résultats d'USPS (service de la poste des états unis) data base comparable à la meilleure approche (1.1% taux d'erreur).
- ✓ Reconnaissance de visage.
- ✓ Images satellite et surveillance.
- ✓ Diagnostic médical (cancer du sein)

**v. Avantages et inconvénients****a. Avantages**

- ✓ Les SVM possèdent des fondements mathématiques solides.
- ✓ Les exemples de test sont comparés juste avec les supports vecteur et non pas avec tous les exemples d'apprentissage.
- ✓ Décision rapide. La classification d'un nouvel exemple consiste à voir le signe de la fonction de décision  $f(x)$ .

**b. Inconvénients**

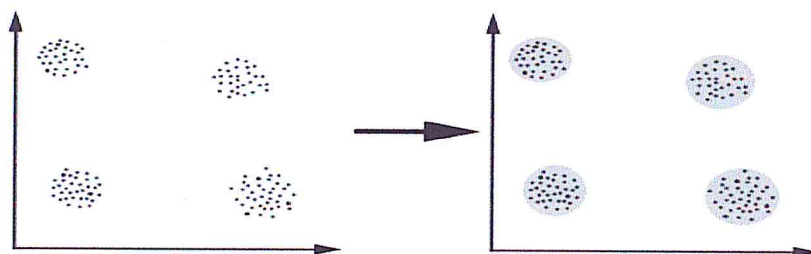
- ✓ Classification binaire d'où la nécessité d'utiliser l'approche un-contre-un.
- ✓ Grande quantité d'exemples en entrées implique un calcul matriciel important.
- ✓ Temps de calcul élevé lors d'une régularisation des paramètres de la fonction noyau.

**I.5.6. La classification Non-supervisée**

Le Clustering aussi connu sous nom (Segmentation) est un regroupement en classes homogènes consistant à représenter un nuage des points d'un espace quelconque en un ensemble de groupes appelé Cluster.

C'est un traitement sur un ensemble d'objets qui n'ont pas été étiquetés par un superviseur. Généralement lié au domaine de l'analyse des données comme ACP (analyse linéaire en composantes principales) (Lebart & all, 2004), ce type de méthodes vise à répondre au problème de : diminution de la dimension de l'espace d'entrée, ou pour le groupement des objets en plusieurs catégories (clusters) non définies à l'avance. Parmi les méthodes qu'on peut trouver dans ce type de classification : les cartes auto-organisatrices de kohonen, GMM . . . etc.

Un «Cluster» est donc une collection d'objets qui sont «similaires» entre eux et qui sont «dissemblables » par rapport aux objets appartenant à d'autres groupes. On peut voir cette définition clairement graphiquement dans l'exemple suivant :



**Figure 1.7 :** Illustration de regroupement en clusters.

Contrairement à la classification (méthodes supervisées), on ne possède pas des connaissances a priori sur les classes prédéfinies des éléments. Donc La division des objets dans les différents groupes (clusters) se procède en se basant sur le calcul de similarité entre les éléments.

Alors que l'objectif des méthodes du Clustering est de grouper des éléments proches dans un même groupe de manière à ce que deux données d'un même groupe soient le plus similaires possible et que deux éléments de deux groupes différents soient le plus dissemblables possible.

Mathématiquement, on a un ensemble  $X$  de  $N$  données décrites chacune par leurs  $P$  attributs. Donc Le Clustering consiste à créer une partition ou une décomposition de cet ensemble en sous parties (clusters) telle que :

- ✓ Les données appartenant au même groupe se ressemblent.
- ✓ Les données appartenant à deux groupes différents soient peu ressemblantes.

On utilise souvent ce type de classification en traitement d'images pour fixer les divers objets qu'elles contiennent (segmentation) : routes, villes, rues, des organes humaines (pour les images médicales).

Les principales exigences qu'un algorithme de clustering doit répondre sont les suivantes :

- ✓ Evolutivité des clusters.
- ✓ traiter les différents types d'attributs.
- ✓ découvrir les clusters de forme arbitraire.
- ✓ exigences minimales pour la connaissance du domaine afin de déterminer les paramètres d'entrée.
- ✓ capacité de composer avec le bruit et les valeurs manquantes traitées les dimensionnalités élevées.
- ✓ l'intelligibilité et la convivialité.



### I.5.7. Les types du clustering

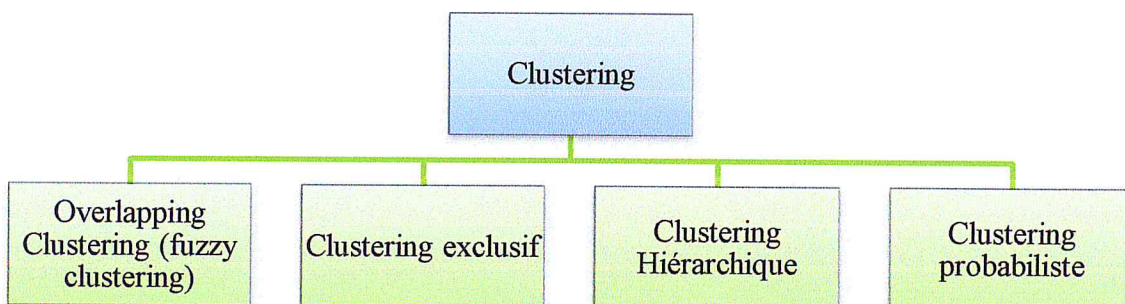


Figure 1.8 : Les différents types de clustering. (David, 2002)

#### i. Clustering exclusif

Les données sont regroupées d'une manière exclusive, de sorte que si une donnée certaine appartient à un amas définie alors il ne pourrait pas être inclus dans un autre cluster. Un simple exemple de cela est montré dans la figure ci-dessous, où la séparation des points est définie par une ligne droite sur un plan bidimensionnel.

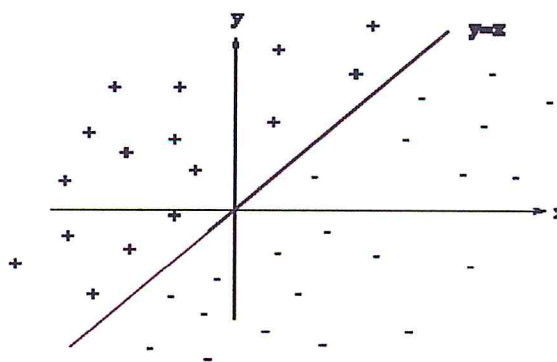


Figure 1.9 : Exemple d'un problème de discrimination à deux classes, avec un séparateur linéaire : la droite d'équation  $y=x$ . Le problème est linéairement séparable.

**ii. Le regroupement Overlapping Clustering (fuzzy clustering)**

Utilise des ensembles flous aux données de cluster, de sorte que chaque point peut appartenir à deux ou plusieurs groupes avec différents degrés d'appartenance. Dans ce cas, les données seront associées à une valeur d'une composition appropriée.

**iii. Clustering hiérarchique**

Un algorithme de clustering hiérarchique consiste à trouver des clusters successifs utilisant des clusters précédemment établis. La première condition est de mettre, au début, chaque objet dans un cluster distinct et les fusionner en clusters successivement plus grand. Après quelques itérations on atteint le final Cluster voulu qui regroupe tous les sous-clusters (sous-partitions).

**iv. Clustering probabiliste**

Utilise une approche complètement probabiliste basant sur la probabilité d'appartenance aux clusters.

**I.5.8. Les algorithmes de Clustering**

Parmi les algorithmes appartient à l'un des types de clustering, K-means est un algorithme de clustering exclusif, pendant que Fuzzy C-means est un algorithme de Overlapping Clustering, alors que clustering hiérarchique il est claire qu'il s'agit de troisième type de clustering, et enfin Mélange de Gaussien est un algorithme de clustering probabiliste.

**i. K-means**

Le principe de « K-means » est le suivant : on part de K données synthétiques (c'est -à-dire des points de l'espace de données D ne faisant pas forcément parti du jeu de données) que l'on nomme des « centres ». Chaque centre caractérise un groupe. A chaque centre sont associées les données qui lui sont les plus proches ; cela crée un groupe autour de chaque centre.

Ensuite, on calcule le centre de gravité de chacun de ces groupes ; ces k centres de gravité deviennent les nouveaux centres et on recommence tant que les groupes ne sont pas stabilisés, i.e. tant qu'il y a des données qui changent de groupe d'une itération à la suivante ou encore, tant que l'inertie varie substantiellement d'une itération à la suivante.

Cet algorithme converge en un nombre fini d'itérations (Jean, 2010). Il existe des versions améliorées des k-moyennes.

La Figure 2.23 illustre le déroulement étape par étape d'un tel algorithme. Lors de la phase d'initialisation, un pixel est sélectionné de façon aléatoire parmi l'ensemble des pixels de l'image afin de représenter son centre de gravité initial et ce pour chaque classe (Figure 2.23 (a)). L'algorithme affecte ensuite à chaque pixel de l'image la classe  $C_i$  dont le centre de gravité est le plus proche (Figure 2.4 (b)).

A chaque fois qu'un pixel de l'image est affecté à sa classe respective, le centre de gravité de la classe est recalculé (Figure 2.23 (c)). Ce processus est réitéré jusqu'à convergence de l'algorithme c'est à dire jusqu'à l'obtention d'un minimum local de la somme des distances entre les points de chaque classe et son centre de gravité (Figure 2.21 (d)).

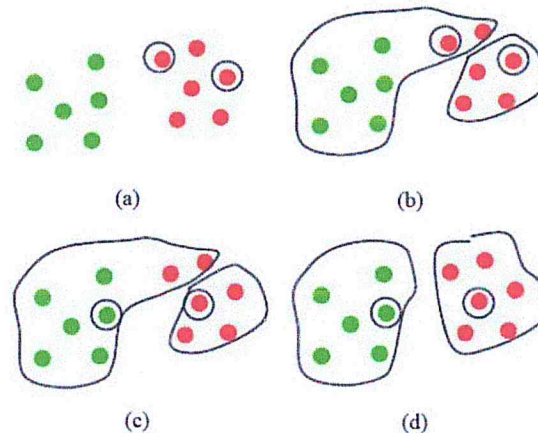


Figure 1.10: K- means. (Jean, 2010)

## ii. Fuzzy C-means

Fuzzy C-means (FCM) est une méthode de clustering qui permet à un objet de données d'appartenir à deux ou plusieurs clusters. Cette méthode dérivée de l'algorithme c-means, identique à l'algorithme k-means décrit précédemment, elle a été développée par Dunn en 1973 et améliorée par Bezdek en 1981, est fréquemment utilisée dans la reconnaissance des formes. Il est basé sur la minimisation de la fonction objective suivante :

$$J_m = \sum_{i=1}^N \sum_{j=1}^C u_{ij}^m \|x_i - c_j\|^2 \quad 1 \leq m < \infty \quad \dots\dots(11)$$

Où  $m$  est un nombre réel ( $> 1$ ),  $U_{ij}$  est le degré d'appartenance de  $X_i$  dans le  $j$  ème Cluster,  $X_i$  est le  $i$ ème élément des données mesurées,  $c_j$  est le centre d'un cluster et  $\|*\|$  est toute norme exprimant la similarité entre les données mesurées et le centre. Ce Partitionnement logique flou (fuzzy) est réalisé grâce à une optimisation itérative de la fonction objectif indiqué ci-dessus, avec la mise à jour de l'appartenance  $u_{ij}$  et les centres des clusters  $C_j$ .

La FCM exige des paramètres d'entrées, et que la matrice de partition floue, doit être initialisée d'une manière appropriée. Ces paramètres sont choisis d'une façon arbitraire, ces paramètres ont une grande influence sur le résultat attendu. Ce qu'il nous oblige de faire une étude approprié sur les données en entrée et le regroupement que l'on souhaite obtenir.

### I.5.9. Les méthodes hiérarchiques

La classification hiérarchique, consiste à effectuer une suite de regroupements en classes de moins en moins fines en agrégeant à chaque étape les objets ou les groupes d'objets les plus proches. Le nombre d'objets n'est pas fixé a priori mais, sera fixé a posteriori. Elle fournit ainsi un ensemble de partitions de l'ensemble d'objets, Il existe deux types de méthodes :

#### ➤ Classification hiérarchique ascendantes (CHA)

Ces méthodes sont les plus anciennes et les plus utilisées dans la classification automatique. Supposons que nous avons  $N$  objets à classer. Les algorithmes agglomératifs suivant cette approche, définissent d'abord une partition initiale en  $N$  classes unitaires. Par la suite, ils fusionnent successivement les classes jusqu'à ce que toutes les entités soient dans la même classe. Dans chaque étape de fusion des classes, le recalcule des dissimilarités entre les nouvelles classes est nécessaire. Le choix des classes se fait selon le critère qui caractérise la méthode. Les méthodes de cette catégorie diffèrent selon le critère local choisi et selon la méthode de calcul des dissimilarités interclasses.

Ce principe est de Fournir un ensemble de partitions de moins en moins fines obtenues par regroupement successifs de parties. Une fois cet algorithme terminé, on ne récupère donc pas Une fois cet algorithme terminé, on ne récupère donc pas directement une partition, mais une hiérarchie de partitions en  $n, \dots, 1$  classes, avec diminution de l'inertie inter-classes à chaque agrégation. On résume ce principe avec ces étapes suivantes :

- ✓ Initialisation : les classes initiales sont les  $n$  singletons-individus. Calculer la matrice de leurs distances deux à deux.
- ✓ Itérer les deux étapes suivantes jusqu'à l'agrégation en une seule classe.
- ✓ Regrouper les deux éléments (classes) les plus proches au sens de la distance entre groupes choisie.
- ✓ Mettre à jour le tableau de distances en remplaçant les deux classes regroupées par la nouvelle et en calculant sa distance avec chacune des autres classes.
- ✓ Nécessité de définir une distance entre groupes d'individus (appelé stratégie d'agrégation).
- ✓ Nécessite de choisir le nombre de classes à retenir.

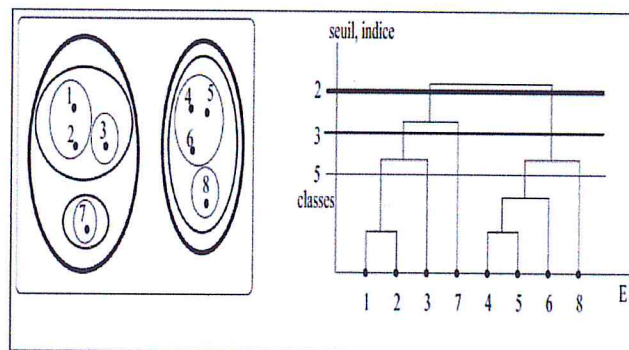


Figure 1.11: Classification hiérarchique descendante. (Carl, 2004)

### ➤ Classification hiérarchique descendantes

Dans le paragraphe précédent, nous avons vu que la classification hiérarchique ascendante se base sur un seul critère à la fois. Ceci engendre uniquement une séparation (méthode du lien simple) ou une homogénéité (méthode du lien complet) optimale des classes. Ce qui risque de donner naissance à l'effet de chaînage (deux entités très dissimilaires appartenant aux points extrêmes d'une longue chaîne, peuvent appartenir à la même classe) ou l'effet de dissection (deux entités très similaires peuvent être dans deux classes différentes). Pour faire face à ces deux problèmes, nous retrouvons les algorithmes divisifs de la classification hiérarchique descendante.

Ces algorithmes commencent par former une seule classe qui englobe tous les objets. Par la suite, ils choisissent une classe de la partition en cours selon un premier critère local. Ils procèdent ensuite à une bipartition successive selon un deuxième critère local des classes

choisies. Cette bipartition continue jusqu'à ce que toutes les entités soient affectées à différentes classes. (Chu, 2005)

#### I.5.10. Les limites de Clustering

Il y a un certain nombre de problèmes avec le clustering. Parmi eux :

- ✓ les techniques de clustering actuelles ne traitent pas tous les besoins de façon adéquate (et simultanément), comme le fait que si nous n'avons pas des variables continuées (la longueur), mais les catégories nominales, comme les jours de la semaine. Dans ces cas encore, la connaissance du domaine doit être faite pour formuler le clustering appropriée.
- ✓ Traitement d'un grand nombre de dimensions et de grand nombre de données, question peut être problématique en raison de la complexité du temps de calcul.
- ✓ L'efficacité de la méthode dépend de la définition de «distance» utilisée.
- ✓ Si la mesure de la distance n'existe pas, nous devons la «définir», ce qui n'est pas toujours facile, surtout dans des espaces multidimensionnels.
- ✓ Le résultat de l'algorithme de clustering peut être interprété de différentes manières.
- ✓ Beaucoup d'algorithmes de clustering exigent la spécification du nombre de clusters à produire en entrée de l'ensemble de données, avant l'exécution de l'algorithme. i.e: connaissance de la valeur correcte à l'avance, la valeur appropriée doit être déterminée.

#### I.6. Conclusion

Dans ce chapitre on a défini la reconnaissance des formes et ces objectifs, présenté le schéma général du processus de la reconnaissance des formes et cité ces différentes techniques et ces étapes.

Dans le prochain chapitre, nous allons aborder les différentes techniques de traitement d'images utilisées dans le domaine de vision par ordinateur.

# *Chapitre 2*

## Traitement d'Images

## II.1. Introduction

Le traitement d'images et ses techniques (étapes) associées sont destinés à exploiter au mieux l'information contenue dans l'image, qui était, depuis l'aube de l'humanité, un élément le plus significatif dans le domaine de l'échange de l'information, et elle l'est toujours d'ailleurs.

Le traitement d'images peut être défini comme l'ensemble des méthodes et techniques opérant sur l'image afin d'extraire les informations les plus pertinentes ou tout simplement pour fournir une image plus perceptible à l'œil humain.

Dans ce chapitre on va traiter les différentes approches de prétraitement et de segmentation d'images ainsi la classification.

## II.2. La chaîne de traitement d'image

Une chaîne de traitement commence par l'acquisition de données physiques sous la forme d'une image. Cette image peut être:

- ✓ Soit numérique obtenue par des systèmes d'acquisition numériques (caméra CCD, capteur CMOS...).
- ✓ Soit analogique d'où la nécessité de passer par l'étape de numérisation avant tout traitement.

La chaîne de traitement est composée de trois étapes essentielles: prétraitement, segmentation et interprétation.

L'image d'origine va donc passer par quatre grandes étapes qui sont le **prétraitement** la **segmentation**, la **classification**, la **représentation** et enfin le **reconnaisseur**.

Nous pouvons visualiser ses étapes comme étant des blocs avec chaque fois une entrée et une ou plusieurs sorties.

Voici donc une description générale de chacun des blocs:

### II.2.1. Prétraitement

Cette phase a lieu juste après l'acquisition des images et a pour objectif d'améliorer la qualité de l'image en vue de sa segmentation et diminuer le bruit des images, ce dernier



désigne les pixels de l'image dont l'intensité est très différente de celle des pixels voisins. Le bruit est donc plutôt de type haut fréquence marquant une grande variation.

Le bruit peut avoir de différentes sources: environnement lors de l'acquisition (poussière, rayures, etc.), qualité du capteur, qualité de l'échantillonnage.

Parmi les méthodes de suppression de bruit les plus courantes, on retrouve le **seuillage** et le **lissage** qu'on va les détailler par la suite.

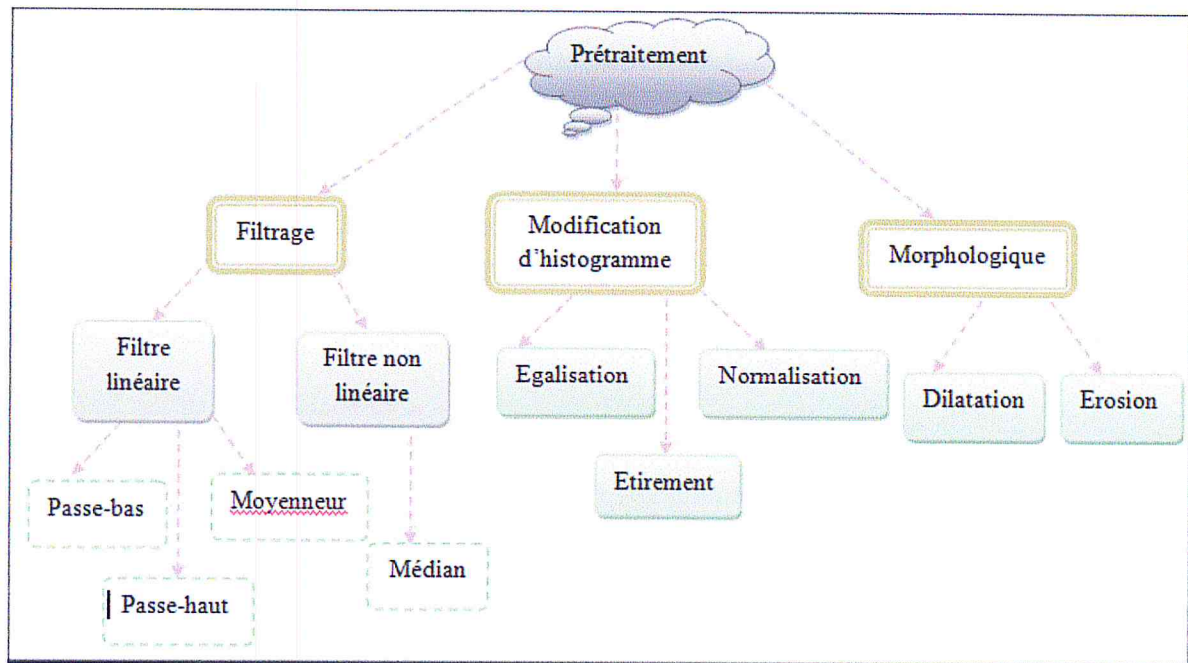


Figure 2.1 : Quelques prétraitements. (Cocquerez & Philipp, 2005)

### II.2.1.1. Le Filtrage

Dans le filtrage, chaque pixel est traité en fonction de sa propre valeur et de celle de ses voisins. On utilise souvent un masque de filtrage (matrice de convolution) qui prend en compte les 8 voisins immédiats du point traité. (Cocquerez & Philipp, 2005)

$$H = \begin{vmatrix} a & b & c \\ d & e & f \\ g & h & i \end{vmatrix}$$

$$I(x,y) = [ a * I(x-1,y-1) + b * I(x-1,y) + c * I(x-1,y+1) + \dots ] / K$$

#### i. filtre passe-bas

Les filtres passe-bas permettent d'atténuer les composantes de l'image ayant de hautes fréquences « pixels claires ». Ils lissent le signal, adoucissent les variations brutales de l'intensité, ils éliminent simplement le bruit, ils modifient assez peu l'image et gardent généralement intacte l'intensité moyenne de l'image.

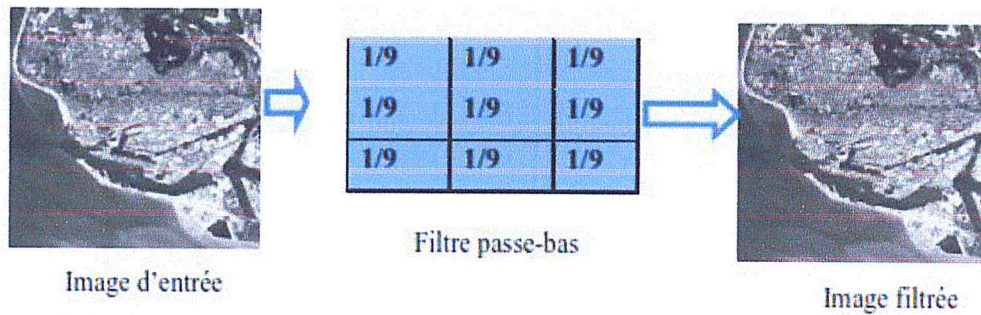


Figure 2.2: Image filtrée (passe-bas). (Houassine, 2012)

### ii. Filtre moyenneur

L'idée est de réaliser une moyenne des niveaux de gris autour du pixel central. Pour cela, on peut utiliser un masque du type :

$$\frac{1}{9} * \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

La taille du masque est un paramètre variable. Plus le masque sera de grande dimension, plus l'effet du filtrage sera fort.



Figure 2.3: Exemple de filtre. (Devars & Milgram, 2002/2003)

### iii. Filtre gaussien

Le filtre gaussien utilise un paramètre  $\sigma$  qu'on dit écart-type, car il correspond à l'écart-type des valeurs de la fonction gaussienne considérée comme une distribution. Ce filtre est très utilisé en traitement du signal et des images, pour de nombreuses raisons. La première liée au théorème centrale.

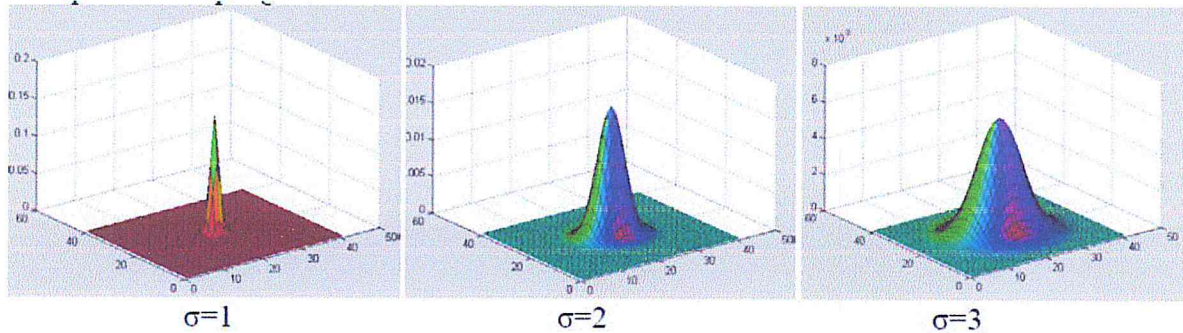


Figure 2.4: Exemple de masque gaussien. (Devars & Milgram, 2002/2003)

iv. **Filtre passe-haut**

Ils permettent d'atténuer les composantes de l'image ayant de basses fréquences. Ce type de filtrage permet d'accentuer les détails et le contraste d'une image numérique. Ils ont pour but de rehausser les frontières. Ils mettent en valeur les points isolés des objets, alors que les autres points sont mis à 0 « en noir ».

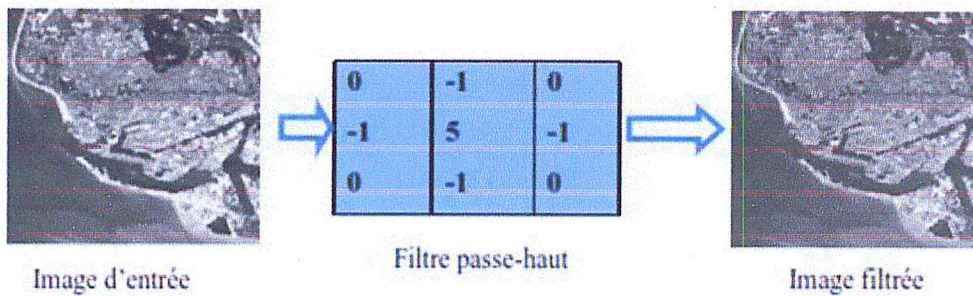


Figure 2.5: Image filtrée (passe-haut). (Hamza & Ibrahim, 2011/2012)

v. **Filtre médian**

Ce filtre est bien adapté à toutes les applications où une reconstruction d'image d'un sous-ensemble de coefficients est nécessaire (par exemple la restauration, compression, reconstruction partielle). La suppression des sous-ensembles de coefficients mène à peu d'objets façonnés dans l'image reconstruite, parce que souvent (Hamroune & Ait Hamou, 2006) Chaque pixel est traité en considérant ses voisins sur un voisinage donné. Le pixel lui-même et ses voisins forment alors un ensemble dont on calcule la « médiane ». Le pixel sera alors remplacé par cette valeur médiane. Nous allons vous donner un exemple illustratif pour un filtre médian en utilisant un voisinage de 3\*3 : (Lamri, avrile 2006)

12	14	5	4	2
4	99	23	56	4
5	126	121	120	5
4	97	12	9	4
2	4	5	4	2

Figure 2.6: Matrice filtrage médian.

Tri : 9 12 23 56 97 99 120 121 126

Valeur médiane 97

### II.2.1.2. L'histogramme

L'histogramme est un vecteur de dimension 256. Chaque élément du vecteur  $h(i)$  représente le nombre de pixels de l'image possédant le niveau de gris  $i$ . On peut donc assimiler l'histogramme à la densité de probabilité des intensités lumineuses (Figure 1.9).

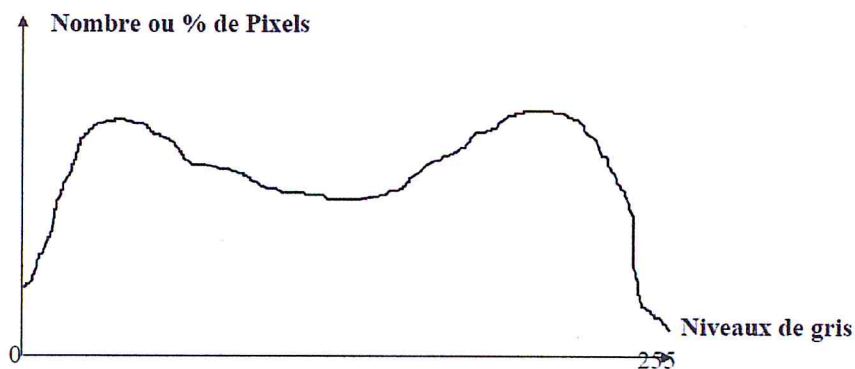


Figure 2.7: Histogramme. (Devars & Milgram, 2002/2003)

L'observation des histogrammes permet de déceler les défauts d'une image (trop claire, trop foncée, manque de contraste) ou de déduire certaines propriétés (séparation des objets du fond) (figure 2.8).

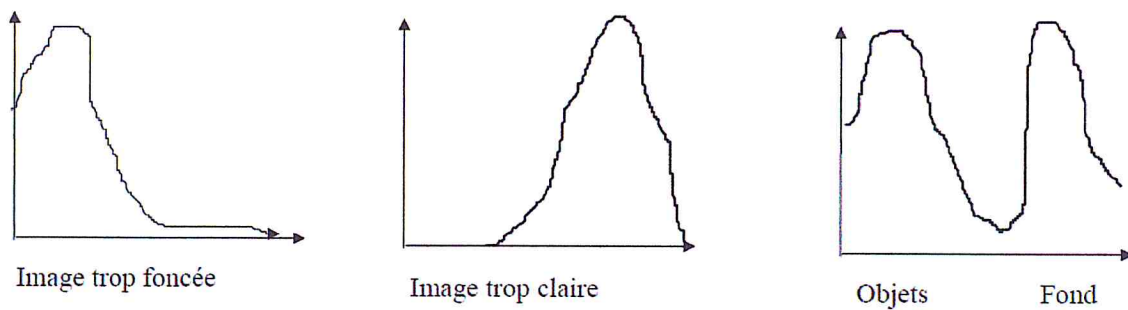


Figure 2.8: Exemple d'histogramme. (Houassine, 2012)

Il existe plusieurs techniques de prétraitements, les plus courantes sont :

### II.2.1.3. Binarisation

Pour une évaluation de différentes méthodes de binarisation, On distingue en général deux approches : les approches à seuillage global, et les approches à seuillage adaptatif. (Mkaouar & Lepage, 2005)

➤ **Seuillage** : l'opération de seuillage consiste à mettre à zéro tous les pixels ayant un niveau de gris inférieur à une certaine valeur, et à la valeur maximale les pixels ayant une valeur supérieure. Ainsi le résultat de seuillage est une image binaire contenant de pixels noirs et blancs, c'est la raison pour laquelle le terme de *binarisation* est parfois employé.

Il est possible de définir deux valeurs de seuil, respectivement borne inférieure et borne supérieure, afin de mettre à la valeur maximale les pixels ayant une valeur comprise entre les bornes et à zéro l'ensemble des autres valeurs.

➤ **Lissage** : C'est un traitement local qui définit une nouvelle valeur de pixel en examinant son voisinage. Il existe plusieurs opérateurs de lissage inspirés de la morphologie mathématique ; ils consistent à remplacer la valeur de chaque point P par une expression en fonction de ses voisins.

## II.2.2. Segmentation

La segmentation consiste à faciliter l'interprétation automatique d'une image de façon similaire à une interprétation humaine. Historiquement, elle a été inspirée du système de perception visuel humain qui utilise les notions de similarité et de différence afin de localiser et délimiter les objets d'une scène, (Salvatore & all, 2010). En général, la segmentation est considérée comme une étape de traitement « bas niveau » de l'image. Elle consiste à partitionner cette dernière en un ensemble de régions homogènes ou à détecter les contours (recherche des discontinuités locales) en se basant sur un calcul de statistiques sur l'image.

### II.2.2.1. Définition de la segmentation

La segmentation consiste à faciliter l'interprétation automatique d'une image de façon similaire à une interprétation humaine.

Historiquement, elle a été inspirée du système de perception visuel humain qui utilise les notions de similarité et de différence afin de localiser et délimiter les objets d'une scène (Salvatore & all, 2010).

On peut définir la segmentation comme étant une partition de l'image  $I$  en un nombre  $N$  de sous-ensembles  $R$  appelés régions, homogènes dans le sens d'un ou plusieurs critères, comme le niveau de gris, tels que :

- ✓ La segmentation doit être complète (c'est-à-dire, chaque pixel doit être affecté à une classe).
- ✓ Les pixels appartenant à la même région doivent être connectés.
- ✓ Les régions doivent être disjointes.

En termes mathématiques :

$$\left\{ \begin{array}{l} \bigcup_{i=1}^n R_i = I \\ R_i \neq \Phi \quad \forall i = 1 \dots n \\ R_i \cap R_j = \Phi \quad \forall i, j \text{ avec } i \neq j \end{array} \right.$$

Où  $R_i$  est l'ensemble des régions formant l'image  $I$ .

### II.2.2.2. Les différentes approches de segmentation

Il est difficile de définir, de manière absolue, une « bonne » segmentation qui fait référence aux notions de différence et de similarité comme les perçoit le système visuel humain et ceci donne naissance à deux approches : contour et région.

- ✓ La notion de « contour » est associée à une variation d'intensité ou à une discontinuité entre les propriétés de deux ensembles connexes de points.
- ✓ La notion de « région » fait référence à des groupements de points ayant des propriétés communes. Dans ce cas on observe une parfaite dualité entre les contours et les régions.

#### II.2.2.2.a. Approche contour

Il existe plusieurs méthodes utilisant l'approche contour, dans ce qui suit, nous allons citer les plus importantes.

➤ **Les méthodes dérivatives**

Les méthodes dérivatives sont les plus utilisées pour détecter des transitions d'intensité par différenciation numérique (Première et deuxième dérivé). Ce sont des méthodes locales qui balayent l'image avec un masque définissant la zone d'intérêt. A chaque position, un opérateur est appliqué afin de détecter les transitions significatives au niveau de l'attribut de discontinuité choisi. Le résultat est une image binaire constituée de points de contours et de points non-contours.

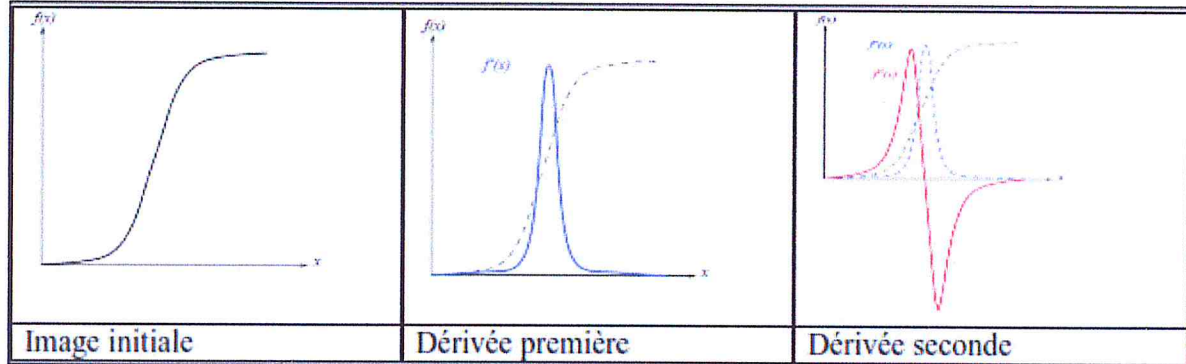


Figure 2.9: Contour et ses dérivées. (Alain, 2009)

De nombreuses techniques d'extraction de contours existent dans la littérature. Elles peuvent être classées comme suit :

- ✓ Les algorithmes basés sur le gradient (ou opérateurs du premier ordre).
- ✓ Les algorithmes basés sur le Laplacien (ou opérateurs du second ordre).
- ✓ Les algorithmes utilisant des filtres sophistiqués.

**i. Utilisation du gradient**

En considérant l'image sur un repère orthogonal et qui est modélisée par une fonction de niveaux de gris, le gradient en tout point est désigné par :

$$\overrightarrow{\text{Grad}} f = \overrightarrow{\nabla} f = \left( \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right)$$

Le principe de cette méthode consiste tout d'abord à calculer le gradient de l'image, puis son module dont la formule est :

$$\|\overrightarrow{\nabla} f\| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

Ce dernier traduit les variations du niveau de gris. Plus la valeur est élevée plus la variation est forte. La direction du gradient est donnée par :

$$\alpha = \text{arctang} \left( \frac{\partial f / \partial y}{\partial f / \partial x} \right)$$

Celle-ci représente l'orientation de la variation. Enfin les contours les plus marqués (les points de plus fort contraste) sont sélectionnés grâce à un seuillage adéquat.

**ii. Les filtres**

Le filtrage est une opération fondamentale en traitement d'images. Il permet d'améliorer la perception de certains détails, de réduire le bruit, de compenser certains défauts du capteur, etc.

Les contours constituent une information essentielle pour certaines applications de traitement d'images. Les contours correspondent en général à des changements brusques de propriétés physiques ou géométriques de la scène. En particulier, les contours d'un objet permettent en général de caractériser sa forme.

La détection de contours peut être réalisée grâce à des filtres dont les coefficients ont été soigneusement choisis. Ces filtres comportent deux masques de convolution, un pour la détection des contours verticaux et un autre pour la détection des contours horizontaux. On obtient après convolution une image des gradients. Cette image pourra par la suite être modifiée dans une phase de seuillage pour ne contenir que des pixels à 1 ou 0 correspondant respectivement à point de contour ou à un non point de contour.

Nous aborderons trois jeux de filtres particuliers, les plus utilisés : les filtres de Prewitt, Roberts, et Sobel.

• **Filtres de Prewitt** : Les filtres de Prewitt sont les suivants :

$$\text{Filtre horizontal : } h = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{pmatrix} \quad \text{Filtre vertical : } v = \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix}$$

• **Filtres de Sobel** : Les filtres de Sobel sont les suivants :

$$\text{Filtre horizontal : } h = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix} \quad \text{Filtre vertical : } v = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}$$

• **Filtres de Robert** : Les filtres de Roberts sont les suivants :

$$\text{Filtre diagonal : } a = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \quad \text{Filtre anti-diagonal : } b = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$$



### iii. Utilisation du Laplacien

Dans cette méthode, le contour est mis en évidence par le passage par zéro de la dérivée seconde (le Laplacien) :

Cette technique est particulièrement sensible au bruit, en raison de la double dérivation. De ce fait, un seuil sera utilisé, afin de ne pas prendre en considération le bruit, qui peut se traduire lui aussi par des oscillations autour de zéro. Ce seuil ne prendra en compte que les passages par zéro les plus marqués (d'amplitude relativement élevée) correspondant aux vrais contours. (Materka & Strzelecki, 2005)

### iv. Transformations morphologiques

Les images binaires permettent également d'appliquer des transformations morphologiques sur les formes. Il peut s'agir par exemple de les épaissir ou des les amincir, de les lisser, d'éliminer des objets trop petits, etc. Ce genre de traitements utilise des filtres morphologiques, tels que la dilatation, l'érosion, le filtre médian. Ces opérations permettent par exemple de remplir des trous laissés vacants dans une forme lors de la binarisation. Dans la même idée, on peut aussi détecter les frontières des formes puis les squelettiser.

#### a. Les modèles déformables

Le principe des contours actifs (ou snakes) est de faire évoluer un contour initial de manière itérative vers une position d'équilibre, c'est -à-dire, en direction des bords de l'objet à détecter. Entre deux itérations, le déplacement des points est régi par une équation mettant en jeu des forces à appliquer au contour. La position d'équilibre peut être définie par la minimisation d'une énergie, calculée sur une ou plusieurs données présentes dans l'image (intensité, gradient...) ou traduisant des régularités exigées sur la solution, de manière à ce que le problème de minimisation devienne un problème bien posé. L'utilité des contours actifs est particulièrement bien illustrée en imagerie médicale (surfaces régulières comme les surfaces de frontière d'organe).

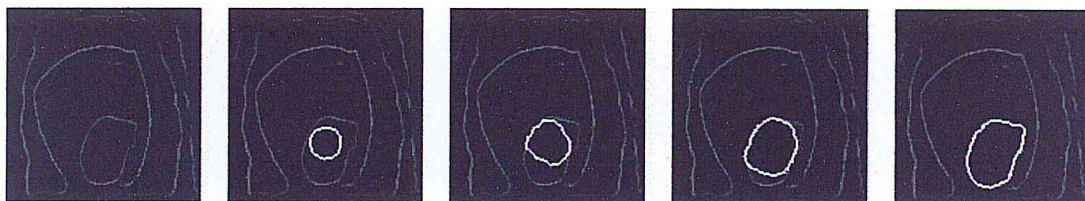


Figure 2.10 : Segmentation par contour actif. (Krystian & Cordelia, oct 2005)

Les domaines d'utilisation sont nombreux tant en 2D qu'en 3D tels : la reconnaissance de formes, la simulation, le suivi de scènes. (Rouquet & all, 2000)

Un contour se matérialise par une rupture d'intensité dans une image. Plusieurs méthodes existent afin de détecter cette rupture ; dans la plupart des cas elles s'effectuent en deux étapes: la détection du contour par le calcul du gradient ou du Laplacien, et l'isolation de ce contour par seuillage.

#### II.2.2.2.b. Approche Région

La segmentation d'images par l'approche région consiste à découper l'image en régions (wikipedia, Nov 2013). Les méthodes appartenant à cette catégorie, contrairement à l'approche contour, se basent sur la similarité entre les pixels. Pour cela, il existe différentes techniques que nous allons présenter dans ce qui suit.

##### i. Segmentation par croissance de régions (Region growing)

La segmentation par croissance de régions appelées aussi agrégation pixels (Stricker & Orengo, 2009) est une méthode ascendante. Initialement, des germes, généralement de la taille d'un pixel, sont choisis. A partir de ces derniers, des régions sont construites par ajout de pixels adjacents et similaires (valeurs identiques de niveau de gris, similarité colorimétrique...etc.), La croissance des régions s'arrête lorsque tous les pixels ont été traités (affectés à une classe). Généralement, une mesure de similarité peut être évaluée par un calcul de distance entre les attributs du pixel candidat et ceux de la région. Lorsqu'une région intègre un nouvel élément, ses attributs sont alors remis à jour. La croissance de région s'interrompt lorsque tous les pixels voisins aux régions ont été affectés.

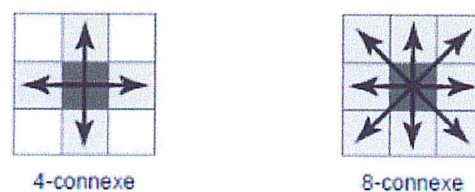


Figure 2.11: Relation entre pixels.

##### ii. Segmentation par division de régions (Split)

La division consiste à partitionner l'image en régions homogènes selon un critère donné (niveau de gris dans le cas des images médicales). Le principe de cette technique est de considérer l'image elle-même comme région initiale, qui par la suite sera divisée en régions.

Le processus de division est réitéré sur chaque nouvelle région (issue de la division) jusqu'à l'obtention de classes homogènes. Le résultat peut être représenté sous forme d'un arbre appelé Quad-tree. (Marr & Hildreth, 1995)

### iii. Segmentation par division-fusion

Ces méthodes consistent à découper « division » itérativement l'image jusqu'à l'obtention de blocs homogènes selon un critère donné. Puis regrouper les blocs voisins de sorte que le bloc résultant respecte un critère d'homogénéité. Ces méthodes peuvent faire appel à la théorie des graphes. (Byngmoon & Panagiotis, 2006)

### iv. Segmentation par graphe d'adjacence

Le graphe d'adjacence est un graphe non orienté où chaque nœud représente une région et chaque branche représente l'adjacence entre les régions :

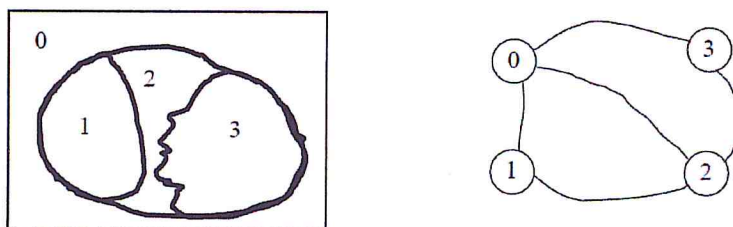


Figure 2.12: Segmentation par graphe d'adjacence.

Grâce au graphe d'adjacence, on n'a pas à remonter au pixel de l'image pour fusionner les régions. (Devars & Milgram, 2002/2003)

### v. Segmentation pyramidale

Horowitz a proposé une approche par pyramide, L'image est représentée sur une pyramide appelée quad-tree, constituée de  $N$  niveaux, l'image originale étant au niveau 0. Chaque pixel au niveau  $v$  a 4 fils au niveau  $v-1$ . Le pixel au niveau  $v$  est la moyenne de ses 4 fils, La segmentation procède directement à un niveau intermédiaire (par exemple  $v=2$ ), et tous les pixels fils sont testés avec le prédicat  $P$ . Si le prédicat n'est pas vérifié, le pixel considéré est étiqueté comme candidat à un partage. Sinon il est candidat à une réunion avec ses voisins. C'est donc une technique de partage et réunion, où l'on profite du passage au niveau supérieur (ici  $v=3$ ) pour accélérer la procédure (figure 2.14).

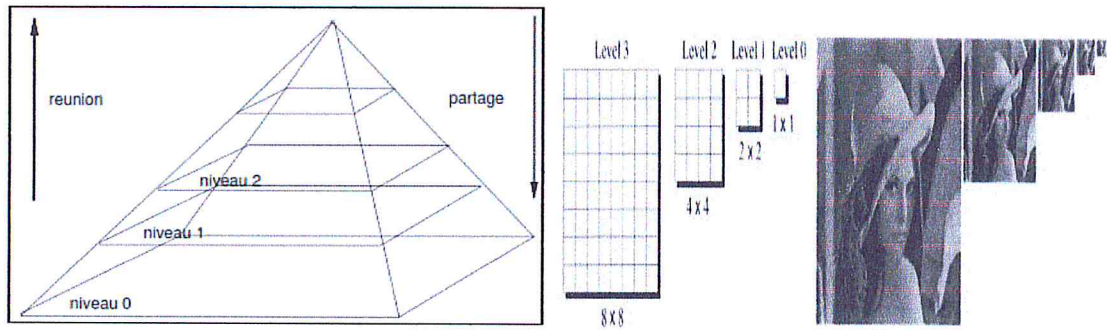


Figure 2.13: Méthode de pyramidage par Quad-tree. (Henri, 2013)

#### vi. Segmentation par seuillage

Cette méthode est parmi les plus simples pour segmenter une image. Elle consiste à regrouper les pixels en régions selon un ou plusieurs seuils. Si nous prenons l'exemple d'un seul seuil, nous pouvons dire que les pixels ayant un niveau de gris inférieur ou égal au seuil appartiennent à la classe C, tandis que les pixels ayant un niveau de gris strictement supérieur au seuil appartiennent à la classe 2. Le principe est le même dans le cas où l'on dispose de plusieurs seuils.

Le problème qui se pose est la détermination des seuils. Ils peuvent soit être introduits par l'utilisateur, on parlera alors de seuillage interactif, ou bien, être calculés automatiquement, on parlera dans ce cas de seuillage automatique.

Une bonne méthode de calcul des seuils consiste à dresser un histogramme qui représente les occurrences de chaque niveau de gris de l'image, les pics de l'histogramme seront considérés comme des seuils. Cette méthode s'applique correctement si l'histogramme contient des pics bien séparés. Elle est très affectée par le bruit. (Materka & Strzelecki, 2005)

#### vii. La segmentation coopérative

Les chercheurs ont essayé de trouver un compromis entre l'approche région et contour et ont donné naissance à ce qu'on appelle la segmentation coopérative. Son principe est de tirer profit de chacune des approches pour un meilleur résultat de segmentation. On distingue en principe trois types de coopération:

1. La coopération série : donne naissance à des algorithmes de segmentation puissants. Cependant le résultat dépend souvent de l'ordonnement de la coopération.

2. La coopération parallèle : elle fait appel à trois mécanismes, la fusion, l'adaptation et la correction. Généralement la modélisation utilisée favorise un de ces trois mécanismes, ce qui ne contribue pas d'une manière générale à des résultats probants.

3. La coopération hybride : permettant de solutionner quelques insuffisances des deux coopérations précédentes. (Guillaume, 2011)

### II.3. Conclusion

La manipulation des images pose cependant des problèmes beaucoup plus complexes que celle du texte. En effet, l'image est un objet à deux dimensions, censé représenter un espace à trois dimensions, ce qui a deux conséquences majeures puisque le volume des données à traiter est beaucoup plus important et la structure de ces données est nettement plus complexe.

Il en résulte que la manipulation, le stockage et la représentation de ces données se heurtent à certaines limitations.

Grâce au traitement d'image, ces contraintes sont levées ou contournées. En effet, ce domaine cherche à détecter la présence de certaines formes, certains contours ou certaines textures d'un modèle connu, c'est le problème de la **segmentation**.

Lorsque l'on cherche à résoudre un problème de classification, on a affaire à des formes susceptibles d'appartenir à des catégories, ou classes différentes. Un classifieur est capable d'attribuer une classe à une forme inconnue qui lui est présentée.

Les performances d'un classifieur dépendent essentiellement de la représentation des formes utilisées, c'est-à-dire du prétraitement effectué sur les données brutes.

# *Chapitre 3*

## **Extraction**

**De**

**Caractéristiques**

### III.1. Introduction

En vision par ordinateur, l'extraction de caractéristiques visuelles (ou visual features extraction en anglais) consiste en des transformations mathématiques calculées sur les pixels d'une image numérique. Les caractéristiques visuelles permettent généralement de mieux rendre compte de certaines propriétés visuelles de l'image, utilisées pour des traitements ultérieurs entrant dans le cadre d'applications telles que la détection d'objets ou la recherche d'images par le contenu.

### III.2. Localisation des caractéristiques

On distingue usuellement les caractéristiques globales qui sont calculées sur toute l'image et les caractéristiques locales qui sont calculées autour de points d'intérêt. On peut aussi parler des caractéristiques semi-locales quand celles-ci sont extraites dans des zones restreintes de l'image, résultant d'une segmentation de l'image en région ou simplement selon une grille arbitraire. (Hervé, 2009)

La distinction entre caractéristiques globales et locales a un intérêt taxinomique. Les caractéristiques locales se distinguent par le fait qu'elles sont distinctes, robustes aux occlusions (car il y en a beaucoup dans une image ou une région) et qu'elles ne nécessitent pas de segmentation, un descripteur local calculé en chaque pixel d'une image ou d'une région obtenue par segmentation, puis accumulé dans un histogramme est donc une description globale de l'image ou de la région. (Krystian & Cordelia, oct 2005)

#### III.2.1. Caractéristiques globales

##### i. Couleur (Histogramme et dérivés)

La prise en compte de la couleur des images a été historiquement l'une des premières caractéristiques employées pour la recherche d'images par le contenu et produit encore des résultats parfois spectaculaires sur certaines bases. C'est en 1991 que Swain et Ballard ont proposé d'utiliser un histogramme couleur pour ce genre de tâche. Il représente la distribution globale des couleurs dans l'image. Son calcul consiste en une quantification de l'espace couleur choisi (RVB par exemple) suivie du calcul de l'histogramme des pixels ainsi transformés. Par exemple, si l'on considère une image RVB classique et que l'on quantifie

chaque plan couleur sur 4 bins, l'histogramme résultant aura une dimension  $4^3 = 64$ . Si l'image est originellement codée sur 24 bits (la valeur de chaque plan est dans l'intervalle [0-255]), l'histogramme couleur sur 64 bins pourrait être représenté par un « cube »  $4 \times 4 \times 4$ .

		Rouge															
		0-63				64-127				128-191				192-255			
		Vert				Vert				Vert				Vert			
		0-63	64-127	128-191	192-255	0-63	64-127	128-191	192-255	0-63	64-127	128-191	192-255	0-63	64-127	128-191	192-255
Bleu	0-63	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	64-127	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	128-191	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	192-255	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Figure 3.1 : Histogramme d'une image couleur sur 64 bins. (Swain & Ballard, 1991)

Chaque couleur représentant l'un des 64 bins quantifiés, et x étant alors le nombre de pixels de l'image correspondant à ce bin. En pratique, ce descripteur est généralement représenté par un vecteur mono dimensionnel à 64 dimensions.

De nombreux descripteurs peuvent être dérivés de la représentation par histogramme, dont plusieurs ont été inclus au modèle expérimental de la norme MPEG-7. Il s'agit notamment de:

**dominant color:** rend compte de la couleur dominante d'une image, calculée comme celle du bin ayant la plus grande valeur. L'espace de couleur peut être quelconque en théorie, mais l'utilisation de HSV donne des résultats intéressants. (Liu, Zhang, & Lu, 2004)

**scalable colour:** il s'agit de la transformée de Haar de l'histogramme couleur dans l'espace HSV, ce qui résulte en un code très compact.

**couleur structure:** il s'agit d'un histogramme couleur «local». Sa construction est identique au principe présenté précédemment (histogramme couleur global), si ce n'est qu'une fenêtre de taille réduite (8x8 pixels) parcourt l'image lors du comptage des couleurs en chaque pixel. Il reflète ainsi une certaine information sur la structure spatiale de la répartition des couleurs.

**colour layout:** la transformée en cosinus discrète de la représentation en histogramme est calculée et seule une partie des coefficients peut être utilisée pour la signature. Cela donne la possibilité d'obtenir un descripteur relativement grossier mais très compact ou bien un descripteur plus fin mais prenant plus de place. Un tel descripteur est intéressant pour des applications nécessitant une certaine scalabilité.



## ii. Forme

Décrire les formes nécessite une identification préalable de régions. Cela peut résulter d'une segmentation de l'image ou de la détection de leurs contours. On peut alors caractériser les régions au moyen de divers indices telle leur orientation principale (par exemple leur premier axe propre), leur symétrie ou encore leurs moments de Zernike (Les polynômes de Zernike sont une série de polynômes qui sont orthogonaux sur le disque unité).

Une méthode de description particulièrement utilisée est celle proposée par Mokhtarian, appelée Curvature Scale Space (**Stricker & Orengo, 2009**). Elle consiste à décrire une région en fonction des variations de courbure de son contour. Pour une courbe plane en coordonnées paramétriques dans un repère orthonormé  $r(t)=(x(t),y(t))$ , la courbure s'exprime par:

$$\gamma(t) = \frac{x'y'' - y'x''}{(x'^2 + y'^2)^{3/2}}$$

Cette description est invariante en rotation et translation. Concernant le zoom, il faut à minima normaliser par la longueur du contour. Plus important, le descripteur est généralement calculé à différentes échelles, chacune correspondant à la convolution par un noyau gaussien.

## iii. Texture

La définition de la texture n'est pas évidente et dépend de l'échelle, On distingue trois approches pour définir la notion de texture. Dans la veine des travaux pionniers de Haralick (**Gevers & Smeulder, 2004**), une première approche considère une texture comme résultant d'un processus stochastique et cherche à les décrire en termes de propriétés statistiques des valeurs et positions relatives des pixels. La seconde approche suppose l'existence de primitives fondamentales permettant de décrire les textures, tels les Textons définis par Julesz. Par suite, une texture donnée est décrite comme une combinaison complexe de ces primitives, exprimée par exemple avec des graphes.

Une approche plus récente s'inspire plutôt de la perception visuelle et tend à rendre compte du processus de formation des textures du point de vue humain. (**Julesz, 2002**)

### III.2.2. Caractéristiques locales

La caractérisation d'une image peut être calculée en un nombre restreint de pixel. Il faut pour cela d'abord détecter les zones d'intérêt de l'image puis calculer en chacune de ces zones un vecteur caractéristique. Ces zones d'intérêt sont par exemple les arêtes ou les points saillants de l'image (zones de fort contraste). Il peut aussi s'agir de points pris aléatoirement ou régulièrement dans l'image (échantillonnage dit dense).

Le vecteur caractéristique contient parfois des données provenant de la détection, telles que l'orientation de l'arête ou la magnitude du gradient dans la zone d'intérêt. Généralement, le vecteur caractéristique en un pixel est calculé sur un voisinage de ce pixel, c'est-à-dire à partir d'une imagerie centrée sur ce pixel. Il peut être calculé à différentes échelles de manière à s'affranchir du facteur de zoom. Parmi les caractéristiques locales couramment calculées, on retrouve des motifs préalablement utilisés globalement, tels que des histogrammes de couleur ou des vecteurs rendant compte de l'orientation des gradients des niveaux de gris.

Certaines méthodes telles SIFT ou SURF incluent à la fois la détection de zone d'intérêt et le calcul d'un vecteur caractéristique en chacune de ces zones. Concernant le vecteur caractéristique, les SIFT sont grossièrement un histogramme des orientations du gradient et les SURF consistent en le calcul d'approximation d'ondelettes de Haar. Dans une veine similaire, les motifs binaires locaux sont un calcul de cooccurrence des niveaux de gris locaux d'une image et les HOG des histogrammes de gradients orientés, assez similaires aux SIFT, calculés selon un échantillonnage dense.

#### III.2.2.1. Détection de zones d'intérêt

En vision par ordinateur et en traitement d'images, la détection de zones d'intérêt d'une image numérique (feature detection en anglais) consiste à mettre en évidence des zones de cette image jugées « intéressantes » pour l'analyse, c'est-à-dire présentant des propriétés locales remarquables. De telles zones peuvent apparaître, selon la méthode utilisée, sous la forme de points, de courbes continues, ou encore de régions connexes rectangulaires ou non et qui constituent le résultat de la détection.

### III.2.2.2. Terminologie

Nous utilisons la locution zone d'intérêt pour traduire dans ce contexte précis le terme feature omniprésent dans la littérature anglo-saxonne, mais qui n'a pas de réelle définition standard. La définition dépend plutôt de la famille de problèmes que l'on cherche à traiter. Comme expliqué plus haut, une zone d'intérêt est une zone « intéressante » d'une image, et peut être utilisée comme point de départ de nombreux algorithmes de traitement d'images. De ce fait, la qualité de l'algorithme utilisé pour détecter les zones d'intérêt conditionne souvent la qualité du résultat de la chaîne de traitement entière que l'on souhaite appliquer à une image. Aussi, la répétabilité, c'est-à-dire le fait que les mêmes zones d'intérêt (ou à peu près) puissent être détectées sur deux images (numériquement) différentes mais représentant la même scène, est une propriété importante et généralement exigée pour tous les algorithmes de détection de zones d'intérêt.

Après la détection, on applique souvent un algorithme de description qui va se concentrer sur chaque zone d'intérêt détectée pour en calculer des caractéristiques (numériques, en général), que les anglo-saxons désignent aussi sous le terme générique de Feature. (Mikolajczyk & all, 2005)

### III.2.2.3. Typologie

Selon la méthode de détection utilisée, les zones peuvent être de différentes natures.

#### i. Contours :

Les premières méthodes proposées historiquement se fondent sur l'analyse des contours et des arêtes, c'est-à-dire des zones où la luminance (ou la couleur) de l'image change brusquement – en d'autres termes, comporte une discontinuité. On peut utiliser pour cela les opérateurs de Canny ou de Sobel, par exemple. Ces détecteurs sont basés sur une dérivation de l'image (au sens mathématique), c'est-à-dire sur le gradient de l'image vue comme une fonction de deux variables réelles, à valeurs dans  $\mathbb{R}$  pour les images en niveaux de gris ou dans  $\mathbb{R}^3$  pour les images en couleurs. Les points de discontinuité connexes trouvés peuvent être reliés entre eux pour former des arêtes (edges), linéaires ou non, par des post-traitements appropriés. (Canny, 1986)

**ii. Points d'intérêt :**

Les algorithmes de détection de points d'intérêt se focalisent en général sur des points particuliers des contours, sélectionnés selon un critère précis.

Ainsi, les coins (corners) sont les points de l'image où le contour (de dimension 1) change brutalement de direction, comme par exemple aux quatre sommets d'un rectangle. Il s'agit de points particulièrement stables, et donc intéressants pour la répétabilité. La méthode la plus répandue pour le détecter est probablement le détecteur de Harris.

Comme celle de Harris, la plupart des autres techniques de détection de points d'intérêt sont basées sur une analyse locale de l'image à l'ordre 2. Ce qui les différencie entre elles est l'opérateur de dérivation utilisé. Nous pouvons par exemple citer les méthodes basées sur l'analyse des DoG (Difference of Gaussians), des LoG (Laplacian of Gaussian) ou des DoH (Difference of Hessians). (Lindeberg T. , 2001)

**iii. Régions d'intérêt**

Il s'agit de zones d'intérêt plus générales que les points, utiles lorsque les structures recherchées dans une image ne correspondent pas à des points saillants ; par exemple, lorsque l'image a subi un lissage important ou lorsque les contours sont épais et progressifs.

Souvent, ces techniques commencent par identifier des points d'intérêt qui vont se révéler être des sortes de barycentres des régions recherchées (blobs en anglais), telles que les méthodes multi-échelles basées sur l'étude des détecteurs de points d'intérêt cités précédemment (Harris, DoG, etc), à différentes échelles de l'image. Ceci permet d'obtenir des régions soit circulaires soit elliptiques, selon le niveau de raffinement voulu. Ces méthodes sont souvent intégrées à des algorithmes plus généraux tels que SIFT ou SURF, qui incluent un descripteur de région d'intérêt en plus d'un détecteur.

Parmi les détecteurs de régions d'intérêt plus généraux existe également MSER (Maximally Stable Extremal Regions).

Plus spécifiquement enfin, la recherche d'objets particulièrement allongés fait souvent appel à des techniques dites de détection de crêtes (ridge detection) qui se prête bien à l'analyse de vues aériennes de réseaux routiers, ou à la détection de vaisseaux sanguins dans l'imagerie médicale par exemple. (Matas & Chum, 2002)

### III.2.3. Scale-Invariant Feature Transform (SIFT)

#### III.2.3.1. Qu'est-ce que SIFT?

Scale-invariant feature transform (SIFT), que l'on peut traduire par "transformation de caractéristiques visuelles invariante à l'échelle", est un algorithme utilisé dans le domaine de la vision assistée par ordinateur pour détecter et identifier les éléments similaires entre différentes images numériques (éléments de paysages, objets, personnes...), ceci ayant pour objectif de décrire des zones d'intérêts dans une image. Il a été développé en 1999 par le chercheur David Lowe, et le propriétaire du brevet est l'Université de la Colombie-Britannique.

##### i. Ses applications

Les applications de la méthode sont nombreuses et ne cessent de s'étendre ; elles couvrent actuellement des domaines tels que la détection d'objet, la cartographie et la navigation, l'assemblage de photos, la modélisation 3D...

##### ii. Principe de base

La méthode proposée par Lowe comprend deux parties :

- ✓ un algorithme de détection de caractéristiques et de calcul de descripteurs
- ✓ un algorithme de mise en correspondance proprement dit.

Le premier est sans doute celui qui a le plus assuré la popularité de la méthode, Il s'agit tout d'abord de détecter sur l'image des zones circulaires « intéressantes », centrées autour d'un point-clé et de rayon déterminé appelé facteur d'échelle. Celles-ci sont caractérisées par leur unité visuelle et correspondent en général à des éléments distincts sur l'image. Sur chacune d'elles, on détermine une orientation intrinsèque qui sert de base à la construction d'un histogramme des orientations locales des contours, habilement pondéré, seuillé et normalisé pour plus de stabilité. C'est cet histogramme qui sous la forme d'un vecteur à 128 dimensions (ou valeurs) constitue le descripteur SIFT du point-clé, et l'ensemble des

une étape de reconvergence et de filtrage qui permet d'améliorer la précision sur la localisation des points-clés et d'en éliminer un certain nombre jugés non pertinents. Chaque point-clé restant est ensuite associé à une orientation intrinsèque, c'est-à-dire ne dépendant que du contenu local de l'image autour du point clé, au facteur d'échelle considéré. Elle permet d'assurer l'invariance de la méthode à la rotation et est utilisée comme référence dans le calcul du descripteur, qui constitue la dernière étape de ce processus. (Haralick, 2004)

### III.2.4.2. Détection d'extrema dans l'espace des échelles

La détection s'effectue dans un espace discret que l'on appelle espace des échelles ("scale space") qui comporte trois dimensions : les coordonnées cartésiennes  $x$  et  $y$  et le facteur d'échelle  $\sigma$ . On appelle "gradient de facteur d'échelle  $\sigma$ " (noté  $L$ ) le résultat de la convolution d'une image  $I$  par un filtre gaussien  $G$  de paramètre  $\sigma$ .



Figure 3.2 : Illustrations de DOG.

Cette convolution a pour effet de lisser l'image originale de telle sorte que les détails trop petits, c'est-à-dire de rayon inférieur à  $\sigma$ , sont estompés. Par conséquent, la détection des objets de dimension approximativement égale à  $\sigma$  se fait en étudiant l'image appelée "différences de gaussiennes" (en anglais "difference of gaussians", DoG).

L'utilisation d'une pyramide est préconisée pour optimiser le temps de calcul des images floutées à un grand nombre d'échelles différentes. La base de la pyramide est en général l'image originale et un niveau donné (on parle d'octave par analogie avec la musique) est obtenu à partir du précédent en divisant la résolution de l'image par 2, ce qui revient à doubler le facteur d'échelle. Au sein d'une même octave, le nombre de convolutions à calculer est constant. (Lindeberg T. , 2001)

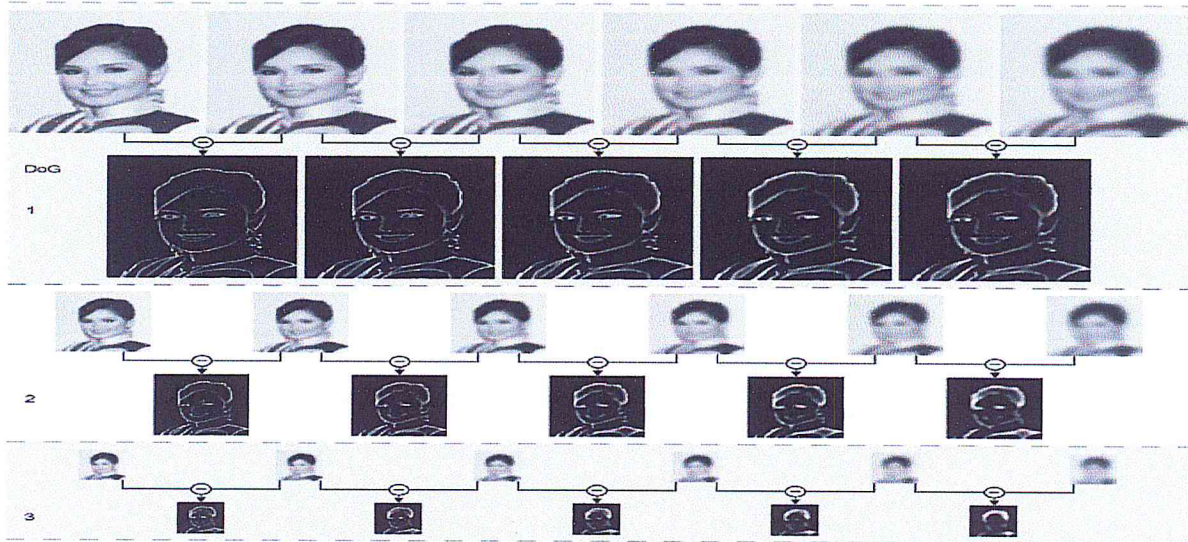


Figure 3.3 : Le calcul des points clés.

Un point-clé candidat est défini comme un point où un extremum du DoG est atteint par rapport à ses voisins immédiats. C'est comme cela que l'invariance par échelle est obtenue aux niveaux des points d'intérêts.

#### III.2.4.3. Amélioration de la précision

L'étape de détection d'extremums produit en général un grand nombre de points d'intérêt candidats, dont certains sont instables. De plus, leur localisation, en particulier aux échelles les plus grandes (autrement dit dans les octaves supérieures de la pyramide où la résolution est plus faible) reste approximative. De ce fait, des traitements supplémentaires doivent être appliqués, pour un objectif double :

- ✓ reconverger la position des points pour améliorer la précision sur  $x$ ,  $y$  et  $\sigma$ .
- ✓ éliminer les points de faible contraste.
- ✓ supprimer les points situés sur des arêtes de contour à faible courbure et donc susceptibles de "glisser" facilement. (Johannes & Niko, 2007)

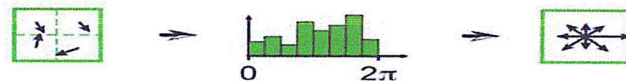




Figure 3.4 : Précision des points clés.

#### III.2.4.4. Assignment d'orientation aux points d'intérêt

L'étape d'assignation d'orientation consiste à attribuer à chaque point d'intérêt de l'image une ou plusieurs orientations déterminées localement (sur l'image) à partir de la direction des gradients dans un voisinage autour du point. Dans la mesure où les descripteurs sont calculés relativement à ces orientations, cette étape est essentielle pour garantir l'invariance de ceux-ci à la rotation : les mêmes descripteurs doivent pouvoir être obtenus à partir d'une même image, quelle qu'en soit l'orientation. (Yan & all, 2009)



#### III.2.4.5. Les descripteurs des points d'intérêt

Une fois que les points d'intérêt associés à des facteurs d'échelles et à des orientations ont été détectés et que leur invariance aux changements d'échelles et aux rotations est assurée, l'algorithme SIFT arrive alors à l'étape de calcul des vecteurs descripteurs, traduisant numériquement chacun de ces points d'intérêt (ou points-clés). Au cours de cette étape, des traitements supplémentaires sont appliqués, permettant d'assurer un surcroît de robustesse de l'algorithme, en rendant les descripteurs invariants à d'autres transformations telles que:

- ✓ la luminosité
- ✓ le changement de point de vue 3D...

Cette étape est réalisée sur l'image lissée avec le paramètre de facteur d'échelle le plus proche de celui du point-clé considéré (Paul & all, 2007).



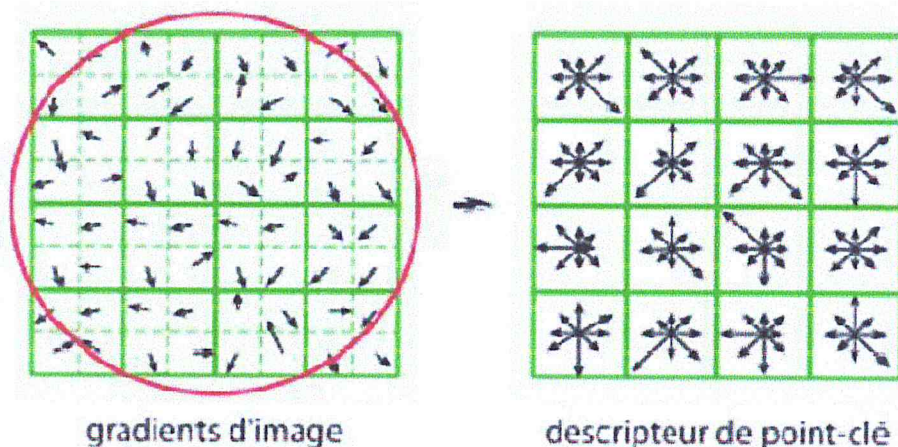


Figure 3.5 : Illustration des descripteurs des points d'intérêt.

#### III.2.4.6. Utilisation pour la recherche d'objets dans des images

La problématique de base pour laquelle la méthode SIFT a été conçue est la suivante : peut-on trouver dans une image donnée (dite image question ou image suspecte), des objets déjà présents dans une collection d'images de référence pré-établie ?

Dans la méthode originale de David Lowe (**David L. , 2002**), les points-clés et les descripteurs SIFT sont tout d'abord extraits des images de référence et stockés dans une sorte de base de données. Un objet est identifié dans l'image question en effectuant une comparaison de ses descripteurs à ceux des images de référence disponibles en base de données, fondée simplement sur la distance euclidienne. Parmi toutes les correspondances ainsi établies, des sous-ensembles (clusters) sont identifiés, au sein desquels la mise en correspondance est cohérente du point de vue des positions des points, des facteurs d'échelle et des orientations. Les clusters contenant au moins trois correspondances ponctuelles sont conservés. Dans chacun d'eux, on modélise la transformation permettant de passer de l'image question à l'image de référence, et on élimine les correspondances aberrantes par simple vérification de ce modèle. Enfin, Lowe applique un modèle probabiliste pour confirmer que la détection d'une correspondance d'objets entre l'image question et l'une des images de référence n'est pas due au hasard, baser sur l'idée que si de nombreux points n'ont pas pu être mis en correspondance c'est que l'on a peut-être affaire à un faux positif. (**Jeffrey & David, 2001**)

### **III.3. Conclusion**

Dans ce chapitre, on a prouvé la puissance de l'algorithme SIFT et son rôle de donner à chaque image une signature unique à travers des descripteurs. Ce qui nous aidera à utiliser ces caractéristiques locales et créer une référence à notre système de classification pour qu'il puisse mieux connaître les images de test en se basant sur cette référence.

Dans le prochain chapitre, on abordera le volé conception de notre système.

*Chapitre 4*

Conception

&

Expérimentation

## IV.1. Introduction

Dans ce chapitre notre objectif est de développer sous Matlab un système complet de classification des images à base de caractéristiques locales.

La figure 4.1. Présente le schéma fonctionnel du système qui résume la conception de notre système de classification, qui nécessite au début l'extraction des caractéristiques de chaque image dans la base des images utilisées (IARP) par la méthode SIFT, puis la classification des descripteurs par un simple classifieur KMEANS afin de construire le vocabulaire visuel. Les K-cluster qui se trouvent dans le vocabulaire visuel et les descripteurs des images de test de notre base servent comme entrée à notre classifieur KPPV ou SVM, qui va classer les images inconnues et les comparer avec les images d'apprentissage de la base pour étiqueter chaque image à sa classe correspondante, à la fin on présente le taux de reconnaissance de notre système.

Dans un premier temps nous avons évalué des classifieurs simples (K-means, KPPV) et complexes (SVM) avec des caractéristiques locales (application de la méthode SIFT) pour la classification des images. Dans un second temps nous avons évalué et étudié l'impact de la combinaison de plusieurs classifieurs, KPPV avec classifieur SVM. Dans le même but, nous avons réappliqué la technique de combinaison en changeant la distance utilisée (Euclidien ou Chi2) pour le classifieur KPPV et la technique (Linéaire ou non Linéaire) pour SVM.

Notre système a été exécuté dans Matlab 2009a sur un processeur Intel(R) Core(TM) i3 CPU à 2,13 GHz avec 4 Go de mémoire DDR3 et le type de système de 64 bits de Windows 7

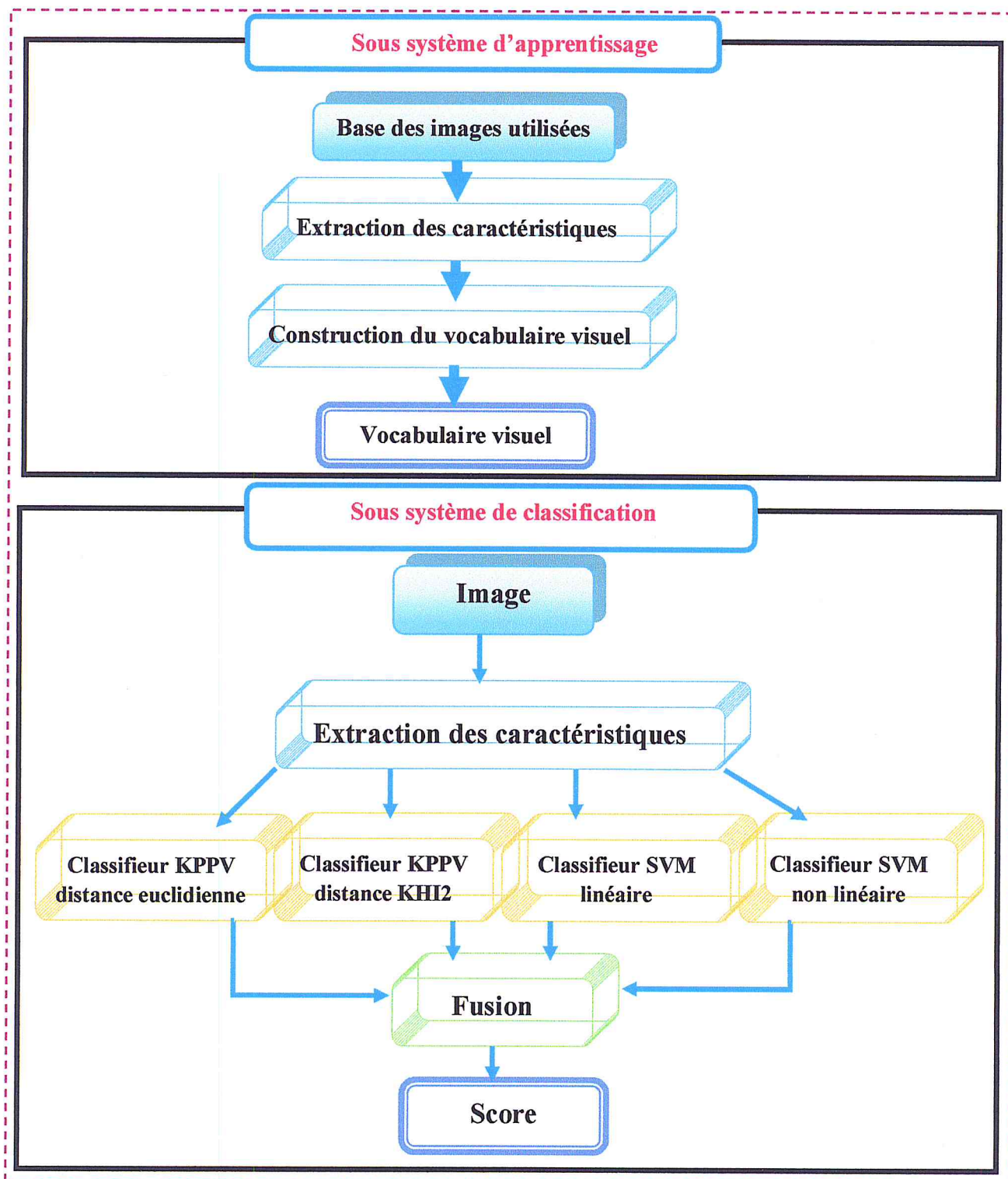
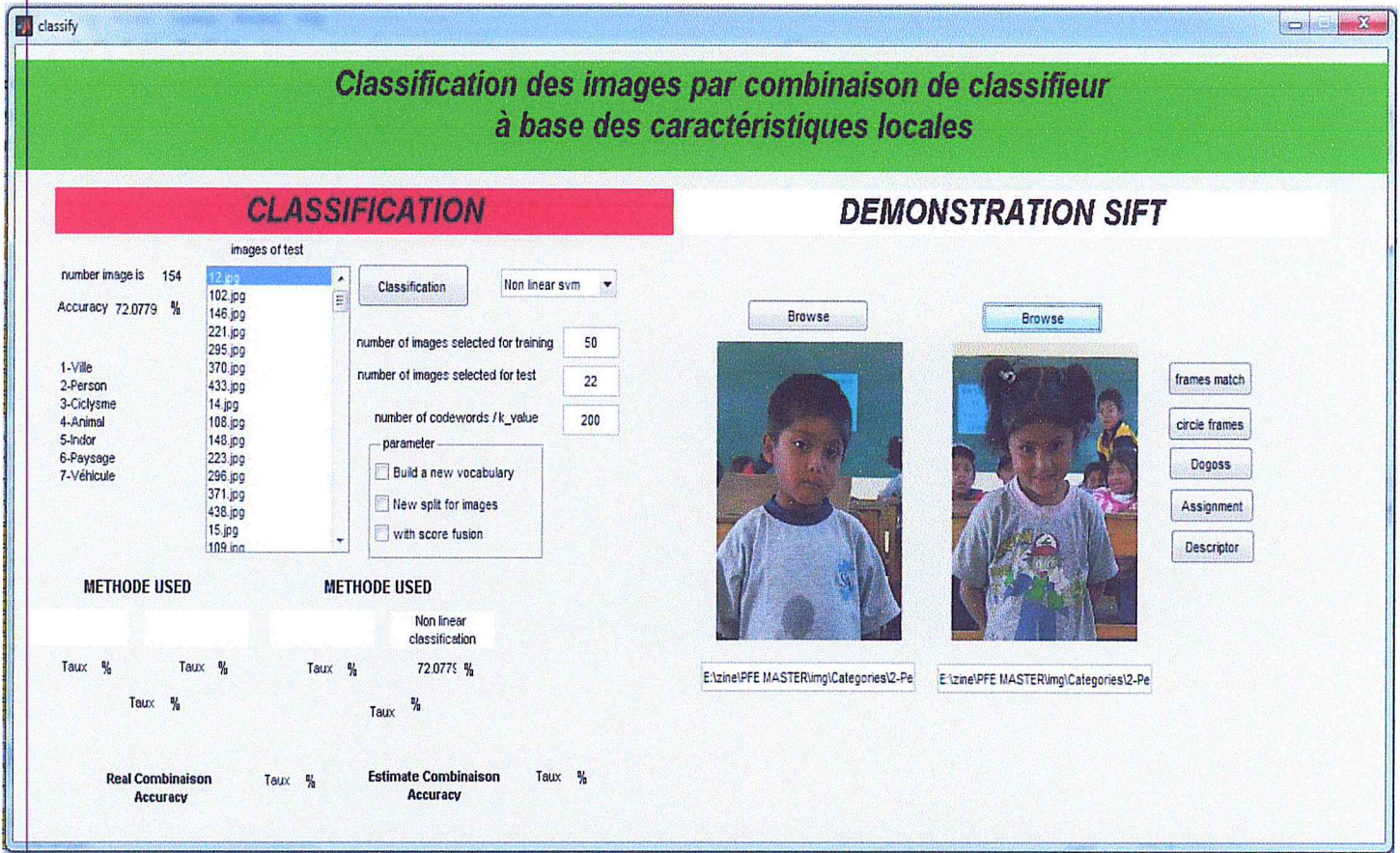


Figure 4.1 : Architecture du système proposé.

## IV.2. Interface

Voici la Figure 4.2 qui montre l'interface de notre programme :



**Figure 4.2:** montre l'interface du programme développé.

### Opération :

On peut sélectionner plusieurs opérations dont :

- ✓ Classification avec SVM linéaire.
- ✓ Classification avec SVM non linéaire.
- ✓ Classification avec KPPV distance euclidienne.
- ✓ Classification avec KPPV distance CHI-2.
- ✓ Combinaison final (lancé la combinaison des 4 classifieurs).
- ✓ Application SIFT Frames match (détecter la similarité entre les points d'intérêts).
- ✓ Application SIFT Circle frames (détecter les coordonnées x, y et r des points d'intérêts).
- ✓ Application SIFT Dogoss (appliquer une convolution par un filtre Gaussien).
- ✓ Application SIFT Assignment (assignation d'orientation aux points d'intérêts).
- ✓ Application SIFT Descriptor (Descripteur de point-clé).

**Paramètre :**

On peut choisir plusieurs paramètres dont :

- ✓ Le nombre des images d'apprentissage.
- ✓ Le nombre des images de test.
- ✓ La taille du vocabulaire.
- ✓ Construction d'un nouveau vocabulaire ou travailler avec un vocabulaire déjà existant.
- ✓ Faire la fusion des scores ou pas.
- ✓ Diviser la base des images à nouveau.

**IV.3. Base des images utilisées**

L'ensemble des données utilisées pour réaliser les expériences a été composée à partir d'un sous-ensemble d'images choisies dans la collection 2012 IAPR. La collection d'images de l'IAPR TC-12 Indice de référence contient des images naturelles prélevées à différents endroits à travers le monde, un fichier texte est associé à chaque image et contient une description textuelle de son contenu. Nous avons travaillé sur des images à partir de sept catégories de personnes présentes, des paysages, des intérieurs, des villes, des animaux, du cyclisme et des véhicules, dans notre travail, nous avons pris un sous-ensemble de 504 images classés dans 7 classes (Personne, Paysage, Animaux, Cyclisme, Véhicule, Ville, Indoor) chaque classe contient 72 images divisées en 50 images pour l'apprentissage et 22 images pour le test. (Informatik.rwth-aachen, 2008)

On obtient donc 350 images pour l'apprentissage et 154 autres pour le test.

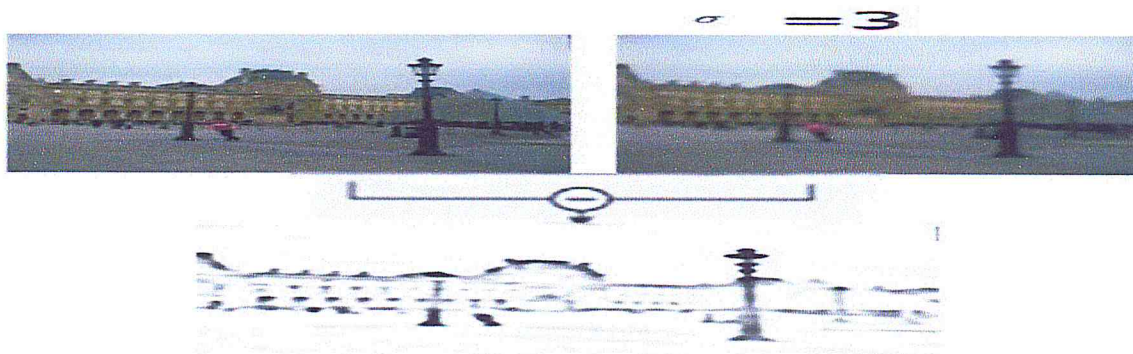
**IV.4. Extraction des caractéristiques**

Dans cette partie on applique la méthode SIFT pour chaque image indépendamment. La première étape de l'algorithme est la détection des points d'intérêt et la seconde étape est le calcul des descripteurs.

**IV.4.1. Détection d'extrema**

Pour chaque image, on applique une convolution par un filtre Gaussien pour obtenir l'image appelée différence de Gaussiens (DoG) pour localiser les points d'intérêts.

Voici une image de notre base après l'application de cette fonction :

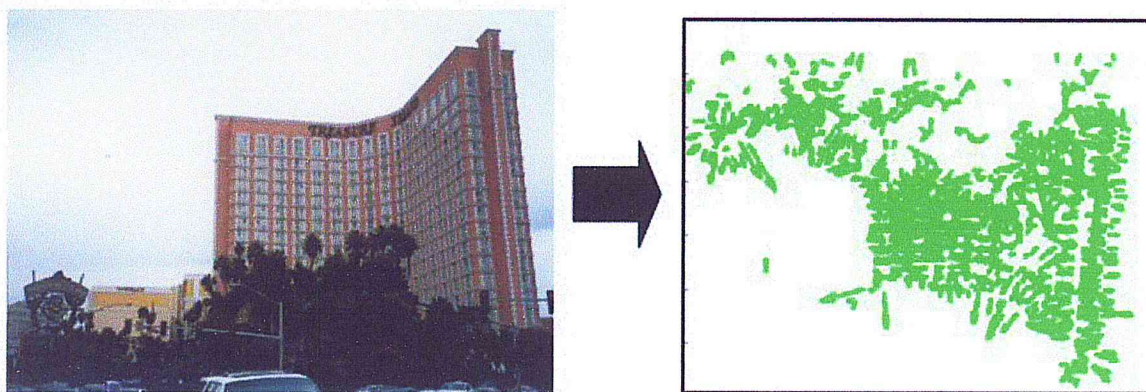


**Figure 4.3 :** Application du filtre gaussien avec  $\sigma=3$  et calcul de la différence de gaussien entre l'image originale et l'image de gaussien.

#### IV.4.2. Assignation d'orientation aux points d'intérêts

L'étape d'assignation d'orientation consiste à attribuer à chaque point d'intérêt de l'image une ou plusieurs orientations déterminées localement (sur l'image) à partir de la direction des gradients dans un voisinage autour du point. Dans la mesure où les descripteurs sont calculés relativement à ces orientations.

Voici une image de notre base après l'application de cette fonction :



**Figure 4.4 :** Détermination des orientations à partir des points d'intérêt de l'image dans le voisinage des points d'intérêt.

#### IV.4.3. Descripteur de point-clé

Pour calculer le descripteur, on estime le gradient par son amplitude (c.-à-d. sa norme), et son orientation.



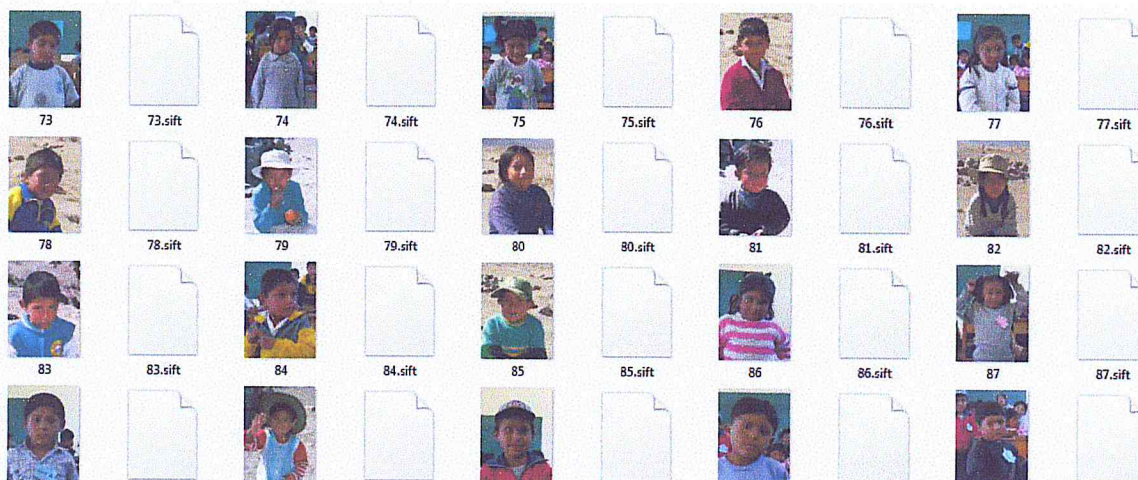


L'algorithme SIFT arrive alors à l'étape de calcul des vecteurs descripteurs, traduisant numériquement chacun de ces points d'intérêts (ou points-clés). La valeur des descripteurs de chaque image sera sauvegardée dans un fichier '.SIFT'. Voici un exemple de fichier SIFT qui contient les valeurs des descripteurs d'une image :

desc_train(1,1).sift <983x128 single>										
	113	114	115	116	117	118	119	120	121	
1	0	2	0	0	1	88	23	0	1	1
2	0	0	0	1	83	72	0	0	0	0
3	1	45	10	2	0	1	8	65	55	1
4	2	5	18	62	61	8	1	3	21	17
5	4	3	2	2	1	2	3	8	2	2
6	1	23	47	13	7	6	19	6	5	7
7	0	4	1	0	2	10	100	26	2	5
8	0	5	2	2	5	8	2	1	1	6
9	4	16	23	33	14	17	22	4	5	15
10	0	0	0	0	2	128	24	0	0	0
11	2	9	6	10	20	46	18	5	14	3
12	2	85	30	0	0	66	110	2	0	30
13	0	0	0	0	6	45	30	1	0	0
14	2	0	0	0	0	23	45	6	0	0
15	0	32	16	33	70	30	11	49	41	94

**Figure 4.5 :** Figure illustre les valeurs des descripteurs d'une image de la base.

Pour chaque image on calcule les descripteurs des points d'intérêts, donc on obtient 504 fichiers SIFT pour la base des images comme le montre la figure :



**Figure 4.6 :** Figure illustre l'ensemble des fichiers SIFT qui représente les descripteurs calculées à partir des images d'apprentissage et de teste de la base.

### IV.5. Classification des descripteurs

Dans cette étape on applique K-means pour la classification des descripteurs de chaque point d'intérêt, qui se trouve dans chaque fichier SIFT.

On obtient par l'application de cet algorithme sur les fichiers SIFT des images d'apprentissage, un vocabulaire visuel qui nous aidera à établir une comparaison entre les points d'intérêts des images de test ainsi que celles des images d'apprentissage.

Les étapes de l'algorithme sont les suivantes :

- ✓ Choisir la taille du vocabulaire (nombre de cluster) compatible avec les classifieurs utilisés pour optimiser les résultats.

Voici une table d'exemple de quelques classifieurs KPPV avec des différentes valeurs de K pour le K-means :

Valeur de K	K = 15	K = 50	K = 100	K = 200	Optimale
KPPV avec Distance Euclidienne	33.76%	46.10%	50.64%	54.94%	54.94%
KPPV avec distance CHI2	40.90%	53.83%	60.38%	63.63%	63.33%

**Tableau 4.1 :** Tableau représentant le résultat de l'application du classifieur KPPV avec différentes valeurs de K.

On remarque que le nombre de cluster idéal pour notre classifieur KPPV est 200.

Et voici une autre table d'exemple de quelques classifieurs SVM avec des différentes valeurs de K pour le K-means :

Valeur de K	K = 15	K = 50	K = 100	K = 200	Optimale
SVM linéaire	44.15%	62.33%	61.03%	64.23%	64.23%
SVM non linéaire	43.50%	63.63%	68.83%	71.25%	71.25%

**Tableau 4.2 :** Tableau représentant le résultat de l'application du classifieur SVM avec différentes valeurs de K.

On remarque que le nombre de cluster idéal pour notre classifieur SVM est 200.

On a choisi donc K = 200 (choix optimal par expérimentation) tel que l'algorithme prends 200 centres initiaux aléatoirement chaque centre représente un descripteur, En suite, il classe les descripteurs de toutes les images dans les groupe dont le centre est le plus proche, en suite recalcule les nouveaux centres pour chaque groupe et refait toute l'algorithme jusqu'à ce qu'il n'y ait aucun changement dans les groupes.

Le résultat de l'application de K-means sera l'ensemble des descripteurs classifiés dans 200 groupes.

- ✓ Enregistrer le vocabulaire visuel dans un tableau.

	113	114	115	116	117	118	119	120	121
174 5674	23.0851	7.1844	12.3617	16.3972	8.3404	14.2340	54.2057	101.3121	15.8652
175 6029	24.5588	30.8824	8.3971	8.5294	73.5074	88.5735	9.4191	6.3088	35.9412
176 0631	25.8559	21.7748	11.0541	20.1892	76.5946	55.5495	17.6036	16.6667	33.8739
177 4778	23.4889	22.9889	6.5111	5.7111	10.8889	15.2444	8.3444	8.0333	18.3667
178 7714	40.1714	79.1619	32.2857	14.0571	13.6762	14.9714	8.8952	8.9619	19.4762
179 5944	10.3427	13.6573	6.8671	4.2098	15.8601	142.1888	69.4755	12.1818	19.3636
180 8824	81.8529	72.8676	10	5.5882	15.9412	21.2206	9.0588	15.2794	37.4853
181 2612	10.6045	16.1567	33.0299	45.5522	42.8731	48.4179	34.7687	11.3284	23.7090
182 6316	69.5684	75.1579	23.2211	12.3789	9.4105	7.4211	11.6737	35.4737	55.0211
183 8263	15.7305	6.7665	6.1617	8.9641	12.5988	7.4551	7.9940	5.7784	16.8802
184 7111	14.7111	17.6963	26.0148	39.0593	68.8741	48.2296	21.3407	14.1926	19.2741
185 2342	60.1441	23.7568	12.8739	10.4234	11.2432	13.7387	36.5405	73.0450	37.8829
186 9518	44.0120	103.3614	42.2771	24.7349	26.5783	19.6024	7.4217	9.3855	31.4578
187 7886	6.0800	97.1314	37.3486	4.5314	8.7371	35.2457	15.0571	3.3714	6.3029
188 6140	3.8070	24.7281	100.3509	15.1667	2.0263	1.2281	1.7632	1.9912	0.0088

**Figure 4.7 :** Figure illustre le résultat de l'application de l'algorithme de K-means sur chaque fichier descripteur.

## IV.6. Représentation de chaque image par histogramme

Dès que chaque point d'intérêt est assigné à un mot visuel (Application du K-Means), on pourra calculer un histogramme qui montre combien de points clés appartenant à chaque classe, et ce dans chaque image.

Voici l'exemple d'un histogramme de l'image \1-person\100.jpg qui montre la distance minimale de chaque point d'intérêt et cluster, et ce dans l'image en question.

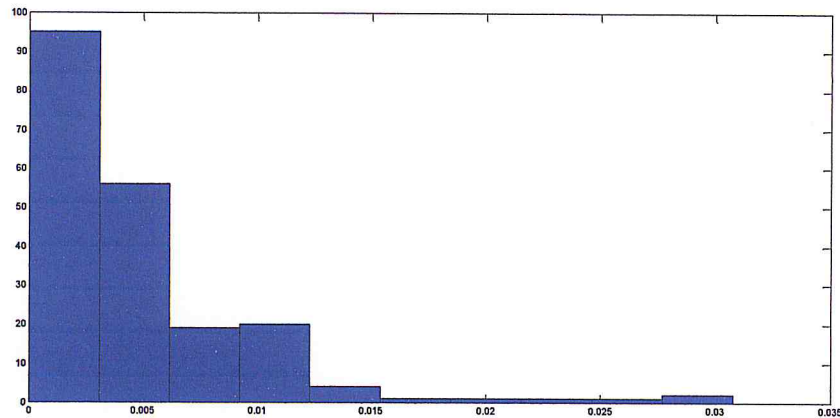


Figure 4.8 : Schématise la distance minimale entre chaque point d'intérêt et cluster d'une image.

## IV.7. Classification

### IV.7.1. Application de KPPV

Dans cette étape, on a utilisé un classifieur KPPV, on a changé en alternance la distance utilisée, on y a procédé grâce à la distance Euclidienne et CHI2 (KHI2) entre les points d'intérêts des images de test obtenues par les fichiers SIFT de ces dernières, et les points déjà classés en utilisant la méthode K-means (vocabulaire visuel).

On prend les 154 fichiers SIFT de la partie test et on les classifie comme suit :

On prend le premier descripteur de test qui représente la première image de test et on calcule la distance euclidienne/CHI2 de ce vecteur par rapport aux 200 vecteurs d'apprentissage (200 cluster) stockés précédemment et ainsi de suite pour toutes les images de test. Le système va calculer les distances et choisir les K plus proches images d'apprentissage (dont on connaît leurs classes) pour chaque image de test.

Si par exemple on choisit la valeur  $K = 5$ , le programme calcule la distance euclidienne/CHI2 de tous les vecteurs de la base d'apprentissage par rapport aux vecteurs

tests, il prend en considération que les distances euclidiennes/CHI2 des 5 images voisines parce qu'on a spécifié le nombre des voisins à cinq et le programme attribue à chaque image sa classe. Si par exemple notre image de test est de la classe PERSONNE, et qu'on a spécifié  $K=5$ , lorsque notre système calcule ces cinq plus proches voisins, ce n'est forcément que ces voisins sont tous des images de la classe PERSONNE, il se peut que sauf les 3 premiers vecteurs représentent les images de la classe PERSONNE et les deux autres vecteurs qui restent représentent une autre classe des images tel que ANIMAUX ou CYCLISME, alors on a utilisé le vote majoritaire pour les 5 résultats pour donner une seule classe dominante. Ensuite on calcule le taux de classification correcte de la classification de toutes les images test en utilisant la formule suivante :

$$\text{Taux de classification} = \frac{\text{Nombre des images test bien classés}}{\text{Nombre des images test}}$$

On prend pour le KPPV :  $K=7$  équivalant au nombre de classe, voici un exemple pour une image de test avec des différentes valeurs de  $K$  (voisin).

Voisin	K1	K2	K3	K4	K5	K6	K7
<b>KPPV distance</b> L-2	0.0673	0.0680	0.0716	0.0716	0.0740	0.0748	0.0757
<b>KPPV distance</b> CHI-2	0.1839	0.1862	0.1918	0.1998	0.2178	0.2244	0.2245

**Tableau 4.3 :** Tableau représentant les distances (euclidiennes et khi-2) d'une image avec ces voisins après l'application du KPPV.

Dans le cas du KPPV on obtient que les distances des voisins et pas les degrés d'appartenance qu'on a besoin pour faire la combinaison, donc il va falloir qu'on fasse une conversion des distances à des degrés d'appartenance de l'image à chaque classe.

On va procéder comme suit :

- ✓ On vérifié les classes des voisins.

Le tableau suivant montre la classe de chaque voisin :

Voisin	K1	K2	K3	K4	K5	K6	K7
<b>KPPV L-2 classe</b>	1	1	1	6	7	7	6
<b>KPPV CHI-2 classe</b>	1	1	1	1	6	6	6

**Tableau 4.4 :** Tableau représentant les classes des voisins de l'image de test.

✓ On fait le vote pour chaque classe pour déduire le degré d'appartenance.

On calcule le nombre d'occurrence de chaque classe et on divise par le nombre des classes.

Exemple : la classe 1 est répéter 3 fois donc  $3/7=0.4286$  où l'image a une possibilité de 42.86% d'appartenir à la classe 1.

Le tableau suivant montre les résultats du vote :

Classe	1	2	3	4	5	6	7
<b>KPPV L-2 degrés</b>	0.4286	0	0	0	0	0.2857	0.2857
<b>KPPV CHI-2 degrés</b>	0.57144	0	0	0	0	0.42858	0

**Tableau 4.5 :** Tableau représentant les degrés d'appartenance d'une image à des différentes classes.

### IV.7.2. Application du SVM

La boîte noire du SVM est appelée en utilisant la bibliothèque libsvm-3.11.

#### IV.7.2.1. SVM linéaire

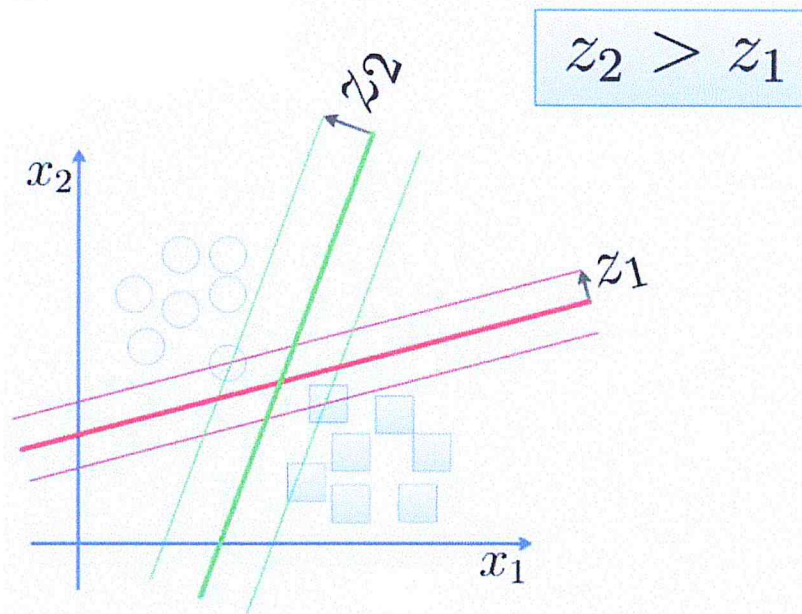
Le but est de générer des vecteurs qui, séparent chaque deux groupe de différents points d'intérêt, ils classent donc tous les descripteurs d'apprentissage en deux classes, puis on choisit les vecteurs qui ont une marge maximale entre chaque deux classes.

À l'aide de la fonction d'apprentissage (svmtrain), on enregistre les paramètres pour notre Classifieur (linéaire SVM), et à la fin on choisit le paramètre optimal pour créer notre model SVM.

En chargeant les images de test, les points clefs de ces dernières vont être comparés aux modèles déjà existants en utilisant la fonction `svmpredict`.

Chaque image de test aura une classe estimée par la fonction de similarité.

La figure 4.8 montre un vecteur à vaste marge qui sépare deux classes différentes.



**Figure 4.9 :** Un dessin montre que le choix du vecteur vert est dû à la vaste marge entre les deux classes.

Le classifieur SVM donne des résultats directs (degré d'appartenance de l'image à chaque classe).

Après l'application du classifieur SVM linéaire on obtient les résultats suivant pour notre image de test :

classes	1	2	3	4	5	6	7
les degrés	0.3045	0.0715	0.0426	0.1773	0.0184	0.3001	0.0856

**Tableau 4.6 :** Tableau représentant les degrés d'appartenance d'une image de test à des différentes classes par l'application du SVM linéaire.

#### IV.7.2.2. SVM non linéaire

La méthode SVM non linéaire fonctionne de cette manière suivante :

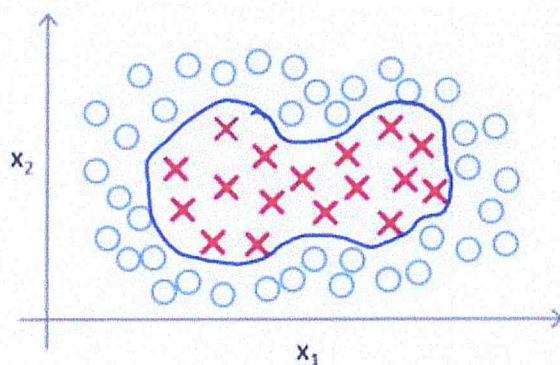
En premier temps :

- ✓ On charge le vocabulaire visuel (les images d'apprentissage) dans un tableau *Ktrain*.
- ✓ On fait un apprentissage à l'aide de la fonction d'apprentissage (*svmtrain*) qui enregistre les paramètres pour notre Classifieur (non linéaire SVM).
- ✓ On choisit le paramètre optimal pour créer notre model SVM.

En deuxième temps :

- ✓ On charge les descripteurs des images de test dans un tableau *Ktest*.
- ✓ On fait appel à la fonction de similarité (*svmpredict*), qui nous mène à une comparaison entre les deux tableaux, *Ktrain* et *Ktest* en se basant sur notre model.
- ✓ Après cette comparaison, chaque image de test aura une image sœur dans les images d'apprentissage et sera étiquetée par la même classe de l'image sœur.

#### frontière non linéaire



Predict  $y = 1$  if

$$\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1 x_2 + \theta_4 x_1^2 + \theta_5 x_2^2 + \dots \geq 0$$

Figure 4.10 : Montre la classification non linéaire d'un classifieur SVM.



Après l'application du classifieur SVM non linéaire on obtient les résultats suivants pour notre image de test :

classes	1	2	3	4	5	6	7
Degrés	0.6289	0.0252	0.0170	0.0843	0.0137	0.1675	0.0634

**Tableau 4.7 :** Tableau représentant les degrés d'appartenance d'une image de test à des différentes classes par l'application du SVM non linéaire.

### IV.8. La combinaison de plusieurs Classifieurs

#### IV.8.1. Fusion des scores

Le principe de la fusion des scores de plusieurs Classifieurs est le fait de connaître les résultats et de les comparer ensuite avec les résultats trouvés pour savoir après si notre combinaison (fusion) elle a atteint le taux maximal qu'on recherche.

Nous avons utilisé 4 classifieurs (KPPV avec distance euclidienne/chi-2 et SVM linéaire/non linéaire) pour la classification et on fait l'union pour les résultats des méthodes utilisées, pour calculer le résultat de la fusion des scores.

Voici une table de quelques exemples de fusion des scores :

<b>KPPV avec Distance Euclidienne</b>	<b>NON Linaire SVM</b>	<b>Linaire SVM</b>	<b>KPPV avec distance CHI2</b>	<b>Classifieurs</b>
<b>54.5455%</b>	<b>71.4286%</b>	<b>58.4416%</b>	<b>63.6364%</b>	<b>Taux</b>
<b>85.7143%</b>				<b>Fusion des scores</b>

**Tableau 4.8 :** Tableau qui représente la fusion des scores entre plusieurs classifieurs.

On voit dans le tableau que les résultats sont faibles avec l'application d'un seul classifieur, mais lors de l'application d'une fusion entre les scores des classifieurs on remarque qu'il y a une bonne amélioration des résultats où le taux de reconnaissance augmente à 85.71% (+14.28%).

On déduit que lorsqu'on va appliquer la combinaison (fusion) normalement elle doit donner des améliorations comme celle de la fusion des scores.

## Conclusion générale

Nous avons proposé dans ce travail un système des images. Il s'articule essentiellement autour d'une partie essentielle celle d'extraction de caractéristique.

Le classifieur est l'élément adaptatif du système de reconnaissance. Pour fonctionner, il doit, au préalable, avoir été réglé par apprentissage. L'apprentissage consiste à donner au classifieur des exemples des images, en lui indiquant, à chaque fois la classe de l'image (un ensemble de données étiquetées que l'on nomme base d'entraînement c'est à dire apprendre la description d'objets a partir d'une base d'exemples).

Le taux de reconnaissance obtenu n'est pas très satisfaisant, lors de l'utilisation de la combinaison de plusieurs classifieurs, comme dans tout logiciel, des améliorations restent toujours à apporter.

On voit comme perspectives :

- ✓ L'optimisation des résultats peut être proposée par le prétraitement.
- ✓ Utilisation de la segmentation des images pour extraire les caractéristiques des images.
- ✓ Utilisation d'un autre type de caractéristique et par combinaison de plusieurs différents classifieurs exploitant ainsi les avantages de chacun par rapport aux autres et de choisir les bons paramètres pour ces classifieurs.
- ✓ Utilisation d'une autre base des images plus réelles telles que les images satellitaires, médicales, etc.

## Bibliographie

- (2008, Dec 18). Récupéré sur Informatik.rwth-aachen: <http://www-i6.informatik.rwth-aachen.de/imageclef/resources/iaprtc12.tgz>
- (2014, mai 22). Récupéré sur wikipedia: <http://fr.wikipedia.org/wiki/Segmentation>
- (2014, mai 17). Récupéré sur WIKIPEDIA:  
[http://fr.wikipedia.org/wiki/Reconnaissance\\_optique\\_de\\_caract%C3%A8res](http://fr.wikipedia.org/wiki/Reconnaissance_optique_de_caract%C3%A8res)
- Achi, S., & Azizi, N. (JUIN 2013). *La détection de polarité d'opinions dans les journaux arabes par KPPV*. Mémoire de Master TIVA.
- Afsar, M. U., & Arif, J. K. (2010). *A Pruned Fuzzy k-Nearest Neighbor Classifier with Application to Electrocardiogram Based Cardiac Arrhythmia Recognition*. Islamabad, Department of Computer & Information Sciences, Pakistan.
- Alain, P. (2009, Juin.). *Contributions à la Classification Sémantique d'Images*. Thèse de doctorat. Ecole Centrale de Lyon.
- Ameur, Z., & all, a. (March 2005). *Segmentation d'images couleur par fusion de régions*. TUNISIA: 3rd International Conference : Science of Electronic,.
- BELAID, A., & y, B. (1992). *"Reconnaissance des formes : Méthodes et applications"*. InterEditions.
- Berrani, & all, a. (2002). *les approches de classifications*.
- Bondugula, R. O., & Duzlevski, D. X. (2010). *Profiles and fuzzy k-nearest neighbor algorithm For protein secondary structure prediction*. Comumbia.
- Byngmoon, K., & Panagiotis, T. (2006). *Image segmentation on cell-center sampled Quadtree and Octree*. Atlanta, USA: Georgia Institute of Technology, "NVIDIA,USA".
- Canny, J. (1986, dec). *décteur de contours de Canny* (Vol. vol. 8). IEEE Transactions on Pattern Analysis and Machine Intelligence.
- Carl, F. (2004, mars 18). *Classification Supervisée avec Rejet (I) Approche Probabiliste*. Laboratoire d'Informatique – Image – Interaction, EA 2118 Université de La Rochelle(séminaire de l'IRIT).
- Chu, T. Q. (2005, Juillet). *Classification automatique des textes vietnamiens*. Institut de la Francophonie pour l'informatique.
- Cocquerez, J.-P., & Philipp, S. (2005). *Analyse d'images: filtrages et segmentation*. Masson.
- David, G. L. (2002). *Local feature view clustering for 3D object recognition*. IEEE Conference on Computer.
- David, G. L. (2004, mai). *Distinctive image features from scale-invariant keypoints* (Vol. vol. 60). International Journal of Computer Vision.

- Devars, J., & M, M. (2002/2003). *Cours traitement d'images* .
- Dubuisson, B. (2006). "*Diagnostic et reconnaissance des formes*". Hermès.
- Gevers, T., & Smeulder, W. M. (2004, fev). *Content-based Image Retrieval: An Overview, from the book Emerging Topics in Computer Vision*.
- Guillaume, W. (2011). *Classification spectrale semi-supervisée. Application à la surveillance de l'ecosystème marin*. Ecole Doctorale Sciences pour l'Ingénieur.
- Hamroune, F., & Ait Hamou, y. (2006). *Segmentation d'image par contours actifs géodésique(CAG)*. Alger: Thèse pour obtention du diplôme d'ingénieur d'état en informatique(I.N.I).
- Hamza, C., & Ibrahim, T. (2011/2012). *Segmentation des masses dans les images mammographiques*. Tlemcen: Memoire de fin d'etude Université Abou Bakr Belkhir .
- Haralick, R. (2004, apr). *Ridges and Valleys on Digital Images, "détection de crêtes"* (Vol. vol. 22).
- Haralick, R., & Shanmugam, K. (2003). *Textural features for image classification, "IEEE Transactions on Systems, Man, and Cybernetics"*.
- Henri, M. (2013). *La segmentation par régions"chapitre 3"*.
- Hervé, L. B. (2009). *Recherche d'information dans les images*. ERMITES.
- Hong L, P. S. (Oct 1999). "*Can Multibiometrics Improve Performance*" (éd. Proceedings Auto ID' 99).
- Houassine, C. (2012). *Segmentation d'images par une approche biomimétrique hybride*. boumerdes: Memoire de Magister Université M'Hamed Bougara.
- Jean, Y. T. (2010). *Classification et Reconnaissance des Formes, "Analyse et Synthèse de l'Information"*.
- Jeffrey, & David, L. (2001). *Shape indexing using approximate nearest-neighbour search in high-dimensional spaces*. San Juan, Porto-Rico: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition.
- Johannes, B., & Niko, S. (2007). *Comparing several implementations of two recently published feature detectors*. Toulouse: Proceedings of the International Conference on Intelligent and Autonomous Systems.
- John, R. S., & Chang, S.-f. (2007, mars). *An Image and Video Search Engine for the World-Wide Web , In Proc. SPIE Storage and Retrieval for Image and Video Databases*.
- Julesz, B. (2002, sept). *Textons, the elements of texture perception, and their interactions*.
- Keller, J. M., & all, &. (1999). *A Fuzzy K-Nearest Neighbor algorithm, "IEEE transaction on systems, man, and cybernetics"*.
- Krystian, M., & all, a. (2005, fev). *comparaison de détecteurs de régions d'intérêt affines (elliptiques)*. International Journal of Computer Vision.

- Krystian, M., & Cordelia, S. (octobre 2005). *A performance evaluation of local descriptors* (Vol. vol. 27). IEEE Transactions on Pattern Analysis and Machine Intelligence.
- Lamri, L. (avrile 2006). *Approche exploratoire sur la classification appliquée aux images*. Québec: Memoire présenté à l'université du Québec.
- Lindeberg, T. (2001, nov). *Edge detection and ridge detection with automatic scale selection* (Vol. vol. 30). International Journal of Computer Vision.
- Lindeberg, T. (2001). *Feature detection with automatic scale selection* (Vol. vol. 30). International Journal of Computer Vision.
- Lindeberg, T. (2001, jan). *laplacien et DoH pour la détection de régions d'intérêt* (Vol. vol. 30). International Journal of Computer Vision.
- Liu, Y. D., Zhang, & Lu, S. G. (2004, décembre). *Region-based image retrieval with perceptual colors*.
- Lowe, D. (1999, sep). *le détecteur SIFT, incluant une détection de régions d'intérêt circulaires* (Vol. vol. 60). International Journal of Computer Vision.
- Marr, D., & Hildreth, H. (1995). *Theory of edge detection*. Proceedings of the Royal Society of London. (mars 2014, mars 08). Récupéré sur wikipedia: [http://fr.wikipedia.org/wiki/Segmentation\\_d'image](http://fr.wikipedia.org/wiki/Segmentation_d'image)
- Matas, J. O., & Chum, M. U. (2002, avr). *le détecteur de régions d'intérêt MSER* (éd. British Machine Vision Conference).
- Materka, A., & Strzelecki, M. (2005). *Texture Analysis Methods - A review*. Bruxelles: Technical University of Lodz, Institute of Electronics.
- Micolajczyk, K., & Schmid, C. (Oct 2005). *A performance evaluation of local descriptors*. Pattenr analysis and machine intelligence, IEEE transaction on, vol 27,no.10,pp.1615,1630.
- Mikolajczyk, K., Schmid, & C. (2005, Oct). *A performance evaluation of local descriptors "Pattern Analysis and Machine Intelligence, IEEE Transactions on"*.
- Mkaouar, M., & Lepage, R. (2005). *Extraction de caractéristiques*.
- Mokhtarian, F., & Mackworth. (2006, January). *Scale Based Description and Recognition of Planar Curves and Two-Dimensional Shapes*.
- (NOV 2013). Récupéré sur wikipedia: [http://fr.wikipedia.org/wiki/Segmentation\\_d%27image](http://fr.wikipedia.org/wiki/Segmentation_d%27image)
- Paul, S., Saad, A., & Mubarak, S. (2007). *A 3-dimensional SIFT descriptor and its application to action recognition*. dans Proceedings of the 15th International Conference on Multimedia.
- Rouquet, C., Bouton, P., & Toumizak, R. (2000). *etude comparative des stratégies de segmentation non supervisée en region par chmaps de Markov*. Traitement du signal.
- Salvatotre, T., Nghyen, T.-O., & Gérald, M. (2010). *Une méthode de binarisation hiérarchique floue*. Thèse doctorat. institut de la francophonie pour l'informatique(IFI).

Stricker, M., & M, O. (2009, Feb). *Similarity of color images*. In *In SPIE Conference on Storage and Retrieval for Image and Video Databases III*.

Swain, M., & Ballard, D. (1991). *Color Indexing*.

Tou, T., & Gonzalez, C. (1997). *Pattern recognition principles julius*. Addison-welsey publishing company.

Unser, M. (1986, oct). *Sum and difference histograms for texture classification*.

Vladimir, V. (1992). *le développement d'une théorie statistique de l'apprentissage*.

Wezka, J., Dyer, C., & Rosenfeld, A. (1996). *A comparative study of texture measures for terrain classification, "IEEE Transactions on Systems, Man, and Cybernetics"*.

Yan, C., Nils, H., & Thorsten, T. (2009). *Scale invariant feature transform with irregular orientation histogram binning*. Halifax (Canada): Proceedings of the International Conference on Image Analysis and Recognition.

