

N° d'ordre : --/2005-M/IN

République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique  
Université des Sciences et de la Technologie Houari Boumédiène  
USTHB – ALGER

Faculté d'Electronique et d'Informatique  
**Département d'Informatique**

## **MEMOIRE**

Présenté pour l'obtention du diplôme de  
**Magister en Informatique**

Spécialité : Intelligence Artificielle et Bases de Données Avancées

Par

**Mr TAFER MEHDI**

Sujet

**Indexation d'une base de descripteurs  
d'images écrits suivant le langage LWDS  
(Language for Writing Descriptors of Shapes)**

Soutenu Le 25-Mai-2005, devant le jury composé de :

|                    |                       |       |              |
|--------------------|-----------------------|-------|--------------|
| Mme A. Aissani     | Professeur            | USTHB | Présidente   |
| Mme Z. Ali Mazighi | Maître de Conférences | USTHB | Examinatrice |
| Melle F. Souami    | Maître de Conférences | USTHB | Examinatrice |
| Mr. K. Achour      | Chargé de Recherche   | CDTA  | Examineur    |
| Mr. S. Larabi      | Maître de Conférences | USTHB | Rapporteur   |

*" L'arbre de la liberté est nourri par le sang des Patriotes et  
des tyrans "*

THOMAS JEFFERSON

## **Résumé**

*La représentation d'images est l'un des problèmes importants pour la reconnaissance d'objets en vision artificielle. En effet, il est nécessaire de disposer d'une base d'images modèles représentant les différentes vues de l'objet. La taille de la base de données augmente considérablement en fonction du nombre d'objets modèles à traiter. Il est donc impératif de réfléchir sur une meilleure représentation de l'image qui garantit un espace mémoire réduit, une facilité de calcul et de traitement.*

*Dans cet axe de recherche, une méthode pour la description de formes en tenant compte du contour externe et des contours internes a été proposée au laboratoire L.R.I.A. Un langage noté LWDS «Language for Writing Descriptors of Shapes» a été aussi proposé pour l'écriture textuelle de ces descriptions en suivant le format XML.*

*En disposant d'une base de données images représentées moyennant le langage LWDS, nous proposons dans ce mémoire une méthode pour l'indexation de ces descripteurs. L'étude est enrichie par une implémentation sous Visual Basic pour valider la proposition*

**Mots clefs :** Descripteur d'images - Silhouette - Forme - Indexation

# **TABLE DES MATIERES**

|                            |   |
|----------------------------|---|
| INTRODUCTION GENERALE..... | 1 |
|----------------------------|---|

---

## **CHAPITRE 1**

### **METHODES DE REPRESENTATION DES FORMES ET BASES DE DONNEES IMAGES**

|                                                               |    |
|---------------------------------------------------------------|----|
| 1.1. INTRODUCTION.....                                        | 5  |
| 1.2. TECHNIQUES DE REPRESENTATION DE FORMES.....              | 5  |
| 1.2.1 Représentations basées sur les contours.....            | 6  |
| 1.2.1.1 Introduction.....                                     | 6  |
| 1.2.1.2 Le descripteur (CSSD) .....                           | 6  |
| 1.2.1.3 Descripteur de Fourier (FD) .....                     | 8  |
| 1.2.2 Représentations basées sur les régions.....             | 9  |
| 1.2.2.1 Introduction.....                                     | 9  |
| 1.2.2.2 Le descripteur grille (Grid Descriptor (GD)) .....    | 10 |
| 1.2.3 Autres primitives pour la description d'image.....      | 10 |
| 1.3. SYSTEMES DE RECHERCHE D'IMAGES.....                      | 11 |
| 1.3.1 Notion de recherche d'image dans une base d'images..... | 11 |
| 1.3.2 Quelques systèmes CBIR.....                             | 12 |
| 1.3.3 Méthodes d'indexation d'images .....                    | 15 |
| 1.3.2.1 Méthodes basées sur le contenu.....                   | 15 |
| 1.3.2.2 Méthodes basées sur l'apparence locale.....           | 16 |
| 1.3.2.3 Méthodes basées sur les relations spatiales.....      | 18 |
| 1.4. CONCLUSION.....                                          | 18 |

---

## **CHAPITRE 2**

### **DESCRIPTION TEXTUELLE D'IMAGES**

|                                                                                      |    |
|--------------------------------------------------------------------------------------|----|
| 2.1 INTRODUCTION.....                                                                | 20 |
| 2.2 DESCRIPTION TEXTUELLE DE SILHOUETTES .....                                       | 20 |
| 2.2.1 Présentation de la méthode.....                                                | 20 |
| 2.2.2 Le langage LWDOS pour l'écriture textuelle des descripteurs de silhouettes ... | 23 |
| 2.2.3 Propriétés du descripteur LWDOS.....                                           | 26 |
| 2.3 METHODE DE DESCRIPTION TEXTUELLE DE FORMES .....                                 | 27 |
| 2.3.1 Présentation de la méthode.....                                                | 27 |
| 2.3.2 Le langage LWDS pour l'écriture textuelle des descripteurs de formes.....      | 30 |

|                      |    |
|----------------------|----|
| 2.4. CONCLUSION..... | 33 |
|----------------------|----|

---

### **CHAPITRE 3**

#### **INDEXATION DE DESCRIPTEURS DE FORMES ECRITS SUIVANT LE LANGAGE LWDS**

|                                                                                       |    |
|---------------------------------------------------------------------------------------|----|
| 3.1 INTRODUCTION.....                                                                 | 35 |
| 3.2 NOTION DE BASE DE DONNEES IMAGES.....                                             | 35 |
| 3.3 INDEXATION DE DESCRIPTEURS DE SILHOUETTES ECRITS SUIVANT LE<br>LANGAGE LWDOS..... | 37 |
| 3.3.1 Premier niveau d'indexation de descripteurs LWDOS .....                         | 37 |
| 3.3.2 Second niveau d'indexation de descripteurs LWDOS .....                          | 40 |
| 3.3.3 Exemple d'indexation de descripteurs LWDOS .....                                | 41 |
| 3.4 INDEXATION DE DESCRIPTEURS DE FORMES ECRITS SUIVANT LE<br>LANGAGE LWDS.....       | 43 |
| 3.4.1 Introduction.....                                                               | 43 |
| 3.4.2 Eléments pour l'indexation de descripteurs LWDS.....                            | 43 |
| 3.4.3 Organisation de la base de données.....                                         | 48 |
| 3.5 CONCLUSION.....                                                                   | 51 |

---

### **CHAPITRE 4**

#### **EXPERIMENTATION**

|                                                              |    |
|--------------------------------------------------------------|----|
| 4.1. DONNEES D'EXPERIMENTATION.....                          | 53 |
| 4.2. TEST EXPERIMENTAUX.....                                 | 54 |
| 4.2.1 Mise en œuvre d'une base de descripteurs LWDOS.....    | 54 |
| 4.2.2 Mise en œuvre d'une base de descripteurs LWDS.....     | 56 |
| 4.3 LOGICIEL DE GESTION DE LA BASE DE DESCRIPTEURS LWDS..... | 60 |
| 4.4 CONCLUSION.....                                          | 63 |
| CONCLUSION GENERALE.....                                     | 65 |
| ANNEXE.....                                                  | 66 |
| REFERENCES.....                                              | 70 |

# **INTRODUCTION**

L'homme possède la faculté de décrire, de définir, de donner un sens aux objets qui l'entoure et cela d'une façon aussi naturelle qu'il n'arrive pas, ou presque à l'expliquer. Il possède des outils très performants à savoir une interface visuelle, une faculté d'interprétation très performante, une mémoire suffisante pour sauvegarder une base de données immense.

Avec la naissance des machines de calcul de plus en plus sophistiquées, un certain nombre de scientifiques s'est attaqué au problème de la vision en essayant de répondre à la question: «Est-il possible de construire un modèle informatique pour la perception visuelle?»

La vision artificielle est donc un processus de traitement de l'information. L'entrée d'un système de vision est constituée par une séquence d'images. Le système lui-même apporte un certain nombre de connaissances. La sortie est une description de l'entrée en terme d'objets et de relations entre ces objets.

Aujourd'hui, la vision est appliquée dans plusieurs problèmes tel que la navigation robotique, la reproduction du mouvement humain, la détection et identification d'intrus, les bases de données médicales, iconographiques, photographies d'art, d'articles archéologiques et bien d'autres domaines variés.

La reconnaissance d'objets est l'un des objectifs de la vision artificielle. C'est un problème difficile en raison du nombre d'images modèles dont on dispose. En effet, à un objet peut être associé des centaines d'images différentes obtenues sous des angles de vues différentes. Il s'agit alors, à partir d'une image requête de trouver dans la base de données l'image ou les images les plus similaires à l'image requête.

Dépendant de l'application, le nombre d'images modèles nécessaires peut varier d'une centaine d'images à des milliards d'images. La mise en œuvre d'une base de données images exige la prise en compte des contraintes suivantes :

- Optimiser l'espace mémoire occupé par chaque image modèle. La solution consiste alors à ne stocker qu'une description de celle-ci.
- Accéder rapidement et directement à l'image modèle à partir d'une requête. L'indexation de la base de données est une solution à ce problème.
- La représentation d'image doit vérifier la facilité de calcul, de traitement

De nombreuses études ont été réalisées sur les techniques de représentation et d'indexation de formes à partir d'images. La méthode la plus référencée et utilisée dans le standard MPEG7 est celle proposée par Mokhtarian [Mokhtarian85]. Cette méthode se base sur la description de l'image d'un objet à partir de la variation de la courbure de sa frontière à

plusieurs échelles. Dans le but d'une description d'images pouvant servir en multimédia et en vision artificielle, une méthode pour la description textuelle d'images a été développée au laboratoire **L.R.I.A.** (Laboratoire de **R**echerche en **I**ntelligence **A**rtificielle de l'U.S.T.H.B.). Cette méthode se base sur la décomposition de la silhouette en éléments simples : partitions, lignes de jonctions, lignes de disjonctions. Chacune de ces éléments est ensuite décrite géométriquement. Un langage noté **LWDOS** est ensuite développé pour l'écriture textuelle de ces descriptions.

L'image d'un objet contient souvent des régions internes. Une méthode est développée pour la description des formes. Un langage noté **LWDS** a été développé pour l'écriture de ces descriptions.

Le travail demandé consiste au développement de méthode pour l'indexation des descripteurs **LWDS**.

Concrètement ce mémoire est scindé en quatre chapitres :

Dans le premier chapitre nous présentons d'abord un état de l'art des principales méthodes proposées dans la littérature pour la description de formes.

Dans le second chapitre, nous étudions les méthode développée au laboratoire **L.R.I.A.** (Laboratoire de **R**echerche en **I**ntelligence **A**rtificielle de l'U.S.T.H.B.) pour la représentation des silhouettes et des formes. Nous étudions aussi les langages **LWDOS**, **LWDS** développés au laboratoire **L.R.I.A.** permettant d'écrire textuellement les descriptions des silhouettes et des formes.

Cet état de l'art est suivi par notre contribution dans le chapitre trois, à savoir, une méthode pour l'indexation d'une base de descripteurs de formes écrits suivant le langage **LWDS** « Language for **W**riting **D**escriptors of **S**hapes ». Une validation de la méthode est ensuite présentée dans le chapitre quatre sur des images artificielles.

# **CHAPITRE 1**

## **METHODES DE REPRESENTATION DES FORMES ET BASES DE DONNEES IMAGES**

## **1.1. INTRODUCTION**

La description d'images est un thème de recherche qui a été abordé durant ces trente dernières années. Au début, ces recherches ont été conduites uniquement pour la modélisation d'objets en vue de leur reconnaissance. En conséquence, les techniques de description ont principalement ciblé des applications particulières. L'efficacité et l'exactitude sont le principal intérêt visé par ces techniques. Par contre, aujourd'hui, la description d'images est devenue une nécessité en raison du développement des technologies de communication et du multimédia. A titre d'exemple, nous citons les applications multimédia de recherche d'images par leur contenu (CBIR : Content-Based Image Retrieval) où l'efficacité est envisagée de même que l'exactitude due à la demande de recherche on-line. C'est ainsi que dans le développement du standard MPEG-7, six exigences ont été définies pour mesurer les performances d'un descripteur de forme:

- Bonne précision de recherche,
- Primitives utilisées compacts,
- Généralité d'application,
- Faible complexité de calcul,
- Performance dans la recherche

Une représentation hiérarchique du global vers le détaillé.

Nous présentons dans les sections suivantes les principales méthodes de description d'images ainsi qu'un aperçu sur les systèmes de recherche d'images faisant appel à la notion de base de données images.

## **1.2. TECHNIQUES DE REPRESENTATION DE FORMES**

Les différentes techniques de représentation de forme proposées dans la littérature peuvent être classées en deux classes [80]: celles qui sont basées sur les contours et celles qui sont basées sur les régions. Les primitives des formes sont extraites à partir des contours uniquement ou à partir de toute la région de la forme (voir figure 1.1). Les différentes méthodes de chaque classe sont répertoriées en deux catégories : globale si la forme est représentée dans son ensemble, et structurelle si la forme est représentée par parties. Il est possible de séparer les différentes méthodes en une seconde hiérarchie en tenant compte du domaine (primitives ou spatial).

## 1.2.1 Représentations basées sur les contours

### 1.2.1.1 Introduction

Les techniques de description de formes basées sur les contours sont importantes. Elles impliquent des descripteurs moins coûteux en temps de calcul et en stockage comparées aux techniques basées sur les régions. La description utilisant ces techniques conviennent à des applications où les contours des formes sont disponibles. Dans la littérature, ces méthodes sont plus connues que celles basées sur les régions, cela est dû à leurs simple implémentation et à leurs propriétés. Ces techniques sont aussi appliquées avec succès à plusieurs applications telle que la reconnaissance d'objets, codage de formes, reconnaissance des caractères. Nous présentons les deux plus importants descripteurs de formes basés sur les contours, à savoir, le descripteur de courbure dans l'espace d'échelle (CSSD) et le descripteur de Fourier (FD).

### 1.2.1.2 Le descripteur (CSSD)

La méthode de la courbure dans l'espace d'échelles a montré qu'elle était très utile dans l'analyse de formes depuis que Asada et Brady l'ont utilisé pour obtenir un croquis primitif à partir d'une forme [5]. Son utilisation dans la recherche de formes a été proposée par Mokhtarian et al et utilisée par d'autres chercheurs [51, 55]. Principalement la méthode CSS traite la frontière de la forme comme un signal unidimensionnel et l'analyse dans l'espace d'échelles. En examinant les passages par zéro de la courbure à différentes échelles, les concavités et convexités du contour de la silhouette sont localisées. Ces concavités et convexités sont très utiles pour la description car elles représentent l'allure géométrique du contour de la silhouette. Le diagramme pour le calcul du CSSD est donné par la figure 1.2.

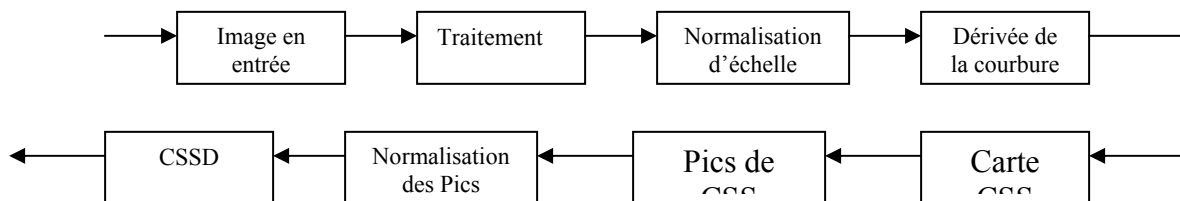


Figure 1.1. Diagramme pour le calcul du CSSD

La sortie de la première étape du processus est l'ensemble des coordonnées de la frontière :

$(x(t), y(t)), t=0,1,2,\dots,N-1$ .

La seconde étape est la normalisation d'échelle permettant de représenter la frontière de la forme par un nombre fixé de points. Ceci permettra d'apparier des formes avec différents nombres de points de haute courbure de la frontière.

Les deux principales étapes du processus sont le calcul des points de haute courbure de la frontière. Ces points sont obtenus en plusieurs étapes en plusieurs échelles des points d'inflexion (passage par zéro de la courbure).

La forme évolue alors vers une autre échelle en appliquant un lissage Gaussien. Plus le paramètre  $\sigma$  croît, plus la frontière devient lisse. Le processus est répété sur la nouvelle frontière, jusqu'à ce qu'aucun passage par zéro ne soit détecté (voir l'exemple de la figure 1.2).

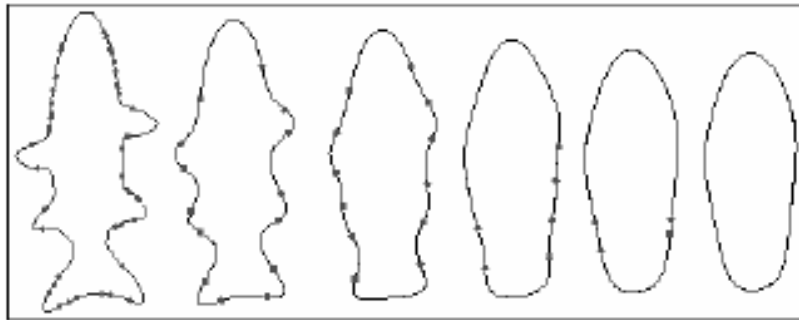


Figure1.2. Evolution du contour externe de la forme en fonction du paramètre  $\sigma$  [55]. De gauche à droite  $\sigma = 1, 4, 7, 10, 12, 14$ . Les points marqués correspondent aux points d'inflexion

La liste des points de contour CSS est composée de tous les points de passage par zéro de la seconde dérivée  $z_c(t, \sigma)$ , où  $t$  est la position et  $\sigma$  est l'échelle avec laquelle le point  $z_c$  est obtenu.

Le CSSD est invariant par translation. L'invariance d'échelle est effectuée en normalisant toutes les formes en un nombre fixé de points de d'inflexion du contour.

Sachant que la rotation de la forme produit des pics circulaires sur l'axe des  $t$ , l'invariance à la rotation est obtenue par un décalage circulaire des pics élevés vers l'origine de la carte CSS.

La ressemblance entre deux formes est alors mesurée par la sommation des différences des pics entre tous les pics mis en correspondance et les valeurs des pics non appariés [55].

### 1.2.1.3 Descripteur de Fourier (FD)

Les méthodes FD sont largement utilisées pour la description du contour de formes et leur analyse. Ils ont été utilisés dans beaucoup d'applications [4, 15, 16, 35, 40, 48, 63, 67, 71]. Nous décrivons ci-après le principe du descripteur de Fourier. Ce descripteur est dérivé en appliquant la transformée de Fourier à un signal 1-D signature de la forme. Différentes signatures ont été utilisées pour obtenir le descripteur de Fourier. Une signature de forme  $u(t)$  est une fonction 1-D représentant une zone 2-D ou une frontière qui décrit la forme d'une façon unique et traduit ses traits perceptuels. En supposant que les coordonnées de la frontière extraites sont  $(x(t), y(t))$ ,  $t = 0, 1, \dots, N-1$ , nous donnons ci-après quelques fonctions utilisées :

#### Fonction de position

La fonction de position, ou coordonnées complexes est le nombre complexe généré à partir des coordonnées de la frontière :

$$z(t) = [x(t) - x_c] + i[y(t) - y_c]$$

Où

-  $(x_c, y_c)$  est le centre de la forme.

-  $z(t)$  est une représentation simple de la frontière de la forme.

$z(t)$  est invariante par translation de la signature. La Rotation cause un décalage circulaire de  $z(t)$ , et le changement d'échelle de la forme produit un changement linéaire dans  $z(t)$ . L'utilisation des coordonnées complexes implique une réduction du temps calcul.

#### Fonction de la distance centrale

La fonction de distance centrale est exprimée par la distance des points de la frontière au point central  $(x_c, y_c)$  de la forme  $r(t) = ([x(t) - x_c]^2 + [y(t) - y_c]^2)^{1/2}$ .  $r(t)$  est aussi invariante par translation. La rotation cause un décalage circulaire de  $r(t)$ , et le changement d'échelle de la forme change linéairement  $r(t)$ . La figure 1.3, montre la signature de la distance centrale pour une forme initiale ainsi que pour la même forme dont le contour externe a subi une perturbation (bruit).

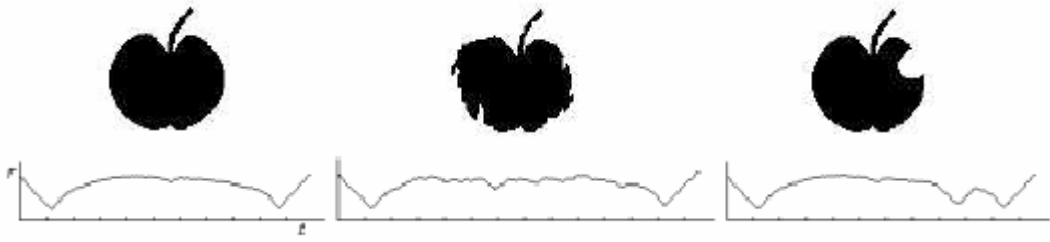


Figure 1.3. Les signatures des trois formes

### Obtention du descripteur de Fourier

Pour chaque fonction signature  $u(t)$ , la formule pour le calcul des paramètres de sa transformée de Fourier discrète est donnée par:

$$a_n = \frac{1}{N} \sum_{t=0}^{N-1} u(t) \exp(-j2\pi nt / N), n = 0, 1, \dots, N-1$$

Les coefficients de Fourier  $\{a_n\}$  calculés constituent la représentation de la forme. Donc les formes générées à travers la rotation, la translation et le changement d'échelle de la même forme sont des formes similaires, la représentation est invariante à ces opérations.

Le nombre de coefficients résultants de la transformée de Fourier est généralement important. Pour la recherche d'images, un petit nombre de coefficients sont sélectionnés pour la représentation de la forme. Les études menées dans ce sens ont montré que le descripteur FD obtenu directement des coordonnées de la frontière ne conviennent pas à la description et la recherche de formes en raison de sa sensibilité aux changements arbitraires de la frontière.

## 1.2.2 Représentations basées sur les régions

### 1.2.2.1 Introduction

Les descripteurs de formes basés sur les contours sont utiles si le contour de la forme est disponible avec précision. Cependant, il est difficile d'utiliser ce type de descripteur dans le cas de formes avec un contenu à l'intérieur. Les méthodes basées sur les régions utilisent le contour ainsi

que l'information de l'intérieur, et n'ont pas besoin en général de connaître le contour de la forme.

Nous présentons ci-après une des ces méthodes.

### **1.2.2.2 Le descripteur grille (Grid Descriptor (GD))**

Le descripteur grille a été proposé par Lu et Sajjanhar [48] et utilisé dans le système de recherche et d'analyse multimédia (MARS) [14] et dans d'autres applications comme [84]. La forme est projetée sur une grille de taille 16x16 cellules par exemple. Aux différentes cellules de la grille est affectée la valeur 1 s'ils appartiennent à la forme, 0 sinon. Le nombre de la forme est défini comme étant la séquence binaire obtenue en scrutant la grille de haut en bas et de gauche à droite. Cette séquence binaire est utilisée comme descripteur de la forme et comme index. Pour la comparaison des formes en utilisant les descripteurs GD, différentes normalisations ont été proposées pour obtenir une invariance au changement d'échelle, à la rotation et à la translation. Le calcul de ce descripteur commence par la recherche de l'axe principal de la forme, c'est à dire, la ligne joignant les deux points les plus éloignés de la frontière. La normalisation à la rotation est effectuée en tournant la forme telle que son axe principal soit parallèle à l'axe des X. La normalisation au changement d'échelle est effectuée en redimensionnant la forme telle que la longueur de l'axe principal soit égale à la largeur de la grille. La représentation est aussi invariante par translation. La seconde étape est l'échantillonnage de la grille. Finalement, le descripteur est généré comme étant la séquence binaire. La différence entre des descripteurs GD est calculée comme étant le nombre d'éléments ayant différentes valeurs.

### **1.2.3 Autres primitives pour la description d'image**

En général, Les primitives pour la description d'image sont regroupées en 3 classes :

- Les descripteurs de formes (décrit dans la section précédente) : descripteurs de Fourier et des moments, segments, courbure et points caractéristiques.
- Les descripteurs liés à la couleur : histogrammes et moments.
- Les descripteurs de texture : matrice de cooccurrence, direction principale et rugosité, et ondelettes.

**La couleur** : Parmi les traits qui permettent de caractériser une image, la couleur tient une bonne place après le contour. Il apparaît donc assez naturel d'intégrer cet aspect dans la description d'images. Cette description se fait par l'histogramme de couleurs ou par les moments des couleurs. Les deux sont des invariants globaux. Pour ce modèle d'approche, les trois couleurs fondamentales de la représentation en couleur les plus utilisées sont : RGB (Red Green Blue). Dans [46] on procède à une segmentation des régions par couleurs. Alors que Lai dans [42] construit une représentation arborescente à base de description de couleur et ce à partir des histogrammes de couleurs. Beaucoup de travaux ont été consacrés à l'invariance de ce descripteur et à son invariance [17, 29, 32, 60, 73].

**La texture** : peut être basée soit sur la description par des ondelettes, soit sur l'approche statistique générant des invariants globaux [61].

La description d'images peut contenir aussi les trois éléments cités (couleur, texture, forme), simultanément. Comme c'est le cas du système IMALBUM d'Idrissi [37].

## **1.3. SYSTEMES DE RECHERCHE D'IMAGES**

### **1.3.1 Notion de recherche d'image dans une base d'images**

Traditionnellement, les attributs textuels de l'image tels que le nom du fichier, la légende et les mots clés sont utilisés pour indexer les bases d'images. La problématique de la recherche d'images est alors translaté vers la recherche des mots.

Ce type d'approche est parfois efficace mais s'avère limité. Le problème repose sur le fait que l'information visuelle est complexe et nécessite l'intervention manuelle d'un opérateur (un expert) pour une description subjective et approximative. Un autre inconvénient subsiste dans la barrière linguistique, une image à une signification plus universelle.

Des recherches en matière de bases de données images ont conduit à développer des techniques de recherche d'images par le contenu. Le but de l'indexation basée sur le contenu est de

pouvoir choisir un groupe d'éléments d'une base de données d'après leur similarité à un élément proposé.

L'orientation suivie par la communauté des chercheurs dans cet axe est la proposition d'une description de l'image qui est par la suite indexée. En conséquence, tout descripteur de formes doit être une entité algorithmique invariant aux transformations géométriques, robuste au bruit, robuste aux occlusions, facilement calculable. Aussi, il est préférable de représenter une forme avec peu de caractéristiques afin que les données soient compressées efficacement.

Une fois que les images de la base de données sont indexées, des moteurs de recherche sont utilisés en une seule fois généralement, où un utilisateur fournit l'information de la requête au système et au retour, le système fournit plusieurs instances de la base de données à l'utilisateur. Un système Feedback permet à l'utilisateur d'indiquer au système quelles sont les instances désirables, ou pertinentes. Basé sur ce feedback (réaction), le système modifie son mécanisme de recherche en vue de satisfaire l'utilisateur.

Un système de recherche à feedback pertinent a plusieurs exigences lui permettant de fonctionner d'une manière efficace.

- Après chaque itération, le système doit rapporter un nombre raisonnable d'images à l'utilisateur.
- Le système doit produire des résultats acceptables après seulement quelques itérations.
- L'extraction devrait être complète dans une période courte de temps.

### **1.3.2 Quelques systèmes CBIR**

Kato [39] est le premier à avoir utilisé le système CBIR (Content-Based Image Retrieval) pour décrire la recherche automatique d'images dans une base de données en utilisant la couleur et les formes. Depuis, ce terme a été largement utilisé pour décrire la recherche d'images dans une collection large d'images en se basant sur les primitives qui peuvent être extraites automatiquement des images.

Différents systèmes commerciaux et de prototypes de recherche CBIR ont été développés. Le système commercial le plus connu et le plus représentatif est le système QBIC développé par

IBM [61], [27]. Il offre une recherche par la combinaison de couleur, texture et les mots clés. Les requêtes images peuvent être formulées par la sélection à partir d'une palette, spécifiant un exemple d'images, ou le squelette désiré de la forme sur l'écran. Les primitives sont organisées en utilisant une structure R\*-tree (voir annexe) pour améliorer l'efficacité de la recherche.

D'autres systèmes bien connus sont Virage [6] qui est utilisé par AltaVisa pour faciliter la recherche d'images et Excalibur [22],[26] qui est adopté par Yahoo pour la recherche d'images.

Photobook [66] est un CBIR prototype de recherche développé au MIT. Comme QBIC, les images dans la base de données sont représentées par la couleur, la forme, la texture. Cependant, il calcule les primitives à partir desquelles les aspects essentiels de l'image originale peuvent être reconstruites. Le système est orienté pour la recherche de textures, de formes de visages.

VisualSEEk est un autre prototype développé à Columbia University par Smith et Chang [74]. Il offre la recherche d'images par régions, par couleurs, par formes ainsi que par mots clés.

Nous citons d'autres prototypes de systèmes de recherche :

- Chabot [62]. Il fournit un accès basé sur le texte et la couleur à une collection de photos numérisées.
- Mars [34] : Développé à l'Université d'Illinois par Huang et al. Le système caractérise chaque objet dans une image par une variété de primitives, et utilise un rang pour différentes mesures de similarités pour comparer la requête et les objets mémorisés.
- Surfimage [59]. Développé à l'INRIA (France). Il utilise de multiples types de primitives images qui peuvent être combinées de différentes manières.
- Netra [49]. Développé par Ma et Manjunath, il utilise la couleur, la texture, la forme et la position spatiale pour fournir une recherche de régions basée sur les propriétés locales d'images dérivées d'une segmentation.

Actuellement, les recherches en CBIR sont focalisées sur trois points : Extraction de primitives, efficacité d'indexation et interface utilisateur.

### ***Extraction de primitives***

L'extraction de primitives inclut aussi les primitives sémantiques ou logiques. Les primitives logiques sont qualitatives de nature et fournissent une représentation générale de la donnée visuelle à différents niveaux de détail. Elles sont extraites manuellement ou semi automatiquement [79]. Une ou plusieurs primitives peuvent être utilisées dans des applications spécifiques. Par exemple, dans le système d'information satellitaire, la texture est importante, alors que les formes et la couleur sont plus importantes dans les systèmes commerciaux. Une fois les primitives ont été extraites, la recherche devient alors une tâche de mesure de similarité entre primitives des images.

### ***Efficacité d'indexation***

Pour obtenir une efficacité dans le processus de recherche suite à une requête, les indices d'images doivent être organisées dans une structure efficace. Des structures de données flexibles doivent être utilisées dans le but de faciliter le stockage et la recherche dans une base de données images. Des structures telles que k-d-tree, R-tree, R\*-tree, quad-tree, et grid file sont utilisés (voire Annexe).

### ***L'interface utilisateur***

Dans les systèmes d'informations visuels, l'interaction de l'utilisateur joue un rôle important dans la plupart de ses fonctions. L'interface utilisateur consiste à fournir des outils graphiques interactifs et des mécanismes pour l'exploration de la base de données et la prise en compte de requêtes.

En résumé deux exigences essentielles doivent être satisfaites par un système de recherche d'image: l'efficacité (aussi appelée l'exactitude) et le rendement. Une méthode de recherche d'images est efficace si les formes retrouvées sont semblables à la requête. Elle est effective si elle retrouve des formes rapidement et en respectant les exigences de l'espace de stockage de la base. Certaines approches font des compromis entre efficacité, rendement, et autres facteurs, tels que la capacité de manier des requêtes d'une silhouette partielle ou occluse. L'objectif est d'accentuer la vitesse en fournissant un niveau raisonnable d'efficacité, et aussi en supportant des requêtes partielles et occluse. Dans les sections suivantes, nous effectuons une présentation de quelques méthodes d'indexation.

### **1.3.3 Méthodes d'indexation d'images**

#### **1.3.3.1 Méthodes basées sur le contenu**

Dans [76], les auteurs présentent une méthode de reconnaissance de formes pour la recherche d'images basée sur le contenu pour les bases de données multimédias. Les formes sont caractérisées par les angles qui sont extraites par la fonction de la courbure de leurs contours. Les séquences d'angles sont alors comparées en utilisant les modèles de Markov afin que le processus soit résistant aux variations des angles et aux bruits. Dans ce travail une méthode de reconnaissance de silhouette est présentée. Elle se base sur un détecteur d'angles moyennant une fonction d'estimation de la courbure. Les points détectés sont résistants aux transformations géométriques et au bruit. L'avantage principal de la méthode proposée est qu'elle exige un index constitué de deux paramètres seulement, l'angle et la distance entre des angles consécutifs. Aucun de ces paramètres ne fait référence à la position du centre de la forme. De là, la méthode ne sera pas sensible aux déformations provoquées par les changements du centre, et par l'occlusion.

Dans [7], les auteurs proposent une méthode d'identification et d'indexation se basant sur une recherche structurelle de la forme par triangulation de Delaunay dans le cas d'identification d'ailes d'insectes. La triangulation de Delaunay est une représentation de formes, invariante aux transformations affines et au changement d'échelle. Outre ces propriétés, elle garantit une certaine stabilité, c'est à dire que des variations locales ne vont pas avoir de répercussions sur l'ensemble de la représentation. Coupler une représentation sous forme de triangles avec un codage simple permet d'obtenir un système d'indexation et d'identification d'images biologiques prenant en compte la totalité de l'information contenue dans la forme.

L'interrogation de bases d'images par le contenu est notoirement onéreuse en temps et ressources. Ce type d'interrogation a été rendu possible par le mariage de techniques bases de données et traitement d'images. Les derniers développements en bases de données visent à accélérer les recherches dans des index multidimensionnels. Ceux en image visent à améliorer les capacités de reconnaissance automatique. Or, les techniques modernes de traitement d'image ont pour conséquence de changer fondamentalement la manière dont la reconnaissance s'effectue: il est nécessaire d'interroger de nombreuses fois la base puis de synthétiser les résultats avant de pouvoir retourner une réponse, et non plus de se contenter d'une seule et unique interrogation. Dans [2] on montre que ce nouveau mode d'interrogation disqualifie toutes les techniques connues en indexation de bases de données. Il montre en particulier que les trois algorithmes donnés comme les plus

performants (à savoir pyramid-tree, VA-Fille et Séquentiel) sont inutilisables en l'état face au besoin de multiples interrogations. Il propose plusieurs pistes de recherche pour permettre l'interrogation de base d'images avec les techniques modernes de traitement d'images

R. Mezhoud [52] propose un système d'indexation des bases de données images par le contenu, ou chaque image est dans un premier temps décrite par un ou plusieurs vecteur de description de formes (Transformée de Fourier-Mellin, Moments géométriques, moment géométrique de Schmid, descripteur de Hu). Ensuite, elle propose d'effectuer une classification bayésienne des vecteurs attributs des images de cette manière, un vecteur de labels pour chaque image est construit.

### **1.3.3.2 Méthodes basées sur l'apparence locale**

Les approches basées sur l'apparence [3], [33], [58], [68], [70] utilisent l'apprentissage visuel pour contourner ce problème de modélisation. L'analyse d'un espace élémentaire pour l'apprentissage et la reconnaissance a été démontrée [58], [77]. Cependant, ces approches ne tiennent pas compte d'occlusion parce que l'espace des caractéristiques est construit pour utiliser la vue complète de l'objet. Pour pallier à ce problème d'occlusion partielle, Ikeuchi et Ohba [38] et Krumm [41] ont proposé une méthode où l'objet est représenté dans un espace élémentaire en terme de petites fenêtres de l'image. Les objets partiellement occlus peuvent être reconnus en se basant sur des fenêtres visibles. Une stratégie a été proposée pour la sélection des caractéristiques des fenêtres efficaces basée sur les critères de détectabilité, unicité et précision.

Cependant, Ikeuchi et Ohba [38] et Krumm [41] ne décrivent aucun mécanisme d'indexation pour accéder à l'espace élémentaire. Ces méthodes exigent, pour sauvegarder l'information un nombre élevé de fenêtres, ce qui induit un temps système important pour trouver les fenêtres appariées.

Une architecture de reconnaissance qui s'attaque aux problèmes susmentionnés de cette approche a été proposée. Une technique d'extraction de caractéristiques pour isoler l'objet de l'arrière plan et pour les rendre invariant à l'illumination [28]. Pour accéder efficacement à l'espace des caractéristiques un mécanisme d'indexation est proposé, et produit un index basé sur l'histogramme de couleur et la géométrie de la fenêtre élémentaire. Ce mécanisme d'indexation élimine l'exigence de comparer une fenêtre candidate à toutes les fenêtres du dictionnaire pour trouver le meilleur appariement dans la phase de reconnaissance, comme dans [38].

Mohit [53] a aussi proposé un mécanisme d'indexation basé sur l'apparence locale. Ayant constaté que dans les systèmes de reconnaissance basée sur les apparences locaux, le processus de recherche devient extrêmement lent en raison de l'augmentation du nombre des points caractéristiques ainsi que le nombre des objets de la base de données, Mohit[53] propose d'indexer les groupes de caractéristiques locaux semblables en utilisant l'information géométrique et la couleur. Pour cela il définit deux caractéristiques pour chaque fenêtre image, à savoir, étiquette contour et étiquette couleur. Ces attributs sont calculés de l'histogramme de la couleur normalisée de la fenêtre après élimination de l'arrière plan.

Dans [11], les auteurs exploitent l'hypothèse de la non fragmentation de la silhouette dans deux directions. En premier lieu, les représentations du contour de la silhouette dans l'espace d'échelles sont utilisées pour la recherche de la structure locale du contour. Deuxièmement, des histogrammes de types de structure locale (c'est des types de parties de la silhouette) sont utilisés comme vecteurs d'index. Chaque composant du vecteur d'index est le nombre d'occurrences d'un type de la partie spécifique. La similarité entre les requêtes et les silhouettes de la base de données sont mesurées en comparant leurs vecteurs d'index. L'indexation structurelle est utilisée pour exécuter les calculs de similarité. Ces trois stratégies — représentation dans l'espace d'échelles, vecteurs d'index, et indexation structurelle — sont combinées pour effectuer une recherche de silhouettes.

L'indexation structurelle réalise l'indexation des modèles ou des silhouettes par parties individuelle. Elle est fréquemment utilisée dans la reconnaissance d'objet.

Dans les méthodes de recherche des formes de Mokhtarian et al. [56] et Del Bimbo et Pala [18] qui utilisent aussi des représentations d'espace d'échelles, des algorithmes d'appariement procèdent en multi-niveaux et font ressortir l'information à différentes échelles dans le calcul de similarité.

Dans [11], la base de données est indexée indépendamment par les parties à différentes échelles, donc des silhouettes peuvent être appariées dans le détail même si leur structure totale n'est pas semblable. Cela est relativement important, comme l'ont suggéré par Dudek et Tsotsos[23].

Le système de récupération de silhouette [11] a trois phases de traitements. Dans la première phase, le contour de la silhouette est segmenté en parties correspondant à des séquences de régions

adjacentes dans l'espace d'échelles. Dans la deuxième phase, les caractéristiques locales des parties de la silhouette sont calculées dans différentes échelles du contour. Dans la troisième phase, la recherche de la silhouette est exécutée en comparant le vecteur d'index de la silhouette de la requête aux vecteurs d'index de la silhouette de la base de données.

L'index vecteur constitué des histogrammes de parties a été utilisé par Hu et Pavlidis [65] pour apparier des réseaux de route et réaliser la reconnaissance des mots manuscrits. Dans [65], entre 5 à 10 applications spécifiques de types de partie ont été utilisés. La conséquence du nombre important des types de parties, donc des vecteurs d'index importants, est qu'une classification effective est exigée pour exécuter les comparaisons des vecteurs de l'index. Ils utilisent l'indexation structurelle pour ce but.

### **1.3.3.3 Méthodes basées sur les relations spatiales**

Certaines approches proposent d'utiliser les relations entre les parties d'un objet comme index. Dans [13] le système réalise une description symbolique de l'objet (ou de la scène) basé sur les propriétés de segments individuels (dédiés à partir d'une organisation de la surface de l'objet) et leurs rapports spatiaux. Les propriétés des segments et les relations sont représentées sous forme d'un graphe dont les nœuds représentent les segments et les liens représentent les rapports géométriques entre les paires de segments.

## **1.4. CONCLUSION**

Dans ce premier chapitre, nous avons présenté les principales méthodes de description d'images proposées dans la littérature. Nous avons aussi présenté les principaux systèmes de recherche d'images existants ainsi que les méthodes d'indexation d'images.

Dans le prochain chapitre nous effectuons une étude focalisée sur la description d'images d'objet à partir de leurs silhouettes, le langage développé pour l'écriture textuelle de ces descriptions, la description d'image contenant des régions internes, et en fin le langage permettant d'écrire textuellement ces descriptions.

## **CHAPITRE 2**

### **DESCRIPTION TEXTUELLE D'IMAGES**

## 2.1 INTRODUCTION

Dans ce chapitre nous présentons tout d'abord la méthode permettant la description d'images d'objet à partir de leurs silhouettes [12]. Elle sera suivie par les éléments du langage développé pour l'écriture textuelle de ces descriptions. Comme les images d'objets contiennent souvent des régions internes, la prise en compte de ces informations a été l'apport de la seconde méthode développée au même laboratoire et permettant de décrire toute forme en tenant compte de ses contours internes. Cette méthode est présentée ainsi que le langage **LWDS** (Language for Writing Descriptors of Shapes) permettant d'écrire textuellement ces descriptions.

## 2.2 DESCRIPTION TEXTUELLE DE SILHOUETTES

### 2.2.1 Présentation de la méthode

Dans [12, 43] les auteurs proposent une méthode pour la description d'images d'objets à partir de leurs silhouettes. Une fois que la silhouette sur l'image est extraite, un processus de partitionnement est opéré en utilisant les points de concavité du contour externe comme étant le lieu de décomposition. Ces points sont d'abord localisés en suivant une direction de balayage comme étant les points de concavité correspondant à un changement de direction du contour allant du Haut ↗ Bas ↘ Haut ou du Bas ↗ Haut ↘ Bas comme illustré par la figure 1.

Chacun de ces points définit une ligne de liaison entre des ensembles contigus de lignes de pixels appelées partitions. Nous obtenons ainsi soit deux partitions ou plus sont jointes avec une autre partition à travers la ligne de jonction ou une partition qui est jointe avec un groupe de partitions à travers une ligne de disjonction.

La silhouette est ainsi décrite par un graphe dont les nœuds sont soit des partitions ou des lignes de jonction et de disjonction. Alors que les arêtes sont représentées par la liaison. Le graphe de la figure 2.2 est par exemple la description associée à la silhouette de la figure 2.4.

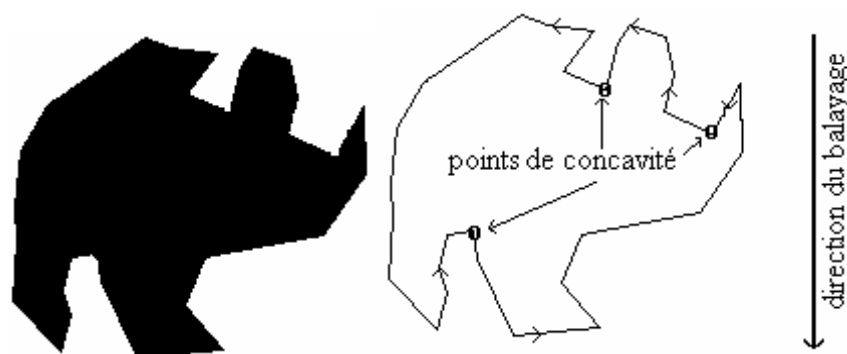


Figure 2.1. A gauche un exemple de silhouette, à droite les points de concavité

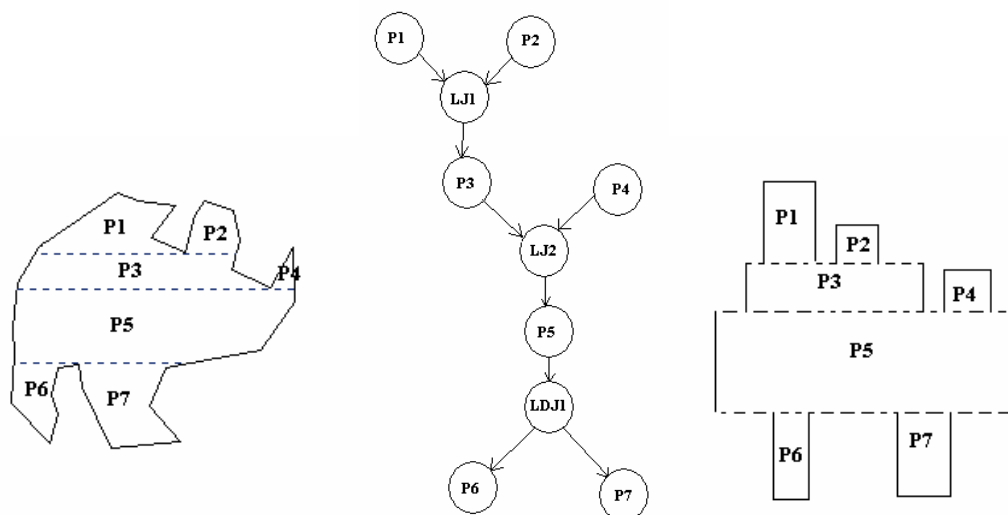


Figure 2.2 Le graphe description des deux silhouettes

Cependant le graphe résultat peut correspondre à plusieurs silhouettes. C'est le cas de la silhouette de la figure 2.1. Il est donc nécessaire de détailler la description de chacun des éléments du graphe pour assurer l'unicité de la description. Nous montrons dans ce qui suit comment chacun de ces éléments est décrit géométriquement.

### Description des partitions

Le contour externe de chaque partition est décomposée en deux : frontière gauche et frontière droite qui commencent chacune au point le plus haut à gauche et se terminent chacune au dernier point du contour (voir Figure 2.3). Ces frontières sont ensuite décomposées en contours élémentaires (ligne ou courbe) au moyen de point de d'inflexion (voir Figure 2.3). Plusieurs algorithmes permettant de localiser ces points d'inflexions ont été proposés. A titre d'exemple l'algorithme de Chetverikof [82]. La figure 2.4 montre les résultats obtenus par cet algorithme sur la silhouette de la figure 2.1.

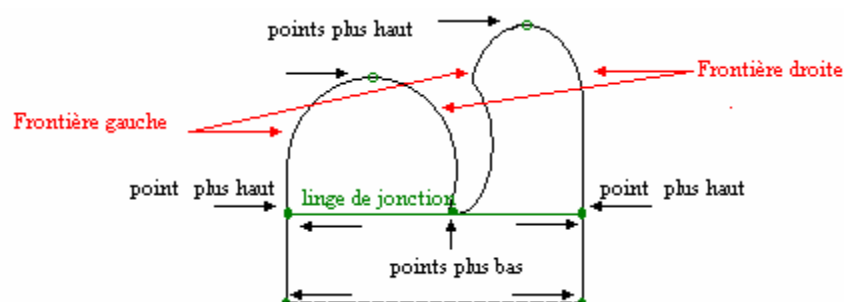


Figure 2.3. Eléments d'une partition

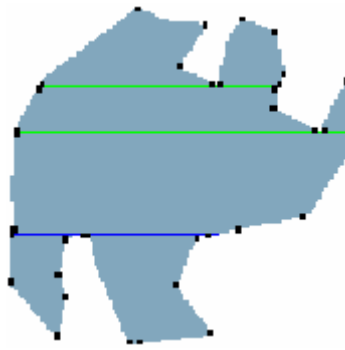


Figure 2.4. Décomposition du contour externe en éléments simples

L'étape suivante consiste en la description de chaque contour élémentaire au moyen des paramètres:

- type (ligne, concave, convexe),
- degré de convexité ou concavité (cas d'un contour courbure)
- angle d'inclinaison,
- longueur

Le calcul des ces attributs est explicité par la figure 2.5. Le degré de concavité ou convexité d'une courbe est le rapport entre  $r$  et la distance entre ses deux extrémités.

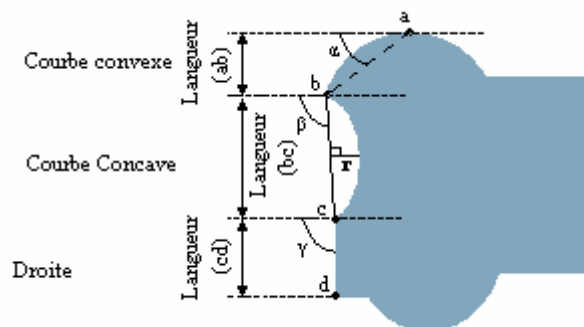


Figure 2.5. Eléments pour la description des contours

### Description des lignes de jonction et de disjonction

En dernier lieu, lignes de jonction et de disjonction sont décomposée en segments et décrits moyennant les attributs suivants : -type du segment qui peut être : jonction lorsqu'il est commun à deux partitions, FreeLow lorsqu'il est libre et liaison uniquement avec la partition du bas et FreeHigh lorsqu'il est libre et en liaison uniquement avec la partition supérieure

- Numéros de partitions qu'ils sont en liaison
- longueur du segment

### 2.2.2 Le langage LWDOS pour l'écriture textuelle des descripteurs de silhouettes

Pour écrire la description de silhouette présentée précédemment, un langage noté LWDOS (Language for Writing Desriptors of Outline Shapes) a été développé [43] et dont la version XML non encore publiée [81] est présentée ci-après :

Chaque partition est décrite en utilisant c'est deux contours comme suit :

**Part** → **<LeftBoundary>** Description du contour gauche **</LeftBoundary>**

**<RightBoundary>**Description du contour droit**</RightBoundary>**

Les contours droit et gauche sont décrits en utilisant leur description de contours élémentaires comme suit :

**LeftBoundary** → **<LeftBoundary>** Description de contour élémentaire 1,

Description de contour élémentaire 2,

..... , Description de contour élémentaire n

**</ LeftBoundary>**

**RightBoundary** → **<RightBoundary>**

Description de contour élémentaire 1,

Description de contour élémentaire 2,

..... , Description de contour élémentaire n

**</RightBoundary>**

Chaque contour élémentaire est décrit comme suit :

**<Description du Contour élémentaire >→**

**cv** Degré de convexité Angle d'inclinaison Longueur /

**cc** Degré de concavité Angle d'inclinaison Longueur /

**r** Angle d'inclinaison Longueur

Où **cv**, **cc**, **r** indiquent respectivement un contour convexe, concave, et droit.

## Description de ligne de jonction et de disjonction

Les lignes de jonction et de disjonction sont décrites comme suit :

**Junction line** → **<Junction>** Description du segment 1  
Description du segment 2  
.....  
Description du segment n  
**</Junction>**

**Disjunction line** → **<Disjunction>** Description du segment 1  
Description du segment 2  
.....  
Description du segment n  
**</Disjunction>**

La description du segment est comme suit:

### Description of segment →

**j** Numéro de la partition haut    Numéro de la partition bas    Longueur /  
**w** Numéro de la partition bas    Longueur /  
**h** Numéro de la partition haut    Longueur

Où **j**, **w**, **h** dénotent respectivement les attributs **Junction**, **FreeLow** et **FreeHigh**.

## Description de la liaison entre partitions

Lorsque deux partitions ou plus se joignent à travers la ligne de jonction ou disjonction, l'ensemble de ces éléments forment une partition composée notée **ComposedPart** qui est décrite par la syntaxe suivante :

**ComposedPart** → **< ComposedPart >**  
Partition<sub>1</sub> Partition<sub>2</sub> .... Partition<sub>n-1</sub>  
**<Junction>** description de la ligne de jonction **</junction>**  
Partition<sub>n</sub>  
**</ ComposedPart>**

Ou bien :

**ComposedPart** → **<ComposedPart>**  
Partition<sub>1</sub>

**<Disjunction>** description de la ligne de disjonction **</Disjunction>**  
 Partition<sub>2</sub> Partition<sub>3</sub> ..... Partition<sub>n</sub>  
**</ ComposedPart>**

Exemple:

Si on prenait l'image de la figure 2.1 comme exemple, son descripteur s'écrirait :

```

< ComposedPart>
  <ComposedPart>
    <ComposedPart> P1 P2 JL1 P3 </ComposedPart> P4 JL2 P5
  </ ComposedPart>
DJL1 P6 P7
</ ComposedPart>

```

## Description de la silhouette

Une silhouette peut être soit une partition ou une partition composée comme cela est décrit par la syntaxe suivante :

Silhouette → **<DLWDOS>**  
     **<Name>** Nom de l'objet **</Name>**  
     **<ComposedPart>** Description de la partition Composée**</ComposedPart>**  
**</DLWDOS>**

Le descripteur de la silhouette de la figure 2.1 est le suivant où les balises des différents éléments ont été réduites en nombre de caractères :

**CP** pour ComposedPart, **L** pour LeftBoundary, **R** pour RightBoundary, **J** pour Junction et **D** pour Disjunction

```

<DLWDOS><Name>Figure 2.1</Name>< Length> 176</ Length>
<CP>
  <CP>
    <P1> <L>r 35 36 </L> <R>cc 6% 167 8 r 55 19 cc 11% 152 9</R> </P1>
    <P2> <L>cv 6% 70 31</L> <R>r 180 1 r 160 6 r 101 20 r 68 5</R> </P2>
    <J> w 3 1 j 1 3 89 w 3 4 j 2 3 29</J>
    <P3><L> r 63 24 </L><R>r 92 23</R></P3>
  </CP>
  <P4><L>r 61 20</L><R>r 83 9 r 152 11</R></P4>
  <J>j 3 5 154 w 5 5 j 4 5 13</J>
  <P5><L>cv 5% 87 45</L><R>r 90 8 r 52 31 r 10 6 </R></P5>
  <D>w 6 1 j 5 6 37 h 5 2 j 5 7 63 h 5 5 </D>

```

```

<CP>
<P6><L> r 87 22 r 132 26</L><R> r 12 9 r 76 17 r 111 10 r 74 18</R></P6>
<P7><L> r 113 51 r 180 5</L><R> r 9 6 r 64 23 r 129 22 r 7 5</R></P7>
</CP>
</CP>
</DLWDOS>

```

### 2.2.3 Propriétés du descripteur LWDOS

Le descripteur LWDOS vérifie les propriétés suivantes :

- invariance à la rotation Cette propriété est obtenue grâce au calcul du descripteur de la silhouette relativement au rectangle minimum englobant la silhouette.
- invariance au changement d'échelle. Cette propriété est obtenue grâce la possibilité de déduire les longueurs relatives des différents éléments contenues dans le descripteur à partir de leurs longueurs absolues et celle du rectangle minimum englobant la silhouette.
- invariance à la translation

## 2.3 METHODE DE DESCRIPTION TEXTUELLE DE FORMES

### 2..3.1 Présentation de la méthode



Figure 2.6. Exemple de silhouette d'objet.

Si on observe l'image de la figure 2.6 nous pouvons interpréter l'image comme étant celle d'un véhicule roulant, il peut être un bus ou un camion. La silhouette n'est pas porteuse d'informations concernant la couleur, la forme interne ainsi que d'autres caractéristiques de l'objet pouvant déterminer avec exactitude le type du véhicule. La silhouette est donc une représentation non suffisante pour la représentation du contenu d'une image. En effet, une image contient d'autres informations aussi pertinentes que la silhouette telle que la couleur, la texture et les régions internes. L'utilisation de ces informations peut être très utile pour obtenir une description complète. Pour enrichir la méthode précédente, une méthode a été proposée par S. Larabi [81] pour la description de formes en tenant compte de son contour externe et ses contours internes. Nous donnons ci-après le principe de base de cette méthode.

#### Notion de région interne

Une région interne est définie comme étant un contour fermé correspondant à une région localisée dans l'image de l'objet (voir Figure 2.7).

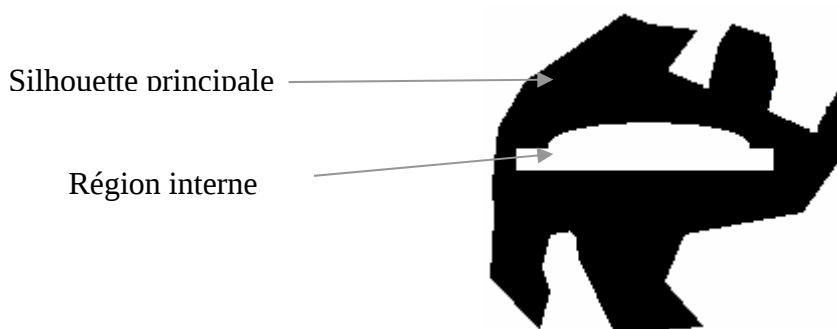


Figure 2.7 : Silhouette interne

La silhouette principale peut donc contenir plusieurs régions internes (voir figure 2.8).



Figure 2.8. La silhouette principale avec plusieurs régions internes

Récursivement, chaque région interne est considérée comme une silhouette et peut contenir des régions internes (voir figure 2.9).



Figure 2.9. La région S contient à son tour deux régions internes

### Notion de Niveaux

Toute région interne ne contenant pas elle-même des régions internes est considérée comme étant une silhouette élémentaire de niveau zéro (voir figure 2.10). Toute région interne contenant des silhouettes élémentaires de niveau 0 est considérée comme étant une silhouette de niveau un. D'une façon générale, une région interne contenant des régions de niveau inférieur ou égal à  $n-1$  est considérée comme étant une silhouette de niveau  $n$ .

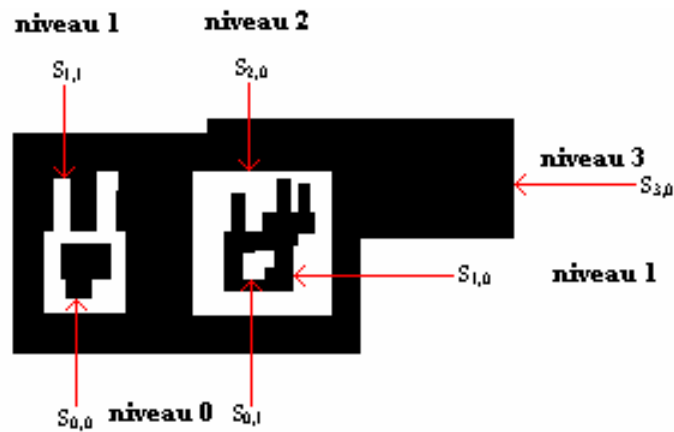


Figure 2.10. Une image et les différentes formes la composant

En notant  $S_{ij}$  la  $j$  silhouette du niveau  $i$ , l'image de la figure 2.10 contient deux silhouettes élémentaires:  $S_{00}$ ,  $S_{01}$ , deux silhouettes de niveau un :  $S_{11}$ ,  $S_{10}$  de niveau 1 et une silhouette de niveau 2:  $S_{20}$ . La silhouette principale :  $S_{30}$  est donc de niveau 3.

Pour la description des différentes régions internes, deux éléments sont à utiliser :

- La position de chaque région interne relativement au repère défini par le rectangle minimum englobant la silhouette principale. Cette position correspond à celle de son point de début (point le plus haut à gauche du rectangle minimum l'englobant) (voir figure 2.11).
- La relation d'inclusion qui permet de décrire comment sont disposées les différentes régions internes.

La description de toute région interne est faite à l'aide du langage LWDOS décrit précédemment relativement au rectangle minimum l'englobant.

Coordonnées de début de  
l'image interne



Figure 2.11. Position des silhouettes internes

### 2.3.2 Le langage LWDS pour l'écriture textuelle des descripteurs de formes

Pour l'écriture du descripteur de formes un langage LWDS (Language for Writing Descriptors of Shapes) a été développé au laboratoire LRIA comme extension du langage LWDOS. La grammaire du LWDOS a été augmentée par un ensemble de règles pour la prise en compte des régions internes (position, inclusion et description LWDOS) en suivant une syntaxe XML [83].

Lorsque la forme est une silhouette élémentaire, elle est décrite moyennant le langage LWDOS. Dans le cas où elle contient des silhouettes internes, elle est décrite comme suit:

#### Internal Shape →

```
<Shape> Description de la silhouette de la forme
    <Internal Shape>Description de la région interne 0 </Internal Shape>
    <Internal Shape>Description de la région interne 1</Internal Shape>
    .....
    <Internal Shape>Description de la région interne n </Internal Shape>
</Shape>
```

Pour la forme générale du descripteur LWDS, elle est comme suit:

#### Description LWDS of Shape → <DLWDS>

```
<Name> Nom de la forme > </Name>
    <Shape>Description de la forme</Shape> /
</DLWDS>
```

Comme exemple, nous donnons la structure générale du descripteur de la forme illustrée par la figure 2.10.

```
<DLWDS>
<Name> Forme de la figure 2.10 </Name>
<Shape>
    Position et Description LWDOS de la silhouette principale S3,0
    <InternalShape>
        <Shape> Position et Description LWDOS de la silhouette principale S2,0
        <InternalShape>
            <Shape> Position et Description LWDOS de S1,0
            <InternalShape>Position et Description LWDOS de S0,1
            </InternalShape>
        </Shape>
    </InternalShape>
</Shape>
</InternalShape>
```

```

    < InternalShape>
      <Shape> Position et Description LWDOS de  $S_{1,1}$ 
      <InternalShape> Position et Description LWDOS de  $S_{0,0}$ </InternalShape >
      </Shape>
    </InternalShape >
  </ Shape >
</DLWDS>

```

### Exemple

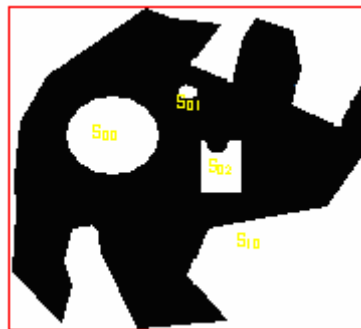


Figure 2.12 Exemple

```

<DLWDS>
<Name> Figure 2.12</Nom>
< Shape > Descripteur LWDOS de  $S_{1,0}$ 
< InternalShape > Position de  $S_{0,0}$  Descripteur LWDOS de  $S_{0,0}$  </ InternalShape >
< InternalShape > Position de  $S_{0,1}$  Descripteur LWDOS de  $S_{0,1}$  </ InternalShape >
< InternalShape > Position de  $S_{0,2}$  Descripteur LWDOS de  $S_{0,2}$  </ InternalShape >
</ Shape >
</DLWDS>

```

Sachant que

**CP** : CompsedPart, **L** : LeftBoundary, **R** : RightBoundary, **J** : Junction et **D** : Disjunction

Le descripteur généré est donc :

```

<DLWDS>
<Name> Figure 2.12</Name>
< Shape >
<DLWDOS><Name>Silhouette  $S_{0,1}$ </Name>

```

< Length> 176</ Length>

<CP>

<CP>

<P1><L>r 35 36 </L><R>cc 6% 167 8 r 55 19 cc 11% 152 9</R></P1>

<P2><L>cv 6% 70 31</L><R>r 180 1 r 160 6 r 101 20 r 68 5</R></P2>

<J>w 3 1 j 1 3 89 w 3 4 j 2 3 29</J>

<P3><L>r 63 24 </L><R>r 92 23</R></P3>

</CP>

<P4><L>r 61 20</L><R>r 83 9 r 152 11</R></P4>

<J>j 3 5 154 w 5 5 j 4 5 13</J>

<P5><L>cv 5% 87 45</L><R>r 90 8 r 52 31 r 10 6 </R></P5>

<D>w 6 1 j 5 6 37 h 5 2 j 5 7 63 h 5 5 </D>

<CP>

<P6><L>r 87 22 r 132 26</L><R>r 12 9 r 76 17 r 111 10 r 74 18</R></P6>

<P7><L>r 113 51 r 180 5</L><R>r 9 6 r 64 23 r 129 22 r 7 5</R></P7>

</CP>

</CP>

</DLWDOS>

< InternalShape > (256,196)

<DLWDOS><Name>Silhouette S0,0</Name>

< Length >40</ Length >

<P1><L>r 9 6 cv 14% 48 16 cv 16% 124 20 r 174 11 </L><R>r 174 11 cv 16% 124 20 cv 14% 48 16 r 9 6 </R></P1>

</DLWDOS>

</ InternalShape >

< InternalShape > (364,192)

<DLWDOS><Name>Silhouette S0,1</Name>

< Length >7</ Length >

<P1><L>r 59 5 r 170 6 </L><R>cv 32% 128 5 r 45 1 </R></P1>

</DLWDOS>

</ InternalShape >

< InternalShape > (490,288)

<DLWDOS><Name>Silhouette S0,2</Name>

< Length >37</ Length >

<CP>

<P1><L>r 90 7 </L><R>r 135 7 </R></P1>

<P2><L>r 63 6 </L><R>r 180 6 r 98 7 </R></P2>

<J>j 1 3 7 w 3 9 j 2 3 10</J>

<P3><L>r 92 28 r 180 25 </L><R>r 90 27 </R></P3>

</CP>

</DLWDOS>

</InternalShape >

</ Shape >

</DLWDS>

## **2.4. CONCLUSION**

Dans ce chapitre nous avons présenté les deux méthodes permettant de décrire les silhouettes et les images d'objets. Nous avons aussi montré comment écrire textuellement leurs descriptions moyennant les langages LWDOS et LWDS. Dans le prochain chapitre nous abordons le problème d'indexation de descripteurs d'images. Nous présentons notre contribution pour la mise en œuvre d'une base de données images : Il s'agit de trouver une solution au problème d'indexation des descripteurs LWDS.

## **CHAPITRE 3**

# **INDEXATION DE DESCRIPTEURS DE FORMES ECRITS SUIVANT LE LANGAGE LWDS**

### 3.1 INTRODUCTION

Dans ce chapitre nous présentons d'abord la notion de base de données images. Nous donnerons ensuite la solution proposée dans [81] pour l'indexation d'une base de descripteurs LWDOS avec quelques améliorations. Nous présentons enfin notre contribution pour l'indexation de descripteurs LWDS.

### 3.2 NOTION DE BASE DE DONNEES IMAGES

Mettre en œuvre une base de données images en vision artificielle nécessite en premier lieu de disposer en mémoire des différentes images (ou de leurs descriptions) possibles de chaque objet modèle. Pour un objet donné, ces images sont prises par une caméra parcourant les cercles d'un hémisphère englobant l'objet pas à pas (voir figure 3.1). Les figures 3.2 et 3.3 illustrent les images d'un objet modèle prises par la caméra MICAM X du laboratoire L.R.I.A. en rotation de 360° pour deux positions différentes de la caméra sur l'hémisphère. La seconde étape est l'organisation de ces données en mémoire et leur indexation dont le but d'assurer un accès rapide aux algorithmes de mise à jour et de recherche d'images.

Toute application faisant intervenir une connaissance nécessite de disposer une base d'images modèles de tous les objets intervenant dans l'application. Elle nécessite aussi le développement des algorithmes pour la comparaison d'une image requête et l'ensemble d'images modèles. Pour la comparaison d'images une mesure de similarité doit être définie pour le classement des images dans un ordre décroissant de similarité.

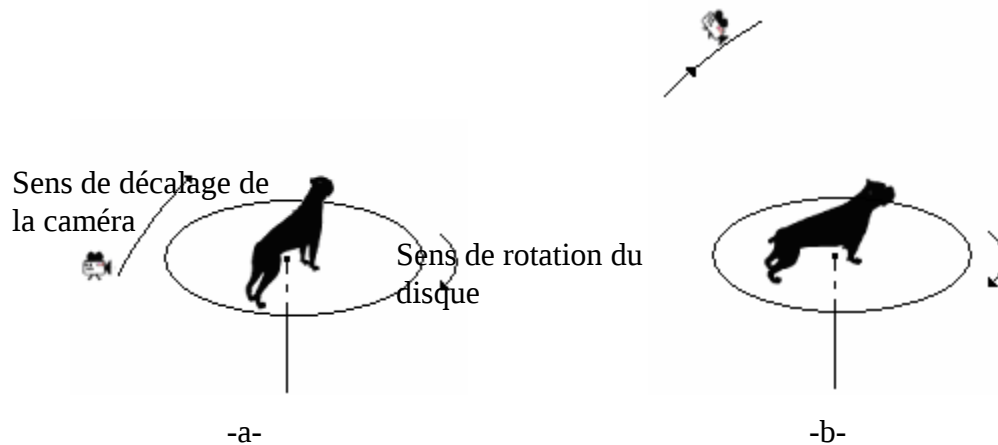


Figure 3.1. Principe de prise de vues d'un objet modèle



Figure 3.2. Exemple d'un ensemble d'images modèles d'un même objet

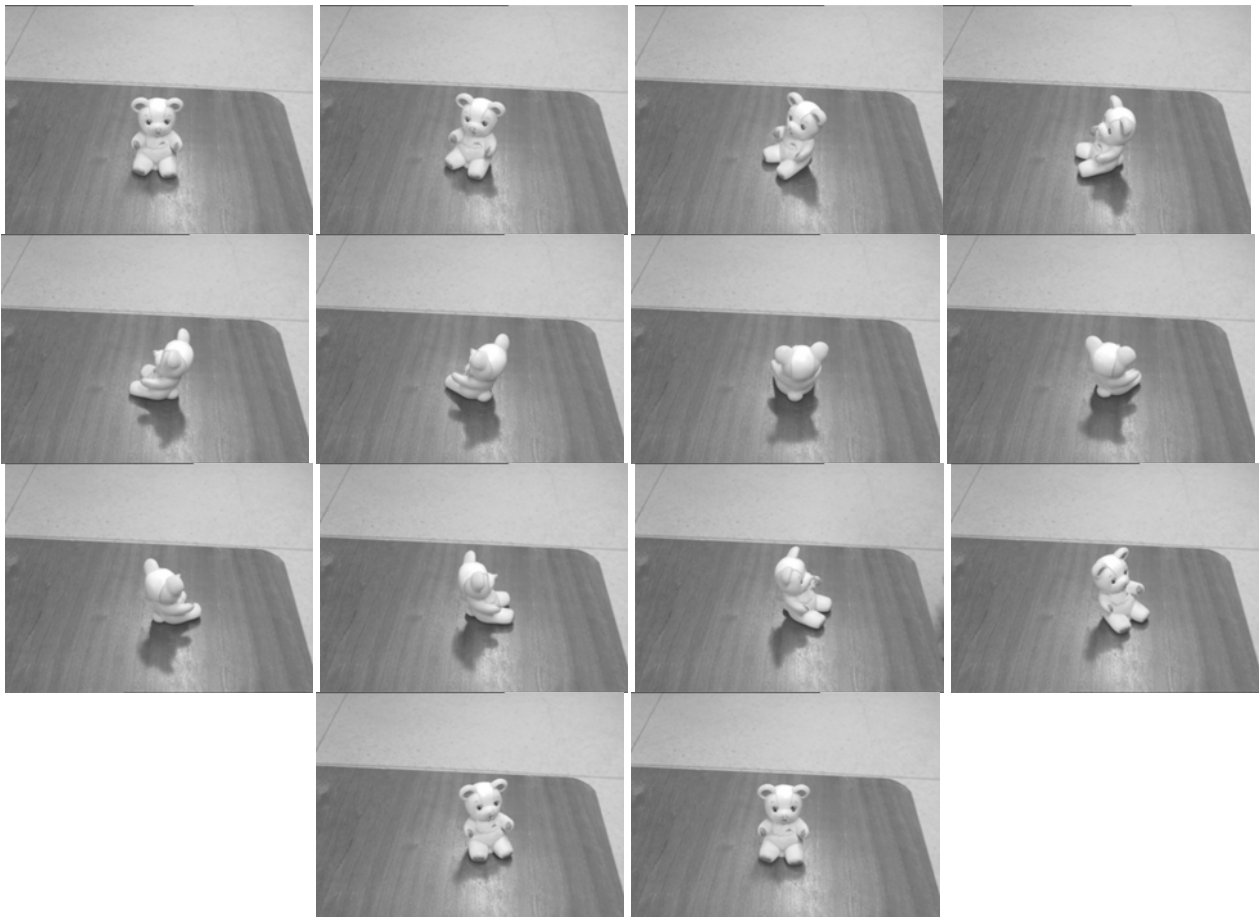


Figure 3.3. Second exemple d'un ensemble d'images modèles du même objet

Pour les applications faisant intervenir un nombre important d'objets modèles, le nombre d'images devient contraignant pour l'espace mémoire qu'elles nécessitent et pour la complexité algorithmique des algorithmes de recherche d'images qui en découle. Une indexation de cette base d'images est alors utile et nécessaire permettant d'accéder rapidement à un ensemble réduit d'images candidates. La figure 3.4 illustre les différents modules d'un système de recherche d'images.

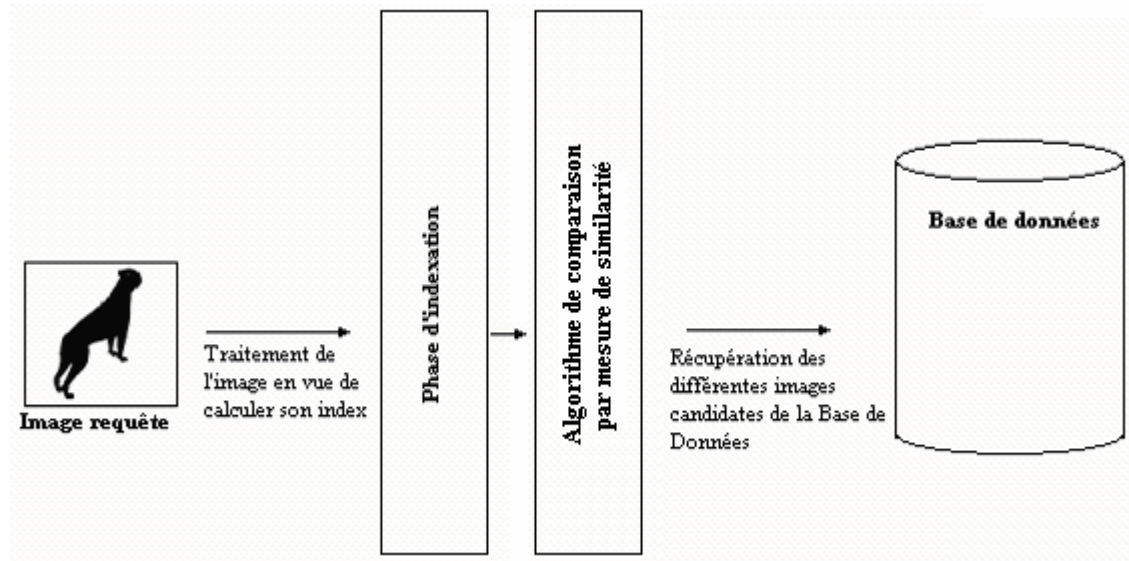


Figure 3.4. Différents éléments d'un système de recherche d'images

Dans la section suivante nous montrons comment mettre en œuvre une base de données de descriptions d'images écrites moyennant le langage LWDOS.

### 3.3 INDEXATION DE DESCRIPTEURS DE SILHOUETTES ECRITS SUIVANT LE LANGAGE LWDOS

#### 3.3.1 Premier niveau d'indexation de descripteurs LWDOS

Nous présentons dans ce mémoire, avec quelques améliorations, la solution proposée dans [81] une indexation de descripteurs LWDOS. L'indexation de ces descripteurs se fait en deux niveaux :

Le premier index est constitué des paramètres suivants :

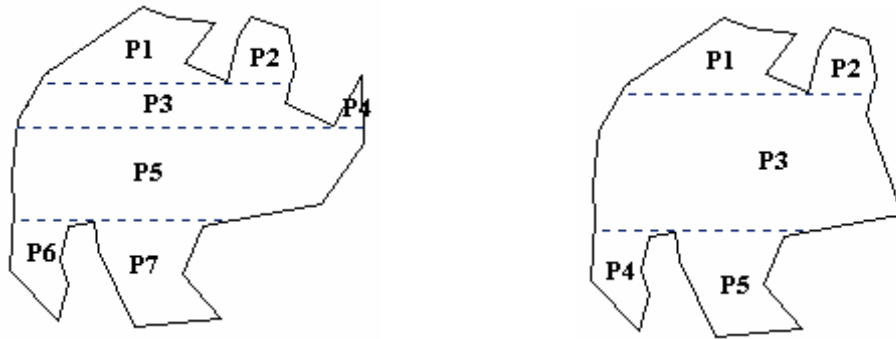


Figure 3.5. Différence de silhouettes par le nombre de partitions

- Le nombre de partitions de la silhouette (NP) compté dans le descripteur LWDOS comme étant le nombre de balises de la forme  $\langle P_i \rangle \langle /P_i \rangle$ . Cet élément permet de distinguer entre silhouettes comme le montre la figure 3.5.

- Les nombres de lignes de jonction (NJL) et de disjonction (NDJL) de la silhouette compté dans le descripteur LWDOS comme étant le nombre de balises de la forme :  $\langle J_i \rangle \langle /J_i \rangle$  et  $\langle D_i \rangle \langle /D_i \rangle$ . Ces deux nombre permettent aussi de distinguer entre silhouettes comme le montre la figure 3.6.

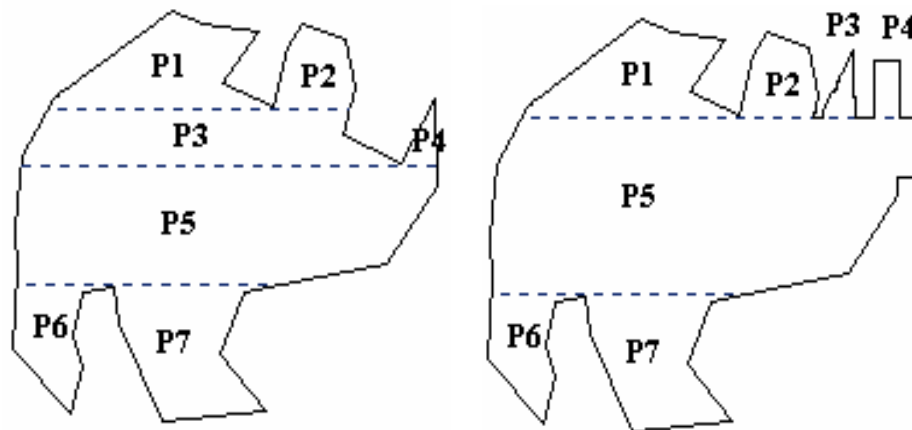


Figure 3.6. Différence de silhouettes par le nombre de ligne de jonction et de disjonction

- L'ordre d'apparition des lignes de jonction et de disjonction dans le descripteur de la silhouette, noté ORDER. Si deux silhouettes ont le même nombre de partitions, de lignes de jonction et de disjonction, cet ordre peut les distinguer comme le montre la figure 3.7. Cet ordre est calculé comme étant une chaîne de bits écrits de gauche à droite en affectant un bit égal à 0 à une ligne de jonction et bit égal à 1 à une ligne de disjonction. La chaîne de bits obtenue représente le nombre

(ORDER). A titre d'exemple, l'ordre pour la silhouette de gauche de la figure 3.7 est : "001", alors que celui de la silhouette de droite de la même figure est "100".

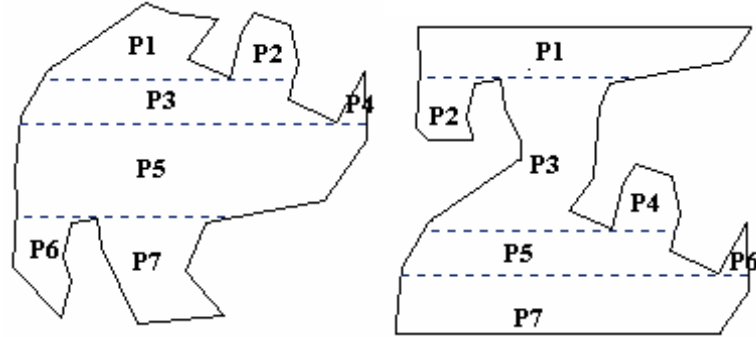


Figure 3.7. Différence de silhouettes par l'ordre des lignes de jonction et de disjonction

Malgré que ce dernier paramètre permette de différencier un nombre important de silhouettes, différentes silhouettes peuvent avoir le même paramètre (voir figure 3.8). Un quatrième paramètre est donc ajouté :

- Le nombre de partitions attachées à chaque ligne de jonction et de disjonction, noté NPA.

Ce nombre est calculé comme suivant :

$$NPA = \sum_{i=1}^{NL+NDJL} NPL_i \times 10^i \text{ Où :}$$

- i est le numéro d'ordre des lignes de jonction et de disjonction

-  $NPL_i$  est le nombre de partitions associées à la  $i^{\text{ème}}$  ligne de jonction ou de disjonction.

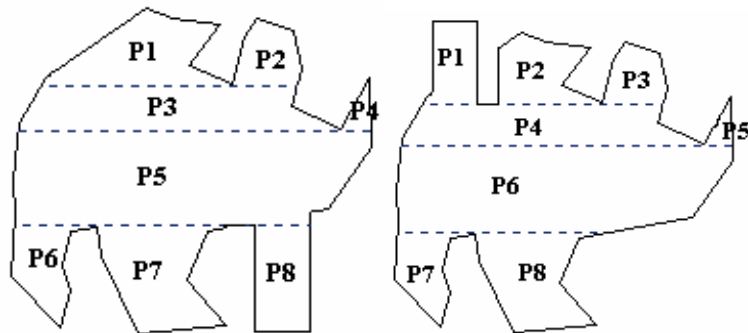


Figure 3.8. Différentes silhouettes ayant le même ordre des lignes de jonction et de disjonction

La valeur NPA de la silhouette de gauche de la figure 3.8 est : 322, alors que celui de la silhouette de droite est : 223. Ces cinq paramètres (NP, NJL, NDJL, ORDER, NPA) constituent un premier index pour la recherche de silhouettes.

### 3.3.2 Second niveau d'indexation de descripteurs LWDOS

Le premier index permet d'accéder à un ensemble de silhouettes ayant une même structure de partitions. Leurs dispositions relativement aux lignes de jonction et de disjonction n'ont pas été prises en compte dans le premier index de même que leurs formes. La figure 3.9 illustre un exemple de telles silhouettes.

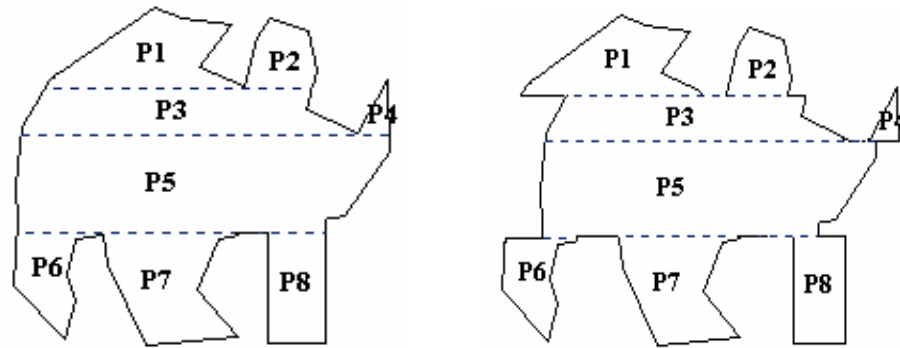


Figure 3.9. Deux silhouettes de même valeurs du premier index (8, 2,1, "001", 322)

Il est donc important de coder la disposition des partitions relativement aux lignes de jonction et de disjonction. Pour ce faire, toutes les configurations possibles ont été recensées et illustrées par la figure 3.10 où w, h, j désignent respectivement les attributs : Low, High et Junction.

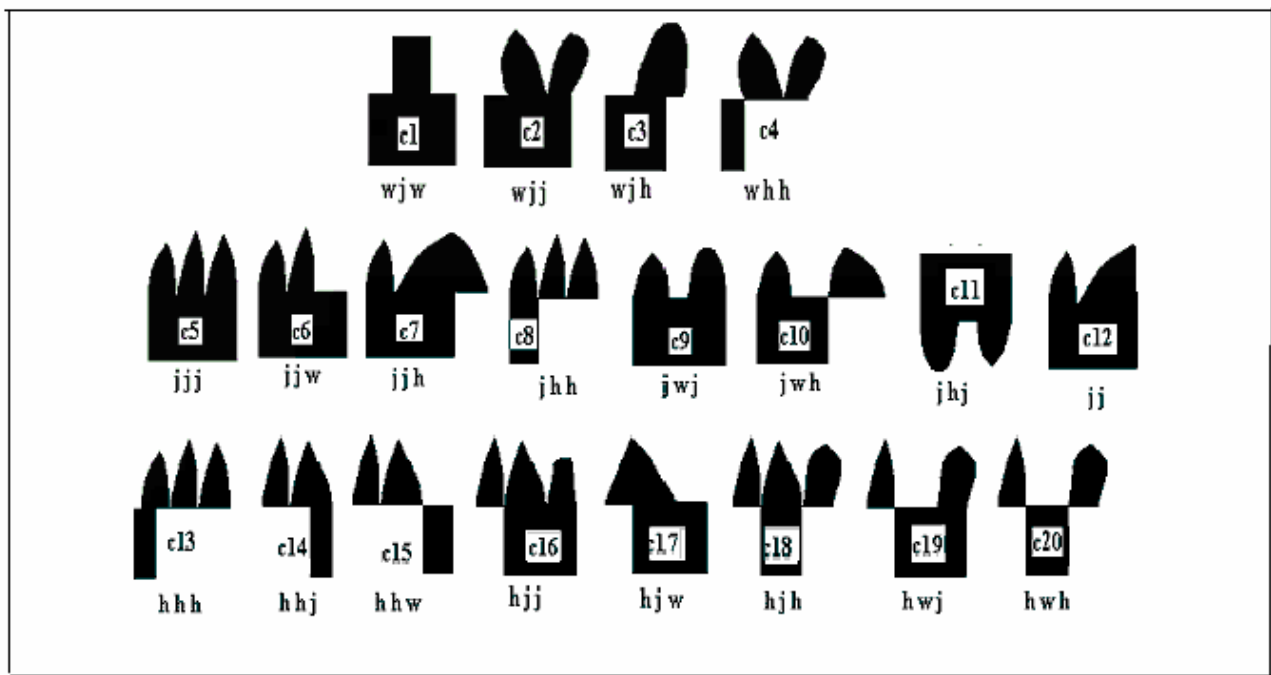


Figure 3.9. Les configurations possibles pour les partitions

A chaque silhouette sera associé une suite de caractères (w, h, j) codant la disposition de ses partitions sur l'ensemble de ses lignes de jonction et de disjonction. Lorsque la silhouette est composée d'une seule partition, la valeur Null est considérée. Cette chaîne de caractères constitue donc le second index.

Pour exemple, à la silhouette de gauche de la figure 3.11 est associé la chaîne suivante : "j j j j j w j", alors que pour celle de droite est associé la chaîne : "j w j w j w j h w j h j h w".

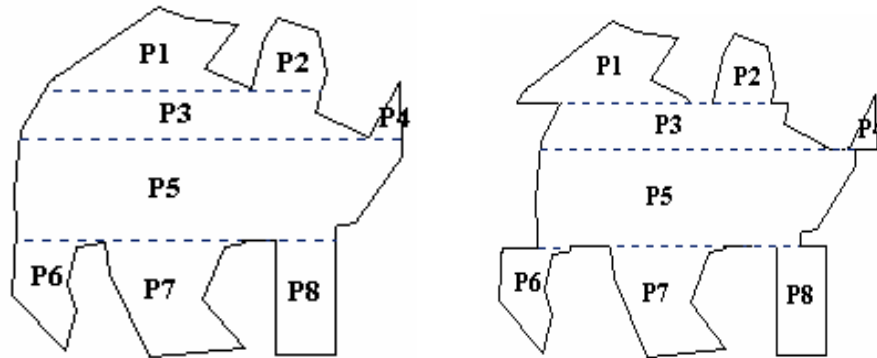


Figure 3.9. Deux silhouettes de même valeurs du premier index (8, 2, 1, "001", 322)

### 3.3.3 Exemple d'indexation de descripteurs LWDOS

Nous donnons dans le tableau 3.1 pour un ensemble de silhouettes exemples leurs deux index calculés.

Nous remarquons que :

- les silhouettes ayant les mêmes valeurs des deux index se différencient uniquement par la forme de leurs partitions, c'est le cas des silhouettes 7 et 8.
- Dans les autres cas, les index des silhouettes se différencient soit par les valeurs du premier ou du second index.

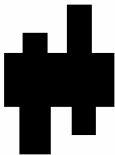
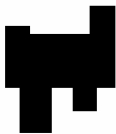
|                                                                                                                                                                                                |                                                                                                                                                                                                |                                                                                                                                                                                                 |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  <p>Silhouette1<br/>Index1=13, 2, 3, 11, 22233<br/>Index2=w j w j w j w j h j h j j w j j<br/>h j j h j</p>   |  <p>Silhouette 2<br/>Index1=15, 3, 3, 22, 223223<br/>Index2=w j w j w j w j h j j h j h j j<br/>h j j w j</p> |  <p>Silhouette 3<br/>Index1=10, 2, 2, 3, 2223<br/>Index2=w j w j w j w j w j w j h j<br/>h j h h j h j h</p> |
|  <p>Silhouette 4<br/>Index1=8, 1, 2, 3, 232<br/>Index2=j j j j j j j</p>                                      |  <p>Silhouette 5<br/>Index1=5, 1, 1, 1, 22<br/>Index2=j j j j</p>                                             |  <p>Silhouette 6<br/>Index1=8, 1, 2, 3, 223<br/>Index2=j j w j j j j j</p>                                   |
|  <p>Silhouette 7<br/>Index1=5, 1, 1, 1, 22<br/>Index2=w j w j w j h j h j h</p>                              |  <p>Silhouette 8<br/>Index1=5, 1, 1, 1, 22<br/>Index2=w j w j w j h j h j h</p>                              |  <p>Silhouette 9<br/>Index1=5, 1, 1, 1, 22<br/>Index2=j w j h j h j h</p>                                   |
|  <p>Silhouette 10<br/>Index1=10, 2, 2, 3, 2232<br/>Index2=w j w j w j w j w j h j h j h<br/>j h j h j h</p> |  <p>Silhouette 11<br/>Index1=8, 1, 2, 3, 322<br/>Index2=j j j j j j j</p>                                   |  <p>Silhouette 12<br/>Index1=5, 1, 1, 1, 22<br/>Index2= w j w j w j h j</p>                                |

Table 3.1. Valeurs des index pour quelques silhouettes

## 3.4 INDEXATION DE DESCRIPTEURS LWDS

### 3.4.1 Introduction

Les descripteurs LWDS des deux formes illustrées par la figure 3.12 ont exactement les mêmes index. Ceci est dû au fait que seuls les contours externes ont été pris en compte dans le calcul des descripteurs. La description en tenant compte des régions internes produit des descripteurs LWDS plus riches en informations. Leur indexation doit donc prendre en compte ces détails internes.

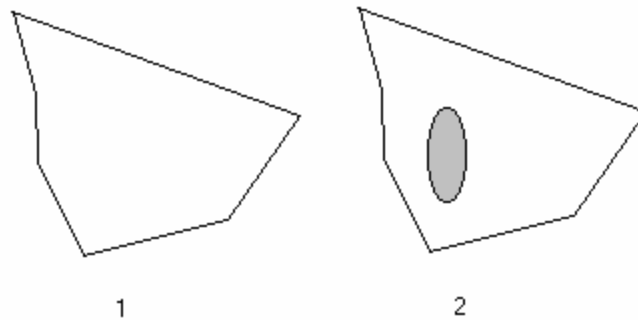


Figure 3.12: Mise en évidence de la région interne.

Dans cette section nous présentons notre solution au problème d'indexation de descripteurs LWDS de formes. Comme le langage LWDS est une application récursive du langage LWDOS sur les régions constituant la forme, notre proposition se base sur la méthode d'indexation des descripteurs LWDOS présentée dans la section précédente. Nous commençons par présenter les éléments qui sont utilisés pour distinguer les formes, nous donnerons ensuite une formulation de l'index proposée.

### 3.4.2 Éléments pour l'indexation de descripteurs LWDS

#### 3.4.2 Éléments pour l'indexation de descripteurs LWDS

- **Nombre de régions**

Le nombre de régions est un élément qui contribue à la différenciation entre formes. La figure 3.13 illustre un exemple de deux silhouettes où la différence est repérée par la présence d'une région interne de plus dans une des deux silhouettes. Concrètement, ce nombre correspond dans le descripteur LWDS au nombre de descripteurs LWDOS.

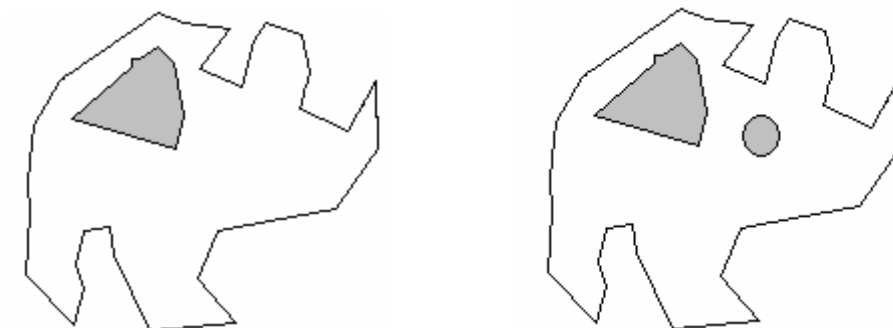


Figure 3.13. Différence entre formes par le nombre de régions internes

- **Nombre de niveaux**

Le nombre de niveaux d'une forme est un élément déterminant pour sa distinction des autres formes. En effet même si deux formes ont la même silhouette externe, les mêmes régions internes en nombre et en forme, l'emplacement de ces régions est un élément déterminant pour la différenciation entre formes. Pour cela, l'information "nombre de niveaux" qui traduit l'imbrication des régions internes dans d'autres régions internes est extraite directement du descripteur LWDS. La figure 3.14 illustre un exemple d'une telle configuration.



Figure 3.14. Mise en évidence de l'importance du nombre de niveaux

- **Inclusion des silhouettes les dans les autres.**

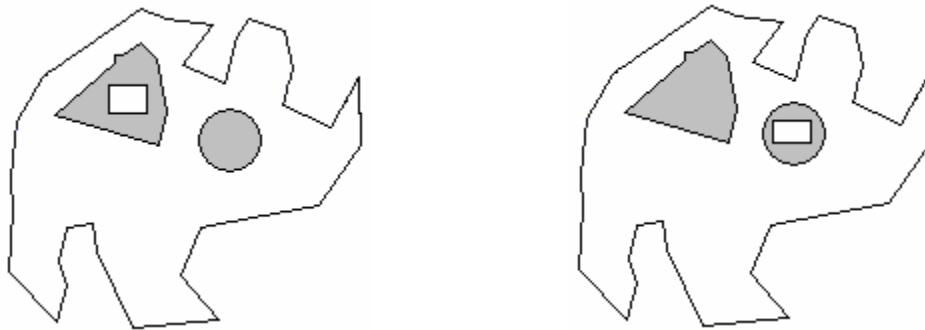


Figure 3.15. Mise en évidence de l'inclusion des silhouettes les dans les autres.

Même si le nombre de niveaux est identique pour deux formes, l'emplacement d'une région par rapport à d'autres est un élément pouvant différencier les formes. Il s'agit donc de traduire la notion d'inclusion et de spécifier pour chaque région la région dans laquelle est incluse.

### 3.4.3 Index de descripteur LWDS

Ayant présenté les éléments importants pour la constitution de l'index d'un descripteur LWDS, nous donnons ci-après l'expression proposée pour le calcul de cet index. La figure 3.16 illustre une image d'une forme contenant plusieurs régions internes et à plusieurs niveaux.

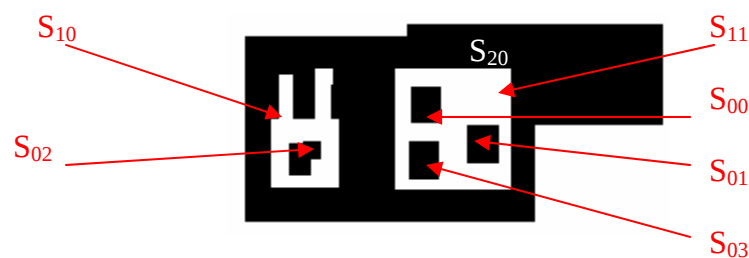


Figure 3.16. Image Exemple

```
<DLWDS>
<Name> Figure 3.15</Name>
  <Shape> Description de la silhouette principale  $S_{2,0}$ 
    <InternalShape>
      <Shape>Position de  $S_{1,0}$  Descripteur LWDOS de  $S_{1,0}$ 
        <InternalShape>Position de  $S_{0,2}$  Descripteur LWDOS de  $S_{0,2}$ </InternalShape>
      </Shape>
    </InternalShape>
  </Shape>
```

```

</InternalShape>

<InternalShape>
  <Shape> Position de S1,1 Descripteur LWDOS de S1,1
    <InternalShape> Position de S0,0 Descripteur de S0,0</InternalShape>
    <InternalShape> Position de S0,1 Descripteur de S0,1</ InternalShape>
    <InternalShape> Position de S0,3 Descripteur de S0,3</ InternalShape>
  </Shape>
</ InternalShape>
</Shape>
</DLWDS>

```

Le descripteur LWDOS de la silhouette S<sub>2,0</sub> qui apparaît deux fois dans la forme est :

```

<DLWDOS> <Name> S2,0 </Name>
<P1><L>r 15 7   r 94 14   r 180 9 </L><R>r 180 7   r 99 6   r 30 3   r 90 7</R></P1>
</DLWDOS>
Son index LWDOS est (Index1=1,0,0,0,Null, index2=Null)

```

Le descripteur LWDOS de S<sub>1,1</sub> qui apparaît quatre fois dans la forme est :

```

<DLWDOS> <Name> S1,1 </Name>
<P1><L>r 93 17   r 180 13</L><R>r 180 13   r 93 16</R></P1></DLWDOS>

```

Son index LWDOS est: (Index1=1,0,0,0,Null, Index2=Null)

Le descripteur LWDOS de S<sub>1,0</sub> qui apparaît une fois dans la forme est :

```

<DLWDOS> <Name> S1,0 </Name>
<CP> <P1><L> r 90 24</L><R>r 180 7   r 90 22</R></P1>
<P2><L>r 90 19</L><R>r 180 5   r 92 21 </R></P2><J> w 3 4 j 2 3 6 w 3 12 j 1 3 7 w 3 4</J>
<P3><L> r 91 30   r 180 32 </L><R>r 90 31 </R></P3> </CP></DLWDOS>

```

Son index LWDOS est: (Index1= 3,1,0,0,2, Index2=wjwjw)

Le descripteur LWDS de la forme de la figure 3.16 peut être représentée moyennant la structure d'arbre comme illustrée par la figure 3.17.

En utilisant ces informations, nous proposons la structure d'un arbre pour la représentation des éléments suivants servant à l'indexation des descripteurs LWDS de formes :

- Index du descripteur LWDOS de chaque région interne et de la silhouette externe.

- Inclusion des silhouettes dans d'autres.
- Nombre de niveaux de l'image
- Nombre de régions internes.

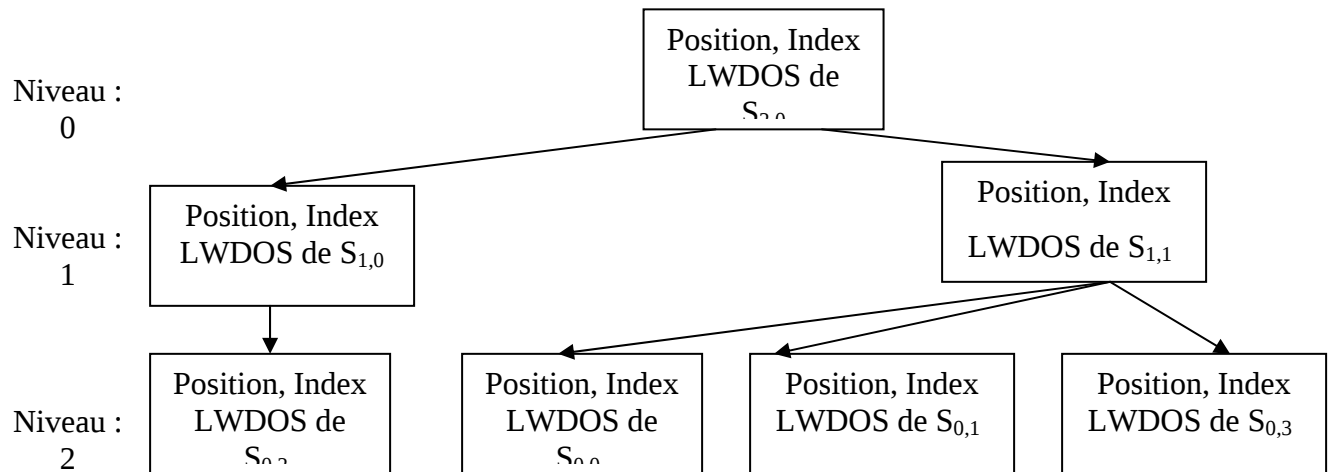


Figure 3.17. Représentation en arborescence du vecteur

A noter que chaque nœud de l'arbre peut apparaître dans d'autres structures associées à d'autres formes.

Pour traduire le nombre de niveaux et l'inclusion des silhouettes, nous proposons de traduire cette structure de la façon suivante où à chaque nœud de niveau  $i$  sera associé le caractère  $i$ :

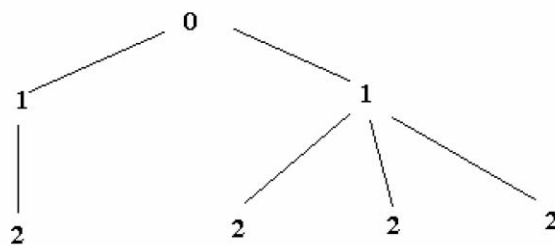


Figure 3.18. Arbre représentant la figure 3.15

Ainsi, la chaîne de caractères correspondant au parcours des nœuds de l'arbre en profondeur donne une description de la forme en termes d'inclusion et de niveau. La chaîne obtenue pour l'arbre de la figure 3.18 : "0 1 2 1 2 2 2" signifie qu'il y a trois niveaux et la silhouette principale contient une silhouette élémentaire et une région contenant trois silhouettes élémentaires.

Pour référencer les silhouettes contenues dans la forme, nous ajoutons à cette chaîne de caractères l'ensemble des entrées dans la table des index LWDOS correspondant à ces silhouettes.

Ainsi, l'index aura la syntaxe suivante :

Index= (0 1 .....carn.) (adresse0, adresse1, ..... adressesn) où adresse i correspond à l'adresse de la silhouette i dans la table des index des descripteurs LWDOS

A partir de cet index, il est possible :

- de compter le nombre de silhouettes de niveau n qui est égal au nombre de caractères n
- de déduire le nombre de niveaux de la forme qui est égal à nombre maximum indiqué par les caractères de la chaîne
- de connaître pour chaque silhouette de la forme, l'ensemble des silhouettes qu'elle contient et la silhouette parent dans laquelle elle est contenue en consultant l'adresse correspondante.

### 3.4.3 Organisation de la base de données

Pour la mémorisation des différents index, nous proposons le schéma illustré par les tables 3.2, 3.3, 3.4 et 3.5.

| Descripteur LWDOS | Identification (Nom de l'objet) |
|-------------------|---------------------------------|
|                   |                                 |
|                   |                                 |
|                   |                                 |

Table 3.2. Descripteurs LWDOS

La première table est réservée pour stocker les descripteurs LWDOS des différentes silhouettes et leurs identification (noms des objets correspondants).

| Index LWDOS | Entrée dans la table des descripteurs LWDOS |
|-------------|---------------------------------------------|
|             |                                             |
|             |                                             |
|             |                                             |

Table 3.3. Index LWDOS

Dans la seconde table seront mémorisés les index des descripteurs LWDOS et leurs entrées dans la table des descripteurs LWDOS.

Dans la troisième table seront mémorisés les descripteurs LWDS des formes et leurs identités (nom des objets correspondants).

| Descripteur LWDS | Identification (Nom de la forme) |
|------------------|----------------------------------|
|                  |                                  |
|                  |                                  |
|                  |                                  |
|                  |                                  |

Table 3.4. Descripteurs LWDS

Enfin, dans la quatrième table seront mémorisés les index des descripteurs LWDS et leurs entrées dans la table des descripteurs LWDS.

| Index LWDS | Entrée dans la table des descripteurs LWDS |
|------------|--------------------------------------------|
|            |                                            |
|            |                                            |
|            |                                            |

Table 3.4. Index LWDS

Pour mieux illustrer cette organisation, nous donnons l'exemple suivant de formes incluant un ensemble de silhouettes élémentaires.

### Exemple d'une base de descripteurs LWDS

Nous commençons par donner la table des descripteurs LWDOS des cinq silhouettes élémentaires. Nous faisons abstraction du détail des partitions. Seuls le descripteur global est donné ainsi que le détail des lignes de jonction et de disjonction.

| Descripteur LWDOS                                                                                                                                                                                                                                                                                      | Identification                                                                                           |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------|
| <p>&lt;DLWDOS&gt; &lt;Name&gt; Silhouette1 &lt;/Name&gt;<br/>         &lt;P1&gt;&lt;/P1&gt; &lt;/DLWDOS&gt;</p>                                                                                                                                                                                        | <p>Silhouette1 </p> |
| <p>&lt;DLWDOS&gt; &lt;Name&gt; Silhouette2 &lt;/Name&gt; &lt;CP&gt; &lt;P1&gt;&lt;/P1&gt; &lt;P2&gt;&lt;/P2&gt;<br/>         &lt;P3&gt;&lt;/P3&gt; &lt;J&gt; w P4 9 j P2 P4 12 w P4 9 j P1 P4 12 w P4 5 j P3 P4 14 w P4 8 &lt;/J&gt;<br/>         &lt;P4&gt;&lt;/P4&gt; &lt;/CP&gt;&lt;/DLWDOS&gt;</p> | <p>Silhouette2 </p> |
| <p>&lt;DLWDOS&gt; &lt;Name&gt; Silhouette3 &lt;/Name&gt; &lt;CP&gt; &lt;P1&gt;&lt;/P1&gt; &lt;P2&gt;&lt;/P2&gt; &lt;J&gt; w<br/>         P3 4 j P2 P3 6 w P3 12 j P1 P3 7 w P3 4&lt;/J&gt; &lt;P3&gt;&lt;/P3&gt; &lt;/CP&gt;&lt;/DLWDOS&gt;</p>                                                        | <p>Silhouette3 </p> |

|                                                                                                                                                                                                                 |                                                                                                 |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------|
| <DLWDOS> <Name> Silhouette4 </Name><br><P1></P1></DLWDOS>                                                                                                                                                       | Silhouette4  |
| <DLWDOS> <Name> Silhouette5 </Name> <CP> <CP> <P1></P1> <P2></P2><br><J> j P1 P4 7 w P4 4 j P2 P4 11</J> <P4></P4> </CP> <P3></P3><J> w P5 9 j<br>P3 P5 12 w P5 9 j P4 P5 22 w 5 27</J><P5></P5> </CP></DLWDOS> | Silhouette5  |

Table 3.6. Table des descripteurs LWDOS

| Index LWDOS              | Entrée dans la table des descripteurs LWDOS |
|--------------------------|---------------------------------------------|
| 1, 0, 0, 0, 1, Null      | 1, 4                                        |
| 4, 1, 0, 0, 3, wjwjwjw   | 2                                           |
| 3, 1, 0, 0, 2, wjwjw     | 3                                           |
| 5, 2, 0, 0, 22, jwjwjwjw | 5                                           |

Table 3.7. Table des index LWDOS

Nous supposons que l'on dispose de quatre formes. Chacune de ces formes contient plusieurs silhouettes dont les descripteurs LWDOS sont mémorisés ainsi que leurs index. Nous donnons dans la table 3.8 les descripteurs LWDS de ces formes en faisant abstraction de descripteurs LWDOS et de leurs positions. La balise <IS> signifie Internal Shape.

| Descripteur LWDS                                                                                                                                                                                                                                                                                                           | Indentification                                                                                 |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------|
| <DLWDS><Name> Forme1 </Name> <Shape><DLWDOS> <Name> Silhouette1<br></Name></DLWDOS> <IS><DLWDOS><Name>Silhouette4</Name></DLWDOS></IS><br><IS><DLWDOS><Name> Silhouette4 </Name></DLWDOS></IS><br><IS><DLWDOS><Name>Silhouette4</Name></DLWDOS></IS><br><IS><DLWDOS><Name>Silhouette4</Name></DLWDOS></IS></Shape></DLWDS> | <br>Forme1 |
| <DLWDS><Name>Forme2</Name><br><Shape><DLWDOS><Name>Silhouette1 </Name></DLWDOS><br><IS><DLWDOS><Name> Silhouette4 </Name></DLWDOS></IS><br><IS><DLWDOS><Name>Silhouette3</Name></DLWDOS></IS></Shape></DLWDS>                                                                                                              | <br>Forme2 |
| <DLWDS><Name>Forme3</Name><Shape><DLWDOS><Name>Silhouette1<br></Name></DLWDOS> <IS> <Shape><DLWDOS><Name>Silhouette3</Name></DLWDOS><br><IS><DLWDOS><Name>Silhouette1</Name></DLWDOS></IS></Shape></IS></Shape><br></DLWDS>                                                                                                | <br>Forme3 |

|                                                                                                                                                                                                                                                                                                                                                                                          |                                                                                               |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------|
| <pre> &lt;DLWDS&gt;&lt;Name&gt;Forme4&lt;/Name&gt; &lt;Shape&gt;&lt;DLWDOS&gt; &lt;Name&gt;Silhouette2 &lt;/Name&gt;&lt;/DLWDOS&gt; &lt;IS&gt;&lt;Shape&gt; &lt;DLWDOS&gt; &lt;Name&gt; Silhouette2 &lt;/Name&gt; &lt;/DLWDOS&gt; &lt;IS&gt; &lt;DLWDOS&gt; &lt;Name&gt; Silhouette1&lt;/Name&gt;&lt;/DLWDOS&gt;&lt;/IS&gt; &lt;/Shape&gt;&lt;/IS&gt;&lt;/Shape&gt;&lt;/DLWDS&gt; </pre> | <br>Forme4 |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------|

Table 3.8. Descripteurs LWDS

Les index de ces formes sont donnés dans la table 3.9. Par exemple, la forme 1 ayant un index = 01111, 14444 est composée :

-d'une silhouette externe (0) dont l'index dans la table des index LWDOS est 1 -de quatre silhouettes internes (de même niveau) et ayant le même entrée dans la table des index de descripteurs LWDOS.

| Index LWDS   | Entrée dans la table des descripteurs<br>LWDS |
|--------------|-----------------------------------------------|
| 01111, 14444 | 1                                             |
| 011,134      | 2                                             |
| 012, 131     | 3                                             |
| 012, 251     | 4                                             |

Table 3.9. Index LWDS

### 3.5 CONCLUSION

Nous avons présenté dans ce chapitre la notion de base de données images et son indexation. Nous avons proposé une solution pour l'indexation d'une base de descripteurs LWDS en se basant sur la méthode d'indexation de descripteurs LWDOS. Dans le prochain chapitre nous présentons les résultats expérimentaux obtenus en utilisant des images réelles de formes. Un logiciel développé sous Visual Basic qui permet d'effectuer les différentes opérations sur la base donnée de descripteurs d'images sous Access sera aussi présenté.

## **CHAPITRE 4**

### **EXPERIMENTATION**

# 4.1 DONNEES D'EXPERIMENTATION

La figure 4.1 illustre les images des formes ayant servi à la validation de la méthode proposée. Les quatre premières images sont celles de silhouettes élémentaires qui ont été utilisées dans la composition des formes illustrées par les onze images restantes.

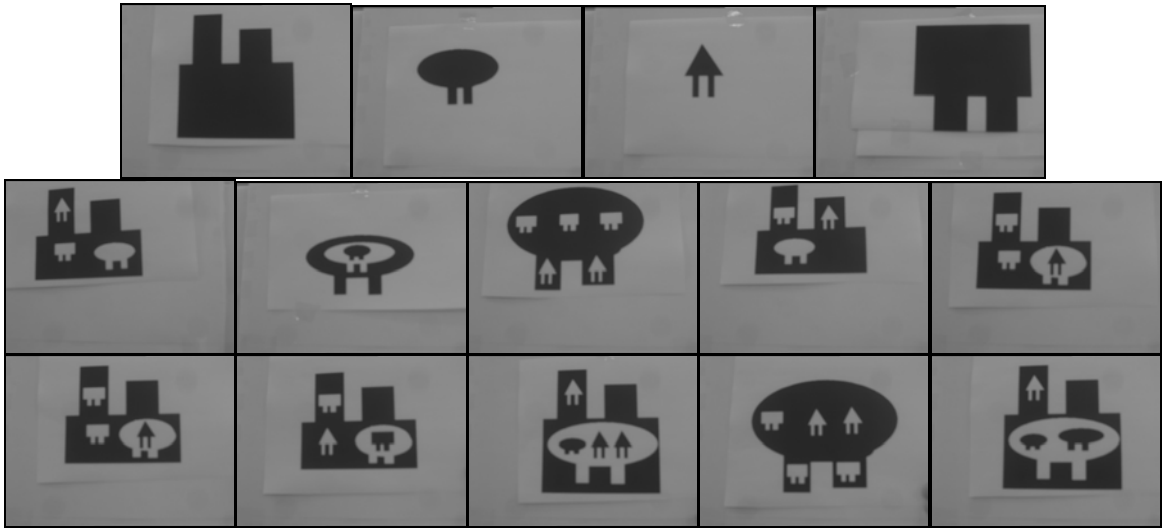


Figure 4.1. Ensemble de formes

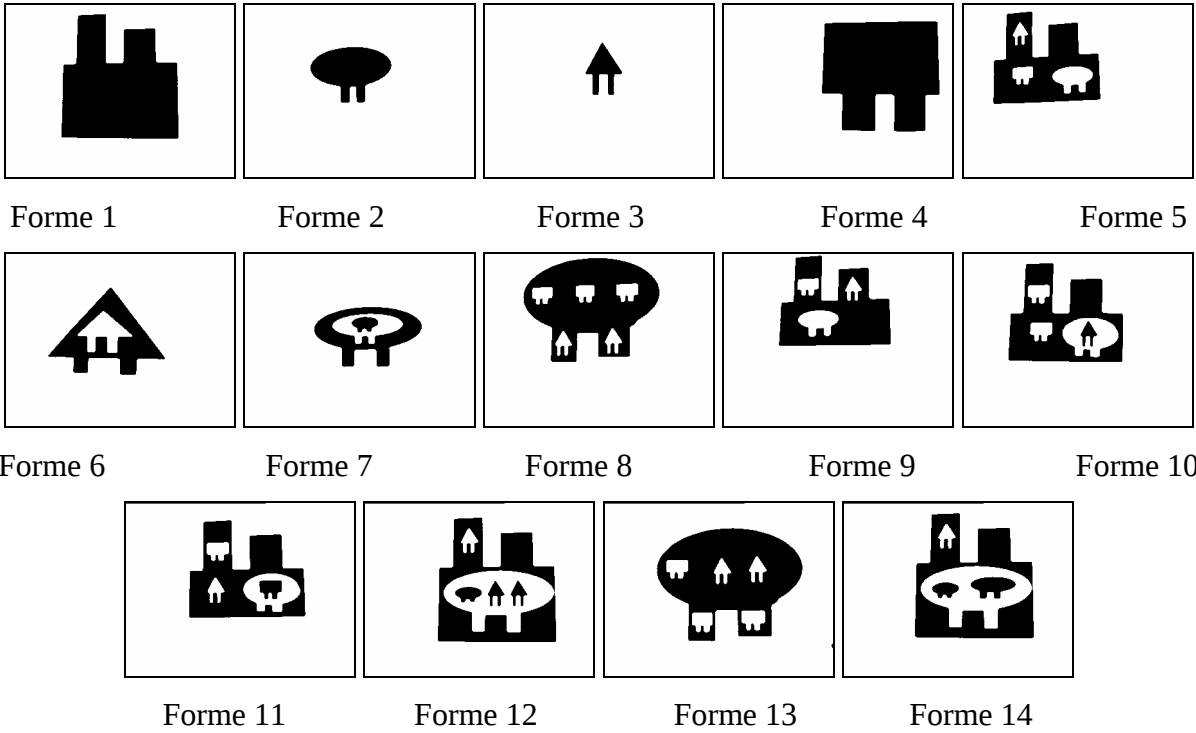


Figure 4.2. Ensemble de formes

Dans la figure 4.2 nous présentons les formes obtenues en appliquant un seuillage (méthode isodata). Cet ensemble de formes constitue la base de données sur laquelle la validation a été faite.

## 4.2 TEST EXPERIMENTAUX

### 4.2.1 Mise en oeuvre d'une base de descripteurs LWDOS

Chaque forme illustrée par les images de la figure 4.1 contient plusieurs silhouettes. Pour la mise en oeuvre de la base de descripteurs LWDOS, chacune de ces formes est traitée dont le but de d'extraire les silhouettes correspondant à chaque niveau et de calculer leurs descripteurs LWDOS. La figure 4.3 montre un exemple d'une forme et de l'ensemble des silhouettes extraites.

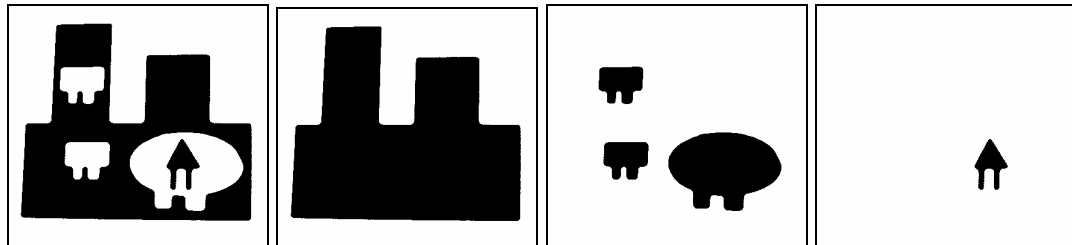


Figure 4.3. Silhouettes extraites pour une forme donnée

Quatre silhouettes ont été recensées et donc quatre descripteurs LWDOS ont été calculés en appliquant d'abord une convolution avec une Gaussienne aux différentes silhouettes dont le but de lisser le contour externe (voir figure 4.4). A noter que cette étape est nécessaire pour éviter des descripteurs avec des partitions ayant une hauteur de quelques pixels ce qui donnera un index erroné.

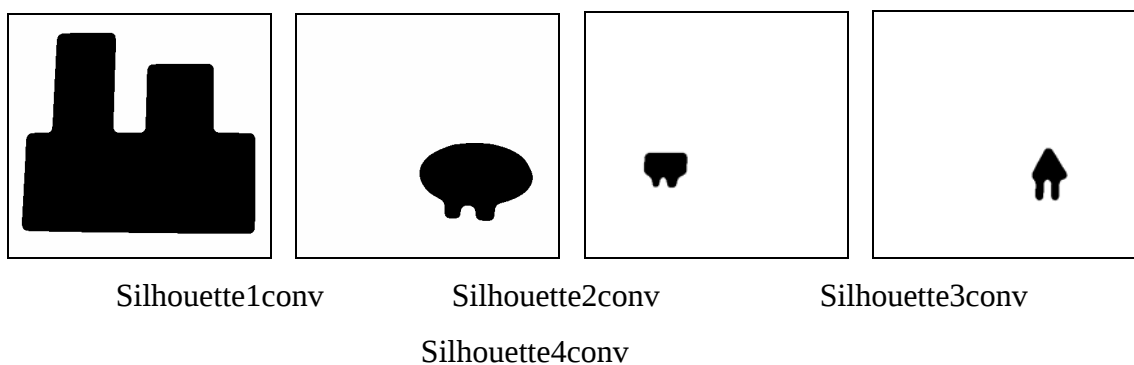


Figure 4.4. Les silhouettes après convolution avec une Gaussienne

Nous donnons ci-après les descripteurs obtenus pour chacune de ces silhouettes élémentaires au nombre de quatre:

#### **Descripteur LWDOS de la silhouette 1**

```
<LWDOS><Name>Silhouette1conv</Name>
<CP><P1><L>r 9 18 r 60 8 r 86 152</L><R> r 0 69 cv 1% 90 156 r 90 5</R></P1>
<P2><L> r 0 9 r 84 109</L><R> r 0 90 r 141 6 r 90 103 r 129 6</R></P2>
<J>w P3 23 J P1 P3 112 w P3 40 J P2 P3 125 w P3 34</J>
<P3><L>r 0 8 r 58 9 r 90 149 r 179 373</L>
<R>r 0 59 r 135 7 r 90 150 r 90 5 r 37 5</R></P3>
</CP></LWDOS>
```

#### **Descripteur LWDOS de la silhouette 2**

```
<LWDOS><Name>Silhouette2conv</Name>
<CP><P1><L>cv 10% 27 82 cv 18% 122 66 r 120 8</L> <R> r 0 45 cv 14% 141 81 r 90 7 r 90 7 r
58 9 cv 6% 26 52 r 31 5</R></P1>
<D>J P1 P2 32 h P1 9 J P1 P3 41</D>
<P2><L> r 90 14 cv 16% 164 17</L><R> r 53 5 r 90 5 r 90 7</R></P2>
<P3><L> r 128 11 r 111 13 r 0 17</L><R> r 90 3 r 90 13 r 45 </R></P3></CP></LWDOS>
```

#### **Descripteur LWDOS de la silhouette 3**

```
<LWDOS><Name>Silhouette3conv</Name>
<CP><P1><L> r 77 30 r 135 14</L><R> r 0 58 r 143 5 r 90 27 r 53 5 r 30 8</R></P1>
<D>J P1 P2 20 J P1 P3 24</D><P2><L> r 117 15</L><R> r 65 14</R></P2>
<P3><L> r 127 5 cv 32% 135 15</L><R> r 90 9 r 51 6</R></P3></CP></LWDOS>
```

#### **Descripteur LWDOS de la silhouette 4**

```
<LWDOS><Name>Silhouette4conv</Name>
<CP><P1><L> r 64 52 r 127 5</L><R> r 0 8 r 123 42 r 90 8 r 45 5</R></P1>
<D>J P1 P2 20 J P1 P3 22</D>
<P2><L> r 106 33</L><R> r 79 31</R></P2>
<P3><L> r 107 34</L><R> r 90 5 r 83 26</R></P3></CP></LWDOS>
```

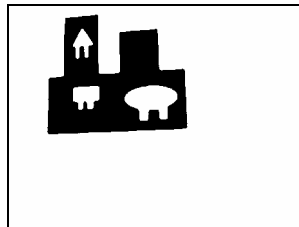
Leurs index calculés pour des quatre descripteurs sont donnés dans la table 4.1.

| Silhouette      | Index1        | Index2 | Adresse |
|-----------------|---------------|--------|---------|
| Silhouette1conv | 3, 1, 0, 0, 2 | wjwjw  | @1      |
| Silhouette2conv | 3, 0, 1, 1, 2 | jhj    | @2      |
| Silhouette3conv | 3, 0, 1, 1, 2 | jj     | @3      |
| Silhouette4conv | 3, 0, 1, 1, 2 | jj     | @4      |

Table 4.1. Table des index de descripteurs LWDOS de silhouettes

#### 4.2.2 Mise en oeuvre de la base de descripteurs LWDS

Une fois que les descripteurs LWDOS ont été organisés, les descripteurs LWDS des différentes formes ont été calculés. Le résultat pour chacune des formes est le suivant en faisant abstraction des descripteurs LWDOS :



Forme5Conv

```
<DLWDS><Name>Forme5Conv</Name>
<Shape>37 139 Silhouette1conv
<Internal Shape>106 148 Silhouette4conv</Internal Shape>
<Internal Shape> 213 295 Silhouette2conv <Internal Shape>
<Internal Shape>232 118 Silhouette3conv </Internal Shape>
</Shape></DLWDS>
```



Forme6Conv

```
<DLWDS><Name> Forme6Conv </Name>
<Shape>17 231 Silhouette4conv
<Internal Shape>96 208 Silhouette4conv
</Internal Shape></Shape></DLWDS>
```



Forme7Conv

<DLWDS><Name> Forme7Conv </Name>

<Shape>48 191 Silhouette2conv

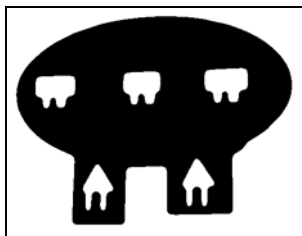
<Internal Shape>

<Shape>72 205 Silhouette2conv

<Internal Shape>81 189 Silhouette2conv

</Internal Shape></Shape>

</Internal Shape></Shape></DLWDS>



Forme8Conv

<DLWDS><Name>Forme8Conv</Name>

<Shape>15 247 Silhouette2conv

<Internal Shape>100 387 Silhouette3conv </Internal Shape>

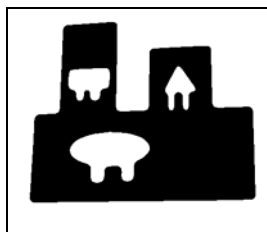
<Internal Shape>106 228 Silhouette3conv </Internal Shape>

<Internal Shape>113 79 Silhouette3conv </Internal Shape>

<Internal Shape>250 315 Silhouette4conv</Internal Shape>

<Internal Shape>260 149 Silhouette4conv</Internal Shape>

</Shape></DLWDS>



Forme9Conv

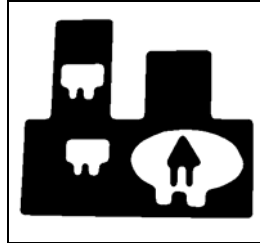
<DLWDS><Name>Forme9Conv</Name>

<Shape>23 165 Silhouette1conv

```

<Internal Shape>96 283 Silhouette4conv </Internal Shape>
<Internal Shape>99 142 Silhouette3conv </Internal Shape>
<Internal Shape>204 169 Silhouette2conv </Internal Shape>
</Shape></DLWDS>

```

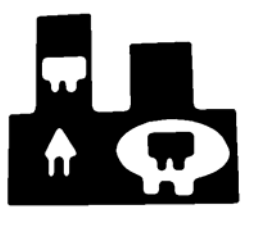


Forme10Conv

```

<DLWDS><Name>Forme10Conv</Name>
<Shape>30 123 Silhouette1conv
<Internal Shape>102 144 Silhouette3conv </Internal Shape>
<Internal Shape>
<Shape>211 283 Silhouette2conv
<Internal Shape>222 287 Silhouette4conv </Internal Shape>
</Shape></Internal Shape>
<Internal Shape>227 104 Silhouette3conv</InternalShape></Shape></DLWDS>

```

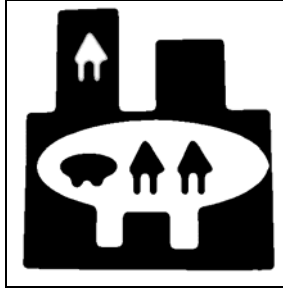


Forme11Conv

```

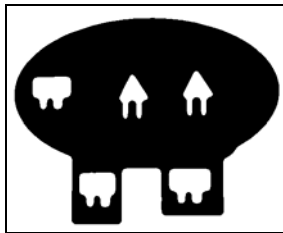
<DLWDS><Name> Forme11Conv </Name>
<Shape>29 161 Silhouette1conv
<Internal Shape>99 157 Silhouette3conv </Internal Shape>
<Internal Shape>
<Shape>203 304 Silhouette2conv
<Internal Shape>225 320 Silhouette3conv </Internal Shape>
</Shape></Internal Shape>
<Internal Shape>218 125 Silhouette4conv </Internal Shape></Shape></DLWDS>

```



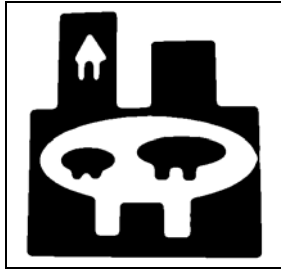
Forme12Conv

```
<DLWDS><Name> Forme12Conv </Name>
<Shape>13 144 Silhouette1conv
<Internal Shape> 51 132 Silhouette4conv </Internal Shape>
<Internal Shape>
<Shape>189 235 Silhouette2conv
<Internal Shape>224 219 Silhouette4conv </Internal Shape>
<Internal Shape>224 293 Silhouette4conv </Internal Shape>
<Internal Shape>243 121 Silhouette2conv </Internal Shape>
</Shape></Internal Shape></Shape></DLWDS>
```



Forme13Conv

```
<DLWDS><Name> Forme13Conv </Name>
<Shape>24 267 Silhouette2conv
<Internal Shape>117 347 Silhouette4conv </Internal Shape>
<Internal Shape>123 232 Silhouette4conv </Internal Shape>
<Internal Shape>129 85 Silhouette3conv </Internal Shape>
<Internal Shape>296 365 Silhouette3conv </Internal Shape>
<Internal Shape>304 148 Silhouette3conv </Internal Shape></Shape></DLWDS>
```



Forme14Conv

```
<DLWDS><Name> Forme14Conv </Name>
<Shape>16 163 Silhouette1conv
  <Internal Shape>54 140 Silhouette4conv </Internal Shape>
  <Internal Shape>
    <Shape>192 241 Silhouette2conv
      <Internal Shape>228 285 Silhouette2conv </Internal Shape>
      <Internal Shape>246 132 Silhouette2conv </Internal Shape>
    </Shape></Internal Shape></Shape></DLWDS>
```

Pour chacune de ces formes nous donnons dans la table 4.2 les index calculés à partir des descripteurs LWDS.

| Forme       | Index1      | Index2       |
|-------------|-------------|--------------|
| Forme5conv  | 0 1 1 1     | @1@4@2@3     |
| Forme6conv  | 0 1         | @4@4         |
| Forme7conv  | 0 1 2       | @2@2@2       |
| Forme8conv  | 0 1 1 1 1 1 | @2@3@3@3@4@4 |
| Forme9conv  | 0 1 1 1     | @1@4@3@2     |
| Forme10conv | 0 1 1 2 1   | @1@3@2@4@3   |
| Forme11conv | 0 1 1 2 1   | @1@3@2@3@3@4 |
| Forme12conv | 0 1 1 2 2 2 | @1@4@2@4@4@2 |
| Forme13conv | 0 1 1 1 1 1 | @2@4@4@3@3@3 |
| Forme14conv | 0 1 1 2 1 1 | @1@4@2@2@2@2 |

Table 4.2. Table des index de descripteurs LWDS de formes

### 4.3 Logiciel de gestion de la base de descripteurs LWDS

La base de données de descripteurs d'images a été implémentée sous ACCESS. L'indexation proposée et les algorithmes de recherche, d'insertion et de suppression ont été

implémentés sous Visual Basic. Nous avons aussi effectué un ensemble de tests de recherche d'images à partir de son index calculée sur l'image requête.

Nous montrons dans ce qui suit avec des illustrations graphiques le gestionnaire de la base des descripteurs LWDS.

Recherche dans la base :

La figure 4.14 illustre une capture d'écran du logiciel développé pour une recherche d'une forme. Dans cette étape, les descripteurs LWDS des différentes silhouette appartenant à l'image sont utilisés.

Données en Entrés

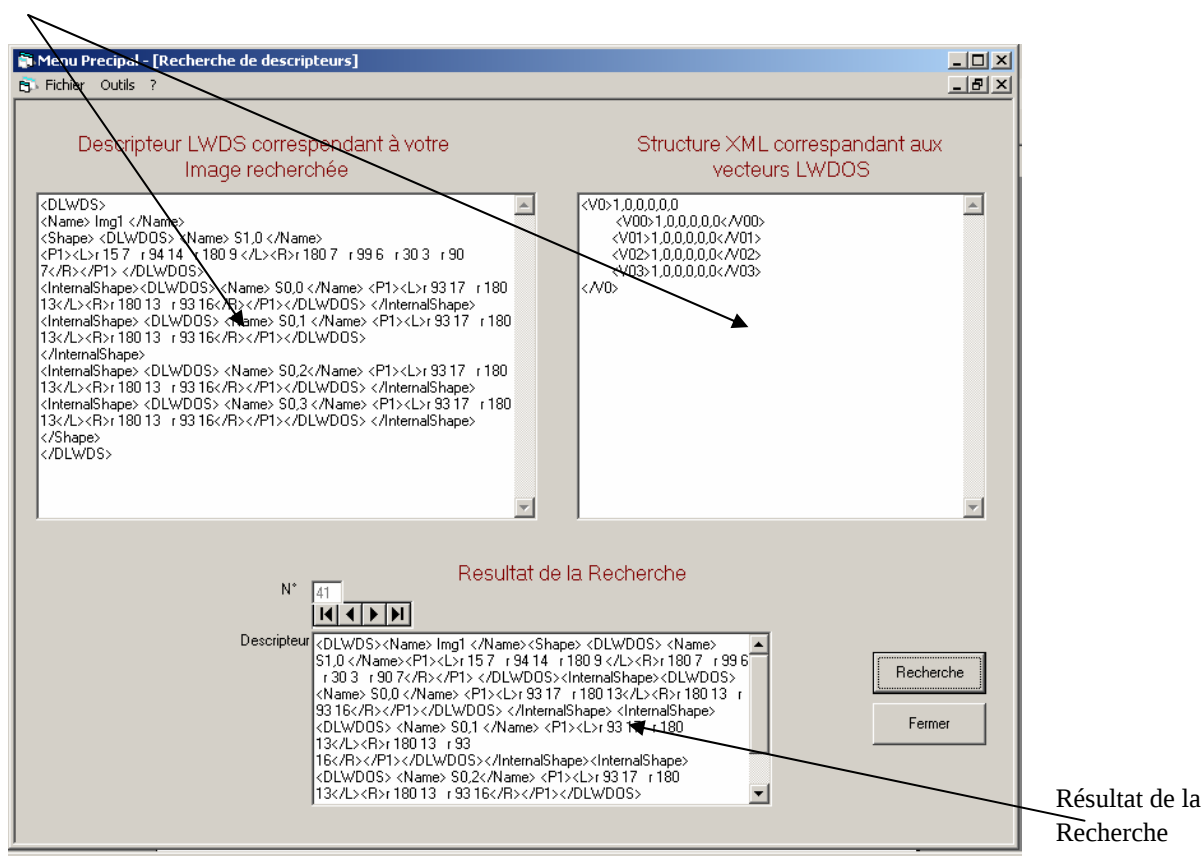


Figure 4.14. Fenêtre de recherche

Le résultat obtenu est affiché comme illustré par la figure 4.15.



Figure 4.15. : Résultats de la recherche

### Insertion dans la base :

Lorsque la recherche ne donne pas de résultat, donc le descripteur n'existe pas dans la base. Le logiciel propose alors son insertion et suivant que les index et les descripteurs LWDOS existent. L'exemple de la figure 4.16 illustre l'insertion du descripteur dans la base.

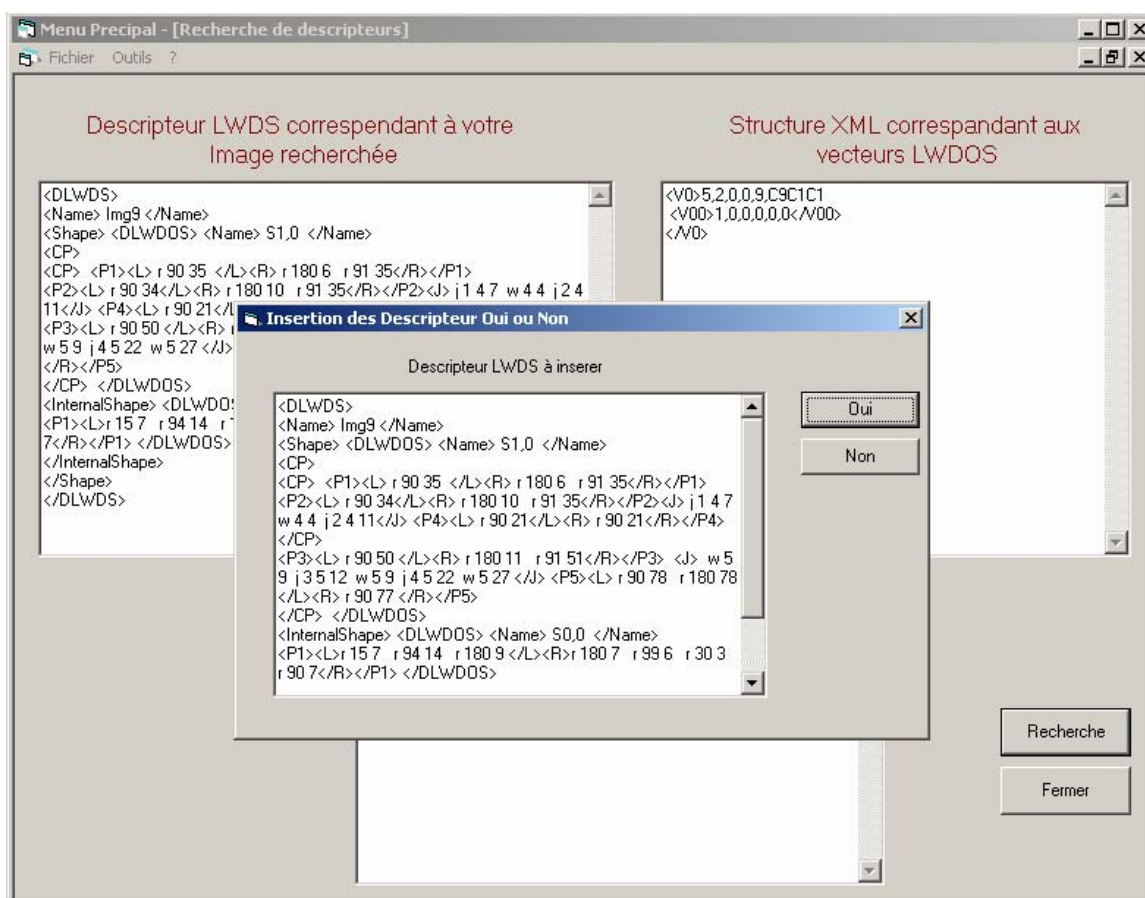


Figure 4.16. Fenêtre d'insertion après une recherche  
L'utilisateur peut aussi insérer un descripteur indépendamment de la recherche.

#### **4.4. CONCLUSION**

Nous avons présenté dans ce chapitre les étapes suivies pour la mise en œuvre d'une base de descripteurs LWDS. Nous avons tout d'abord montré comment à partir d'images réelles, extraire des silhouettes élémentaires, et constituer une base des descripteurs LWDOS. Nous avons aussi montré comment exploiter cette base pour la mise en œuvre des descripteurs LWDS de formes. La solution proposée pour l'indexation a été implémentée et validée sur des images réelles.

# **CONCLUSION**

Dans ce mémoire nous avons présenté d'abord un aperçu de ce qui se fait dans le domaine de la représentation et l'indexation des images en vue de mettre en œuvre une base de données images. Nous avons aussi présenté les méthodes de descriptions de silhouettes et de formes ainsi que les langages LWDOS « **L**angage for **W**riting **D**escriptors of **O**utline **S**hapes » et LWDS « **L**angage for **W**riting **D**escriptors of **S**hapes » permettant d'écrire textuellement ces descriptions.

Nous avons proposé une solution pour l'indexation d'une base de descripteurs LWDS en se basant sur la méthode d'indexation de descripteurs LWDOS. Cette proposition a été ensuite validée sur des images réelles de formes.

Nous avons montré avec notre contribution qu'il est possible d'utiliser les descripteurs LWDS pour la mise en oeuvre d'une base de données images en raison de la facilité qu'ils offrent dans leur indexation.

La suite logique à ce travail est la définition d'une mesure de similarité entre formes en se basant sur les descripteurs LWDS. Ceci permettra de répondre à toute requête de recherche d'images d'objets avec plusieurs réponses classées par ordre décroissant de ressemblance.

Un autre problème ouvert dans cet axe de recherche consiste à résoudre le problème de recherche de formes avec occlusion et recouvrement. Une solution doit être apportée à ce problème en utilisant le descripteur LWDS.

# **ANNEXE**

## 1. Différentes Structures d'indexations

Représenter une image par des vecteurs de descripteurs est une approche très pratique pour la reconnaissance. Mais la recherche de similarités dans des espaces de données de grande dimension est un domaine loin d'être bien maîtrisé. Plusieurs techniques ont été proposées, chacune utilise des concepts différents pour indexer les données.

Les structures d'indexation se classent en deux grandes catégories : celles qui regroupent les données en fonction de leur proximité (data-partitioning index methods), et celles qui découpent à priori l'espace multidimensionnel et qui ensuite stockent les données selon ce découpage (space-partitioning index methods)

### - B-arbre :

Le B-arbre (Arbre balancé) est la structure d'indexation la plus utilisée dans les bases de données unidimensionnelles. Cette structure a été formulée en 1970 par R. Bayer [Bay71]. L'efficacité de cette technique par rapport au séquentiel indexée apparaît clairement dans le cas où de nouvelles insertions ou suppressions existent, en évitant d'effectuer des réorganisations fréquentes des données.

### - Quad-tree :

Le quad-tree est une méthode d'indexation de données utilisée pour le stockage d'un ensemble de données bidimensionnelles. C'est un arbre à précision infinie, où chaque nœud peut avoir jusqu'à quatre fils.

### - R-tree :

Le R-tree développé par A. Guttman[Gut84], est la première méthode d'indexation de données multidimensionnelles, elle est basée sur les b-arbres. Cette méthode permet de consulter efficacement des données pouvant être décrites sous forme de points ou encore de régions dans un espace de grande dimension.

### - X-tree :

Le X-tree (eXtended tree) [Ber96] est une structure d'indexation qui supporte le problème de la recherche de similarités dans un espace de données multidimensionnelles. C'est la variante la plus récente du R-tree. Le problème majeur qui apparaît dans les bases de données structurées par un R-tree est le chevauchement entre les zones contenant les données,

et ceci à différents niveaux de l'arbre ; ce qui implique une détérioration des performances avec l'augmentation de la dimension.

- Kd-tree :

Le kd-tree est un arbre utilisé pour indexer des espaces multidimensionnels. Il est caractérisé principalement par la simplicité de sa mise en œuvre et ses opérations de recherche et d'insertion.

- M-tree :

Le M-tree est une technique d'indexation pour la recherche des similarités dans des espaces de grande dimension dotés d'une métrique.

- TV-tree :

Le TV-tree (Telescopic-Vector Tree) repose sur l'hypothèse qu'on n'a pas besoin de toutes les composantes d'un vecteur pour l'indexer. Il suffit alors d'utiliser un nombre variable de dimensions pour l'indexation, et qui est adapté au nombre d'objets à indexer. Donc, des vecteurs "télescopiques" seront utilisés pour l'indexation.

- Pyramid-tree :

Berchtold, Bohm et Kriegel [Bec90] proposent le pyramid-tree, une technique d'indexation de vecteurs imprécis dans un espace de données de grande dimension. Cette technique consiste à transformer l'espace de données vers l'espace canonique de données multidimensionnel  $[0,1]^d$ . Ensuite cet espace est divisé en deux étapes : au début, il est découpé en  $2^d$  pyramides. Chaque pyramide a son sommet placé au centre de l'espace  $(0.5, 0.5, \dots, 0.5)$  et sa base est l'hyperplan de dimension  $(d-1)$ . En second lieu, chaque pyramide est divisée en plusieurs tranches parallèles à sa base, chacune de ses tranches correspond à une page de données d'un B+-tree.

- VA-File :

Cette technique vient pour parer au problème d'indexation d'espaces de grande dimension. Elle a, en effet, été conçue dès le départ de façon à ce que son comportement ne dégénère pas trop rapidement lorsque des données ayant de nombreuses dimensions sont indexées ; et ce en améliorant les performances de la recherche séquentielle, puisque celle-ci

s'avère la plus efficace en grandes dimensions. La technique du VA-File utilise une première phase d'approximation permettant de diminuer le nombre de descripteurs à comparer. Elle gère deux ensembles de données : un fichier contenant les descripteurs et un autre contenant une approximation géométrique de ces descripteurs.

## Références

- [1]Ronald Alfarez and Yuan-Fang Wang. "Image Database Indexing using a combination of invariant shape and color descriptions" The second international conference on information fusion Sunnyvale CA. USA 1999.
- [2]Laurent Amsaleg , Patrick Gros , Rim Mezhoud "Mise en base d'images indexées par des descripteurs locaux : problèmes et perspectives " INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE Rapport de recherche Mars 2000.
- [3]Andersen C., Jones S., Crowley J. "Appearance based processes for visual navigation" In: Proc. 5th Internat. Symp. on Intelligent, Robotic, Sweden, July. 1997.
- [4]K. Arbter, W. E. Snyder. H. Burkhardt and G. Hirzinger. "Application of Affine-Invariant Fourier Descriptors to Recognition of 3-D Objects". IEEE Trans. PAMI 12(7):640-647, 1990.
- [5]H. Asada and M. Brady. "The Curvature Primal Sketch". MIT AI Memo 758, 1984.
- [6]J. R. Bach et al. "Virage Image Search Engine: An Open Framework for Image Management". SPIE Conf. On Storage and Retrieval for Image and Video Databases IV, San Jose, CA, pp.76-87, 1996.
- [7]Barbeau Jérôme, Stamon Georges et Vignes-Lebbe Régine. "Indexation d'images biologiques par triangulation de Delaunay : Application à l'indexation d'ailes d'insectes". TAIMIA 2001, Atelier de traitements d'images, organisé par l'INRA
- [8]R. Bayer. "Binary B-trees for virtual memory" Proceedings of the ACM 1971 SIGFIDET November 1971
- [9]N. Beckmann, H.P. Kriegel, R. Shneider, B. Seeger. "The R\*-tree: An efficient and robust access method for points and rectangles." Proc. ACM SIGMOD International conference on Management of data, Atlantic City, New Jersey, 1990.
- [10]S. Berchtold, D. A. Keim, and H.-P. Kriegel. "The X-tree : An Index Structure for High-Dimensional data" 1996.
- [11]Boaz J. Super." Fast Retrieval of Isolated Visual Shapes" Computer Vision and Image understanding, 85(1)1-21, 2002
- [12]S. Bouagar, "Utilisation de la connaissance en vision stéréoscopique", thèse de magister en Informatique, USTHB, Juillet 2003.
- [13]Kim L. Boyer, Ravi Srikantiah, and Patrick J. Flynn "Saliency Sequential Surface Organization for Free-Form Object Recognition" Computer Vision and Image understanding, 88, 152-188(2002).
- [14]K. Chakrabarti, M. O. Binderberger, K. Porkaew and S. Mehrotra. Similar Shape Retrieval in MARS. In Proc. of IEEE Int. Conf. On Multimedia and Expo (CD-ROM), New-York, USA, 2000.
- [15]Charles T. Zahn and Ralph Z. Roskies. Fourier Descriptors for Plane closed Curves. IEEE

Trans. On Computer, C-21(3):269-281, 1972.

[16]R. Chellappa and R. Bagdazian. "Fourier Coding of Image Boundaries". IEEE Trans. PAMI-6(1):102-105, 1984.

[17]Daniel Berwick, and Sang Wook Lee, "Spectral gradients for color-based object recognition and indexing", Computer Vision and Image Understanding (2004)

[18]A. Del Bimbo and P. Pala, "Shape indexing by multi-scale representation, Image Vision Comput". 17, 1999, 245–261.

[19] A.Del Bimbo, P. Pala, and S. Santini, "Visual Image Retrieval by elastic Deformation of Objects Shapes", Proc. IEEE VL'94, Intl Symp. Visual languages, St Louis, Mo, Oct. 1994.

[20]Stéphane Derrode , Rim Mezhoud et Faouzi Ghorbel "Comparaison de deux familles complètes de descripteurs de formes pour l'indexation de bases d'objets 2D à niveaux de gris" Département Image et Traitement de l'Information, Ecole Nationale Supérieure des Télécommunications de Bretagne, BP 832, 29285 Brest cedex, France. Avril 2000

[21]S. Derrode "Représentation de formes Planes à niveaux de gris par différentes approximations de Fourier-Mellin analytique en vue d'indexation de base d'images." Thèse de Doctorat à l'Université de Rennes I, Décembre 1999.

[22]J. Dowe. "Content-based Retrieval in Multimedia Imaging". In Proc. SPIE Storage and Retrieval for Image and Video Databases, pp.164-167, 1993.

[23]G. Dudek and J. K. Tsotsos, "Shape representation and recognition from multiscale curvature", Comput. Vision Image Understand. 68, 1997, 170–189.

[24]Arnaud DUPUIS, Pascal VASSEUR, Josselin HARP "Détection d'objets basée sur une approche multi primitives " ORASIS2001 IRIT Université Paul Sabatier 118 rue de Narbonne 31062 Toulouse Cedex4.

[25]D. H. Eberly, "Geometric Methods for analysis of Ridges in N-Dimensional Images" PhD Thesis, Univesity of North Carolina at Chapel Hill, 1994.

[26]J. Feder. "Towards Image Content-based Retrieval for the World-Wide Web". Advanced Imaging 11(1):26-29, 1996.

[27]M. Flickner et al. "Query by Image and Video Content: the QBIC System". IEEE Computer 28(9):23-32, 1995.

[28]Funt, B.V-, Finlayson, G.D., "Color constant color indexing". IEEE Trans. Pattern Anal. Machine Intell. 17 (5), 522-529. 1995.

[29]B.V. Funt, G.D. Finlayson, "Color constant color indexing", PAMI 17 (5) (1995) 522–529.

[30]A. Guttman. "R-trees: A dynamic index structure for spatial searching." Proc. ACM SIGMOD conference, Boston, June 1984.

[31]Hayit Greenspan, Guy Dvir, and Yossi Rubner, " Context-dependent segmentation and

matching in image databases”, Computer Vision and Image Understanding (2004)

[32]G. Healey, D. Slater, “Global color constancy: recognition of objects by use of illumination-invariant properties of color distributions”, JOSA 11 (11) (1994) 3003–3010.

[33]Horst, B., Aies, L., “Robust recognition of scaled eigenimages through a hierarchical approach”. In: Proc. CVPR98. IEEE Computer Soc. Press, Silver Spring, MD, pp. 664-670. 1998.

[34]T. S. Huang, S. Mehrotra and K. Ramachandran. “Multimedia Analysis and Retrieval System (MARS) Project”. In Proc. of 33rd Annual Clinic on Library Application of Data Processing—Digital Image Access and Retrieval, pp.101-117, 1996.

[35]C.-L. Huang and D.-H. Huang. “A Content-based Image Retrieval System”. Image and Vision Computing, 16:149-163, 1998.

[36]Chien-Yuan Huang, Octavia I.Capms, and Tapas Kanungo. “Object Recognition Using Appearance-Based Part and Relations” Proc. Of IEEE Conf. on Computer Vision and Pattern Recognition 877- PUERTO RICO 1997.

[37]Khalid Idrissi, Guillaume Lavoué, Julien Ricard, and Atilla Baskurt “Object of interest-based visual navigation, retrieval, and semantic content identification system” Article in press Computer Vision and Image Understanding 2004

[38]Ikeuchi, K., Ohba, K., “Detectability, uniqueness and reliability of eigen Windows for stable verification of partially occluded objects”. IEEE Trans. Pattern Anal. Machine Intell. 19 (9), 1043-1048. 1997.

[39]T. Kato. “Database Architecture for Content-based Image Retrieval”. In Image Storage and Retrieval Systems, Proc SPIE 1662, pp.112-123, 1992.

[40]H. Kauppinen, T. Seppanen and M. Pietikainen. ”An Experimental Comparison of Autoregressive and Fourier-Based Descriptors in 2D Shape Classification”. IEEE Trans. PAMI-17(2):201-207, 1995.

[41]Krumm, J., “Eigen features for planar pose measurement of partially occluded objects”. In: Proc. IEEE CS Computer Vision and Pattern Recognition, pp. 55-60. 1996.

[42]Ting-Sheng Lai “CHROMA: A Photographic Image Retrieval System“, School of Computing, Engineering and Technology University of Sunderland Sunderland United Kingdom, January 2000

[43]S. Larabi, S. Bouagar, F. M. Trespaderne, E. F. Lopez, “LWDOS: Language for Writing Descriptors of Outline Shapes”, In the LNCS proceeding of Scandinavian Conference on Image Analysis, June 29 - July 02, Gotborg, Sweden, 2003.

[44]Longin Jan Latecki and Rolf Lak” amper and Ulrich Eckhardt “Shape Descriptors for Nonrigid Shapes with a Single Closed Contour.” IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 424-429, 2000.

[45]G. Lavoué K. Idrissi A. Baskurt “Accès aux bases d’images par navigation visuelle“ In Proc. Of colloque CORESA’03, Lion, France, pp 55-58, 2003.

- [46]Lifeng Lin, Stan Selaroff “Color region grouping and shape recognition with deformable models”. Boston University Computer Science technical Report 97-019, November, 1997.
- [47]Erwan Loisant , José Martinez, Hiroshi Ishikawa , Manabu Ohta, Kaoru Katayama<sup>emme</sup><sub>29</sub>  
 “Recherche d’Images par la Navigation “ Journée Science et Technologie JSF2003 , Tokyo.
- [48]G. J. Lu and A. Sajjanhar. “Region-based Shape Representation and Similarity Measure Suitable for Content-based Image Retrieval”. *Multimedia Systems*, 7(2):165-174, 1999.
- [49]W. Y. Ma and B. S. Manjunath. “Netra: A Toolbox for Navigating Large Image Databases”. In *Proc IEEE International Conference on Image Processing (ICIP97)*, Vol.I, pp.568-571, 1997.
- [50]Sean D. MacArthur, Carla E. Brodley, Avinash C. Kak, and Lynn S. Broderick. “Interactive Content-Based Image Retrieval Using Relevance Feedback” *Computer Vision and Image Understanding* 88, 55-75 (2002)
- [51]S. Matusiak, M. Daoudi “Indexation Robuste et Recherche d’Images par le Dessin“ Compression et représentation des signaux audiovisuels (CORESA 98), LANNION, FRANCE, 9-10 JUIN 1998.
- [52]R. Mezhoud, L. B. Hamida, S. Derrode, S. Mhiri, F. Ghorbel “Nouvelle approche  
<sup>7èmes</sup>  
 d’indexation de bases d’images par le contenu”journée d’étude et d’échange CORESA2001, Dijon, Novembre 2001.
- [53]Mohit Agarwai, Gaurav Jain, Santanu Chaudhury “Indexing for local appearance-based recognition of planar objects”
- [54]Farzin Mokhtarian “Silhouette-Based Isolated Object Recognition through Curvature, Scale Space” *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, VOL. 17, NO. 5, MAY 1995.
- [55]F. Mokhtarian, S. Abbasi and J. Kittler. “Robust and Efficient Shape Indexing Through Curvature Scale Space”. *Proc. British Machine Vision Conference*, pp.53-62, Edinburgh, UK, 1996.
- [56]F. Mokhtarian, S. Abbasi, and J. Kittler, “Efficient and robust retrieval by shape content through curvature scale space”, in *Image Databases and Multi-Media Search* (A. Smeulders and R. Jain, Eds.), pp. 51–58, World Scientific, New Jersey, 1997.
- [57]Maurizio Cibelli, Michele Nappi, Maurizio Tucci “ABI: analogy-based indexing for content image retrieval”, *Image and Vision Computing* 2004.
- [58]Murase, H., Nayar Shree, K., “Visual learning and recognition of 3D objects from appearance”. *Internat. J. Comput. Vision* 14 (1), 5-24. 1995.
- [59]C. Nastar et al. “Surfimage: A Flexible Content-Based Image Retrieval System”. In *Proc. ACM Multimedia’98* , Bristol, UK, 1998.

- [60]S. Nayar, R. Bolle, "Reflectance based object recognition", IJCV 17 (3) (1996) 219–240.
- [61]W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, P. Yanker, C. Faloutsos and G. Taubin. "The QBIC Project: Querying Image By Content Using Color, Texture and Shape". In Proc. SPIE Storage and Retrieval for Image and Video Databases, vol.1908, pp.173-187, 1993.
- [62]V. E. Ogle and M. Stonebraker. "Chabot: Retrieval from a Relational Database of Images". IEEE Computer, 28(9):40-48, 1995.
- [63]P. J. van Otterloo. "A contour-Oriented Approach to Shape Analysis". Prentice Hall International (UK) Ltd. pp.90-108, 1991.
- [64]T. Pavlidis. "Algorithms for Shape Analysis of Contours and Waveforms". IEEE Trans. PAMI-2(4):301-312, 1980.
- [65]J. Hu and T. Pavlidis, "A hierarchical approach to efficient curvilinear object searching", Comput. Vision Image Understand. 63, 1996, 208–220.
- [66]A. Pentland et al. "Photobook: Tools for Content-Based Manipulation of Image Databases". International Journal of Computer Vision, 18(3):233-254, 1996.
- [67]E. Persoon and K. S. Fu. "Shape Discrimination Using Fourier Descriptors". IEEE Trans. On Systems, Man and Cybernetics, Vol.SMC-7(3):170-179, 1977.
- [68]Pope, A.R., Lowe, D.G., "Learning appearance models for object recognition". In: Ponce, J., Zisserman, A., Hebert, M. (Eds.), Proc. Internat. Workshop on Object Representation for Computer Vision II, Cambridge, UK. Springer, Berlin, pp. 201-219. 1996.
- [69]B.G. Prasad, K.K. Biswas, et S.K. Gupta Region-based image retrieval using integrated color, shape, and location index. Computer Vision and Image Understanding (2004)
- [70]Rajesh Rao, P.N., "Dynamic appearance-based recognition". In: Proc. 1997 Conf. on Computer Vision and Pattern Recognition (CVPR '97), pp. 199-205.
- [71]T. W. Rauber. "Two-Dimensional Shape Description". Technical Report: GR UNINOVA-RT-10-94, University Nova de Lisboa, Portugal, 1994.
- [72]Y. Rui, T. S. Huang., and S. Mehrotra, "Content-based image retrieval with relevance feedback in MARS", in Proceedings of the IEEE International Conference on Image Processing, 1997.
- [73]D. Slater, G. Healey, "The illumination-invariant recognition of 3d objects using local color invariants", PAMI 18 (2) (1996) 206–210.
- [74]J. R. Smith and S. F. Chang. "VisualSEEK: A Fully Automated Content-based Image Query System". ACM Multimedia'96, Boston, MA, Nov.20, 1996.
- [75]Fridtjof Stein, and Gerard Medioni "Structural Indexing: Efficient 3-D Object Recognition". IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 14, NO. 2, FEBRUARY 1992

- [76]C. de Trazegnies , C. Urdiales, A. Bandera, F. Sandoval, "Planar shape indexing and retrieval based on Hidden Markov Models". Dpto. Tecnologia Electronica, E.T.S.I. Telecomunicacion, Universidad de Malaga, Campus de Teatinos, Malaga 29071. 2001
- [77]Turk, A., Pentland, A.P., "Face recognition using eigen-faces". In: Proc. CVPR 1991, pp. 586-591
- [78]P. Vandeborre, M. Daoudi, C. Chaillou "3D Models Recognition from a 2D sketch" CESA'98 IMACS Multiconference, Nabeul-Hammamet, Tunisia, April 1-4, 1998
- [79]J. Yang, L. Wenyin, H. Zhang and Y. Zhuang. "Thesaurus-aided Approach for Image Browsing and Retrieval". In Proc. IEEE Conf. on Multimedia and Expo, pp.313-316, Tokyo, Japan, August 22-25, 2001.
- [80]Dengsheng Zhang, "Image Retrieval Based on Shape", PhD thesis, Faculty of Information Technology, Monash University, USA, March, 2002.
- [81]S. Larabi, Textual description of shapes, submitted to ICIP'2005
- [82] D. Chetverikov: "A Simple and Efficient Algorithm for Detection of High Curvature Points in Planar Curves". 10<sup>th</sup> International Conference, CAIP 2003, Groningen, The Netherlands, August 25-27, 2003
- [83]Extensible Markup Language. World Wide Web Consortium. "[HTTP: //www.w3.org/xml/](http://www.w3.org/xml/)"
- [84] M. SAFAR, C. SHAHABI AND X. SUN. "Image Retrieval by Shape: A Comparative Study". IN PROC. OF IEEE INT. CONF. ON MULTIMEDIA AND EXPO (CD-ROM), NEW YORK, USA, 2000.

## ***Abstract***

*Images representation is an important problem for object recognition process in artificial vision. Indeed, it is necessary to have a database of images representing the different views of the object in order to identify any object in the acquired image. The size of this database increases considerably according to the number of objects models to treat. It is therefore necessary to store instead of the image its description in order to reduce space memory and algorithm complexity.*

*We present in this thesis the two methods developed in LRIA laboratory of USTHB University for describing shapes using both the interior and exterior contours. The language LWDS (Language for Writing Descriptors of Shapes) allowing writing textually these descriptions is also presented. As contribution, we propose a method for indexing descriptors of shapes written using the language LWDS. A validation of our approach using real images is also presented and discussed.*

***Key words:*** *Image description - Silhouette - Shape – Indexing - Language*