



# Contribution à la description des formes par la transformée en ondelettes

## THÈSE

présentée et soutenue publiquement le 17 Mai 2004

pour l'obtention du

Doctorat de l'université de La Rochelle  
(spécialité informatique)

par

Kimcheng KITH

### Composition du jury

<i>Rapporteurs :</i>	Patrice DALLE	Université Paul Sabatier, Toulouse III
	Mohamed DAOUDI	Ecole Polytechnique de Tours
<i>Examineurs :</i>	Guy WALLET	Université de La Rochelle
	Bertrand VACHON	Université de La Rochelle
	El-hadi ZAHZAH (directeur de thèse)	Université de La Rochelle
	Laurent MASCARILLA	Université de La Rochelle





# Remerciements

Ce travail a été réalisé au sein du Laboratoire Informatique, Image et Interaction (L3I) de l'Université de La Rochelle et de l'École Doctorale de La Rochelle à l'aide du financement du Ministère de l'Éducation Nationale, de la Recherche et de la Technologie (MENRT). Je remercie Monsieur le Professeur P. ESTRAILLIER, Directeur de L3I, de m'avoir accueilli dans son laboratoire.

J'exprime toute ma gratitude à Monsieur El-Hadi ZAHZAH, Maître de Conférence à l'Université de La Rochelle, pour avoir proposé et encadré ce sujet pendant ces années de travail au sein du Laboratoire L3I. Je le remercie de m'avoir encouragé et soutenu pendant cette période ; je lui suis très reconnaissant de l'intérêt qu'il a porté à mon travail, de même que pour son aide et sa disponibilité. Grâce à lui, j'ai découvert un domaine de recherche qui aujourd'hui me passionne.

J'adresse mes plus vifs remerciements à Monsieur Patrice DALLE, Professeur à l'Université Paul Sabatier, et à Monsieur Mohamed DAOUDI, Professeur à École Polytechnique de l'Université de Tours, pour avoir accepté d'être les rapporteurs de ce mémoire, pour la pertinence de leurs remarques, et leur participation à ce jury.

Je souhaite exprimer toute ma reconnaissance à Monsieur Guy WALLEY, Professeur à l'Université de La Rochelle pour son accord qui m'a permis de suivre son cours sur la THÉORIE DES ONDELETTES ET APPLICATIONS, qui est précieux pour cette thèse, et sa participation à ce jury.

Je remercie également Monsieur Bertrand VACHON, Professeur à l'Université de La Rochelle, et Monsieur Laurent MASCARILLA, Maître de Conférence à l'Université de La Rochelle, de me faire l'honneur de participer au jury.

Merci à R. COUDRAY, pour le partage du bureau et pour son soutien dans mon travail. Merci aussi aux re-lecteurs et correcteurs de cette thèse : T. BOUWMANS, D. SEMANI-DELMY, C. PEYRAMAURE et F. CHENEVIÈRE.

Merci à toutes les personnes qui m'ont entourées et aidées (directement ou indirectement) durant ces quatre années.



*Je dédie cette thèse à mes parents,  
aux autres membres de ma famille  
et à mes amis  
qui m'ont soutenu au long de ce parcours.*



# Table des matières

<b>Remerciements</b>	<b>i</b>
<b>Table des figures</b>	<b>ix</b>
<b>Liste des tableaux</b>	<b>xi</b>
<b>Chapitre 1 Introduction générale</b>	<b>1</b>
1.1 Contexte . . . . .	1
1.2 Approche proposée . . . . .	3
1.3 Contributions . . . . .	4
1.4 Plan du mémoire . . . . .	5
<b>Chapitre 2 Les outils de la reconnaissance des formes</b>	<b>7</b>
2.1 Introduction . . . . .	7
2.2 La représentation du contour d'un objet 2D . . . . .	8
2.2.1 La représentation par un nombre complexe ou par les coordonnées cartésiennes ou polaires . . . . .	8
2.2.2 La représentation par la chaîne des codes de direction . . . . .	8
2.2.3 L'approximation polygonale . . . . .	9
2.2.4 La distance au centre de gravité . . . . .	11
2.2.5 La fonction angulaire cumulative . . . . .	11
2.2.6 La représentation par la courbure . . . . .	12
2.2.7 La représentation par la surface fermée . . . . .	13
2.3 Paramétrisations invariantes . . . . .	13
2.3.1 Paramétrisation par abscisse curviligne ou la longueur d'arc . . . . .	14
2.3.2 Paramétrisation par la longueur affine . . . . .	14
2.3.3 Paramétrisation par la surface fermée . . . . .	15
2.3.4 Discussion . . . . .	16
2.4 Fixation du point de départ . . . . .	16
2.4.1 Maximum distance . . . . .	17

2.4.2	Maximum courbure . . . . .	17
2.4.3	L'axe principal . . . . .	18
2.4.4	L'axe naturel . . . . .	19
2.4.5	Discussions . . . . .	20
2.5	Méthode générale pour obtenir l'invariance . . . . .	21
2.5.1	Le calcul d'invariant . . . . .	21
2.5.2	Exemple . . . . .	22
2.6	Mesure de similarité et performance . . . . .	23
2.7	Mesure de similarité . . . . .	23
2.8	Mesure de Performance . . . . .	26
2.8.1	Rappel et Précision ( $RP$ ) . . . . .	27
2.8.2	Le pourcentage de poids des morceaux . . . . .	27
2.8.3	Le pourcentage du classement de similarité . . . . .	27
2.8.4	Discussions . . . . .	28
2.9	Evaluation des mesures de distance . . . . .	29
2.10	Conclusion . . . . .	29
<b>Chapitre 3 Descripteur de formes</b>		<b>31</b>
3.1	Introduction . . . . .	31
3.2	Descripteur de moment . . . . .	31
3.2.1	Les moments géométriques . . . . .	32
3.2.2	Les moments complexes . . . . .	33
3.2.3	Les moments orthogonaux . . . . .	33
3.3	Descripteur de Fourier . . . . .	36
3.3.1	Série de Fourier . . . . .	36
3.3.2	La transformée de Fourier continue . . . . .	37
3.3.3	La transformée de Fourier discrète . . . . .	37
3.3.4	Le descripteur de Fourier par rapport à la transformation similitude . . . . .	39
3.3.5	Le descripteur de Fourier par rapport à la transformation affine . . . . .	40
3.4	Descripteur de Fourier à fenêtre glissante . . . . .	40
3.5	Descripteur d'espace-échelle de courbure (CSSD) . . . . .	41
3.5.1	Fondement de base de la théorie d'espace-échelle . . . . .	41
3.5.2	Propriété de la non-crédation des nouvelles caractéristiques . . . . .	42
3.5.3	CSSD Normal . . . . .	43
3.5.4	CSSD modifié . . . . .	46
3.5.5	Algorithme d'appariement . . . . .	47
3.6	Conclusion . . . . .	49



---

<b>Chapitre 4 Transformation en ondelettes</b>	<b>51</b>
4.1 Introduction . . . . .	51
4.2 La transformée continue en ondelettes . . . . .	52
4.3 Bases orthonormées d'ondelettes . . . . .	53
4.3.1 Le passage à la transformée discrète en ondelettes . . . . .	53
4.3.2 Analyses multirésolutions et bases orthonormées d'ondelettes . . . . .	53
4.4 Des bases orthonormées à l'algorithme de Mallat . . . . .	56
4.5 Les quatre filtres . . . . .	57
4.6 Calcul efficace des coefficients . . . . .	58
4.7 Complexité de l'algorithme . . . . .	58
4.8 Transformée invariante par translation . . . . .	59
4.8.1 DWT $\varepsilon$ -décimée . . . . .	60
4.8.2 Calcul de la SWT . . . . .	60
4.9 Propriété de la transformation en ondelettes . . . . .	61
4.10 Descripteur d'ondelettes invariant à la transformation similitude . . . . .	64
4.11 Descripteur d'ondelettes invariant à la transformation affine . . . . .	67
4.12 Complexité du calcul des descripteurs . . . . .	72
4.13 Conclusion . . . . .	73
<b>Chapitre 5 Expérimentations, évaluations et comparaisons</b>	<b>77</b>
5.1 Introduction . . . . .	77
5.2 Contexte des expérimentations . . . . .	78
5.3 Evaluation du descripteur $WD$ . . . . .	79
5.3.1 Résultat concret . . . . .	79
5.3.2 L'effet du nombre de moments nuls . . . . .	80
5.3.3 Comparaison des $WDs$ basée sur différents atomes d'ondelettes . . . . .	81
5.3.4 Comparaison des méthodes de fixation du point de départ . . . . .	83
5.4 Evaluations et comparaisons des descripteurs invariants à la transformation simi- litude . . . . .	84
5.4.1 Comparaison des $FDs$ . . . . .	84
5.4.2 Comparaison des descripteurs $FD$ , $SFD$ et $CSSD$ . . . . .	86
5.4.3 Comparaison du $WD$ et du $FD$ . . . . .	87
5.5 Evaluation du descripteur affine . . . . .	88
5.6 Conclusion . . . . .	90
<b>Chapitre 6 Conclusion Générale</b>	<b>91</b>
6.1 Synthèse des travaux effectués et apport de la thèse . . . . .	91

*Table des matières*

---

6.2 Perspectives . . . . .	92
<b>Bibliographie</b>	<b>95</b>

# Table des figures

2.1	Approximation polygonale du contour . . . . .	9
2.2	(a) Le contour avec deux points de départ $S1$ et $S2$ (b) Les deux signatures superposées. En bas (resp. en haut), c'est la signature correspondant au point de départ $S1$ (resp. $S2$ ) . . . . .	11
2.3	(a)Le contour avec les composants principaux et le point de départ (b)On place les composants principaux sur l'axe abscisse et l'axe ordonnée . . . . .	19
2.4	Le collier de $N$ perles de rayon un associé à l'ensemble ordonnée $S$ . . . . .	19
2.5	(a) La distance Euclidienne (b) La distance cosinus (c) La distance $L1$ ( $d_1 = d_{11} + d_{12}$ ) . . . . .	25
3.1	Polynôme de Legendre dans l'espace 2D (a) $P_2(x)P_2(y)$ (b) $P_4(x)P_4(y)$ (c) $P_6(x)P_6(y)$ (d) $P_8(x)P_8(y)$ . . . . .	33
3.2	Polynôme de Legendre dans l'espace 2D (a) $P_2(x)P_4(y)$ (b) $P_2(x)P_6(y)$ (c) $P_4(x)P_8(y)$ (d) $P_6(x)P_8(y)$ . . . . .	34
3.3	Polynôme de Zernike (a) $ V_{20} $ (b) $ V_{40} $ (c) $ V_{80} $ (d) $ V_{12,0} $ . . . . .	35
3.4	Polynôme de Zernike (a) $ V_{11} $ (b) $ V_{22} $ (c) $ V_{31} $ (d) $ V_{42} $ . . . . .	35
3.5	(a)Le signal origine $f(x)$ en bas et ses versions lissées sont au-dessus, ici $t$ est l'échelle de la fonction de lissage. (b)L'arbre d'intervalle obtenu des zero-crossings des deuxièmes dérivées des signaux lissés dans (a), chacun des points $(x, t)$ dans l'arbre d'intervalle correspond à un point de zero-crossing à la position $x$ et à l'échelle $t$ du signal [Lin96] . . . . .	43
3.6	Les points d'inflexions détectés sur : 1) le contour d'un poisson (en bas à gauche) et 2) la courbure du poisson (en bas à droite) . . . . .	44
3.7	Les contours du poisson ont été filtrés avec le noyau Gaussien avec des valeurs $\sigma \in \{1.5, 3.5, 5.5, 7, 5\}$ . Ensuite, les point d'inflexions ont été détectés après le filtrage . . . . .	44
3.8	Le résultat de la détection des points inflexions à partir de la courbure des contours filtrés de la figure 3.7 . . . . .	44
3.9	L'exemple de l'image du CSS . . . . .	45
3.10	(a)Un contour avec la concavité peu profonde et (b)son CSS image. (c)Le contour avec la courbure très forte et (d) son CSS image. . . . .	46
4.1	Translation et translation-dilatation d'une ondelette . . . . .	52
4.2	Premier pas de $DWT$ , où $\boxed{F}$ représente la convolution par $F$ et $\boxed{\downarrow 2}$ la décimation . . . . .	58
4.3	Représentation d'un pas de décomposition de la $DWT$ (pour $j = 0, a^j = X$ ) . . . . .	58

4.4	(a) $S_1$ est le signal d'origine. Le signal $S_i$ est un décalage circulaire de 50 points du signal $S_{i-1}$ , $2 \leq i \leq 5$ (b) Les coefficients de $DWT$ de $S_i$ . On constate que les coefficients de $DWT$ des 5 signaux sont différents. . . . .	59
4.5	$SWT$ 1D : décomposition au niveau 1 . . . . .	60
4.6	$SWT$ 1D : algorithme de décomposition . . . . .	74
4.7	(0) Contour de départ (1-8) Les approximations du contour de départ de la résolution la plus fine vers la résolution la plus grossière [CK96] . . . . .	74
4.8	Energie du signal et de ses approximations sur plusieurs niveaux de décomposition	74
4.9	La représentation graphique de l'énergie en moyenne de la décomposition des signatures de la base par $DWT$ sur sept niveaux . . . . .	75
4.10	Contre-exemple : la relation d'ordre de la longueur des points du contour n'est pas préservée dans le cas de la transformation affine . . . . .	75
5.1	(a) 70 contours qui représentent 70 classes (b) 20 contours de la même classe. Tous ces contours font partie de la sous-base B de la grande base CE-1 MPEG-7	78
5.2	Les 140 premiers contours de la sous-base C de la grande base CE-1 MPEG7 . .	78
5.3	Les 140 contours suivants de la sous-base C de la grande base CE-1 MPEG7 . . .	78
5.4	Les figures (a) et (b) représentent les vingt premières réponses aux deux premières requêtes de la figure 5.1a. Les figure (c) et (d) représentent les vingt premières réponses aux deux premières requêtes de la figure 5.1b . . . . .	79
5.5	La fonction d'échelle d'ondelette db6, coif2, sym6 et dmey . . . . .	81
5.6	La fonction d'échelle de décomposition (à gauche) et de reconstruction (à droite) d'ondelette (a) bior1.5, bior3.5 et bior5.5 (b) rbio1.5, rbio3.5 et rbio5.5 . . . . .	81
5.7	La comparaison de la performance en moyenne de la recherche en utilisant différents atomes d'ondelettes (a) Coif2, db6, dmey, sym6 (b) bior1.5, bior3.5 et bior5.5 (c) rbio1.5, rbio3.5 et rbio5.5 . . . . .	81
5.8	La comparaison de la performance de la recherche entre les quatre techniques pour fixer le point de départ . . . . .	83
5.9	Les vingt premières réponses à la requête de la quatrième sous-figure de la figure 5.1a en utilisant (a) $FD_1$ (b) $FD_2$ (c) $FD_3$ (d) $FD_4$ . La figure (e) donne le résultat de comparaison des quatre descripteurs de Fourier $FD_1$ , $FD_2$ , $FD_3$ et $FD_4$ . . . . .	84
5.10	Les valeurs moyennes de Précision et Rappel de 1400 requêtes utilisant (a) Le descripteur de Fourier et le descripteur de Fourier à fenêtre glissante (b) Le descripteur de Fourier et le descripteur d'espace-échelle de courbure sur la sous-base B de la base CE-1 MPEG7 [Zha02] . . . . .	86
5.11	La comparaison de la performance de la recherche entre $WD$ et $FD$ . . . . .	87
5.12	(a) Les contours de requête, $P_1$ est le contour origine, $P_2$ et $P_3$ sont des transformations affines de $P_1$ et $P_4$ est le contour $P_1$ avec un bruit uniforme entre $[-1,1]$ (b) les résultats de la requête $(P_i)_{1 \leq i \leq 3}$ (c) les résultats de la requête $P_4$ . . . . .	88
5.13	(a) Le taux de succès de $AWD$ pour la base $(DB_i)_{1 \leq i \leq 3}$ (b) La mesure de performance P/R sur la sous-base B de la base CE-1 MPEG7 . . . . .	88

# Liste des tableaux

2.1	Résultat d'évaluation rappel-précision du système de reconnaissance utilisant le descripteur de Fourier sur l'ensemble B de la base standard CE1-MPEG7 . . . .	29
4.1	Le pourcentage en énergie moyenne des signatures de la base sur chaque niveau de décomposition : chaque élément du tableau représente la valeur de $\dot{e}_k$ , $k = 0..7$	67
5.1	Tableau de comparaison des performances en moyenne de la recherche en utilisant les atomes d'ondelettes de la même famille mais avec le nombre de moments nuls différents. Les éléments de ce tableau sont des valeurs moyennes de Précision (P) calculées en utilisant 1400 requêtes. . . . .	80
5.2	Propriétés des familles d'ondelettes . . . . .	82
5.3	Tableau de comparaison des performances en moyenne de la recherche en utilisant différents atomes d'ondelettes et en utilisant la distance maximum au centre de gravité pour fixer le point de départ. Les éléments de ce tableau sont des valeurs moyennes de Précision (P) calculées en utilisant 1400 requêtes. . . . .	82
5.4	Tableau de comparaison de la performance en moyenne des quatre techniques pour fixer le point de départ. Les éléments de ce tableau sont des valeurs moyennes de Précision (P) calculées en utilisant 1400 requêtes . . . . .	83
5.5	Tableau de comparaison de la performance en moyenne de notre méthode et de la méthode de Fourier. Les éléments de ce tableau sont des valeurs moyennes de Précision (P) calculées en utilisant 1400 requêtes . . . . .	87



# Chapitre 1

## Introduction générale

Les travaux présentés dans ce rapport de thèse s'inscrivent dans le domaine de la reconnaissance des formes, plus précisément de la reconnaissance des contours d'objet 2D. Il s'agit d'un domaine fondamental et très vaste de la vision par ordinateur. Il recouvre des problèmes très variés comme ceux de l'analyse mathématique, de l'intelligence artificielle, des techniques de traitement ou/et de l'analyse du signal et d'image, et des techniques de gestion de base de données...

L'approche proposée dans cette thèse propose une solution aux problèmes de la reconnaissance des formes par l'utilisation de la *transformation en ondelettes discrètes*. Dans ce chapitre, nous présentons d'abord le contexte dans lequel nous nous plaçons, ensuite une description de l'approche proposée. Enfin, les contributions de ce travail puis un plan de ce document terminent ce chapitre.

### 1.1 Contexte

Jusqu'à présent, l'exploitation d'un fond photographique faisait largement appel à des techniques de descriptions manuelles et textuelles : un documentaliste saisissait à la main un ensemble de mots clés ou une courte notice décrivant le contenu des images avant archivage. Ceci étant fait, il est alors possible d'interroger la collection d'images à partir d'un ensemble de mots décrivant l'image recherchée. Ceci permettait de retrouver, par exemple, les images relatives à un thème particulier ou bien celles où apparaît telle personnalité.

Ce procédé de description porte beaucoup de sémantique de par la description textuelle des images par un humain. Il est cependant subjectif à cause du très grand nombre d'interprétations pouvant être prêtées au contenu d'une même image et aux différents niveaux de descriptions possibles. Deux documentalistes peuvent ainsi donner deux descriptions différentes d'une même image. De plus, ce mode de description étant basé uniquement sur le texte, il est intrinsèquement limité au contenu exprimable à l'aide de mots. Ce schéma est également laborieux et onéreux à mettre en oeuvre lorsque l'on doit décrire de très grandes collections d'images. Ces limites rendent, par conséquent, son utilisation de plus en plus difficile dans une application réelle, géant une très grande collection d'images.

Il existe d'autres techniques de description qui sont, elles, automatiques. Se passant d'interventions humaines, elles peuvent servir à décrire de très larges collections d'images. Ces techniques sont basées sur l'extraction de propriétés visuelles des images (couleur, texture, forme...)

et, ainsi tendent à en traduire le contenu. Décrire automatiquement le contenu visuel des images permet alors l'émergence d'un nouveau mode d'interrogation : l'interrogation par l'exemple. Dans ce cas, le principe est de retrouver les images qui ressemblent fortement à l'image donnée en exemple, c'est-à-dire celles qui ont une forte similarité visuelle avec l'exemple. Retrouver ainsi des images où apparaît telle personnalité ne requiert plus d'avoir par avance saisi son nom, mais demande alors de fournir un exemplaire du visage recherché pour initier la recherche.

Clairement, la nature des propriétés visuelles à choisir pour la description automatique dépend du type de ressemblances que l'on désire : trouver des visages similaires fait appel à des propriétés différentes de celles mises en jeu si l'on cherche à retrouver les images ressemblantes à des images de paysages par exemple.

Mettre en oeuvre ces techniques de description automatique basées sur la ressemblance visuelle entre images dans un contexte de très grandes collections d'images fait appel à des techniques développées dans deux domaines différents : l'*analyse d'images* et les *bases de données*. Cette mise en oeuvre s'effectue dans le cadre d'un *système de recherche d'images par le contenu* (SRIC) offrant la possibilité d'archiver, de décrire automatiquement une collection d'images hors ligne, et d'exécuter, en ligne, des interrogations par l'exemple.

Les méthodes d'*analyse d'images* permettent d'extraire à partir des images, des descripteurs permettant de décrire certaines de leurs propriétés visuelles. L'enjeu principal de ces méthodes est de mettre au point des schémas de description permettant de traduire, le plus fidèlement possible, la notion de similarité visuelle requise par l'application en une mesure de similarité entre descripteurs. Généralement, les descripteurs se présentent sous la forme de vecteurs numériques auxquels est associée une mesure de similarité (typiquement une distance). Cette mesure permet de quantifier la proximité entre descripteurs et, de manière indirecte, la similarité visuelle entre images. Ainsi, la proximité entre descripteurs se veut proportionnelle à la similarité visuelle entre images.

Lors d'une interrogation, un descripteur est tout d'abord extrait à partir de l'image fournie en exemple (image requête). Ce descripteur requête est ensuite utilisé pour retrouver les descripteurs stockés dans la base qui lui sont les plus proches en terme de distance. Une fois retrouvés, ces descripteurs permettent d'obtenir les images auxquelles ils sont associés et qui sont censées être similaires à l'image requête. Ce mode de recherche se traduit le plus souvent par la recherche des voisins les plus proches du descripteur requête au sens de la mesure de similarité associée aux descripteurs. On parle ainsi de *recherche de plus proches voisins*. Ces recherches sont cependant extrêmement coûteuses lorsqu'elles s'appliquent à de très grandes collections d'images. Il est donc fait appel à des méthodes issues du domaine des bases de données pour tenter d'accélérer les interrogations.

Cette accélération repose sur le principe très général de limiter au maximum la quantité des descripteurs à comparer avec le descripteur requête. Confiner la recherche permet de réduire la quantité des données à lire depuis le disque et de diminuer le nombre d'opérations de calcul de similarités à effectuer. Ceci réduit les temps de réponse de la recherche. Les stratégies permettant de limiter l'étendue des recherches sont issues des algorithmes proposés pour l'indexation multidimensionnelle.

Ainsi, un SRIC complet requiert l'intégration dans un même système, de techniques issues



de ces deux domaines. De nombreuses solutions ont été proposées pour résoudre les différents problèmes liés à SRIC. Elles ont donné lieu à des applications variées. En ce qui concerne le domaine d'analyse d'images, les techniques utilisées pour extraire des descripteurs sont très différentes. Cependant, elles présentent de fortes limitations par rapport à la norme de MPEG-7 [KK00][ZL04][ZL03a], c'est à dire qu'elles ne satisfont pas la plupart des critères suivants :

1. Bonne précision de la recherche
2. Descripteur compact
3. Application générale
4. Faible complexité par rapport au temps de calcul utilisé pour extraire et comparer les descripteurs
5. Faible dégradation de performance quand on augmente la taille de la base des données
6. Représentation hiérarchique (ou multi-échelle ou multirésolution)

L'objet de cette thèse a été de proposer une méthode innovante par rapport à ces méthodes et à leurs limitations.

## 1.2 Approche proposée

Les techniques de représentation et de description des formes peuvent être généralement regroupées en deux classes de méthodes :

- Des méthodes basées sur le contour
- Des méthodes basées sur la région

Cette classification est fondée sur le fait que le descripteur de forme est obtenu ou non uniquement à partir des informations sur le contour. Si la réponse est oui, alors la méthode concernée est considérée comme une méthode basée sur le contour. Si le descripteur est obtenu à partir des informations sur la région de la forme, alors elle est considérée comme une méthode basée sur la région. En plus sur chaque classe, selon [ZL04], les différentes méthodes sont divisées en deux approches :

- Approche structurelle
- Approche globale

La distinction entre les sous-classes est basée sur le type de représentation de la forme : si la forme est représentée dans son ensemble on parle d'approche globale, sinon, si la forme est divisée et représentée en segments (ou sections), on parle d'approche structurelle.

Dans cette thèse, les travaux que nous avons menés appartiennent à la classe des méthodes basées sur le contour. L'avantage des méthodes basées sur le contour par rapport aux méthodes basées sur la région est qu'elles nécessitent moins d'espace pour stocker des données et par conséquent moins de temps de calcul. De plus, intuitivement et visuellement, la perception humaine est plus sensible au contour qu'au contenu de la forme.

Notre approche, que nous détaillons dans la suite, fait partie des approches globales, elle s'appuie sur la *transformation en ondelettes discrètes*. Les avantages d'utilisation de cette transformation par rapport aux autres transformations sont nombreux. En ce qui concerne les critères de la norme MPEG-7, d'une manière évidente, la transformation en ondelettes discrètes est une

transformation multi-résolution (critère numéro 6) et le temps de calcul des coefficients d'ondelettes (critère numéro 4) est très faible ( $O(n)$ ). De plus, il est très facile à implémenter et il est possible de :

- Changer le noyau d'ondelettes. On peut choisir le noyau pré-défini ou bien construire son propre noyau. De plus, on a la possibilité d'adapter le nombre de moments nuls associé à l'ondelette selon nos besoins
- Reconstruire le contour origine (reconstruction parfaite)
- L'analyse utilisant la transformation en ondelettes est une analyse qui possède deux caractéristiques en même temps : il s'agit d'une analyse à la fois locale et globale
- En fonction des objectifs du traitement par ondelettes, on peut préférer la transformée continue à la transformée discrète si la redondance peut être mise à profit pour analyser le signal. On fait le choix contraire si une compression du signal est souhaitée

Nous verrons dans la suite que le descripteur que nous avons proposé satisfait tous les critères de la norme MPEG-7.

### 1.3 Contributions

Cette thèse comporte cinq contributions :

1. Nous proposons un *nouveau descripteur simple* invariant à la transformation en similitude en utilisant la transformation en ondelettes discrètes. Il est important de souligner les deux termes *nouveau* et *simple*. Certes, il y a déjà des travaux qui ont été menés en utilisant la transformation en ondelettes discrètes, mais le descripteur obtenu est différent de celui que nous proposons [CK96][TB97a][HSYL98]. En plus, notre descripteur est très *simple* à calculer car il n'est rien d'autre que les coefficients de la transformation en ondelettes discrètes.
2. Nous proposons d'exploiter l'histogramme de la moyenne d'énergie des signatures de la base pour sélectionner automatiquement le niveau de décomposition qui est indispensable pour la transformation en ondelettes discrètes. La connaissance du niveau de décomposition est important, il nous permet de choisir l'ensemble des coefficients à retenir pour représenter les contours.
3. Nous proposons également un *nouveau descripteur* invariant à la transformation affine. Contrairement aux descripteurs affines proposés dans [KB01][KB02], le descripteur que nous proposons n'est pas redondant. Ce point (la suppression de la redondance) est vital car il influence directement sur les critères 2,4 et 5 de la norme MPEG-7. La technique proposée pour supprimer la redondance est très générale, il est possible d'utiliser cette technique pour supprimer la redondance des invariants proposée par M. Khalil et al. [KB01][KB02].
4. Lors de la phase d'appariement entre les contours <sup>1</sup> la tâche à accomplir est de trouver la correspondance de chaque élément du premier contour avec celui du deuxième contour. Concernant le temps de calcul, il est possible de calculer toutes les correspondances possibles (recherche exhaustive) pour trouver la solution optimale. Cependant, il est souhaitable de trouver une autre solution pour améliorer le temps de calcul et par la même occasion satisfaire aux conditions 4 et 5 de la norme MPEG-7. Pour cette raison, nous proposons d'étudier et de comparer quatre méthodes pour fixer le *point de départ*. Ces quatre méthodes sont :

---

<sup>1</sup>ou descripteur des contours

- (a) La longueur maximum
  - (b) La courbure maximum
  - (c) L'analyse en composantes principales (ACP)
  - (d) L'orientation de l'axe naturel
5. Enfin, nous avons développé et évalué l'ensemble des contributions décrites précédemment sur la base CE-1 de MPEG7 et plus précisément sur la base des contours B de CE-1. Ceci sera décrite en détail dans le chapitre 5. Le résultat de notre évaluation prouve que dans le cas de la transformation similitude, notre descripteur est meilleur que le descripteur Curvature-Scale-Space et il a la même performance que le descripteur de Fourier.

## 1.4 Plan du mémoire

Ce rapport présente d'abord dans le **chapitre 2**, les outils qui serviront dans la suite de nos travaux à développer, à proposer nos contributions.

Dans le **chapitre 3**, nous présentons quelques descripteurs des formes les plus connus et les plus populaires dans le domaine de la reconnaissance des formes. L'évaluation et la comparaison de certains de ces descripteurs feront l'objet d'expérimentations détaillées dans le chapitre 5.

Dans le **chapitre 4**, nous commençons par présenter le fondement de base de la transformation en ondelettes en séparant le cas continu du cas discret. Le lien et le passage du premier au deuxième cas se font en introduisant l'analyse multi-résolution. Dans la deuxième partie de ce chapitre, nous consacrons entièrement la présentation à la démarche progressive de nos contributions par rapport aux deux nouveaux descripteurs d'ondelettes, à la méthode de sélection du niveau de décomposition et à la méthode de suppression de la redondance.

Dans le **chapitre 5**, nous présentons l'étude expérimentale, l'évaluation et la comparaison de certains descripteurs que nous avons présentés dans le chapitre 3 et de nos descripteurs.

A la fin de ce mémoire, dans le **chapitre 6**, nous concluons ces travaux de recherche, nous donnons quelques pistes pour les améliorations possibles et nous dressons quelques perspectives scientifiques.



# Chapitre 2

## Les outils de la reconnaissance des formes

### 2.1 Introduction

Fondamentalement, la reconnaissance d'images basée sur leur forme consiste à mesurer la similarité entre les formes qui sont représentées par leurs vecteurs caractéristiques. Donc, afin de réaliser un système capable d'effectuer cette tâche, nous devons d'abord et avant tout, matérialiser le système par un module capable d'extraire les vecteurs caractéristiques qui représentent les formes et ensuite nous devons définir des outils permettant de mesurer la similarité entre les vecteurs caractéristiques qui ont été extraits. Donc, deux étapes essentielles concernant le système de reconnaissance des formes sont : l'extraction des vecteurs caractéristiques et la mesure de similarité entre ces vecteurs. En ce qui concerne les méthodes d'extraction, ces méthodes ont recours à divers outils de pré-traitement comme la représentation du contour, la paramétrisation, la fixation du point de départ, le calcul d'invariant etc. L'objectif de ce chapitre a pour but de présenter ces outils qui nous serviront dans la suite et surtout à l'extraction des vecteurs caractéristiques.

En parallèle, bien que l'extraction des vecteurs caractéristiques soit cruciale pour la reconnaissance des formes, la mesure de similarité entre les vecteurs caractéristiques est aussi importante. Le système de reconnaissance des formes est robuste et performance exige que les formes perceptuellement semblables soient également semblables en termes des vecteurs caractéristiques. Dans la littérature, il existe déjà plusieurs mesures de similarité. Dans ce chapitre, nous allons présenter un certain nombre de ces mesures et nous allons aussi présenter les mesures de performance afin de comparer la robustesse des mesures de similarité et la robustesse des vecteurs caractéristiques issus des différentes méthodes d'extraction.

Après avoir exposé les différentes représentations du contour les plus connues et les plus souvent utilisées dans la section 2.2, on s'intéressera aux techniques de paramétrisation, section 2.3, et aux techniques de fixation du point de départ (section 2.4). La discussion à la fin de chacune de ces sections permet de clarifier un certain aspect du point de vue de l'application. Nous exposons également la méthode générale de calcul de l'invariant à la section 2.5. Nous allons ensuite exposer et discuter les mesures de similarité et les mesures de performance respectivement dans la section 2.7 et 2.8. Enfin, la section 2.10 présente le résumé et conclut le chapitre.

## 2.2 La représentation du contour d'un objet 2D

Après l'opération de segmentation qui consiste à différencier l'objet d'intérêt du fond, on obtient une image comportant des régions correspondant à l'objet. Même si certaines méthodes de reconnaissance des formes peuvent être appliquées directement à ce type d'images, dans la pratique, le plus souvent on préfère utiliser la représentation de données plus compactes car cela permet de réduire considérablement le temps de calcul du descripteur et nécessite en même temps moins de mémoire à utiliser. Pour cela, on doit appliquer l'algorithme de suivi de contour afin de récupérer les coordonnées des points sur le contour d'objet et ensuite on peut choisir la représentation ad-hoc. Dans cette section, on discute de nombreuses approches sur la représentation du contour.

### 2.2.1 La représentation par un nombre complexe ou par les coordonnées cartésiennes ou polaires

La représentation du contour la plus naturelle et la plus simple, est la représentation des points du contour par les coordonnées cartésiennes  $(x_k, y_k)$  ou par les coordonnées polaires  $(r_k, \theta_k)$  où  $r_k$  et  $\theta_k$  représentent respectivement la distance du point  $(x_k, y_k)$  à l'origine du repère orthonormé et l'angle entre l'axe abscisse et la droite passant par le point  $(x_k, y_k)$  et l'origine du repère. La relation entre les deux représentations est la suivante :

$$\begin{cases} x_k = r_k \cos \theta_k \\ y_k = r_k \sin \theta_k \end{cases} \quad \text{et} \quad \begin{cases} r_k = \sqrt{x_k^2 + y_k^2} \\ \theta_k = \arctan \frac{y_k}{x_k} \end{cases} \quad (2.1)$$

Mais il est parfois très utile de représenter les deux valeurs coordonnées  $x_k$  en abscisse et  $y_k$  en ordonnée par une seule valeur complexe  $x_k + iy_k$ . Dans certains cas, il est aussi préférable d'utiliser les coordonnées cartésiennes (ou le nombre complexe) par rapport au centre de gravité de l'objet. Formellement, si  $n$  est le nombre de points sur le contour, cela veut dire qu'au lieu d'utiliser les coordonnées cartésiennes  $(x_k, y_k)$  (ou le nombre complexe  $x_k + iy_k$ ), on utilise les coordonnées  $(x_k - x_c, y_k - y_c)$  (ou le nombre complexe  $[x_k - x_c] + i[y_k - y_c]$ ) avec :

$$x_c = \frac{1}{n} \sum_{k=1}^n x_k, \quad y_c = \frac{1}{n} \sum_{k=1}^n y_k \quad (2.2)$$

De la même façon, on peut aussi utiliser les coordonnées polaires par rapport au centre de gravité, c'est à dire qu'on place l'origine du repère au centre de gravité du contour puis on calcule les coordonnées polaires selon l'équation 2.1.

### 2.2.2 La représentation par la chaîne des codes de direction

Le concept de la chaîne des codes de direction (CCD) a été introduit en premier par [Fre61]. La représentation par la CCD est une représentation qui est basée sur le codage d'une séquence de segments orientés, en 4 ou 8 directions ou plus généralement  $n$  directions [FS78], et sur la notion de connexité. Ce codage consiste à associer à chaque direction un nombre entier. En fait, les pixels d'une image numérique sont placés sur une grille, donc on peut générer la CCD en suivant le contour, disons dans le sens des aiguilles d'une montre, et à chaque segment joignant deux pixels adjacents ou connexes, on associe un nombre correspondant à la direction de ce segment. Mais cela pose deux problèmes :

- la longueur de la CCD obtenue est relativement longue
- s'il y a des perturbations à cause du bruit ou de mauvaises segmentations, cela entraîne le changement sur la CCD qui peut ne pas correspondre au contour de l'objet

Une approche souvent utilisée pour éviter ce problème, est d'utiliser une grille plus grande (voir [GW93]). Un autre problème aussi important est la dépendance de la représentation vis-à-vis du choix du point de départ sur le contour, c'est-à-dire que si on a deux points différents  $P$  et  $Q$  sur le contour d'un même objet, la représentation du contour par la CCD en commençant le code à partir du point  $P$  est différente de la représentation de cet objet par la CCD en commençant le code à partir du point  $Q$ . Heureusement, on peut le (le choix du point de départ) normaliser par une procédure très simple : on choisit le point de départ de sorte que la valeur de la CCD résultant du choix effectué soit la plus grande. On peut aussi le normaliser par rapport à la rotation en utilisant la *différence* successive de la CCD au lieu d'utiliser la chaîne code elle-même (voir [GW93]).

Iivarinen et Visa [IV96] ont utilisé l'histogramme de la CCD (CCH) pour faire la reconnaissance des contours. Le CCH est défini par la fréquence d'apparition du code  $n_k$ , assimilé à une probabilité  $p_k = n_k/n$ , avec  $n_k$  le nombre de segments de direction  $k$  et  $n$  le nombre de segments du contour. Par rapport à cette définition, CCH représente la probabilité des différentes directions présentes dans la CCD. Le CCH est invariant par rapport au changement d'échelle et à la translation mais il n'est pas invariant à la rotation. Pour faire face au problème lié à la rotation, Iivarinen et Visa [IV96] ont proposé une solution alternative en utilisant le CCH normalisé (NCCH). Dans le NCCH, ils ont redéfini la probabilité  $p_k$  par  $p_k = l_k n_k / l$  où  $l_k$  représente la longueur du segment de direction  $k$ ,  $l$  représente la longueur du contour et la définition de  $n_k$  est la même que dans CCH. La méthode NCCH est très simple à mettre en oeuvre et permet de résoudre le problème de la longueur de la CCD, elle est cependant très sensible au bruit et de plus elle ne prend pas en compte la relation spatiale ou la propriété locale de chaque segment.

### 2.2.3 L'approximation polygonale

Dans la section 2.2.1, pour représenter un contour, on enregistre toutes les coordonnées de tous les points du contour, mais cette représentation pose de sérieux problèmes pour certaines applications où le temps de réponse et la taille d'espace utilisé sont limités. Une solution très simple à ce problème est de réduire les données (le nombre de points du contour), c'est à dire qu'on remplace le contour par son *approximation polygonale* autrement dit *approximation linéaire par morceaux*. Chaque segment du polygone représente une portion du contour original. Il faut remarquer que l'approximation polygonale peut être utilisée dans le cadre du filtrage du bruit ou dans le cadre de la compression de données.

Le but de l'approximation polygonale est de trouver les sommets du polygone le long du contour de sorte que le polygone résultant reflète le plus fidèlement possible la forme du contour original en essayant de réduire au maximum le nombre de sommets de ce polygone. Les méthodes d'approximation polygonale ou les méthodes de détection des sommets sont regroupées en deux catégories : les méthodes globales et les méthodes locales. Les méthodes globales sont généralement basées sur l'approximation polygonale de sorte que certaines fonctions d'erreur entre le polygone et le contour original soient minimales. Par contre, les méthodes locales sont basées assez souvent sur la détection des points du contour de forte courbure [dFCJ01].

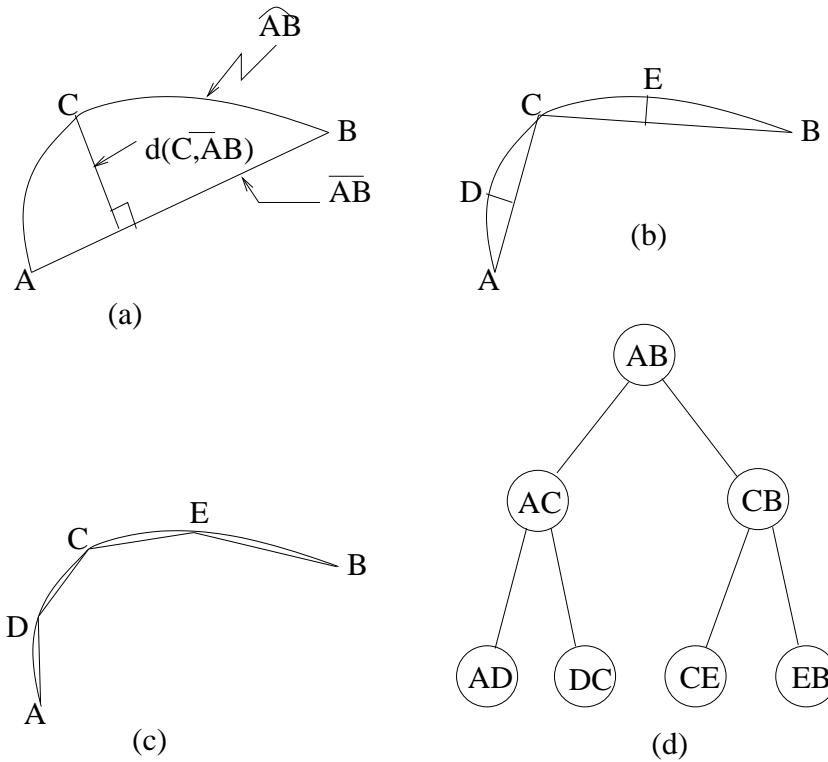


FIG. 2.1 – Approximation polygonale du contour

Un exemple de la fonction d'erreur utilisée dans les méthodes globales est la plus grande distance entre les points du contour et le segment du polygone [dFCJ01]. On peut utiliser d'autres fonctions d'erreur comme par exemple :

- Périmètre minimum [GW93]
- Surface minimum
- Surface extérieure du polygone minimum

Certaines de ces méthodes ont été développées pendant les années 1970. Ramer, [Ram72], a développé une méthode très élégante basée sur un algorithme de *fractionnement* ou *partitionnement* (*split*). Le point de départ de l'algorithme consiste à initialiser le paramètre d'erreur  $\varepsilon$  et une solution initiale (polygone) de l'algorithme. La distance maximum entre les points du contour et le segment du polygone correspondant est utilisée comme l'erreur d'approximation. Pour mieux comprendre le fonctionnement de la méthode, prenons un exemple de la figure 2.1(a).

Pour simplifier l'explication, notons  $\widehat{AB}$  (resp.  $\overline{AB}$ ) la portion du contour (resp. le segment du polygone d'approximation) comprise entre le point  $A$  et le point  $B$ . Pour savoir si l'approximation  $\overline{AB}$  de  $\widehat{AB}$  est valide, l'algorithme calcule la distance de tous les points de  $\widehat{AB}$  au segment  $\overline{AB}$ . Dans l'exemple de la figure 2.1(a), la distance entre le point  $C$  de  $\widehat{AB}$  est indiquée par  $d(C, \overline{AB})$ . Supposons que

$$C = \max_{P \in \widehat{AB}} d(P, \overline{AB})$$



Si  $d(C, \overline{AB}) < \varepsilon$ , on peut décider que  $\widehat{AB}$  peut être remplacé (approximé) par  $\overline{AB}$ . Il est important de noter, concernant le seuil d'erreur  $\varepsilon$  du paramètre **entree** de l'algorithme, que plus  $\varepsilon$  est grand plus le niveau d'approximation est grossier et le nombre de sommets et de segments du polygone est relativement faible.

A l'inverse si  $\varepsilon$  est petit, le polygone résultant est très proche du contour origine mais le nombre de sommets et de segments du polygone est relativement élevé. Dans le cas contraire, si  $d(C, \overline{AB}) \geq \varepsilon$ , alors la portion du contour  $\widehat{AB}$  ne peut pas être représentée par le segment de droite  $\overline{AB}$  car la plus grande distance entre un point  $C$  du contour au segment  $\overline{AB}$  est supérieure à l'erreur  $\varepsilon$  autorisée. Dans ce cas, on doit *partitionner* le contour  $\widehat{AB}$  en deux. Bien qu'il y ait plusieurs façons de partitionner le contour  $\widehat{AB}$  en deux, le partitionnement le plus naturel est le partitionnement au point  $C$ , c'est à dire qu'il faut partitionner au point où sa distance au segment  $\overline{AB}$  est la plus grande. Le résultat du partitionnement de la figure 2.1(a) est montré dans la figure 2.1(b) où le contour  $\widehat{AB}$  est approximé par deux segments polygonaux  $\overline{AC}$  et  $\overline{CB}$ . L'étape suivante de l'algorithme consiste à appliquer les mêmes traitements aux contours  $\widehat{AC}$  et  $\widehat{CB}$  (figure 2.1(b)). La répétition de ce processus continue tant que la distance maximum entre le contour et le segment de polygone correspondant est supérieure à  $\varepsilon$  et il s'arrête dans le cas contraire (figure 2.1(c)).

On peut voir la méthode de partitionnement comme une structure hiérarchie d'arbre binaire (figure 2.1(d)). Dans cette structure, chaque noeud d'arbre correspond au couple de sommet du polygone et les feuilles correspondent aux segments du polygone final. Chaque niveau de l'arbre correspond à l'échelle d'approximation, cela signifie que le niveau près de la racine correspond à l'approximation grossière et quand on descend vers les feuilles, la qualité d'approximation devient de plus en plus proche du contour original.

On peut faire deux reproches importants à la méthode de partitionnement de Ramer. Premièrement, la qualité d'approximation de la méthode proposée dépend du choix du polygone initial (solution initiale). Les explications de l'exemple de la figure 2.1 montrent très bien que les deux sommets  $A$  et  $B$  du contour qu'on a fixé au début de l'algorithme restent toujours jusqu'à la fin de l'algorithme. Si ces sommets ont été choisis par erreur, il n'est possible de corriger cette erreur. Le second reproche porte sur le fait qu'il n'évalue pas la qualité des segments obtenus, par exemple dans la figure 2.1(c), il est parfois utile de fusionner les deux portions du contour  $\widehat{DC}$  et  $\widehat{CE}$  et ensuite appliquer à nouveau l'algorithme de partitionnement de Ramer. Afin de palier à ces deux lacunes, Pavlidis et Horowitz, [PH74], ont proposé un algorithme de *partitionnement-et-fusionnement* (*split-and-merge*) pour étendre la méthode de Ramer.

#### 2.2.4 La distance au centre de gravité

Cette représentation est un cas particulier de la représentation par les coordonnées polaires relatives au centre de gravité dans laquelle on ignore la deuxième coordonnée  $\theta_k$  et on utilise uniquement la distance des points du contour au centre de gravité  $r_k$ . On rappelle que cette représentation est invariante en translation et en rotation. Le changement d'échelle entraîne un changement linéaire de la distance entre le point du contour et centre de gravité. Un point important de cette représentation est le choix du point de départ : un changement du point de départ entraîne un décalage circulaire de la représentation. La figure 2.2 illustre cette idée : la figure 2.2(a) montre un exemple de contour avec deux points de départ différents et la figure 2.2(b) montre les deux signatures superposées du contour de la figure 2.2(a). En bas (resp. en

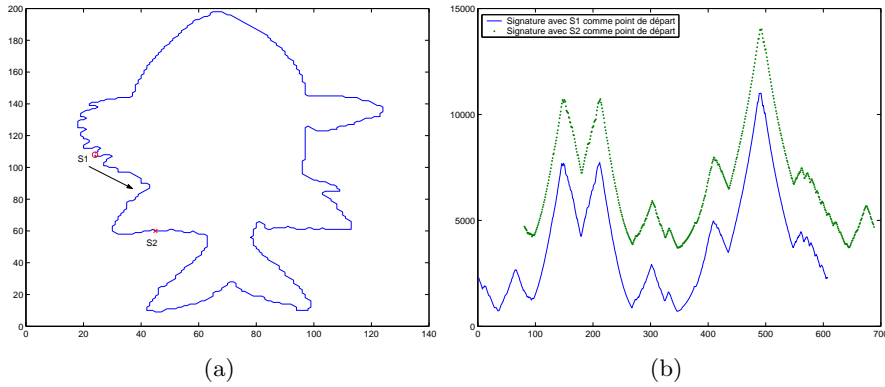


FIG. 2.2 – (a) Le contour avec deux points de départ  $S1$  et  $S2$  (b) Les deux signatures superposées. En bas (resp. en haut), c'est la signature correspondant au point de départ  $S1$  (resp.  $S2$ )

haut) c'est la signature qui correspond au point de départ  $S1$  (resp.  $S2$ ).

### 2.2.5 La fonction angulaire cumulative

Intuitivement, l'angle créé par la droite tangente au contour et l'axe abscisse (ou l'angle tangent) indique le changement de la direction angulaire du contour. Le changement de la direction angulaire est important dans la perception humaine. C'est pour cela qu'on peut représenter le contour par l'angle tangent défini par :

$$\theta(t) = \arctan \frac{y(t) - y(t-w)}{x(t) - x(t-w)}$$

où  $w \in \mathcal{N}$  est une fenêtre de calcul utilisée dans la pratique. Mais le problème de cette représentation est que la valeur de la fonction  $\theta(t)$  appartient à l'intervalle  $[-\pi, \pi]$  ou  $[0, 2\pi]$ , par conséquent la fonction  $\theta(t)$  possède une discontinuité de période  $2\pi$ . Pour éviter ce problème (la discontinuité), la fonction angulaire cumulative a été proposée. La *fonction angulaire cumulative*  $\varphi(t)$  est la quantité de changement de direction angulaire entre le point de départ  $\Gamma(0)$  et le point courant  $\Gamma(t)$  du contour modulo  $2\pi$  :

$$\varphi(t) = [\theta(t) - \theta(0)] \text{ mod } 2\pi$$

Avec cette définition, il est facile de voir que  $\varphi(0) = 0$  et  $\varphi(L) = -2\pi$  si  $L$  est la longueur totale du contour et le sens du parcours correspond au sens trigonométrique inverse. La fonction  $\varphi(t)$  est continue à l'endroit où  $\theta(t)$  a une valeur multiple de  $2\pi$ .

La version normaliser  $\psi(t)$  de  $\varphi(t)$  est proposée par [ZR72] :

$$\psi(t) = \phi\left(\frac{Lt}{2\pi}\right) - t$$

La soustraction de  $t$  à l'angle cumulatif  $\phi$  entraîne  $\psi(t) \equiv 0$  pour le cercle et  $\psi(t) \neq 0$  pour les autres contours. Il est important de voir que  $\psi(t)$  est invariant par rapport à la transformation similitude. Cependant, l'inconvénient de cette représentation est qu'elle possède toujours des points de discontinuité car en fait l'angle cumulatif est défini à partir de la direction de la tangente et qui est lui même défini à partir de la dérivée première des coordonnées du contour.

### 2.2.6 La représentation par la courbure

La *courbure* est une caractéristique importante et elle est très souvent utilisée pour représenter le contour. Pour un contour paramétré par un paramètre  $t$ , par définition, la courbure est définie par :

$$k(t) = \frac{d\theta(t)}{dt} \quad \text{avec} \quad \theta(t) = \arctan \frac{x(t+h) - x(t)}{y(t+h) - y(t)}$$

où  $h$  représente le pas utilisé pour le calcul. On peut utiliser une autre définition équivalente de la courbure qui est définie par :

$$k(t) = \frac{\dot{x}(t)\ddot{y}(t) - \ddot{x}(t)\dot{y}(t)}{(\dot{x}(t)^2 + \dot{y}(t)^2)^{3/2}} \quad (2.3)$$

où  $\dot{x}(t)$  et  $\ddot{x}(t)$  représentent respectivement la dérivée première et seconde de  $x$  au point  $t$ . De même  $\dot{y}(t)$  et  $\ddot{y}(t)$  représentent respectivement la première et la seconde dérivée de  $y$  au point  $t$ . De l'équation 2.3, comme on peut le constater, pour calculer la courbure  $k(t)$ , on doit calculer la dérivée première et seconde de  $x(t)$  et  $y(t)$  et on sait très bien dans la pratique que le calcul direct des dérivées des signaux ou des images est très sensible au bruit. Pour résoudre ce problème, Cesar et Costa, [JdFC95][JdFC96], ont proposé une estimation de la courbure par la transformée de Fourier.

Si on pose  $u(t) = x(t) + i y(t)$ , alors les dérivées première et seconde de  $u$  par rapport à la variable  $t$  sont définies par :

$$\dot{u}(t) = \dot{x}(t) + i \dot{y}(t) \quad \text{et} \quad \ddot{u}(t) = \ddot{x}(t) + i \ddot{y}(t)$$

et donc on a :

$$\begin{aligned} \dot{u}(t)\ddot{u}^*(t) &= \dot{x}(t)\ddot{x}(t) + \dot{y}(t)\ddot{y}(t) - i[\dot{x}(t)\ddot{y}(t) - \ddot{x}(t)\dot{y}(t)] \\ |\dot{u}(t)|^3 &= [\dot{x}(t)^2 + \dot{y}(t)^2]^{3/2} \end{aligned}$$

où  $z^*$  représente le nombre complexe conjugué du nombre complexe  $z$ , d'où l'équation 2.3 devient :

$$k(t) = -\frac{\text{Im}[\dot{u}(t)\ddot{u}^*(t)]}{|\dot{u}(t)|^3} \quad (2.4)$$

Si  $U(f)$ ,  $\dot{U}(f)$  et  $\ddot{U}(f)$  sont les transformées de Fourier de  $u(t)$ ,  $\dot{u}(t)$  et  $\ddot{u}(t)$ , alors en utilisant la propriété de la dérivée dans le cadre de la transformée de Fourier, on a les relations suivantes :

$$\dot{U}(f) = 2\pi i f U(f) \quad \text{et} \quad \ddot{U}(f) = -(2\pi f)^2 U(f) \quad (2.5)$$

Finalement, pour calculer la courbure, il suffit de calculer la transformée de Fourier  $U(f)$ , puis en déduire  $\dot{U}(f)$  et  $\ddot{U}(f)$  en utilisant les deux relations 2.5, ensuite on applique la transformée de Fourier inverse sur  $\dot{U}(f)$  et  $\ddot{U}(f)$  pour calculer  $\dot{u}(t)$  et  $\ddot{u}(t)$ .

### 2.2.7 La représentation par la surface fermée

Sous une transformation en similitude, la distance change linéairement, de même sous la transformation affine, la surface change également de façon linéaire. La linéarité est une propriété importante pour représenter le contour car la normalisation de la linéarité est équivalente

à la normalisation par rapport au changement d'échelle qui est beaucoup plus simple que la normalisation par rapport à la transformation affine. C'est pour cette raison que certains descripteurs utilisent la représentation par la surface fermée. Formellement, supposons que  $\Gamma$  est un contour dont le centre de gravité coïncide avec l'origine du repère, si  $(x_i, y_i)_{1 \leq i \leq n}$  sont des coordonnées des points sur le contour  $\Gamma$ , alors la représentation de  $\Gamma$  par la surface fermée est une représentation  $(s_i)_{1 \leq i \leq n}$  telle que :

$$s_i = \begin{cases} 0 & \text{pour } i = 1 \\ \frac{1}{2}|x_{i-1}y_i - x_iy_{i-1}| & \text{pour } 2 \leq i \leq n \end{cases} \quad (2.6)$$

## 2.3 Paramétrisations invariantes

Ici, on s'intéresse à la paramétrisation des contours 2D. En physique, on peut considérer une courbe paramétrique comme une évolution d'un point (ou d'une petite particule) qui se déplace dans un espace 2D, la position de ce point sur le plan peut être exprimée par sa position (vecteur)  $\vec{\Gamma}(t) = (x(t), y(t))$ , où  $t$  est une valeur réelle qu'on appelle le *paramètre* de la courbe. Un exemple concret de cette idée, est la courbe paramétrique :

$$\vec{\Gamma}(t) = (x(t), y(t)) = (\cos(t), \sin(t))$$

La trajectoire définie par cette courbe pour la valeur  $t$  entre 0 et  $2\pi$  est un cercle de rayon unité. On constate que le paramètre  $t = 0$  correspond au point de référence d'une trajectoire et dans le cas général le paramètre  $t$  correspond à l'angle du point courant et l'axe abscisse.

Pour un contour donné, il peut y avoir plusieurs types de paramètres possibles. De ce fait, pour décrire un contour qui ne dépend que de la courbe de ce contour et invariant par rapport aux groupes de transformations sur ce contour, on doit reparamétriser le contour par rapport à ces groupes de transformations. Les transformations qu'on rencontre le plus souvent en reconnaissance des formes sont : *translation*, *rotation*, *changement d'échelle*, *étirement*, *transformation Euclidienne* (translation ou rotation), *transformation similitude* (transformation Euclidienne ou changement d'échelle), *transformation affine* (transformation similitude ou étirement).

Dans cette section, on s'intéresse à la paramétrisation invariante par rapport à la transformation similitude et à la transformation affine. Pour obtenir une paramétrisation invariant par rapport à la transformation similitude, on utilise l'*abscisse curviligne* ou la *longueur d'arc* du contour qu'on présentera dans la section suivante. Pour obtenir une paramétrisation invariant par rapport à la transformation affine, il y a deux possibilités, ou bien on utilise la paramétrisation par la *longueur affine* (*affine arc length*) (section 2.3.2) ou bien la paramétrisation par la *surface fermée* (*enclosed area*) (section 2.3.3). Le choix entre ces deux paramètres sera discuté dans la section (2.3.4).

### 2.3.1 Paramétrisation par abscisse curviligne ou la longueur d'arc

Supposons que nous ayons les coordonnées des points du contour  $\Gamma = (x_i, y_i)_{1 \leq i \leq n}$  où  $n$  est le nombre de points sur le contour et l'indice  $i$  représente le numéro de points du contour. Pour reparamétriser le contour par la longueur d'arc nous devons calculer :

$$l(i) = \begin{cases} 0 & \text{pour } i = 1 \\ \sum_{k=2}^i \sqrt{(x_k - x_{k-1})^2 + (y_k - y_{k-1})^2} & \text{pour } 2 \leq i \leq n \end{cases} \quad (2.7)$$

et remplacer l'indice  $i$  par l'indice  $l(i)$  pour  $1 \leq i \leq n$ . La valeur  $l(i)$  représente la longueur du segment de contour entre le point de départ  $(x_1, y_1)$  et le point  $(x_i, y_i)$  et  $l(n)$  représente la longueur totale du contour.

Pour obtenir l'invariance du paramètre par rapport au changement d'échelle, on normalise la longueur totale du contour  $l(n)$  à un, c'est à dire qu'on remplace les coordonnées  $(x_i, y_i)$  (resp. le paramètre  $l(i)$ ) par  $(x_i/l(n), y_i/l(n))$  (resp.  $l(i)/l(n)$ ) pour  $1 \leq i \leq n$ . Il faut noter aussi que le choix arbitraire du point de départ sur le contour entraîne simplement un décalage sur le paramètre.

De façon générale, la paramétrisation curviligne invariante par rapport à la transformation similitude sur le contour consiste à calculer :

$$l(t) = \int_a^t \sqrt{x'(u)^2 + y'(u)^2} du, \forall t \in [a, b]$$

où  $x'$  et  $y'$  représentent la dérivée première de  $x$  et  $y$ ,  $a$  et  $b$  représentent le point de référence et l'extrémité du contour.

### 2.3.2 Paramétrisation par la longueur affine

Dans le cas de la transformation affine, on ne peut pas utiliser la paramétrisation par la longueur d'arc car celle-ci ne varie pas linéairement sous l'action affine. On doit alors la remplacer par d'autres paramètres qui transforment linéairement par rapport à la transformation affine. En fait, il y a deux types de paramètres qui se transforment linéairement sous l'action affine, ce sont la *longueur affine* [CV91] et la *surface fermée* [ASBH90] (section suivante).

Le premier paramètre peut être obtenu par la propriété du déterminant [CV91]. Supposons qu'on ait une courbe dans un espace 2D  $(x(t), y(t))$  avec un paramètre arbitraire  $t \in [a, b]$ . Soit  $(x_a(t_a), y_a(t_a))$  représentant une courbe paramétrique après une transformation affine, c'est à dire que :

$$\begin{bmatrix} x_a \\ y_a \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_1 \\ t_2 \end{bmatrix} = A \begin{bmatrix} x \\ y \end{bmatrix} + T \quad (2.8)$$

où  $A$  une matrice carrée non-singulière représente la rotation, le changement d'échelle et l'étirement et le vecteur  $T$  représente la translation.

En notant la dérivée par rapport à  $t$  (respectivement  $t_a$ ) par un point (resp. un prime) on a :

$$\begin{bmatrix} x' & x'' \\ y' & y'' \end{bmatrix} = \begin{bmatrix} \dot{x}t' & \ddot{x}t'^2 + \dot{x}t'' \\ \dot{y}t' & \ddot{y}t'^2 + \dot{y}t'' \end{bmatrix}. \quad (2.9)$$

Comme  $(x(t), y(t))$  et  $(x_a(t_a), y_a(t_a))$  sont liés par une transformation affine, alors, de l'équation (2.8) et (2.9), la relation entre la première et la seconde dérivée par rapport au paramètre  $t_a$  peut s'écrire par :

$$\begin{bmatrix} x'_a & x''_a \\ y'_a & y''_a \end{bmatrix} = A \begin{bmatrix} x' & x'' \\ y' & y'' \end{bmatrix}. \quad (2.10)$$

La longueur affine  $\tau$  est définie par [CV91] :

$$\tau = \int_a^b |\sqrt[3]{\dot{x}\ddot{y} - \ddot{x}\dot{y}}| dt \quad (2.11)$$

Si les deux contours avant et après une transformation affine sont paramétrés par la longueur affine, alors, encore d'après la relation (2.9), (2.10) et (2.11), on peut montrer que les deux paramètres sont liés par une relation :

$$\tau_a = |\sqrt[3]{\det(A)}| \tau$$

Cette dernière relation nous dit que, la longueur affine est une invariante relative par rapport à la transformation affine, autrement dit, il se transforme linéairement sous l'action de la transformation affine.

### 2.3.3 Paramétrisation par la surface fermée

Le deuxième paramètre invariant par rapport à la transformation affine est le paramètre représentant la surface fermée qui est obtenu en se basant sur la propriété de la transformation affine : *sous une transformation affine, toutes les surfaces changent avec la même proportion*. Cela veut dire que si  $F_a$  est une image de la figure  $F$  par une transformation affine, alors leurs surfaces sont liées par la relation :

$$s_a = ks$$

où  $k$  est une constante et  $s$ ,  $s_a$  sont respectivement les surfaces de  $F$  et  $F_a$ . En se basant sur cette propriété, Arbter et al. [ASBH90] définissent un paramètre  $\sigma$ , qui se transforme linéairement sous l'action d'affine, par :

$$\sigma(u) = \frac{1}{2} \int_a^u |x(t)\dot{y}(t) - y(t)\dot{x}(t)| dt, \quad u \in [a, b]. \quad (2.12)$$

où  $t$  est un paramètre initial,  $x(t)$  et  $y(t)$  sont les coordonnées des points sur le contour en considérant que l'origine du système (repère) se trouve au centre de gravité du contour. Dans le cas discret, la dérivée première de  $x$  et de  $y$  peut être calculée par les équations suivantes :

$$\dot{x}(t) = x(t+1) - x(t) \quad \text{et} \quad \dot{y}(t) = y(t+1) - y(t)$$

et donc l'équation (2.12) devient :

$$\sigma(k) = \begin{cases} 0 & \text{si } k = 1 \\ \frac{1}{2} \sum_{t=1}^{k-1} |x(t)y(t+1) - x(t+1)y(t)| & \text{sinon} \end{cases} \quad (2.13)$$

Le paramètre  $\sigma$  représente essentiellement l'accumulation de la somme des surfaces engendrées en reliant le centre de gravité et deux points successifs sur le contour.

### 2.3.4 Discussion

Les deux paramètres  $\tau$  et  $\sigma$  se transforment linéairement par rapport à la transformation affine. On peut rendre ces paramètres complètement invariants en les normalisant par rapport à la totalité ou bien de la longueur affine ou bien de la surface fermée du contour. L'illustration et la signification géométrique des deux paramètres sont données dans [TB97b].

Les deux paramètres invariants à une transformation affine peuvent être utilisés pour paramétrer le contour d'un objet. Mais les propriétés de ces deux paramètres sont différentes et on peut choisir l'un des deux paramètres en fonction de notre application. Les dérivées, première et seconde, sont nécessaires pour calculer la longueur affine  $\tau$  et il est clair que la dérivée est très sensible aux bruits. Donc, si le contour est corrompu par les bruits, le débruitage de contour par un filtre passe-bas est nécessaire avant de paramétrer le contour. La proportion des bruits à éliminer par le processus de débruitage reste à déterminer. C'est pour cette raison que la longueur affine n'est pas souvent utilisée. Par contre, l'utilisation de la surface fermée souffre du problème de non-invariance par rapport à la translation, mais ce problème peut être évité simplement en plaçant le centre de gravité du contour à l'origine du système (repère).

## 2.4 Fixation du point de départ

Dans cette section, on s'intéresse aux différentes méthodes de fixation du point de départ sur les contours. Ces méthodes sont utilisées principalement lors de la phase d'appariement entre les contours <sup>2</sup> ou juste avant la phase de construction du descripteur. La tâche à accomplir par ces méthodes est de trouver la correspondance de chaque élément du premier contour avec celui du deuxième contour. En temps de calcul, il est possible de calculer toutes les correspondances possibles (la recherche exhaustive) pour trouver la solution optimale. Cependant, il est souhaitable de trouver une autre solution pour améliorer le temps de calcul et par la même occasion satisfaire aux conditions 4 et 5 de la norme MPEG-7. Pour cette raison, nous proposons d'étudier et de comparer quatre méthodes pour fixer le *point de départ*. Ces quatre méthodes sont :

1. La longueur maximum
2. La courbure maximum
3. L'analyse en composantes principales (ACP)
4. L'orientation de l'axe naturel

Il faut préciser que, dans le cas où les contours ou descripteurs ne sont pas corrompus par les bruits, les quatre méthodes ci-dessus permettent de trouver le correspondant optimal. Mais dans le cas pratique, les contours que nous obtenons ne correspondent pas exactement aux contours réels des objets c'est à dire qu'ils ne sont pas parfaits et par conséquent la distance qui mesure le degré de similarité obtenu en appliquant ces méthodes ne reflète pas vraiment la distance réelle. Elle donne plutôt une distance sous estimée. Pour obtenir la distance réelle, il faut utiliser le minimum des mesures de corrélations suivantes :

$$d(I_1, I_2) = 1 - \max_{m=0..n} \sum_{i=1}^n \frac{I_1(i)I_2(i+m \bmod n)}{\|I_1\| \|I_2\|}$$

où  $I_1$  et  $I_2$  sont des vecteurs de dimension  $n$ . Dans le cas continu, la distance ci-dessus s'écrit :

$$d(I_1, I_2) = 1 - \max_{\tau \in \mathcal{R}} \frac{\int I_1(t)I_2(t+\tau)dt}{\|I_1\| \|I_2\|}$$

Dans ce qui suit, nous décrivons les quatre méthodes pour fixer le point de départ et ensuite nous discutons et commentons certains aspects de la mise en oeuvre de ces quatre méthodes.

---

<sup>2</sup>ou descripteur des contours

### 2.4.1 Maximum distance

La solution la plus simple pour fixer le point de départ consiste à considérer le point du contour où sa distance par rapport au centre de gravité du contour est maximale. Cela veut dire que le point  $(x_{i_{max}}, y_{i_{max}})$  est considéré comme point de départ si

$$i_{max} = \arg \max_{1 \leq i \leq Nr} (i)$$

### 2.4.2 Maximum courbure

Une autre solution similaire à la précédente consiste à sélectionner le point du contour où sa courbure, en valeur absolue, est maximale. Par définition, la courbure  $k_i$  du contour en un point  $(x_i, y_i)$  est donnée par l'expression :

$$k_i = \frac{\dot{x}_i \ddot{y}_i - \ddot{x}_i \dot{y}_i}{(\dot{x}_i^2 + \dot{y}_i^2)^{3/2}} \quad (2.14)$$

où  $(\dot{x}_i, \dot{y}_i)$  et  $(\ddot{x}_i, \ddot{y}_i)$  représentent respectivement la première et la seconde dérivée du contour au point  $(x_i, y_i)$ . Il existe plusieurs façons pour calculer la courbure  $k_i$  de l'équation (2.14), mais l'approche que nous adoptons ici est de remplacer la première et seconde dérivée du contour par la convolution avec la dérivée première et seconde de la fonction de Gauss <sup>3</sup>  $g(i, \sigma)$ , d'où l'équation (2.14) devient :

$$k_i = \frac{\dot{x}(i, \sigma) \ddot{y}(i, \sigma) - \ddot{x}(i, \sigma) \dot{y}(i, \sigma)}{(\dot{x}^2(i, \sigma) + \dot{y}^2(i, \sigma))^{3/2}}$$

avec

$$\dot{x}(i, \sigma) = x_i * \dot{g}(i, \sigma) \quad \text{et} \quad \ddot{x}(i, \sigma) = x_i * \ddot{g}(i, \sigma)$$

et on a la même formule pour  $\dot{y}(i, \sigma)$  et  $\ddot{y}(i, \sigma)$ .

### 2.4.3 L'axe principal

On peut également considérer le point d'intersection entre le contour avec l'axe principal de ce contour comme point de départ. Pour déterminer l'orientation de l'axe principal, nous calculons le vecteur propre qui correspond à la plus grande valeur propre de la matrice covariance  $C$  :

$$C = \sum_{i=1}^N (x_i - \dot{x}, y_i - \dot{y})^t (x_i - \dot{x}, y_i - \dot{y})$$

Si on fait la normalisation en plaçant l'origine du repère au centre de gravité du contour, c'est à dire qu'on remplace les  $(x_i, y_i)$  par  $(x_i - \dot{x}, y_i - \dot{y})$ , on a donc

$$(\dot{x}, \dot{y}) = (0, 0)$$

d'où

$$C = \frac{1}{N} \sum_{i=1}^N (x_i, y_i)^t (x_i, y_i) \quad (2.15)$$

$$= \begin{pmatrix} \sum_{i=1}^N x_i^2 & \sum_{i=1}^N x_i y_i \\ \sum_{i=1}^N x_i y_i & \sum_{i=1}^N y_i^2 \end{pmatrix} \quad (2.16)$$

$$= \begin{pmatrix} c_{11} & c_{12} \\ c_{12} & c_{22} \end{pmatrix} \quad (2.17)$$

---

<sup>3</sup>Dans notre implémentation, nous avons fixé  $\sigma$  à 1.4



Supposons que  $(\lambda_i)_{1 \leq i \leq 2}$  tel que  $\lambda_1 \geq \lambda_2$  sont des valeurs propres de la matrice covariance  $C$ . On peut les calculer en résolvant l'équation de second degré  $\det(C - \lambda I) = 0$ , d'où ces deux valeurs propres sont données par :

$$\lambda_1 = \frac{c_{11} + c_{22} + \sqrt{(c_{11} - c_{22})^2 + 4c_{12}^2}}{2} \quad (2.18)$$

$$\lambda_2 = \frac{c_{11} + c_{22} - \sqrt{(c_{11} - c_{22})^2 + 4c_{12}^2}}{2} \quad (2.19)$$

On peut donc calculer les deux vecteurs propres associés à ces deux valeurs propres.

Soient  $(e_i = (e_{ix}, e_{iy})^t)_{1 \leq i \leq 2}$  les deux vecteurs propres associés aux deux valeurs propres  $(\lambda_i)_{1 \leq i \leq 2}$  de la matrice covariance  $C$ . Ces deux vecteurs propres satisfont les équations suivantes :

$$\begin{pmatrix} c_{11} - \lambda_i & c_{12} \\ c_{12} & c_{22} - \lambda_i \end{pmatrix} \begin{pmatrix} e_{ix} \\ e_{iy} \end{pmatrix} = 0 \quad \text{pour } 1 \leq i \leq 2 \quad (2.20)$$

et en imposant  $(\|e_i\| = \sqrt{e_{ix}^2 + e_{iy}^2} = 1)_{1 \leq i \leq 2}$ , nous obtenons :

$$e_i = \begin{pmatrix} e_{ix} \\ e_{iy} \end{pmatrix} = \begin{pmatrix} \frac{c_{12}}{\sqrt{(\lambda_i - c_{11})^2 + c_{12}^2}} \\ \frac{\lambda_i - c_{11}}{\sqrt{(\lambda_i - c_{11})^2 + c_{12}^2}} \end{pmatrix} \quad (2.21)$$

De l'équation 2.20, on remarque si  $e_i = (e_{ix}, e_{iy})^t$  est un vecteur propre, alors  $-e_i$  est aussi le vecteur propre. De plus comme la matrice  $C$  est réelle symétrique, on sait que les deux vecteurs propres sont orthogonaux, c'est à dire que  $e_{1x}e_{2x} + e_{1y}e_{2y} = 0$ , on a donc  $e_2 = (-e_{1y}, e_{1x})$  ou  $e_2 = (e_{1y}, -e_{1x})$ .

Par définition, l'axe principal et l'axe secondaire du contour d'un objet sont des axes qui passent par le centre de gravité de l'objet et qui se dirigent selon  $e_1$  et  $e_2$  respectivement et aussi par définition, l'excentricité  $E$  du contour est un rapport entre la valeur propre maximale et la valeur propre minimale :

$$E = \lambda_1 / \lambda_2.$$

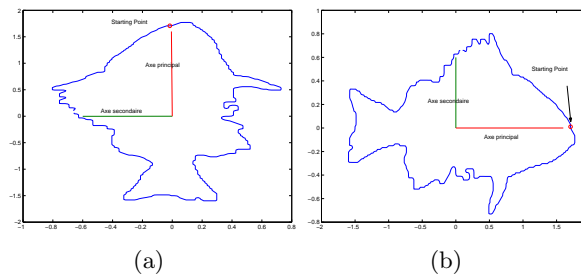


FIG. 2.3 – (a)Le contour avec les composants principaux et le point de départ (b)On place les composants principaux sur l'axe abscisse et l'axe ordonnée

La figure 2.3(a) montre le contour, les composants principaux (l'axe principal et l'axe secondaire) et le point de départ. La figure 2.3(b) montre le contour après avoir placé l'axe principal et

l'axe secondaire sur l'axe abscisse et l'axe ordonnée respectivement, c'est à dire faire la rotation du contour. Cette dernière transformation (la rotation du contour pour que les composants principaux se trouvent sur l'axe abscisse et l'axe ordonnée) est l'équivalence à normaliser le contour par rapport à la rotation.

#### 2.4.4 L'axe naturel

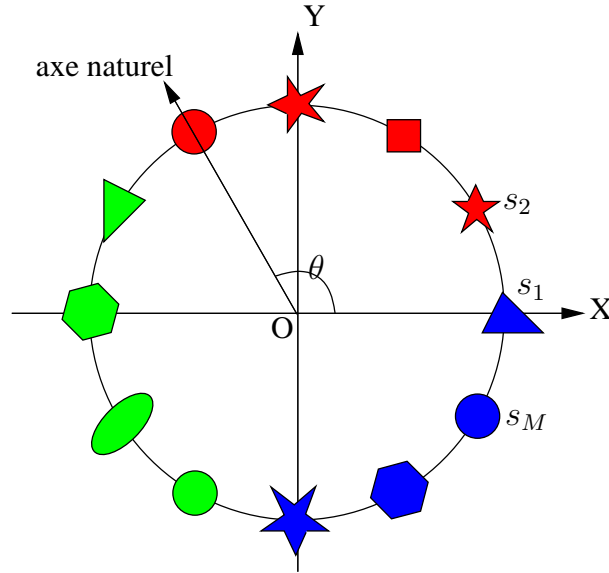


FIG. 2.4 – Le collier de  $N$  perles de rayon un associé à l'ensemble ordonné  $S$

Une autre possibilité est de considérer le point d'intersection entre la direction d'*axe naturel* et le contour comme point de départ. Le principe d'orientation d'axe naturel est le suivant (voir la figure 2.4). On part d'un ensemble d'ordonnées  $S = \{s_i, 1 \leq i \leq N\}$ , on associe à l'ensemble  $S$  un collier de rayon un et  $N$  perles bien réparties uniformément équidistants autour du collier, c'est à dire que, la distance d'une perle à une autre est de longueur  $2\pi/N$ . Le poids de la  $i$  ème perle est égal à  $s_i$  unité de mesure. Alors l'*orientation d'axe naturel* de l'ensemble d'ordonnée  $S$  ou du collier de  $N$  perles est défini par [ShWI99] :

$$\Theta_{naturel} = \text{angle}\left(\sum_{m=1}^N s_m e^{2\pi i(m-1)/N}\right) \quad (2.22)$$

Placer le point de départ à l'intersection entre l'orientation d'axe naturel et le contour est équivalent à choisir le point  $(x_{i_{naturel}}, y_{i_{naturel}})$  comme point de départ, où l'indice  $i_{naturel}$  est défini par

$$i_{naturel} = \left\lfloor \frac{\Theta_{naturel}}{2\pi/N} \right\rfloor + 1 \quad (2.23)$$

avec  $\text{angle}(z)$  renvoie l'angle entre l'intervalle  $[0, 2\pi]$  du nombre complexe  $z$  et  $\lfloor x \rfloor$  renvoie le nombre entier  $i$  tel que  $i \leq x < i+1$ .  $\Theta_{naturel}$  correspond à l'orientation naturelle de la signature  $(s_i)_{1 \leq i \leq N}$  du contour.

**Remarque 1**

Si on regarde la définition d'orientation d'axe naturel de plus près, on constate que l'angle  $\Theta_{naturel}$  correspond à la phase du premier coefficient de la transformée de Fourier de  $S$ .

**2.4.5 Discussions**

Comme on peut le constater, il existe plusieurs méthodes pour fixer le point de départ. Le choix d'une méthode dépend de l'application concernée. Par exemple, il est intéressant de choisir la première méthode, la distance maximum, pour fixer le point de départ si on utilise la représentation de la section 2.2.4, la distance au centre de gravité, ou la représentation par les coordonnées polaires de la section 2.2.1. Par contre, il est plus intéressant d'utiliser l'axe principal si on utilise la représentation par les coordonnées cartésiennes.

Cependant, on sait que la distance maximum est très sensible au bruit. Pour rendre cette méthode plus robuste, il est possible de trier la distance des points du contour par rapport au centre de gravité dans l'ordre décroissant de leurs distances et de choisir les deux ou trois ou plus premiers points comme point de départ. Un point parmi les points sélectionnés est considéré comme le vrai point de départ si la mesure de distance entre le descripteur obtenu en fixant ce point comme point de départ avec un autre descripteur donne la valeur minimum. Donc, cela veut dire que le vrai point de départ du contour est relatif par rapport à un autre contour en comparaison. On peut faire le même constat pour la méthode de maximum courbure. Dans le cas de l'axe principal et de l'axe naturel, au lieu de choisir le deuxième ou le troisième point où la distance au centre de gravité ou la courbure sont de valeur maximum, on peut choisir le point voisin du point d'intersection entre l'axe principal ou l'axe naturel et le contour. L'avantage de la méthode utilisant l'axe principal pour fixer le point de départ est qu'on peut utiliser l'excentricité comme caractéristique globale du contour.

Pour chacune des quatre méthodes ci-dessus, nous pouvons nous poser la question suivante : Peut-on toujours déterminer le point de départ ? Par exemple, dans certains cas :

1. les points de distance maximum au centre de gravité, ou de courbure maximum sont nombreux
2. l'axe principal ou l'axe naturel n'est pas déterminé

Dans les deux cas, il suffit de choisir un point au hasard parmi les choix multiples, ou bien choisir le point où la mesure de distance entre le descripteur obtenu en fixant ce point comme point de départ avec un autre descripteur donne la valeur minimum, comme décrit dans le paragraphe précédent. Dans le cas d'axe naturel, comme dans la remarque 1, au lieu de choisir la transformée de Fourier de la suite des valeurs  $s_i$ , on peut choisir la transformée de Fourier de la suite des valeurs  $s_i^p$ ,  $p > 0$  et au lieu de choisir la phase du premier coefficient de la transformée de Fourier comme la direction d'axe naturel, on peut choisir la phase du *premier coefficient non nul* de la transformée de Fourier comme direction d'axe naturel.

Une autre possibilité est de combiner ces méthodes pour fixer le point de départ. S'il y a l'ambiguïté sur les choix multiples par rapport à la distance maximale, on peut fixer ce point en utilisant la courbure maximale, et s'il y a ambiguïté sur les choix multiples par rapport à la courbure maximale, on peut le fixer en utilisant l'axe principal et ainsi de suite.

## 2.5 Méthode générale pour obtenir l'invariance

Dans cette section, on s'intéresse spécialement aux méthodes permettant d'obtenir l'invariance par la *normalisation* [IRR94][IRV96]. La méthode qu'on va présenter pour obtenir toutes les invariants par normalisation est celle des invariants globaux. La détermination des caractéristiques des invariants reste un problème important dans le système de reconnaissance. Les invariants sont des propriétés géométriques qui restent inchangées sous un certain groupe de transformations. La difficulté principale pour reconnaître un objet par son image est que l'apparence de l'image dépend du point de vue et de la transformation exercée sur l'image. La méthode de normalisation qu'on va présenter fonctionne de façon générale et elle est très facile à appliquer car elle procède étape par étape d'une manière presque mécanique.

### 2.5.1 Le calcul d'invariant

La méthode de normalisation commence par décrire la fonction  $f(x, y)$  (l'image, l'objet...) auquel on s'intéresse par un *descripteur*. Ce descripteur décrit un objet ou une image dont on veut déterminer les propriétés invariantes. La tâche principale pour trouver l'invariant est d'éliminer les effets des transformations qui agissent sur le descripteur.

Soit

$$(x, y) \rightarrow (x', y') \text{ avec } x' = u(x, y, \xi_1, \dots, \xi_n) \text{ et } y' = v(x, y, \xi_1, \dots, \xi_n)$$

une fonction de  $\mathcal{R}^2$  dans  $\mathcal{R}^2$ , où les fonctions  $u$  et  $v$  décrivent l'ensemble des transformations  $\mathcal{T}$  qui dépendent des paramètres  $\xi_1, \dots, \xi_n$ . On suppose qu'il existe des transformations inverses  $u'$  et  $v'$  telles que :

$$f(x, y) \rightarrow f(u'(x', y', \xi_1, \dots, \xi_n), v'(x', y', \xi_1, \dots, \xi_n)) = f'(x', y')$$

La fonction  $f'$  est l'image ou objet résultant de la transformation  $u, v \in \mathcal{T}$ . Il faut souligner que l'application de cette méthode n'est pas restreinte uniquement dans le cas des fonctions bi-dimensionnelles, elle fonctionne aussi dans le cas de dimensions supérieures.

Comme cela a été indiqué précédemment, l'invariant représente une quantité qui reste inchangée même en présence de transformations. Pour une fonction  $f$ , une telle quantité est définie comme une fonction à valeur complexe :

$$F_k(f) = \int f(x, y)k(x, y)dx dy,$$

où  $k$  est un noyau (kernel) de la transformation. Le choix d'un noyau particulier joue un rôle très important dans le processus de la détermination d'invariant.

Si  $f$  subit un certain nombre de transformations, alors la fonction  $F_k(f)$  se transforme en une fonction :

$$F_k(f') = \int f'(x', y')k(x', y')dx' dy'.$$

Une fonction  $I$  est dite *invariante absolue* par rapport à un ensemble de transformations si on a :

$$I_k(f) = I_k(f').$$

La méthode générale pour déterminer l'invariant repose sur quatre étapes [IRR94][IRV96] :

- a) Description de la fonction** : On cherche un descripteur bijectif  $\gamma_{k,l}$  de  $f$
- b) Transformation** : On applique une transformation  $u, v \in \mathcal{T}$  sur le descripteur  $\gamma_{k,l}$ . On obtient un nouveau descripteur  $\gamma'_{k,l}$ . Dans la plupart des cas, on peut écrire  $\gamma'_{k,l}$  comme une de  $\gamma_{k,l}$  et des paramètres de la transformation  $\xi_1, \dots, \xi_n$  de façon explicite.
- c) Présentation des paramètres de la transformation** : Si l'ensemble de la transformation  $\mathcal{T}$  dépend de  $n$  paramètres  $\xi_1, \dots, \xi_n$  alors on doit résoudre un système de  $n$  équations avec  $n$  inconnues  $\xi_1, \dots, \xi_n$  suivants :

$$\begin{cases} \gamma'_{k_1, l_1} = c_1 \\ \vdots \\ \gamma'_{k_n, l_n} = c_n \end{cases}$$

- d) Le calcul de l'invariant** : On remplace les paramètres de transformations  $\xi_1, \dots, \xi_n$  dans la formule  $\gamma'_{k,l}$  obtenue à l'étape **b)**

### 2.5.2 Exemple

Pour illustrer le fonctionnement de la méthode, on prend un exemple très simple. A partir de la définition générale des moments  $m_{pq}$  pour  $p, q = 0, 1, \dots, \infty$  par :

$$m_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x, y) dx dy$$

on va montrer comment on peut arriver à obtenir le moment central qui est invariant par rapport à la translation.

**L'étape a)** Les moments généraux  $m_{pq}$  sont des descripteurs qui décrivent de façon unique la fonction  $f(x, y)$

**L'étape b)** Pour une translation  $x' = x + a$  et  $y' = y + b$ , les moments (descripteur) transformés  $m'_{pq}$  (pour  $p, q = 0, 1, \dots, \infty$ ) sont :

$$m'_{pq} = \sum_i \sum_j \binom{p}{i} \binom{q}{j} a^{p-i} b^{q-j} m_{ij}$$

**L'étape c)** Le choix de fixer les deux moments  $m'_{10}$  et  $m'_{01}$  à 0 est raisonnable pour éliminer les deux paramètres  $a$  et  $b$  :

$$\begin{cases} m'_{10} = m_{10} + a m_{00} = 0 \\ m'_{01} = m_{01} + b m_{00} = 0 \end{cases} \quad \text{d'où} \quad \begin{cases} a = -\frac{m_{10}}{m_{00}} \\ b = -\frac{m_{01}}{m_{00}} \end{cases}$$

**L'étape d)** Les moments centraux :

$$\mu_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \left(x - \frac{m_{10}}{m_{00}}\right)^p \left(y - \frac{m_{01}}{m_{00}}\right)^q f(x, y) dx dy$$

pour  $p, q = 0, 1, \dots, \infty$  sont invariants par rapport à la translation.

Cet exemple montre très bien la simplicité de la méthode. Dans les deux articles de Irène et al. [IRR94][IRV96], les auteurs ont montré dans plusieurs exemples que la plupart des invariants qu'on trouve dans la littérature peuvent être obtenus par la méthode décrite ci-dessus. De plus, ils ont montré que certains de ces invariants sont instables. Les auteurs ont montré aussi qu'on peut obtenir de nombreux nouveaux invariants plus stables que les anciens invariants (voir [IRR94][IRV96] pour plus de détails).

## 2.6 Mesure de similarité et performance

Une fois les images indexées dans la base en utilisant leurs descripteurs, la recherche d'image se résume essentiellement à la détermination de la mesure de similarité entre l'image requête et les images de la base. Cette mesure de similarité correspond principalement à la distance entre les vecteurs caractéristiques (descripteur) qui représentent les images. Il est préférable que cette mesure reflète la perception humaine. Cela veut dire que si deux images sont perceptuellement similaires, leur distance doit être faible et, si deux images sont perceptuellement différentes, leur distance doit être importante (grande). Donc, pour un vecteur caractéristique donné, plus l'exactitude de recherche est élevée, plus la mesure de distance est meilleure. Naturellement pour la recherche en temps réel, on doit aussi tenir compte de l'efficacité du calcul pour choisir une distance.

De nombreuses distances ont été utilisées dans la recherche d'images, elles incluent la distance *city-block*, la distance *Euclidienne*, la distance *cosinus* [Smi97], la distance *d'intersection d'histogramme* [SB91][Smi97], la distance *statistique*  $\chi^2$  [Rub], la distance *quadratique* [Den99] et la distance *Mahalanobis* [Smi97]. Dans cette section, ces distances seront présentées et évaluées. Le but de cette évaluation est de déterminer la mesure de similarité qui correspond le mieux à la perception d'un être humain.

## 2.7 Mesure de similarité

Normalement, la mesure de similarité est définie comme la distance dans un espace métrique. Dans cette section, les différentes mesures de similarité sont décrites en détail.

### La distance d'espace métrique

Pour un espace métrique  $\mathcal{E}$ , si pour tout élément  $x$  et  $y$  de  $\mathcal{E}$ , il existe une fonction  $d : \mathcal{E} \rightarrow \mathcal{R}^+$ , qui satisfait les propriétés suivantes :

$$\begin{array}{ll} \forall x, y \in \mathcal{E} & d(x, y) \geq 0 & \text{(non négativité)} \\ \forall x, y \in \mathcal{E} & d(x, y) = 0 \text{ si et seulement si } x = y & \text{(identité)} \\ \forall x, y \in \mathcal{E} & d(x, y) = d(y, x) & \text{(symétrie)} \\ \forall x, y, z \in \mathcal{E} & d(x, z) \leq d(x, y) + d(y, z) & \text{(inégalité triangulaire)} \end{array}$$

$d$  est une distance de  $\mathcal{E}$ .

### La distance de Minkowski

La distance de Minkowski est définie en se basant sur la norme d'espace métrique  $\mathcal{L}^p$  :

$$d_p(X, Y) = \left( \sum_{i=1}^N (X_i - Y_i)^p \right)^{1/p}$$

avec  $X = (X_1, X_2, \dots, X_N)$  et  $Y = (Y_1, Y_2, \dots, Y_N)$  sont respectivement des descripteurs de l'image requête et de l'image candidate.

Pour  $p = 1$ ,  $d_1(X, Y)$  est appelé la distance *city-block* ou la distance *Manhattan* ( $\mathcal{L}^1$ )

$$d_1(X, Y) = \sum_{i=1}^N |X_i - Y_i|$$

Pour  $p = 2$ ,  $d_2(X, Y)$  est une distance *Euclidienne* ( $\mathcal{L}^2$ )

$$d_2(X, Y) = \left( \sum_{i=1}^N (X_i - Y_i)^2 \right)^{1/2}$$

Pour  $p \rightarrow \infty$ , on obtient une distance sur  $\mathcal{L}^\infty$  définie par

$$d_\infty(X, Y) = \max_{1 \leq i \leq N} \{|X_i - Y_i|\}$$

### La distance en cosinus

La distance cosinus est une distance qui est obtenue par le calcul de la différence entre les directions de ses vecteurs sans tenir compte de la longueur de chacun de ses vecteurs. La distance est définie par l'angle entre les deux vecteurs en paramètre. D'après la formule du produit scalaire on a :

$$X \cdot Y = X^t Y = |X| |Y| \cos \theta$$

$$d_{\cos}(X, Y) = 1 - \cos \theta = 1 - \frac{X^t Y}{|X| |Y|}$$

La figure (2.5) montre la différence entre la distance cosinus et les deux distances  $L^1$ ,  $L^2$  dans un espace à deux dimensions. Comme on peut le voir, la distance Euclidienne prend en compte à la fois l'angle et en même temps le module des deux vecteurs pour calculer cette distance (figure 2.5(a)), par contre la distance cosinus prend uniquement l'angle entre les deux vecteurs en compte. Par conséquent,  $Q_1$  et  $Q$  ont la même distance par rapport à  $T$ , c'est à dire que  $d_{\cos}(Q, T) = d_{\cos}(Q_1, T)$  (figure (2.5)(b)). En ce qui concerne la distance  $d_1$ , les différences entre les composants de chaque dimension sont prises en compte (figure 2.5(c)).

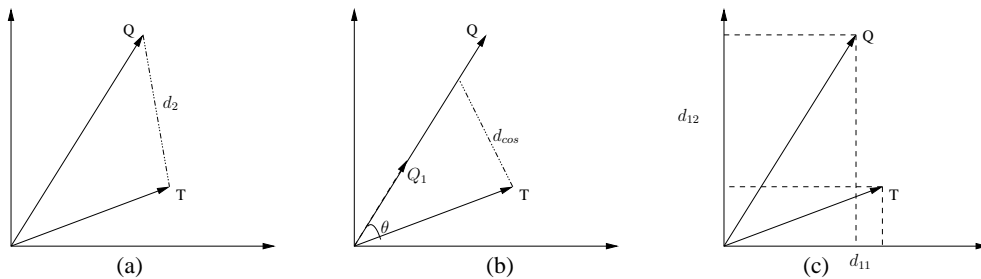


FIG. 2.5 – (a) La distance Euclidienne (b) La distance cosinus (c) La distance  $L^1$  ( $d_1 = d_{11} + d_{12}$ )

## La distance $\chi^2$

La distance  $\chi^2$  statistique est définie par :

$$d_{\chi^2}(X, Y) = \sum_{i=1}^N \frac{(X_i - Y_i)^2}{m_i}$$

où  $m_i = \frac{X_i + Y_i}{2}$ . Cette quantité mesure la non-vraisemblance entre les deux distributions en paramètre [Rub].

## Intersection d'histogramme

L'intersection entre l'histogramme a été proposé par Swain et Ballard [SB91]. L'objectif est de retrouver l'image dans une base d'images en utilisant l'histogramme de couleur. L'origine de la définition de l'intersection d'histogramme est la suivante :

$$d_{hi}(X, Y) = 1 - \frac{\sum_{i=1}^N \min(X_i, Y_i)}{|X|}$$

Il a étendu plus tard dans [Smi97] en

$$d_{hi}(X, Y) = 1 - \frac{\sum_{i=1}^N \min(X_i, Y_i)}{\min(|X|, |Y|)}$$

## La distance quadratique

Les mesures de distances présentées précédemment ne prennent en compte uniquement que les correspondantes de chaque dimension et elles n'utilisent pas les informations de l'ensemble des dimensions. La distance quadratique est proposée pour palier à cette lacune [Smi97]. La distance quadratique entre deux vecteurs caractéristiques  $X$  et  $Y$  est définie par :

$$d_{qad}(X, Y) = [(X - Y)^t A (X - Y)]^{1/2} \quad (2.24)$$

où  $A = [a_{ij}]$  est une matrice de dimension  $N * N$ , et  $a_{ij}$  sont les coefficients de similarité entre les indices (dimension)  $i$  et  $j$ .  $a_{ij}$  est définie par :

$$a_{ij} = 1 - d_{ij}/d_{max} \quad \text{avec} \quad d_{ij} = |X_i - Y_j|$$

Pour calculer, la distance quadratique est réécrite ([Den99]) comme suit :

$$d_{qad}(X, Y) = \left( \sum_{i=1}^N \sum_{j=1}^N a_{ij} X_i X_j + \sum_{i=1}^N \sum_{j=1}^N a_{ij} Y_i Y_j + \sum_{i=1}^N \sum_{j=1}^N X_i Y_j \right)$$

## La distance Mahalanobis

La distance Mahalanobis est un cas particulier de la distance quadratique dans lequel la matrice de transformation est définie par la matrice de covariance qui est obtenue par apprentissage sur l'ensemble des vecteurs caractéristiques, c'est à dire que  $A = \Sigma^{-1}$ . L'application de la distance Mahalanobis consiste à considérer les vecteurs caractéristiques comme des variables aléatoires  $V = [V_1, V_2, \dots, V_N]$  et ensuite on calcule la matrice de corrélation  $C = [c_{ij}]$  avec  $c_{ij} = E(V_i V_j)$  où  $E(X)$  est l'espérance mathématique de la variable  $X$ . Donc on peut calculer



la matrice de covariance  $\Sigma = [\sigma_{ij}^2]$  avec  $\sigma_{ij}^2 = c_{ij} - E(V_i)E(V_j)$ .

La distance Mahalanobis entre deux vecteurs caractéristiques  $X$  et  $Y$  est obtenue en remplaçant  $V_X = X$  and  $V_Y = Y$  dans l'équation 2.24, on a donc :

$$d_{mah} = [(V_X - V_Y)\Sigma^{-1}(V_X - V_Y)]^{1/2}$$

Dans le cas particulier où les  $V_i$  sont statistiquement indépendants, mais ont des variances différentes,  $\Sigma$  est une matrice diagonale :

$$\Sigma = \begin{pmatrix} \sigma_1^2 & & & 0 \\ & \sigma_2^2 & & \\ & & \ddots & \\ 0 & & & \sigma_N^2 \end{pmatrix}$$

Dans ce cas, la distance Mahalanobis est réduite à une forme simplifiée :

$$d_{mah}(X, Y) = \sum_{i=1}^N \frac{(X_i - Y_i)^2}{\sigma_i^2}$$

qui est une version pondérée de la distance de  $\mathcal{L}^2$ , qui donne plus de poids à la dimension où la variance est faible et moins de poids à la dimension où la variance est grande.

## 2.8 Mesure de Performance

Avant d'évaluer la performance des différents algorithmes de recherche d'images, on doit avant tout définir ce qu'on appelle la *mesure de performance*. Plusieurs mesures de performance ont été proposées dans [LS98][Bim99]. Les mesures de performance sont généralement basées sur des tests subjectifs de statistiques, et cela entraîne différentes définitions de l'efficacité de la recherche. Ces mesures de performance sont présentées dans ce qui suit.

### 2.8.1 Rappel et Précision (*RP*)

*RP* est l'une des mesures de performance qui est largement utilisée dans la littérature. Elle est basée sur le critère de sélection binaire. Dans cette méthode, chaque image de la base est considérée ou bien similaire ou bien non-similaire par rapport à la requête et par rapport aux critères pré-définis. Ces critères peuvent être par exemple le jugement d'un certain nombre d'individus. Si le nombre d'individus, qui considèrent que l'image est similaire, est supérieur à un seuil alors l'image concernée est considérée comme *similaire*, sinon elle est considérée comme *non-similaire*. Le résultat final du système par rapport à une requête est qu'on arrive à regrouper les images de la base en deux groupes, un groupe qui ressemble à la requête et un autre groupe non. Le *rappel* et la *précision* sont définis par :

$$R = \frac{r}{n_2} = \frac{\text{nombre d'images similaires trouvées}}{\text{nombre images similaires dans la base}}$$

$$P = \frac{r}{n_1} = \frac{\text{nombre d'images similaires trouvées}}{\text{nombre images recherchées}}$$

La *précision* mesure l'exactitude de la recherche tandis que le *rappel* mesure la capacité du système à retrouver l'image similaire dans la base. Le *rappel* et la *précision* sont inversement proportionnels, c'est à dire que la *précision* diminue quand le *rappel* augmente.

### 2.8.2 Le pourcentage de poids des morceaux

Le pourcentage de poids des morceaux (*PWH* <sup>4</sup>) est similaire au mesure *RP*. Le test subjectif est le même que celui de *RP*, mais au lieu de prendre la décision sur la similarité de chaque image par rapport à la requête, on s'intéresse au nombre d'individus qui considèrent l'image comme similaire. Le poids ( $w_i$ ) de chaque image est égal à ce nombre d'individus. Le résultat final du test subjectif est qu'on arrive à donner un poids de similarité de chaque image par rapport à une requête. Pour mesurer la performance d'un système de reconnaissance en utilisant cette mesure de performance, on commence d'abord par fixer le nombre d'images recherchées, et ensuite on calcule le pourcentage de poids des morceaux  $P$  qui est défini par :

$$P = \frac{\sum_{i=1}^n w_i}{\sum_{j=1}^N w_j}$$

où  $n$  est le nombre d'images recherchées et  $N$  est le nombre total d'images dans la base,  $w_i$  et  $w_j$  sont des poids de similarité des images  $i$  et  $j$  par rapport à la requête.

On peut constater que le pourcentage de poids des morceaux  $P$  ressemble à la mesure *rappel* ou *recall* sauf qu'il prend en compte le nombre d'individus. En pratique, pour mesurer la performance d'un système de reconnaissance en utilisant cette mesure, plusieurs requêtes sont utilisées et la valeur moyenne des valeurs  $P$  par rapport à ces requêtes est considérée comme la mesure de performance du système.

### 2.8.3 Le pourcentage du classement de similarité

Le pourcentage du classement de similarité, *PSR* <sup>5</sup> est proposé par Del Bimbo et Pala [Bim99]. Dans cette mesure, chaque individu fait son classement de similarité sur toutes les images de la base par rapport à une requête au lieu de dire que oui ou non l'image est similaire à la requête comme dans *RP* et *PWH*. Le résultat final de ce test subjectif est qu'on obtient une matrice  $\{Q_j(i, k)\}$ , où  $Q_j(i, k)$  indique le nombre d'individus qui considèrent que l'image  $i$  est à la  $k$ -ième position par rapport à la requête  $j$ . On peut donc calculer la moyenne  $\bar{p}_j(i)$  et la variance  $\bar{\sigma}_j(i)$  de chaque ligne de la matrice.  $\bar{p}_j(i)$  et  $\bar{\sigma}_j(i)$  représentent la valeur moyenne du classement de l'image  $i$  par rapport à la requête  $j$  et la variation du classement autour de  $\bar{p}_j(i)$ .

Si pour une requête  $j$ , le système de reconnaissance renvoie l'image  $i$  à la position  $p_j(i)$ , alors la performance du système d'être en accord avec le jugement des individus est mesurée par le *pourcentage du classement de similarité* :

$$S_j(i) = \sum_{k=p_j(i)-\frac{\sigma_j(i)}{2}}^{p_j(i)+\frac{\sigma_j(i)}{2}} Q_j(i, k)$$

---

<sup>4</sup>Percentage of Weighted Hits

<sup>5</sup>Percentage of Similarity of Ranking

La courbe de la fonction  $S_j(i)$ , qui dépend de la valeur  $p_j(i)$ , représente la performance du système de reconnaissance : plus la valeur de  $S_j(i)$  est grande, meilleure est la qualité du résultat de la reconnaissance.

#### 2.8.4 Discussions

Sur les trois mesures de performance précédentes, *PWH* prend en compte le nombre d'individus qui considèrent l'image concernée comme similaire. Comme on le souhaitait, cela correspond bien aux jugements donnés par des humains, mais en même temps *PWH* ne mesure pas la capacité de rejeter l'image non-similaire. La valeur de *PWH* peut être élevée alors qu'il y a plusieurs images non-similaires dans la liste des résultats de la reconnaissance. Un autre désavantage de *PWH* est qu'il suppose qu'on fixe auparavant le nombre d'images recherchées. Cela est contraignant car différentes requêtes peuvent avoir un nombre d'images similaires différent.

Pour *PSR*, on prend en compte le nombre et la qualité de classement des individus. Mais pour une requête, si le pourcentage d'individus qui a classé l'image particulière à une position donnée est assez élevé, alors la variance pour cette requête sera faible et par conséquent le *PSR* sera mauvais si le classement donné par le système de reconnaissance est différent de celui du classement donné par des individus. Par contre si la variance est élevée, alors le *PSR* sera lui aussi élevé même si la différence entre le classement donné par le système de reconnaissance et le classement donné par les individus est important.

Pour *RP*, on mesure à la fois la capacité de retrouver les images similaires et aussi la capacité de rejeter les images non-similaires. Le seul inconvénient de *RP* c'est qu'il ne prend pas en compte le degré de similarité de l'image par rapport à la requête. Mais cet inconvénient n'est pas significatif si les images de la base sont classifiées.

Globalement, sur l'ensemble de cette discussion, il est préférable d'utiliser *RP* comme mesure de performance plutôt que *PWH* et *PSR*. L'utilisation *RP* comme mesure de performance est particulièrement intéressante dans le cas où la base d'images est classée et dans le cas où la quantité d'images de cette base est importante.

## 2.9 Evaluation des mesures de distance

Dans la section 2.7, nous avons présenté et discuté plusieurs mesures de distance. Afin de déterminer quelle est la mesure de distance qui convient le mieux pour le système de reconnaissance, Zhang [Zha02][ZL03b] a évalué et testé la performance de ce système de reconnaissance utilisant le descripteur de Fourier sur chacune de ces distances sur l'ensemble B de la base standardisée CE1-MPEG7 (voir [LJLE00][Zha02]). Le tableau 2.1 donne le résultat d'évaluation de comparaison des mesures de similarité.

On comprend à partir du résultat d'évaluation, que l'intersection d'histogramme est très mauvaise en performance pour mesurer les caractéristiques d'images qui n'utilisent pas d'histogramme. La performance du système utilisant la distance quadratique et la distance Mahalanobis est relativement faible par rapport à la distance Euclidienne. Cela résulte du fait que le poids associé à chaque dimension du vecteur descripteur ne reflète pas exactement l'importance de cette dimension. En général, la distance Euclidienne, la distance *city-block* et la distance statistique

Rappel	10	20	30	40	50	60	70	80	90	100	Moyenne
City Block	93.4	81.1	71.2	65.7	59.6	54.1	47.5	43.0	34.3	25	57.5
Euclidien	93.0	81.0	70.8	65.6	58.7	53.4	47.0	41.8	33.3	24.8	57.0
L-infinie	91.6	77.8	66.5	60.2	53.0	47.3	40.3	35.3	28.0	20.2	52.0
Cosinus	90.0	73.8	62.5	56.8	51.2	46.2	41.1	36.6	30.0	21.4	51.0
$\chi^2$ statistique	93.7	82.5	72.1	66.7	60.5	56.0	50.0	44.5	36.4	26.9	58.9
Histo. Inter.	37.3	27.9	23.1	21.6	20.3	19.0	18.0	16.6	15.1	12.5	21.1
Quadratique	83.3	59.5	45.1	36.3	29.8	24.2	19.7	16.0	12.5	8.6	33.5
Mahalanobis	77.9	54.5	41.4	34.0	28.2	23.1	18.9	15.0	11.3	7.3	31.2

TAB. 2.1 – Résultat d'évaluation rappel-précision du système de reconnaissance utilisant le descripteur de Fourier sur l'ensemble B de la base standard CE1-MPEG7

$\chi^2$  sont préférables en terme de performance. En temps de calcul, mise a part la distance quadratique dont la complexité est de l'ordre  $O(N^2)$ , toutes les autres distances sont de complexité  $O(N)$  ( $N$  est la dimension du vecteur descripteur).

En résumé, en terme de complexité et d'efficacité, la distance *city-block*, la distance Euclidienne et la distance statistique  $\chi^2$  sont meilleures que les autres distances pour mesurer la similarité entre les vecteurs descripteurs. Mais la distance *city-block* est plus simple à utiliser car elle est la plus simple à calculer par rapport à la distance Euclidienne et la distance statistique  $\chi^2$ , ce qui est recherché pour les applications en temps réel.

## 2.10 Conclusion

Dans ce chapitre, nous avons présenté un certain nombre d'outils qui devront nous servir dans la suite pour l'algorithme d'extraction des vecteurs descripteurs (chapitre 3 et 4). En particulier, nous avons présenté plusieurs façons pour représenter et paramétrer le contour. Nous avons aussi présenté quatre méthodes pour choisir le point de départ. Ces méthodes sont indispensables pour certaines techniques de calcul d'invariants. Le détail sur la discussion et la comparaison de ces quatre méthodes feront l'objet d'études expérimentales dans le chapitre 5.

Nous avons aussi présenté une méthode générale, illustrée par un exemple concret, pour déterminer l'invariant. Cette méthode est très intéressante car dans le système de reconnaissance, les indexes utilisées pour retrouver un objet ou pour indexer les objets de la base sont des invariants.

Dans ce chapitre, nous avons également présenté et discuté des mesures de similarités et des mesures de performances. Les différentes mesures de similarités ont été évaluées en utilisant la base de données standard des contours. Le résultat d'expérimentations montre que la distance *city-block* est la plus appropriée pour la reconnaissance des formes basée sur le contour. Par conséquent, cette distance sera utilisée lors de nos prochaines expérimentations.

# Chapitre 3

## Descripteur de formes

### 3.1 Introduction

Le descripteur de formes basé sur le contour est une technique très importante parmi les techniques de reconnaissance des formes. Le descripteur de formes basé sur le contour nécessite, généralement, moins de temps de calcul, moins d'espace de mémoire à utiliser et à gérer par rapport, par exemple, aux méthodes de reconnaissance des formes basées sur la région. Le descripteur de formes basé sur le contour peut être très bien utilisé dans les applications où les informations sur le contour sont disponibles. Dans la littérature, le descripteur de formes basé sur le contour est la technique de description de formes la plus populaire en raison de sa simplicité au niveau de l'implémentation et de la compréhension. Les descripteurs de formes basés sur le contour ont été appliqués avec succès dans plusieurs applications par exemple, la reconnaissance d'objets, le codage de formes, la reconnaissance des caractères, etc.

Dans le chapitre précédent, nous avons présenté les principaux outils qui vont servir lors de la phase d'extraction des vecteurs descripteurs de formes. Nous avons aussi présenté des mesures de similarité entre ces vecteurs et des mesures de performance. Dans ce chapitre, nous nous intéressons aux méthodes proprement dit permettant d'extraire les vecteurs descripteurs de formes. Nous commençons tout d'abord par la présentation de la méthode d'extraction des vecteurs caractéristiques la plus ancienne et la plus connue dans la section 3.2, le descripteur de moment, ensuite également ancien, le descripteur de Fourier dans la section 3.3, suivi du descripteur de Fourier à fenêtre dans la section 3.4. Avant de finir, la section 3.5 présente le descripteur de contour basé sur la courbure multi-échelle et enfin, la section 3.6 conclut et termine le chapitre.

### 3.2 Descripteur de moment

Les moments et les fonctions des moments font partie des descripteurs les plus populaires. Ils sont largement utilisés dans plusieurs types d'applications pour réaliser un système de reconnaissance d'images 2D ou 3D. Dans cette section, plusieurs types de moments et leurs propriétés sont présentés.

### 3.2.1 Les moments géométriques

Les *moments géométriques* 2D d'ordre  $(p + q)$  d'une fonction  $f(x, y)$  peuvent être considérés comme la décomposition de la fonction  $f(x, y)$  sur l'ensemble des fonctions de base  $x^p y^q$ , et ils sont définis par :

$$M_{pq} = \int_{a_1}^{a_2} \int_{b_1}^{b_2} x^p y^q f(x, y) dx dy, \quad p, q \in \mathcal{N} \quad (3.1)$$

L'utilisation des moments pour identifier et analyser une image a été inspirée par Hu [Hu62]. Hu a énoncé un théorème très important qui dit que :

#### Théorème 1

Si la fonction  $f(x, y)$  est continue par morceaux et définie sur une région bornée, alors la suite des moments  $\{M_{pq}\}$  est déterminée de manière unique par la fonction  $f(x, y)$ , et inversement, la fonction  $f(x, y)$  est déterminée de manière unique par la suite des moments  $\{M_{pq}\}$ .

Vu le fait que la fonction  $f(x, y)$  qui représente le contour ou l'image vérifie parfaitement les hypothèses du théorème, donc on peut représenter le contour ou l'image par une suite infinie de moments. Mais comme dans le cadre du descripteur de Fourier, il est important de ne sélectionner qu'un nombre fini de coefficients des moments les plus significatifs pour représenter le contour.

De façon similaire au moment géométrique  $M_{pq}$ , le *moment central* d'ordre  $p + q$  est défini par :

$$\mu_{pq} = \int_{a_1}^{a_2} \int_{b_1}^{b_2} (x - \bar{x})^p (y - \bar{y})^q f(x, y) dx dy, \quad p, q \in \mathcal{N}$$

avec

$$\bar{x} = \frac{M_{10}}{M_{00}}, \quad \bar{y} = \frac{M_{01}}{M_{00}}$$

L'intérêt du moment  $\mu_{pq}$  par rapport au moment  $M_{pq}$  c'est que  $\mu_{pq}$  est invariant en translation, mais dans la pratique cette propriété n'est pas suffisante. Un des premiers travaux les plus significatifs dans le domaine de la reconnaissance utilisant les moments a été effectué par Hu [Hu62]. En se basant sur la théorie d'invariants algébriques, Hu a défini les sept fonctions suivantes, calculées en utilisant des moments centraux jusqu'à l'ordre trois, qui sont invariants par rapport à la transformation similitude :

$$\begin{aligned} \phi_1 &= \mu_{20} + \mu_{02} \\ \phi_2 &= (\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2 \\ \phi_3 &= (\mu_{30} - 3\mu_{12})^2 + (3\mu_{21} - \mu_{03})^2 \\ \phi_4 &= (\mu_{30} - \mu_{12})^2 + (\mu_{21} - \mu_{03})^2 \\ \phi_5 &= (\mu_{30} - 3\mu_{12})(\mu_{30} + \mu_{12})[(\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu_{03})^2] \\ &\quad + (3\mu_{21} - \mu_{03})(\mu_{21} + \mu_{03})[3(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2] \\ \phi_6 &= (\mu_{20} - \mu_{02})[(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2] \\ &\quad + 4\mu_{11}(\mu_{30} + \mu_{12})(\mu_{21} + \mu_{03}) \\ \phi_7 &= (3\mu_{21} - \mu_{03})(\mu_{30} + \mu_{12})[(\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu_{03})^2] \\ &\quad - (\mu_{30} - 3\mu_{12})(\mu_{21} + \mu_{03})[3(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2] \end{aligned}$$

### 3.2.2 Les moments complexes

La notion des moments complexes a été introduite dans [AMP84]. De façon similaire au moment géométrique, le moment complexe d'ordre  $(p, q)$  de la fonction image  $f(x, y)$  est défini par :

$$C_{pq} = \int_{a_1}^{a_2} \int_{b_1}^{b_2} (x + jy)^p (x - jy)^q f(x, y) dx dy, \quad p, q \in \mathcal{N} \text{ et } j = \sqrt{-1}$$

Le moment complexe d'ordre  $(p, q)$  est une combinaison linéaire, avec des coefficients complexes, de tous les moments géométriques  $\{M_{nm}\}$  pour les indices  $n$  et  $m$  tel que  $n + m = p + q$ .

Dans le cas des coordonnées polaires, les moments  $C_{pq}$  peuvent s'écrire :

$$C_{pq} = \int_0^{2\pi} \int_0^\infty \rho^{p+q} e^{j(p-q)\theta} f(\rho \cos \theta, \rho \sin \theta) dx dy$$

Si les moments complexes, dans les mêmes coordonnées polaires, de l'image originale et sa transformation par une rotation d'angle  $\theta$ , sont notés par  $C_{pq}$  et  $C_{pq}^r$ , alors ils vérifient la relation suivante :

$$C_{pq}^r = e^{-j(p-q)\theta} C_{pq}$$

Les invariants des moments complexes peuvent s'écrire sous la forme :

$$C_{rs} C_{tu}^k + C_{sr} C_{ut}^k, \quad \text{avec } (r - s) + k(t - u) = 0$$

Cependant, ces invariants ne sont pas, en général, un bon descripteur et ils souffrent de la perte, de la suppression, et de la redondance de l'information qui limitent leur puissance de discrimination.

### 3.2.3 Les moments orthogonaux

La définition du moment géométrique 3.1 a la forme de la projection de la fonction  $f(x, y)$  sur la base des polynômes  $x^p y^q$ . Cependant, cette base, bien que complète, n'est pas orthogonale. Le critère d'orthogonalité de la base est très important : si une base ne vérifie pas cette condition, la représentation par la suite des moments est redondante.

Dans cette section, on présentera deux types de moments orthogonaux, les moments de Legendre et les moments de Zernike.

#### 3.2.3.1 Les moments de Legendre

Les moments de Legendre d'ordre  $(p + q)$  de la fonction  $f(x, y)$  sont définis par :

$$L_{mn} = \frac{(2m + 1)(2n + 1)}{4} \int_{-1}^1 \int_{-1}^1 P_m(x) P_n(y) f(x, y) dx dy, \quad m, n \in \mathcal{N}$$

Le polynôme de Legendre  $\{P_m(x)\}$  sont des bases orthogonales complètes sur l'intervalle  $[-1, 1]$  :

$$\int_{-1}^1 P_m(x) P_n(x) dx = \frac{2}{2m + 1} \delta_{mn}$$

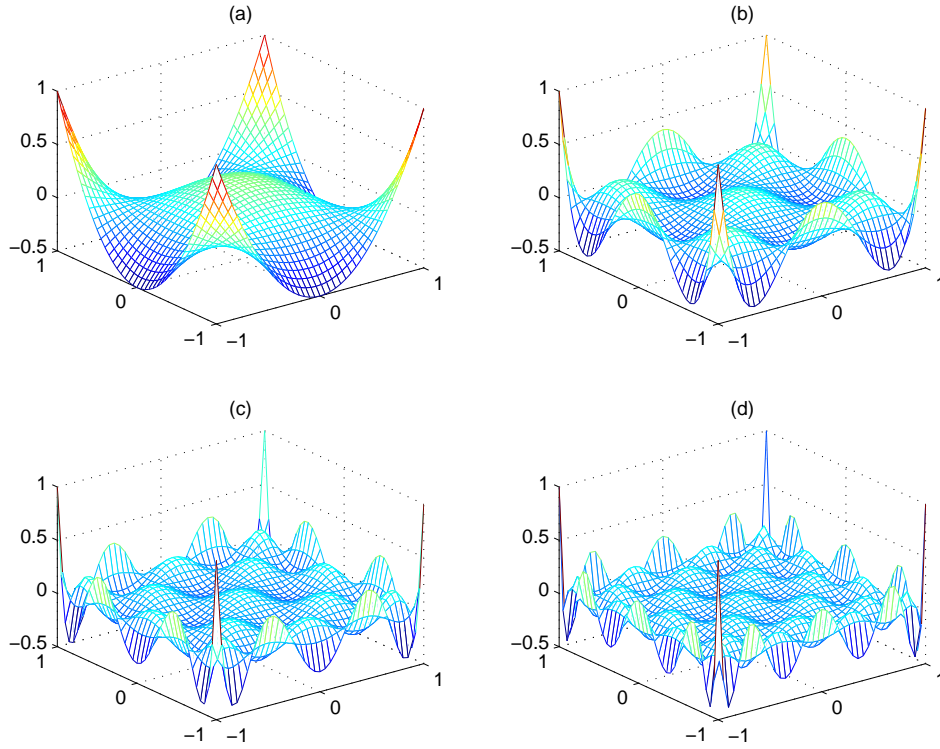


FIG. 3.1 – Polynôme de Legendre dans l'espace 2D (a)  $P_2(x)P_2(y)$  (b)  $P_4(x)P_4(y)$  (c)  $P_6(x)P_6(y)$  (d)  $P_8(x)P_8(y)$

où  $\delta_{mn}$  est un symbole de Kronecker. Le polynôme de Legendre d'ordre  $n$  est défini par :

$$P_n(x) = \frac{1}{2^n} \sum_{m=0}^{n/2} (-1)^m \frac{(2n-2m)!}{m!(n-m)!(n-2m)!} x^{n-2m} \quad (3.2)$$

ou simplement :

$$P_n(x) = \sum_{m=0}^n C_{nk} x^k$$

où les coefficients de Legendre,  $C_{nk}$ , sont définis par :

$$C_{nk} = (-1)^{(n-k)/2} \frac{1}{2^n} \frac{(n+k)!}{[(n-k)/2]! [(n+k)/2]! k!}, \quad \text{avec } n-k = \text{pair}$$

En fait, l'équation 3.2 est exactement la même que l'équation :

$$P_n(x) = \frac{1}{2^n n!} \frac{d^n}{dx^n} (x^2 - 1)^n$$

Ce polynôme est généré à partir de la fonction génératrice :

$$\frac{1}{\sqrt{1-2rx+r^2}} = \sum_{s=0}^{\infty} r^s P_s(x), \quad r < 1$$



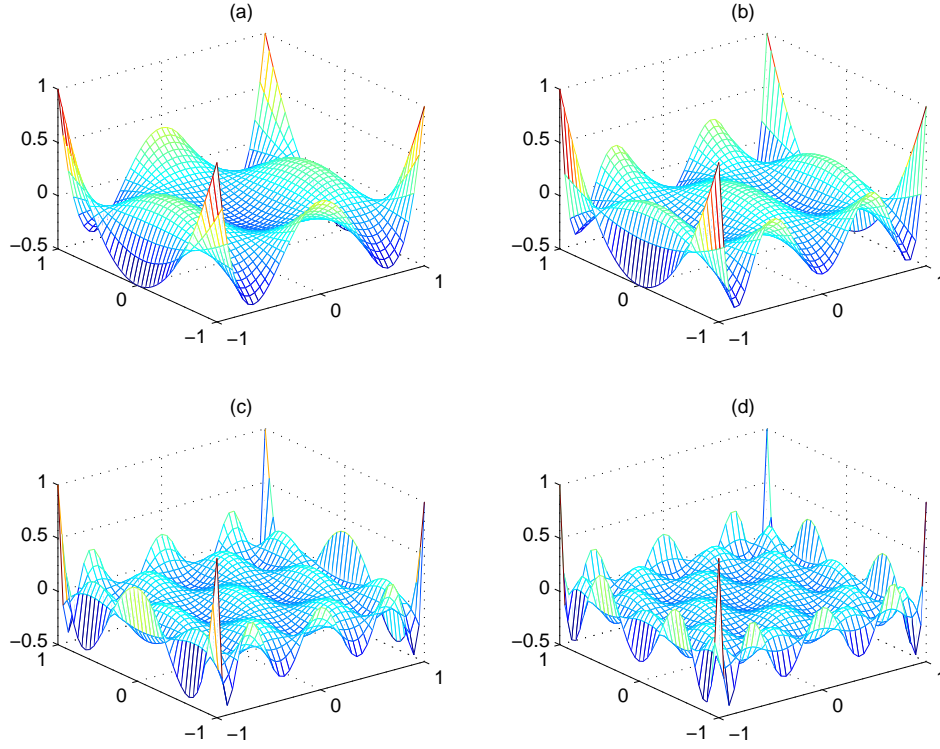


FIG. 3.2 – Polynôme de Legendre dans l'espace 2D (a)  $P_2(x)P_4(y)$  (b)  $P_2(x)P_6(y)$  (c)  $P_4(x)P_8(y)$  (d)  $P_6(x)P_8(y)$

En appliquant la dérivée sur cette équation, on peut déduire la formule de récurrence du polynôme de Legendre par :

$$\begin{aligned} \frac{x-r}{(1-2rx+r^2)^{3/2}} &= \sum_{s=0}^{\infty} sr^{s-1}P_s(x) \\ (x-r) \sum_{s=0}^{\infty} r^s P_s(x) &= (1-2rx+r^2) \sum_{s=0}^{\infty} sr^{s-1}P_s(x) \\ xP_k(x) - P_{k-1}(x) &= (k+1)P_{k+1}(x) - 2xkP_k(x) + (k-1)P_{k-1}(x) \end{aligned}$$

On obtient donc la formule de récurrence suivante :

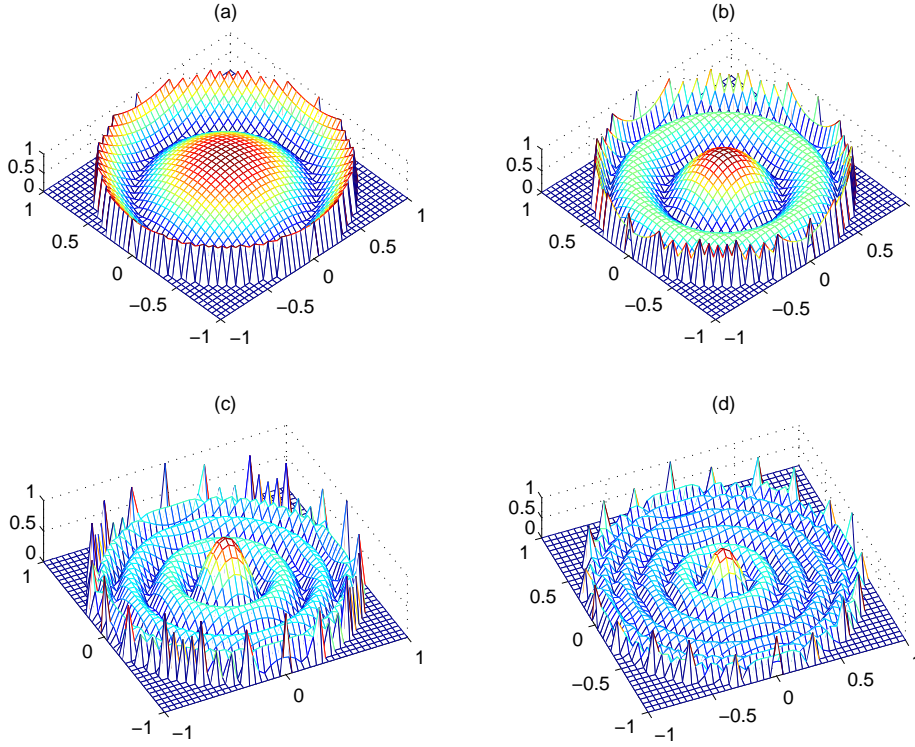
$$P_{n+1}(x) = \frac{2n+1}{n+1}xP_n(x) - \frac{n}{n+1}P_{n-1}(x)$$

La figure 3.1 et 3.2 montre certains polynômes de Legendre dans l'espace 2D.

### 3.2.3.2 Les moments de Zernike

Le moment complexe de Zernike d'ordre  $n$  à  $m$  répétitions est défini par :

$$A_{nm} = \frac{n+1}{\pi} \int_0^{2\pi} \int_0^1 \rho f(\rho, \theta) V_{nm}^* d\rho d\theta \quad (3.3)$$


 FIG. 3.3 – Polynôme de Zernike (a) $|V_{20}|$  (b) $|V_{40}|$  (c) $|V_{80}|$  (d) $|V_{12,0}|$ 

où  $V_{nm}(\rho, \theta)$  sont des polynômes de Zernike :  $V_{nm}(r, \theta) = R_{nm}(r)e^{jm\theta}$  avec

$$n \in \mathcal{N}, m \in \mathcal{Z} \text{ tel que } n - |m| = \text{pair et } |m| \leq n \quad (3.4)$$

et  $R_{nm}$  sont des polynômes de radial définis par

$$R_{nm} = \sum_{s=0}^{(n-|m|)/2} (-1)^s \frac{(n-s)!}{s! \left(\frac{n+|m|}{2} - s\right)! \left(\frac{n-|m|}{2} - s\right)!} r^{n-2s} \quad (3.5)$$

La figure 3.3 et la figure 3.4 montrent quelques exemples des polynômes de Zernike.

On peut constater que les polynômes  $R_{nm}$  de l'équation 3.5 satisfait les relations suivantes :

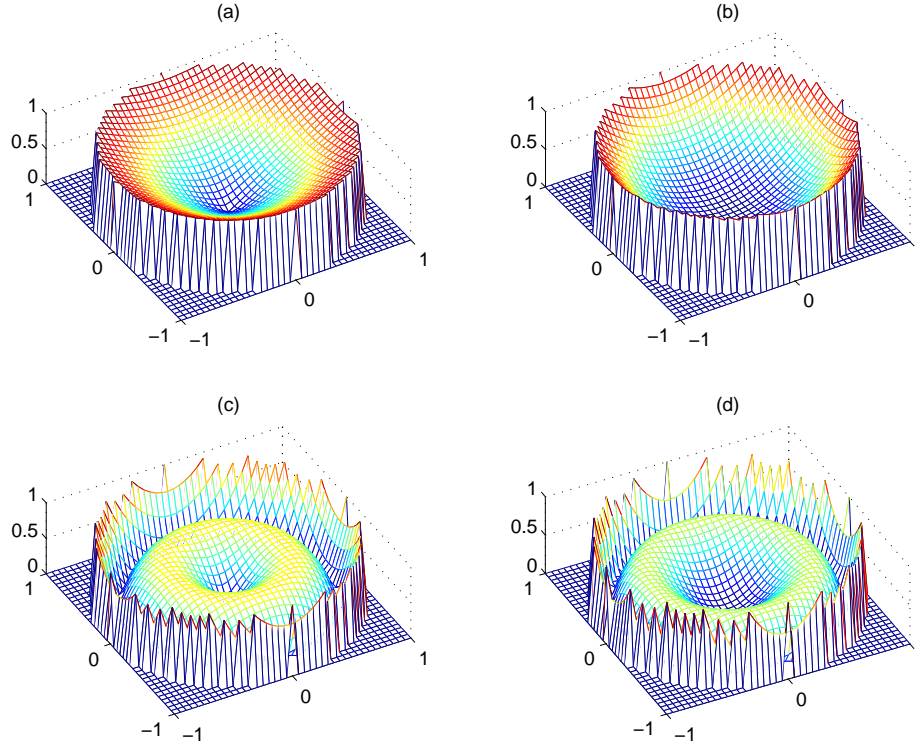
$$\begin{aligned} R_{n,-m}(r) &= R_{n,m}(r) \\ \int_0^1 R_{nm}(r) R_{pq}(r) r dr &= \frac{1}{2(n+1)} \delta_{np} \end{aligned}$$

La fonction génératrice du polynôme  $R_{nm}(r)$  est l'équation suivante :

$$\frac{[1 + t - \sqrt{1 - 2t(1 - 2r^2) + t^2}]^m}{(2tr)^m \sqrt{1 - 2t(1 - 2r^2) + t^2}} = \sum_{s=0}^{\infty} t^s R_{m+2s,m}(r)$$

Pour  $m = 0$ , il est intéressant de voir que l'équation précédente transforme en :

$$\frac{1}{\sqrt{1 - 2t(1 - 2r^2) + t^2}} = \sum_{s=0}^{\infty} t^s P_s(1 - 2r^2)$$


 FIG. 3.4 – Polynôme de Zernike (a) $|V_{11}|$  (b) $|V_{22}|$  (c) $|V_{31}|$  (d) $|V_{42}|$ 

et il devient la fonction génératrice des polynômes de Legendre d'argument  $2r^2 - 1$  :

$$R_{2n,0}(r) = P_n(2r^2 - 1)$$

Comme les polynômes de Legendre, les polynômes de Zernike sont complets et orthogonaux :

$$\int \int_{x^2+y^2 \leq 1} [V_{nm}(r, \theta)]^* V_{pq}(r, \theta) r \, dr d\theta = \frac{\pi}{n+1} \delta_{np} \delta_{mq}$$

### Remarque 2

1. Pour  $n$  fixé, il y a  $(n+1)(n+2)/2$  polynômes de Zernike  $V_{nm}$  indépendants tel que les conditions 3.4 soient vérifiées
2. D'après l'équation 3.3, on a  $A_{nm}^* = A_{n,-m}$  donc  $|A_{nm}| = |A_{n,-m}|$  et on a besoin simplement de calculer  $|A_{nm}|$  pour  $m \geq 0$
3. Si  $f^r(\rho, \theta)$  est une rotation d'angle  $\alpha$  de  $f(\rho, \theta)$ , ie.  
 $f^r(\rho, \theta) = f(\rho, \theta - \alpha)$ , alors on a  $A_{nm}^r = A_{nm} \exp(-jm\alpha)$

### 3.2.3.3 Les Pseudo Moments Zernike

Si on supprime la condition  $n - |m| = \text{pair}$  dans 3.4, et si on remplace les polynômes radiaux  $R_{nm}$  par :

$$R_{nm}(r) = \sum_{s=0}^{n-|m|} (-1)^s \frac{(2n+1-s)!}{s!(n-m-s)!(n+m-s)!} r^{n-s}$$

les polynômes  $\{V_{nm}\}$  deviennent l'ensemble des pseudo-polynômes de Zernike. Les pseudo-polynômes de Zernike ont les mêmes propriétés que les polynômes de Zernike.

### 3.3 Descripteur de Fourier

#### 3.3.1 Série de Fourier

Pour une fonction périodique  $f(x)$  de période  $T$  et sommable sur cette période, la série de Fourier de  $f(x)$  est définie par

$$f(x) \sim a_0 + \sum_{n=1}^{\infty} \left( a_n \cos \frac{2\pi nx}{T} + b_n \sin \frac{2\pi nx}{T} \right)$$

avec

$$\begin{aligned} a_0 &= \frac{1}{T} \int_0^{\infty} f(x) dx \\ a_n &= \frac{2}{T} \int_0^{\infty} f(x) \cos \frac{2\pi nx}{T} dx \\ b_n &= \frac{2}{T} \int_0^{\infty} f(x) \sin \frac{2\pi nx}{T} dx \quad n = 1, 2, \dots \end{aligned}$$

En utilisant la formule d'Euler  $e^{ix} = \cos x + i \sin x$ , on peut réécrire la série de Fourier sous forme complexe par :

$$f(x) \sim \sum_{n=-\infty}^{\infty} c_n e^{2\pi i n x / T}$$

avec

$$c_n = \frac{1}{T} \int_0^{\infty} f(x) e^{2\pi i n x / T} dx \quad n = 0, \pm 1, \pm 2, \dots$$

Il faut souligner que le contour fermé est naturellement périodique. Dans le cas général, on peut étendre le contour (ou l'image) en un signal périodique par divers moyens : symétrisation, extrapolation...

#### La convergence de la série de Fourier

Par définition, si  $f(x)$  est périodique et sommable, alors la série de Fourier existe, mais il ne garantit pas que la série de Fourier converge vers  $f(x)$ . Heureusement, due aux conditions de Dirichlet pour la série de Fourier, dont la plupart des fonctions périodiques et sommables ces conditions seront en général vérifiées. Les conditions de Dirichlet pour la série de Fourier disent que pour une fonction continue par morceaux  $f(x)$  qui a :

1. Un nombre fini de points de discontinuité et
2. Un nombre fini d'extrema (maximum ou minimum)

alors la série de Fourier converge vers  $f$  aux points où la fonction  $f$  est continue et vers la moyenne de la limite à gauche et limite à droite de  $f$  aux points de discontinuité, c'est à dire que :

$$\sum_{n=-\infty}^{\infty} c_n e^{2\pi i n x / T} = \begin{cases} f(x) & \text{si } f(x) \text{ est continue en } x \\ \frac{1}{2}[f(x^+) + f(x^-)] & \text{si } f(x) \text{ est discontinue en } x \end{cases}$$

Dans le cas de contours ou d'images, les conditions de Dirichlet sont vérifiées. Donc le contour ou l'image peuvent être représentés par la série de Fourier.

### 3.3.2 La transformée de Fourier continue

Si la fonction peut être représentée par la série de Fourier, donc on peut déterminer  $f$  de façon unique par les coefficients de Fourier,  $c_n$ . Inversement, si on connaît les coefficients de Fourier de  $f$ , on peut reconstruire  $f$  à partir de ses coefficients. La série de Fourier établit une relation bijective entre la fonction  $f$  et ses coefficients de Fourier. Cette relation est exprimée par la transformée de Fourier continue de  $f(x)$  :

$$F(u) = \int_{-\infty}^{\infty} f(x) \exp(-2\pi i u x) dx \quad (3.6)$$

leur correspondance, la transformée de Fourier inverse (dans le cas continue) est définie par :

$$f(x) = \int_{-\infty}^{\infty} F(u) \exp(2\pi i u x) du \quad (3.7)$$

### 3.3.3 La transformée de Fourier discrète

La transformée de Fourier discrète est particulièrement intéressante pour le traitement et l'analyse d'images car les images sont discrètes. Pour réécrire l'équation (3.6) et (3.7) sous forme discrète, on doit échantillonner la fonction  $f(x)$  en  $N$  échantillons sur l'intervalle  $[0, T]$  [GW93] :

$$f(x_0), f(x_0 + \Delta x), f(x_0 + 2\Delta x), \dots, f(x_0 + (N-1)\Delta x)$$

$\Delta x$  est un pas d'échantillonnage dans le domaine spatial. La transformée de Fourier discrète et son inverse de la fonction  $f(x)$  sont définies par :

$$F(u) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) \exp(-2\pi i u x / N) \quad u = 0, 1, 2, \dots, N-1 \quad (3.8)$$

$$f(x) = \sum_{u=0}^{N-1} F(u) \exp(2\pi i u x / N) \quad x = 0, 1, 2, \dots, N-1 \quad (3.9)$$

Le pas d'échantillonnage  $\Delta u$  dans le domaine fréquentielle et le pas d'échantillonnage  $\Delta x$  dans le domaine spatiale sont liés par la relation suivante :

$$\Delta u \geq \frac{1}{N\Delta x}$$

Cette relation est un principe d'incertitude, cela veut dire que la précision sur la position dans le domaine spatial et la précision sur la position dans le domaine fréquentiel sont inversement proportionnels. Autrement dit, si la résolution de l'image dans le domaine spatial est élevée ( $\Delta x$  petit), alors la résolution de l'image dans le domaine fréquentiel est faible ( $\Delta u$  grand). C'est important de noter que ces deux valeurs sont liées, elles ne peuvent pas varier de façon indépendante et on doit prendre cette relation en compte pour extraire les caractéristiques des images discrètes.

### 3.3.3.1 La transformée de Fourier discrète dans le cas 2D

Dans le cas de la fonction à deux variables  $f(x, y)$  qui est définie pour les valeurs  $0 \leq x, y \leq N$ , comme dans le cas 1D, par définition sa transformée de Fourier est :

$$F(u, v) = \frac{1}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \exp[-2\pi i(ux + vy)/N], \quad u, v = 0, 1, 2, \dots, N - 1 \quad (3.10)$$

même chose pour son inverse :

$$f(x, y) = \frac{1}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} F(u, v) \exp[2\pi i(ux + vy)/N], \quad x, y = 0, 1, 2, \dots, N - 1$$

Bien que le nombre de coefficients  $F(u, v)$  résultant de la définition de la transformée de Fourier 2D de l'équation (3.10) (ou  $F(u)$  dans l'équation 3.8) soit relativement élevé, le nombre de coefficients significatifs (dans le sens où leur amplitude ou magnitude est grande) est peu élevé. Cela parce que les coefficients de haute fréquence (coefficients moins significatifs) représentent seulement la partie **détails** de l'image, détails qui ne sont pas souvent utiles dans la plupart des applications et, en particulier, dans les systèmes de reconnaissance. Cela veut dire qu'on peut reconstruire l'approximation de  $f(x, y)$  (ou  $f(x)$ ) en utilisant très peu de coefficients significatifs  $F(u, v)$  (ou  $F(u)$ ). Cette idée a été exploitée par les systèmes de reconnaissance utilisant la transformée de Fourier.

### 3.3.3.2 Propriétés de la transformée de Fourier discrète

Par définition, une suite discrète d'échantillon  $(f_n)_{n \in \mathcal{Z}}$  est dite *paire* (resp. *impaire*) si on a  $f_{-n} = f_n$  (resp.  $f_{-n} = -f_n$ ) pour tout  $n \in \mathcal{Z}$ . On suppose que la suite des valeurs  $F_u$  représente la transformée de Fourier discrète de la suite d'échantillons  $f_n$ , en utilisant la définition de la parité et la définition de la transformée de Fourier discrète et son inverse, on peut montrer que [CP00][Zha02] :

- i)  $f_n$  est paire (resp. impaire)  $\Leftrightarrow F_u$  est aussi paire (resp. impaire)
- ii)  $f_n$  est réelle  $\Leftrightarrow \forall n \in \mathcal{Z} \quad F_{-u} = F_u$
- iii)  $f_n$  est réelle paire  $\Leftrightarrow F_u$  est réelle paire
- iv)  $f_n$  est réelle impaire  $\Leftrightarrow F_u$  est imaginaire pure et impair
- v) Translation :  $f_{n+k} \Leftrightarrow \exp(2\pi iku)F_u$   
 Cette relation signifie qu'une translation de  $k$  unités dans le domaine spatial entraîne une modification de la phase des coefficients de la transformée de Fourier de  $2\pi ku$  dans le domaine fréquentiel.
- vi) Changement d'échelle :  $f_{\alpha n} \Leftrightarrow F(u/\alpha)$ .  
 Cette relation exprime qu'un facteur de changement d'échelle  $\alpha$  dans le domaine temporel entraîne un changement d'échelle en proportion de  $1/\alpha$  dans le domaine fréquentiel. Concrètement cela veut dire que l'augmentation la taille d'image implique l'apparition des détails (haute fréquence) dans le domaine spectral (fréquentiel).
- vii) Dans le cas 2D : Les deux propriétés suivantes concernent uniquement le cas 2D ou multi-dimensionnel.

- a) **Séparabilité** : La transformée de Fourier discrète 2D de l'équation 3.10 peut se réécrire sous forme

$$F(u, v) = \sum_{x=0}^{N-1} \left( \exp[-2\pi i u x / N] \sum_{y=0}^{N-1} f(x, y) \exp[-2\pi i v y / N] \right).$$

Cela signifie qu'on peut séparer la transformée de Fourier 2D en deux transformées de Fourier 1D successives. L'avantage de cette propriété est qu'on peut l'exploiter en utilisant la *transformée de Fourier rapide*.

- b) **Rotation** : Si on utilise la coordonnée polaire

$$x = r \cos \theta, \quad y = r \sin \theta, \quad u = \omega \cos \varphi, \quad v = \omega \sin \varphi$$

En effectuant un changement de variable dans l'équation 3.10, on obtient la relation suivante :

$$f(r, \theta + \theta_0) \Leftrightarrow F(\omega, \varphi + \theta_0)$$

autrement dit, pour une rotation d'angle  $\theta$  de  $f(x, y)$  dans le domaine spatiale entraîne aussi une rotation d'angle  $\theta$  dans le domaine fréquentiel.

### 3.3.4 Le descripteur de Fourier par rapport à la transformation similitude

Nous avons vu dans la section 3.3.3, la transformée de Fourier discrète d'une signature  $u(t)$  est définie par :

$$a_n = \frac{1}{N} \sum_{t=0}^{N-1} u(t) \exp[-2\pi i n t / N], \quad n = 0, 1, \dots, N - 1 \quad (3.11)$$

Le résultat de cette transformation, les coefficients  $a_n$ , est une autre représentation de la signature. Comme on s'intéresse à la reconnaissance du contour par rapport à la transformation similitude (translation, rotation et changement d'échelle), il est important de pouvoir représenter les contours similaires, par le même ensemble de coefficients de Fourier. Cela signifie que la représentation doit être invariante par rapport à la transformation similitude.

Or on a vu les propriétés de la transformée de Fourier dans la section précédente. On peut montrer que la relation générale entre les coefficients de Fourier d'une signature transformée par une transformation similitude et les coefficients de Fourier de la signature d'origine est définie par [Zha02] [ZL03a] :

$$a_n = \exp[in\tau] \cdot \exp[i\varphi] \cdot s \cdot a_n^{(0)}, \quad n \neq 0 \quad (3.12)$$

où  $\exp[in\tau]$ ,  $\exp[i\varphi]$  et  $s$  sont des termes correspondent au changement du point de départ, à la rotation et au changement d'échelle respectivement et  $a_n$  (resp.  $a_n^{(0)}$ ) sont des coefficients de Fourier de la signature après (resp. avant) la transformation similitude. On remarque que la relation 3.12 n'est pas valable pour le composant DC ( $a_0$ ), le reste des coefficients n'est pas affecté par la translation. Maintenant si on définit les coefficients  $b_n$  par :

$$\begin{aligned} b_n &= \frac{a_n}{a_0} = \frac{\exp[in\tau] \cdot \exp[i\varphi] \cdot s \cdot a_n^{(0)}}{\exp[i\tau] \cdot \exp[i\varphi] \cdot s \cdot a_0^{(0)}} \\ &= \frac{a_n^{(0)}}{a_0^{(0)}} \exp[i(n-1)\tau] = b_n^{(0)} \exp[j(n-1)\tau] \end{aligned} \quad (3.13)$$

Ici, les coefficients  $b_n$  et  $b_n^{(0)}$  sont des coefficients de Fourier normalisés du contour transformé et du contour origine. L'équation 3.13 montre que le rapport entre le coefficient  $b_n$  du contour transformé et le coefficient  $b_n^{(0)}$  du contour origine est le nombre complexe  $\exp[j(n-1)\tau]$  de module un. Si on ignore la phase des coefficients  $b_n$  et si on prend uniquement en compte l'amplitude des  $b_n$ , on a égalité entre  $b_n$  et  $b_n^{(0)}$ . Autrement dit, l'amplitude des coefficients  $|b_n|$  est invariante par rapport à la transformation similitude. Donc, on peut utiliser l'amplitude des coefficients de Fourier normalisés  $\{|b_n|, 0 < n < N\}$  comme descripteur de contour par rapport à la transformation similitude. Il faut remarquer que la normalisation est effectuée en utilisant le composant DC ( $a_0$ ) car il représente la moyenne d'énergie du signal et donc les amplitudes  $|b_n|$  sont inférieures ou égales à un.

### 3.3.5 Le descripteur de Fourier par rapport à la transformation affine

Arbter et al. [ASBH90][ODA03] ont proposé un descripteur affine en utilisant la transformée de Fourier. Cet descripteur est obtenu à travers une analyse mathématique complexe.

Supposons que le contour est représenté par  $u(t) = (x(t), y(t))$ ,  $X_k$ ,  $Y_k$  sont des coefficients de Fourier de  $x(t)$ ,  $y(t)$  où  $t$  représente le paramètre qui transforme linéairement sous l'action de la transformation affine (section 2.3). Arbter et al. ont montré que les coefficients normalisés suivants sont invariants par rapport à la transformation affine :

$$Q_k = \frac{X_k Y_p^* - Y_k X_p^*}{X_p Y_p^* - Y_p X_p^*}$$

où  $p$  est une constante  $p \neq 0$ .

## 3.4 Descripteur de Fourier à fenêtre glissante

Le descripteur de Fourier contient à la fois les informations grossières (globales) et les informations les plus fines (locales). Cependant, pour les informations locales, le descripteur de Fourier contient seulement la grandeur (magnitude) et il ne contient pas des précisions sur leurs localisations (ou positions). Eichmann et al. [ea90] utilise la transformée de Fourier à fenêtre glissante. Contrairement à la transformée de Fourier, elle permet de localiser les caractéristiques locales du contour.

Dans la transformée de Fourier à fenêtre glissante, le signal (la signature du contour) est multiplié par une fenêtre (une fonction), qu'on appelle le *filtre d'analyse*, dont les valeurs sont nulles sauf pour la partie à laquelle on s'intéresse, c'est à dire :

$$a_{nm} = \frac{1}{T} \int_0^T u(t) g(t - nt_0) \exp(-j2\pi mt/T) dt$$

où  $t_0$  représente la taille d'étape de filtre. Cela est équivalent à projeter le signal  $u(t)$  sur une famille de fonctions :  $g(t - nt_0) \exp(-j2\pi mt/T)$ . Généralement, la fenêtre glissante  $g(t)$  est une *fonction de Gauss* ou une *fonction rectangulaire*. La différence entre la fonction de Gauss  $g(t)$  et la fonction rectangulaire  $rect(t)$  est que la fonction  $g(t)$  possède un support infini alors que  $rect(t)$  possède un support fini. Selon les résultats d'expérimentation de D. S. Zhang, la



fenêtre glissante rectangulaire est plus attractive pour extraire les caractéristiques du contour. La fonction rectangulaire est définie par :

$$rect(t) = \begin{cases} 1 & \text{si } -t_0/2 < t < t_0/2 \\ 0 & \text{sinon} \end{cases}$$

La normalisation de ces coefficients, pour chaque fenêtre, est la même que pour les coefficients de Fourier décrits dans la section 3.3.4. Après la normalisation, nous obtenons l'ensemble des caractéristiques  $SFD = \{SFD_{nm}, 0 \leq n \leq N-1, 0 \leq m \leq M-1\}$ , où  $N$  et  $M$  sont respectivement des résolutions spatiales et fréquentielles. Pour éliminer la dépendance de  $SFD$  par rapport au choix du point de départ, les méthodes présentées dans la section 2.4 peuvent être utilisées. De plus, comme le descripteur  $SFD$  n'est pas invariant par rapport à la rotation, pour mesurer la similarité entre le  $SFD$  de la requête  $Q$  et celui du modèle  $T$ , cela nécessite d'utiliser la distance suivante :

$$d = \min_{0 \leq k \leq N} \left\{ \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} |SFD_{nm}^Q - SFD_{(n+k),m}^T| \right\}$$

### 3.5 Descripteur d'espace-échelle de courbure (CSSD)

Avant de présenter les outils utilisés pour décrire le contour par le *descripteur de courbure d'espace-échelle* (*Curvature Scale-Space descriptor* ou simplement *CSSD*), il est nécessaire de faire rapidement le tour sur la théorie d'*espace-échelle* (*Scale-Space*) qui est le fondement de base du CSSD. En comparaison avec la transformée de Fourier, la théorie d'espace-échelle est relativement un outil nouveau pour la vision par ordinateur et depuis ces deux dernières décennies, cette théorie a gagné en popularité et s'impose de plus en plus dans divers domaines d'application. Dans cette section, nous faisons un petit résumé de la théorie Scale-Space et ensuite nous présenterons la méthode utilisée pour calculer le CSSD d'un contour.

#### 3.5.1 Fondement de base de la théorie d'espace-échelle

La philosophie derrière la modélisation d'espace-échelle est l'observation d'un objet à une échelle différente sachant que les objets n'ont un sens que dans un certain intervalle d'échelle. Un exemple simple de ce concept est l'observation d'une branche d'un arbre. Elle a un sens seulement, disons, de quelques centimètres à quelques mètres. Il n'y a pas de sens si l'on dit que la taille de l'arbre mesure quelques nanomètres ou quelques kilomètres [Lin96]. En parallèle avec le traitement d'images pour extraire des informations de l'image, on doit utiliser certains opérateurs pour traiter les informations dans l'image. Mais les questions fondamentales qu'on doit se poser concernant les opérateurs sont les suivantes : Quel type d'opérateur faut-il utiliser ? où doit-on appliquer l'opérateur ? Comment peut-on adapter l'opérateur par rapport à une tâche particulière ? Il est évident que le type d'information qu'on peut obtenir est très largement déterminé par la relation entre la taille de la structure des données (ou information) qui réside dans l'image et la taille (résolution) de l'opérateur.

Dans certains cas où l'on peut contrôler la situation, on a la connaissance à priori de l'échelle à choisir. Par exemple en physique, le physicien choisit l'échelle de façon correcte, pour modéliser une situation, souvent par intuition. Mais généralement, surtout en vision par ordinateur, on ne connaît pas a priori la situation, c'est à dire qu'on a aucune connaissance sur la scène à analyser.

Un argument principal pour construire l'espace échelle est que, si on a pas de connaissances au préalable à disposition sur l'échelle adéquate à utiliser pour un ensemble de données, le seul moyen raisonnable pour faire face à une telle situation est de représenter les données en plusieurs niveaux d'échelle [Lin96]. Autrement dit, dans un environnement libre, si aucune information n'est disponible sur la taille des données à traiter, la solution immédiate est de laisser évoluer la taille comme un paramètre.

Witkin [Wit83] a été le premier à introduire le terme espace-échelle et a lié et inter-connecter systématiquement la structure d'image sur plusieurs niveaux d'échelle. Il a suggéré qu'une dimension supplémentaire associée à l'échelle, doit être ajoutée pour représenter l'espace de sorte qu'on puisse étudier les structures dans l'espace sur plusieurs échelles différentes. En particulier, il a associé à un signal  $f(x) : \mathcal{R}^n \rightarrow \mathcal{R}$  une famille de signaux continus  $\{L(x, t) | t > 0\}$ , ce qu'on appelle l'*espace-échelle* de  $f(x)$ , qui sont des versions lissées du signal  $f(x)$ . Formellement, cela signifie que :

$$L(x, t) = g(x, t) * f(x) = \int_{u \in \mathcal{R}^n} g(u, t) f(x - u) du \quad (3.14)$$

avec la convention  $L(x, 0) = f(x)$  et la fonction  $g(x, t)$  est une fonction de lissage (ou filtre passe bas).

Linderberg, [Lin94], a montré que parmi les fonctions de lissage, le noyau Gaussien a des propriétés intéressantes pour générer l'espace-échelle, ie.

$$g(x, t) = \frac{1}{(2\pi t)^{n/2}} e^{-x^T x / 2t}$$

Dans le cas discret, la formule 3.14 s'écrit :

$$L(x, t) = \sum_{n=-\infty}^{\infty} \frac{1}{\sqrt{2\pi t}} e^{-n^2/2t} f(x - n)$$

### 3.5.2 Propriété de la non-crétion des nouvelles caractéristiques

La propriété la plus importante de l'espace-échelle est la propriété de la non-crétion des nouvelles caractéristiques. Cela signifie que la transformation de l'échelle la plus fine à l'échelle la plus grossière constitue une simplification du signal telle que les caractéristiques à l'échelle la plus fine disparaissent progressivement en augmentant la valeur d'échelle. L'augmentation de la valeur d'échelle rend le signal mono-dimensionnel de plus en plus lisse et il rend le signal à deux dimensions plus flou.

Les caractéristiques qui sont particulièrement intéressantes sont des points de passage par zéro (zero-crossing points) de la dérivée spatiale d'ordre- $n$ . En pratique, la dérivée d'ordre deux du signal est utilisée pour analyser les signaux ou les images, car la dérivée seconde représente la courbure du signal. La courbure a été reconnue comme une caractéristique très intéressante pour analyser un pattern. Les points de passage par zéro de la seconde dérivée sont des points d'inflexion qui sont des caractéristiques saillantes pour un pattern. Pour un signal mono-dimensionnel, si l'espace échelle est formé par un noyau Gaussien, les points de passage par zéro du signal sur tous les niveaux d'échelles forment ce qu'on appelle une *empreinte digitale* ou l'*arbre d'intervalle* (voir figure 3.5.2). A cause de la propriété de la non-crétion des nouvelles caractéristiques, la hauteur d'arbre d'intervalle est finie. Witkin [Wit83] a interprété cet arbre

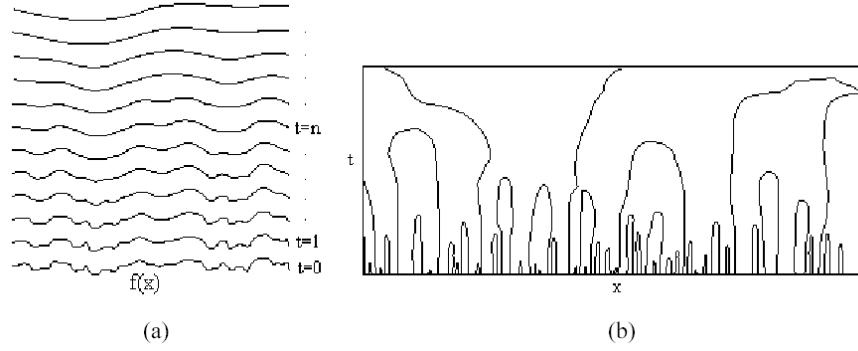


FIG. 3.5 – (a) Le signal origine  $f(x)$  en bas et ses versions lissées sont au-dessus, ici  $t$  est l'échelle de la fonction de lissage. (b) L'arbre d'intervalle obtenu des zero-crossings des deuxièmes dérivées des signaux lissés dans (a), chacun des points  $(x, t)$  dans l'arbre d'intervalle correspond à un point de zero-crossing à la position  $x$  et à l'échelle  $t$  du signal [Lin96]

d'intervalle avec une observation empirique selon laquelle les branches dans l'arbre d'intervalle correspondent aux caractéristiques perceptuelles dans le signal. Mokhtarian, [Mok95], a utilisé le sommet de l'arbre intervalle comme descripteur pour indexer les signaux. Nous allons présenter et étudier en détail ce qu'on appelle *Curvature Scale-Space Descripteur* ou simplement *CSSD* dans la section suivante.

### 3.5.3 CSSD Normal

Supposons que nous avons un contour  $\vec{\Gamma}(s) = (x(s), y(s))$ , où  $s$  est un paramètre qui représente la longueur d'arc sur le contour, que nous avons présenté dans la section 2.3.1. Nous supposons aussi que nous avons normalisé la longueur totale du contour à une constante fixe et également nous ré-échantillonons uniformément le contour en  $N$  points, où la valeur de  $N$  est fixé une fois pour toute à fin de faciliter l'appariement. La courbure en chaque point du contour  $\Gamma$  est, par définition, égale à la dérivée d'angle tangent au contour par rapport à la longueur d'arc :

$$k(s) = \lim_{h \rightarrow 0} \frac{\phi}{h},$$

où  $\phi$  est l'angle entre le vecteur  $\vec{t}(s)$  et le vecteur  $\vec{t}(s+h)$ , et  $\vec{t}(s)$  représente le vecteur tangent au point  $\Gamma(s)$  du contour. Pour calculer la courbure, on peut utiliser la définition équivalence 2.3 :

$$k(s) = \frac{\dot{x}(s)\ddot{y}(s) - \ddot{x}(s)\dot{y}(s)}{(\dot{x}(s)^2 + \dot{y}(s)^2)^{3/2}} \quad (3.15)$$

Si  $g(s, \sigma)$  est un noyau Gaussien mono-dimensionnel de largeur  $\sigma$  :

$$g(s, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp(-s^2/2\sigma^2)$$

on a alors  $X(s, \sigma)$  et  $Y(s, \sigma)$  sont des versions évoluées de  $x(s)$  et  $y(s)$  avec :

$$X(s, \sigma) = x(s) * g(s, \sigma), \quad Y(s, \sigma) = y(s) * g(s, \sigma)$$

où  $*$  représente la convolution. D'après la propriété de la convolution, la dérivée de toutes les composantes peut être calculée facilement par :

$$X_s(s, \sigma) = x(s) * g_s(s, \sigma), \quad X_{ss}(s, \sigma) = x(s) * g_{ss}(s, \sigma)$$

et nous avons les mêmes formules pour  $Y_s(s, \sigma)$  et  $Y_{ss}(s, \sigma)$ . L'avantage des deux dernières formules réside sur le fait qu'on connaît la formule explicite de la fonction  $g_s(s, \sigma)$  et  $g_{ss}(s, \sigma)$  et par conséquent on peut calculer facilement la courbure de la version évoluée du contour  $\Gamma$  par :

$$k(s, \sigma) = \frac{X_s(s, \sigma)Y_{ss}(s, \sigma) - Y_s(s, \sigma)X_{ss}(s, \sigma)}{(X_s(s, \sigma) + Y_s(s, \sigma))^{3/2}} \quad (3.16)$$

La fonction implicite définie par l'équation :

$$k(s, \sigma) = 0$$

est l'image du CSS du contour  $\Gamma$  [MM86][MM92][Mok95].

### 3.5.3.1 La construction d'image du CSS

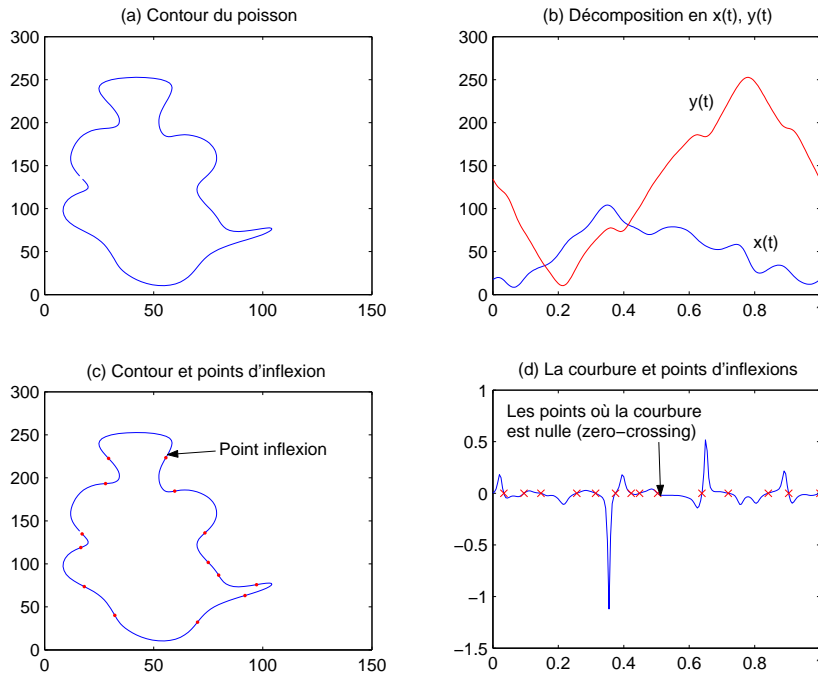


FIG. 3.6 – Les points d'inflexions détectés sur : 1) le contour d'un poisson (en bas à gauche) et 2) la courbure du poisson (en bas à droite)

Si on calcule les passages par zéros de la courbure de la version évoluée  $\vec{\Gamma}_\sigma(s, \sigma) = (X(s, \sigma), Y(s, \sigma))$ , c'est à dire que si on calcule l'ensemble des points <sup>6</sup>

$$\{(s, \sigma) \in [0, 1] \times \mathcal{R}_+^* \text{ tel que } k(s, \sigma) = 0\},$$

<sup>6</sup>On suppose que la longueur totale du contour est normalisée à 1

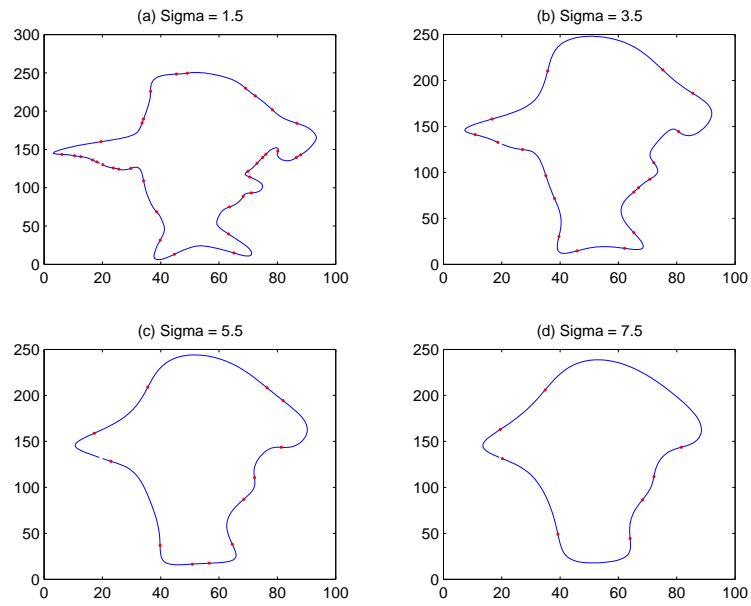


FIG. 3.7 – Les contours du poisson ont été filtrés avec le noyau Gaussien avec des valeurs  $\sigma \in \{1.5, 3.5, 5.5, 7, 5\}$ . Ensuite, les point d'inflexions ont été détectés après le filtrage

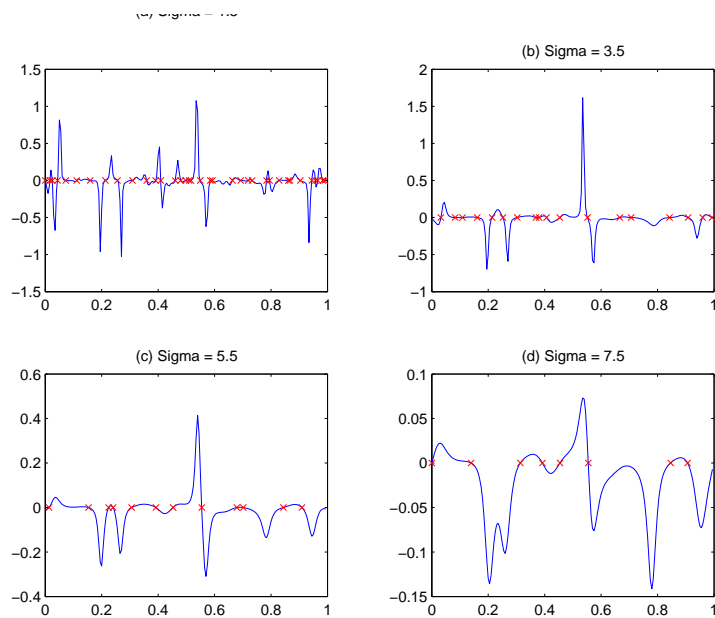


FIG. 3.8 – Le résultat de la détection des points inflexions à partir de la courbure des contours filtrés de la figure 3.7

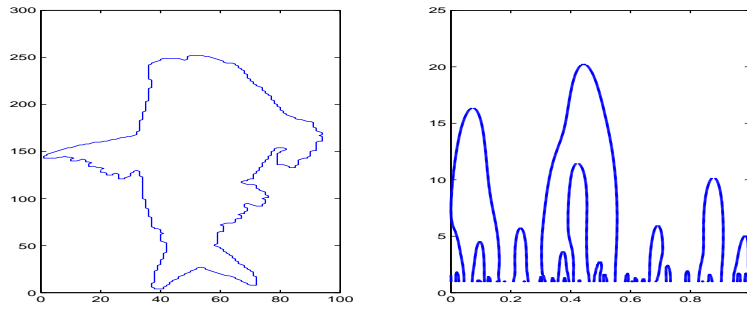


FIG. 3.9 – L'exemple de l'image du CSS

on peut visualiser ces points sur le plan  $(s, \sigma)$ .

Prenons un exemple pour donner une idée sur la construction de l'image du CSS d'un contour. La figure 3.5.3.1(a) représente le contour d'un poisson dont on veut construire l'image du CSS. Pour ce faire, nous récupérons d'abord les coordonnées  $(x(t), y(t))$  du contour (figure 3.5.3.1(b)) et nous re-paramétrisons et ré-échantillonons comme nous l'avons déjà expliqué dans la section 2.3.1 et section 3.5.3. Ensuite nous devons calculer la courbure en utilisant la formule 3.16 pour la valeur initiale de  $\sigma$  disons égale à 1 ( $\sigma = 1$ ). Cette courbure  $k(s, 1)$ , figure 3.5.3.1(d), nous permet de détecter et localiser de façon précise les points inflexions (figure 3.5.3.1(c)) de la version évoluée  $\Gamma_1$  de la figure 3.5.3.1(a). Ces points inflexions correspondent aux points d'intersection de la courbure  $k(s, 1)$  avec l'axe abscisse  $s$  (passage par zéro ou zero-crossing). Comme on peut constater dans la figure 3.7, pour chaque valeur de  $\sigma$ , nous avons le contour correspondant  $\Gamma_\sigma$  qui lui-même a un certain nombre de points inflexions. Encore une fois, comme dans le cas où  $\sigma = 1$ , ces points d'inflexions sont détectés et localisés en utilisant la courbure  $k(s, \sigma)$  (les courbures de la figure 3.7 sont présentées dans la figure 3.8). En augmentant la valeur de  $\sigma$ , le contour  $\Gamma_\sigma$  devient de plus en plus lisse et le nombre des points d'inflexions diminue. Quand la valeur de  $\sigma$  devient suffisamment grande, le contour  $\Gamma_\sigma$  devient convexe et dans ce cas le contour  $\Gamma_\sigma$  ne possède aucun point d'inflexions car la courbure  $k(s, \sigma)$  est strictement positive (ou strictement négative, ça dépend du sens de parcours) et donc on peut cesser le processus de construction de l'image du CSS. Enfin, pendant l'évolution, c'est à dire au fur et à mesure que la valeur de  $\sigma$  augmente, si on mémorise les couples de valeurs  $\{(s, \sigma) \text{ tel que } s \in [0, 1], \sigma \geq 1 \text{ et } k(s, \sigma) = 0\}$ , on peut les afficher sur le plan  $(s, \sigma)$ , figure 3.9, et l'ensemble de ces points correspondent exactement à l'image du CSS.

### Remarque 3

Il faut souligner que :

1. le corps de l'image du CSS est continu
2. les sommets du CSS sont généralement fermés
3. pour le contour fermé, le nombre des points inflexions est pair
4. les points d'inflexions sont invariants par rapport à une transformation affine. De plus si  $\Gamma_a$  est une transformation affine du contour  $\Gamma$ , alors il y a bijection entre l'ensemble des points d'inflexions de  $\Gamma_a$  et l'ensemble des points d'inflexions  $\Gamma$  [MA02]
5. les points d'inflexions sont regroupés deux à deux et correspondent aux différentes portions

---

**Algorithm 1** : Extraction des maximums du CSS
 

---

**Require:** L'image du CSS**Ensure:** Normal CSSD

```

1: %% pas est une constante utilisée pour incrémenter
2: %% la variable  $\sigma$  lors de la construction de l'image du CSS
3: for  $i = \sigma_{max}$  à 1 do
4:   if un zero-crossing est trouvé au point  $(i, s_j)$  then
5:     vérifiez les points voisins ci-dessus  $(i + pas, s_{j-1}), (i + pas, s_j)$  et  $(i + pas, s_{j+1})$ 
6:     if les trois voisins ci-dessus ne sont pas des zero-crossings then
7:        $(i, s_j)$  est un candidat aux maxima du CSS
8:       trouvez tous les autres candidats aux maximums tel que  $\sigma = i$ 
9:     end if
10:    Pour chaque candidat  $(i, s_j)$ , vérifiez les candidats voisins
11:    if  $(i, s_k)$  est un candidat voisin then
12:      if  $|j-k| > 5$  then
13:         $(i, s_j)$  est un maximum du CSS
14:      else
15:         $(i, (j + k)/2)$  est un maximum du CSS
16:      end if
17:    end if
18:  end if
19:   $i \leftarrow i - pas$ 
20: end for

```

---

convexes ou concaves du contour  $\Gamma_\sigma$  et plus la convexité<sup>7</sup> est profonde (resp. large), plus la hauteur (resp. largeur) de l'image du CSS correspondant est élevée (resp. large).

### 3.5.3.2 Extraction des maxima du CSS

L'idée cachée derrière l'utilisation des maximums du CSS est d'obtenir l'appariement au niveau brut (coarse-level match) en utilisant la caractéristique locale du CSS. C'est pour cette raison que l'algorithme d'appariement est très efficace et fiable. Ce choix est justifié par le fait que les maxima du CSS sont les plus significatifs parmi les points sur le corps de l'image du CSS, ils contiennent les informations à la fois sur l'échelle mais aussi sur la localisation du contour (du CSS ou de l'image). De plus, les maxima sont des points isolés très faciles à détecter. Un autre avantage et il est particulièrement intéressant, est que les maxima du CSS pour chaque modèle<sup>8</sup> ne sont pas nombreux et donc il est relativement simple pour résoudre le problème de la mise en correspondance.

Intuitivement, pendant le processus de construction de l'image du CSS, il est impossible de savoir si les points inflexions correspondent ou pas aux maxima du CSS. Pour extraire les maxima du CSS, nous utilisons l'algorithme 1.

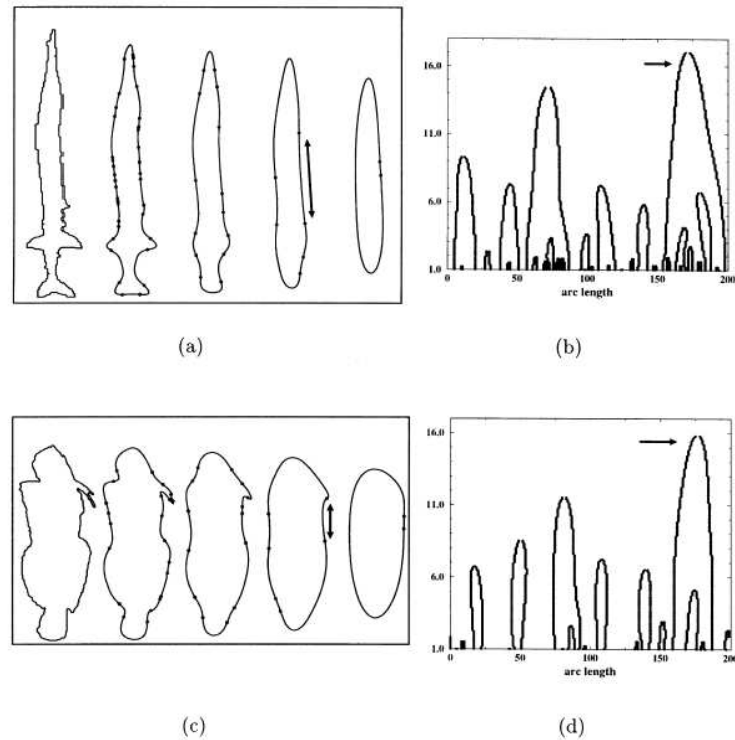


FIG. 3.10 – (a)Un contour avec la concavité peu profonde et (b)son CSS image. (c)Le contour avec la courbure très forte et (d) son CSS image.

### 3.5.4 CSSD modifié

Le CSSD présenté dans la section 3.5.3 pose de sérieux problèmes pour les contours qui ont des concavités peu profondes. Pour cela prenons un exemple de la figure 3.10[SAK00]. Comme le montre la figure 3.10(a), la partie du contour marquée par une flèche a une concavité peu profonde, le maximum CSS correspondant est également marqué par une flèche dans la figure 3.10(b). Ce maximum a la même hauteur que le maximum de la figure 3.10(d) qui correspond à une portion de contour avec une courbure très forte marquée dans la figure 3.10(c). Par conséquent, bien que les deux contours de la figure 3.10(a) et 3.10(c) soient très différents, les images du CSS des deux contours sont relativement proches et donc lors de l'appariement de ces deux contours, il n'est pas surprenant que la figure 3.10 fasse partie de l'ensemble des contours les plus similaires à la requête de la figure 3.10(a), ce qui est complètement faux. L'explication de ce phénomène est la suivante. Quand la valeur de  $\sigma$  augmente, à cause de la faible courbure, les deux points zero-crossing convergent vers le même point de façon très lente. Par contre dans la partie où il y a la forte courbure, la convergence est très rapide, mais pour convertir cette partie en un segment, chaque point de cette portion du contour doit subir un mouvement très fort. Par conséquent, le maximum CSS des deux portions du contour aura la même hauteur.

Pour résoudre ce problème, Abbasi et al. [SAK00] ont proposé trois solutions. Dans la suite nous adoptons la première solution, la méthode d'ajustement de la hauteur des maxima, comme

<sup>7</sup>ou concavité

<sup>8</sup>Pour simplifier, l'image de la requête est appelée *image* et l'image de la base s'appelle *modèle*



la version *modifiée de CSSD*. Le critère d'ajustement est basé sur la détection de la courbure maximum sur partie de la courbe sur chaque niveau d'échelle, si le maximum courbure à l'échelle  $\sigma_0$  est inférieur à un seuil autorisé, la hauteur du maximum est réduite à la valeur  $\sigma_0$  (voir [SAK00] pour plus de détail).

### 3.5.5 Algorithme d'appariement

---

**Algorithm 2** : Match

---

**Require:**  $CSSD_A$  et  $CSSD_B$

**Ensure:**  $Dist$

- 1: On regroupe les sommets de chaque descripteur en deux groupes, ceux qui ont leur correspondant ( $E$ ) et ceux qui n'ont pas leur correspondant ( $F$ ). Un sommet  $p_i^1$  du descripteur  $CSSD_A$  a un sommet correspondant  $p_i^2$  dans le descripteur  $CSSD_B$  si la distance des deux sommets  $d_i$  par rapport à  $s$ -coordonnée est inférieur à un seuil de tolérance.

2:

$$Dist \leftarrow \sum_{i \in E} (|hauteur(p_i^1) - hauteur(p_i^2)| + d_i) + \sum_{j \in F} hauteur(p_j)$$


---

Après avoir effectué l'extraction des maxima des contours du CSS (ou les maxima du CSS) de tous les modèles et de l'image, suivi éventuellement par l'ajustement de la hauteur des sommets (ou des maxima) du CSS dans le cas de CSSD modifié, les sommets du CSS de chaque modèle sont stockés et *triés dans l'ordre décroissant* par rapport à la valeur de  $\sigma$  (on l'appelle coordonnée verticale ou simplement le  $\sigma$ -coordonnée). Après cette opération, nous obtenons ce qu'on appelle le CSSD. Il faut souligner que le CSSD est invariant par rapport à une translation. Pour que le CSSD soit invariant par rapport au changement d'échelle, il suffit de fixer la longueur totale du contour à une constance. La rotation du contour entraîne un décalage circulaire du CSSD, donc pour que le CSSD soit invariant par rapport à la rotation, il suffit d'effectuer un décalage circulaire de sorte que le sommet le plus haut de l'image du CSS se trouve sur l'axe  $\sigma$ . Donc, pour calculer le degré de similarité entre deux contour  $A$  et  $B$ , il suffit de calculer la somme des différences entre la hauteur des sommets qui se chevauchent et la hauteur des sommets isolés (si on superpose l'image du CSS de  $A$  et de  $B$ ) [MM86][Mok95]. De plus, pour renforcer la robustesse, quatre plans de décalage circulaire ont été appliqués pour tolérer la variation des positions en  $s$ -coordonnée des maxima du CSS. Les quatre plans sont :

1. Décaler le sommet principal de  $A$ <sup>9</sup> (les autres sommets de  $A$  sont également décalés) de sorte que sa position en  $s$ -coordonnée coïncide avec la position en  $s$ -coordonnée du sommet principal de  $B$ .
2. Décaler le sommet principal de  $A$  de sorte que sa position en  $s$ -coordonnée coïncide avec la position en  $s$ -coordonnée du sommet secondaire de  $B$ .
3. Décaler le sommet secondaire de  $A$  de sorte que sa position en  $s$ -coordonnée coïncide avec la position en  $s$ -coordonnée du sommet principal de  $B$ .
4. Décaler le sommet secondaire de  $A$  de sorte que sa position en  $s$ -coordonnée coïncide avec la position en  $s$ -coordonnée du sommet secondaire de  $B$ .

Par ailleurs, comme le contour en miroir est aussi similaire au contour de départ, il est important que l'algorithme d'appariement prenne en compte cet aspect. Donc en tout, huit plans

---

<sup>9</sup>Le sommet principal est le sommet (ou le maximum) du CSS dont la hauteur est la plus grande

---

**Algorithm 3** : Appariement de deux descripteurs CSSD

---

**Require:**  $CSSD1$  et  $CSSD2$

**Ensure:**  $Dist(CSSD1, CSSD2)$

- 1: Décaler le sommet principal des deux descripteurs  $CSSD1$  et  $CSSD2$  de sorte que ses coordonnées sur l'axe abscisse (l'axe  $s$ ) aient la même valeur. Après cette opération, on obtient deux nouveaux descripteurs  $CSSD_A$  et  $CSSD_B$ .

$$Dist_{pp} \leftarrow Match(CSSD_A, CSSD_B)$$

- 2: Décaler le sommet principal du descripteur  $CSSD1$  et le sommet secondaire du descripteur  $CSSD2$  de sorte que ses coordonnées sur l'axe abscisse (l'axe  $s$ ) aient la même valeur. Après cette opération, on obtient deux nouveaux descripteurs  $CSSD_A$  et  $CSSD_B$ .

$$Dist_{ps} \leftarrow Match(CSSD_A, CSSD_B)$$

- 3: Décaler le sommet secondaire du descripteur  $CSSD1$  et le sommet principal du descripteur  $CSSD2$  de sorte que ses coordonnées sur l'axe abscisse (l'axe  $s$ ) aient la même valeur. Après cette opération, on obtient deux nouveaux descripteurs  $CSSD_A$  et  $CSSD_B$ .

$$Dist_{sp} \leftarrow Match(CSSD_A, CSSD_B)$$

- 4: Décaler le sommet secondaire du descripteur  $CSSD1$  et le sommet secondaire du descripteur  $CSSD2$  de sorte que ses coordonnées sur l'axe abscisse (l'axe  $s$ ) aient la même valeur. Après cette opération, on obtient deux nouveaux descripteurs  $CSSD_A$  et  $CSSD_B$ .

$$Dist_{ss} \leftarrow Match(CSSD_A, CSSD_B)$$

- 5: Apparier le descripteur  $CSSD1$  et le miroir du descripteur  $CSSD2$  en utilisant l'étape 1-4. Le résultat de cet appariement donne la distance  $Dist_{miroir}$

- 6:  $Dist(CSSD1, CSSD2) \leftarrow \min\{Dist_{pp}, Dist_{ps}, Dist_{sp}, Dist_{ss}, Dist_{miroir}\}$
-

de décalage circulaire sont nécessaires pour accomplir complètement l'appariement entre deux CSSD. De plus, comme les sommets du CSSD de deux contours similaires ne se trouvent pas exactement à la même position en  $s$ -coordonnée, il est indispensable que l'algorithme d'appariement accepte une certaine tolérance au niveau de la variation de position en  $s$ -coordonnée. Toutes ces considérations nous amènent à écrire la méthode d'appariement présentée dans l'algorithme 3.

## 3.6 Conclusion

Nous avons présenté, dans ce chapitre, plusieurs descripteurs de formes les plus populaires et les plus utilisés dans le domaine de la reconnaissance des formes. Ces descripteurs sont : le descripteur de moment, descripteur de Fourier, descripteur de Fourier à fenêtre glissante et descripteur de courbure multi-échelle. A ce jour, bien qu'il y ait déjà un certain nombre de travaux qui sont effectués afin de comparer les performances de ces descripteurs, les résultats de ces comparaisons ne sont pas très convaincants pour diverses raisons : la base de données utilisée pour l'évaluation n'est pas classée ou bien ne contient pas assez d'objets, la comparaison n'est pas impartiale etc.

L'étude comparative des trois derniers descripteurs avec le descripteur d'ondelette (chapitre 4), que nous espérons plus précise, fera l'objet de nos expérimentations détaillées dans le chapitre 5.



## Chapitre 4

# Transformation en ondelettes

### 4.1 Introduction

Les ondelettes sont un outil récent de traitement du signal ou d'image permettant l'analyse, à plusieurs échelles de temps, des propriétés locales de signaux complexes pouvant présenter des zones non-stationnarités. Elles donnent lieu à de nombreuses applications dans des domaines variés, comme par exemple la géophysique, l'astrophysique, les télécommunications, l'imagerie et le codage vidéo. Elles sont à la base de nouvelles techniques d'analyse et de synthèse d'images et trouvent de belles applications pour des problèmes généraux tels que la compression et le débruitage. Contrairement à l'analyse de Fourier, l'analyse en ondelettes permet une liberté supplémentaire puisque le choix des atomes de la transformée, déduits de l'ondelette analysante, est laissé à l'utilisateur. En outre, en fonction des objectifs du traitement par ondelettes, on peut préférer la transformée continue à la transformée discrète si la redondance peut être mise à profit pour analyser le signal. On fait le choix contraire si une compression du signal est souhaitée. Dans ce second cas, on doit se restreindre aux ondelettes à filtres alors que, dans le premier cas, presque n'importe quelle fonction d'intégrale nulle convient.

Ce chapitre a pour objectif de donner un bref rappel sur la théorie des ondelettes en commençant par la transformation continue en ondelette dans la section 4.2, puis dans la section 4.3 on explique le passage de la transformation continue vers la transformation en ondelette discrète en passant par l'analyse multi-résolution. Les trois sections suivantes sont consacrées à l'aspect pratique au niveau du calcul efficace des coefficients d'ondelettes en exploitant d'une manière intelligente les propriétés d'analyse multirésolution. La complexité de l'algorithme rapide et discret de la transformation en ondelettes est présenté dans la section 4.7. Dans la section 4.8, on reste toujours dans le cadre de la transformation discrète en ondelettes, mais ce fois-ci, on s'intéresse à la propriété d'invariance en translation de cette transformation. Dans la section 4.9, on essaye de dégager un certain nombre de propriétés des coefficients d'ondelettes par rapport à la transformation similitude. Ce chapitre contient également une part importante de contributions de cette thèse, la section 4.10 et la section 4.11 présentent respectivement les descripteurs d'ondelettes invariants à la transformation similitude et à la transformation affine. Enfin, la section 4.13 donne la conclusion et termine le chapitre.

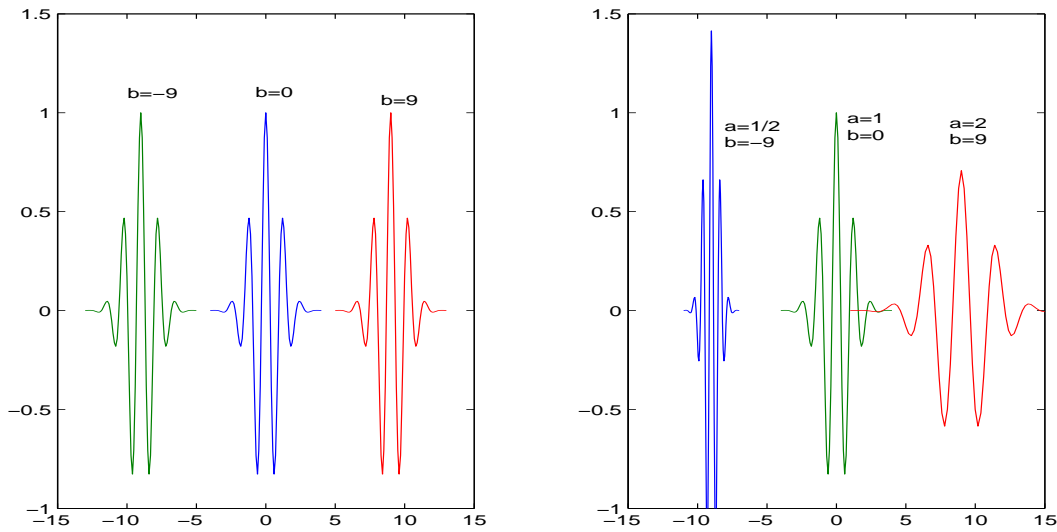


FIG. 4.1 – Translation et translation-dilatation d’une ondelette

## 4.2 La transformée continue en ondelettes

On appelle *ondelette* (ou *ondelette mère*) une fonction  $\psi \in \mathcal{L}^1 \cap \mathcal{L}^2$ , admettant  $n+1$  moments nuls (où  $n \in \mathbb{N}$ ), c’est à dire vérifiant :

$$\int_{\mathcal{R}} t^p \psi(t) dt = 0 \quad p = 0, \dots, n$$

La fonction est d’intégrale nulle. Elle est aussi orthogonale aux polynômes de degré inférieur ou égal à  $n$ . La fonction  $\psi$  oscille, prenant des valeurs positives et négatives. Le nombre  $n$  contrôle les *oscillations* de  $\psi$ , au sens où plus  $n$  est grand, plus  $\psi$  oscille.

Par *translation* et *dilatation* de l’ondelette  $\psi$ , on définit les *atomes* de la transformée en ondelettes. Pour toute *échelle*  $a \in \mathcal{R}^{+*}$  et toute *position*  $b \in \mathcal{R}$ , on définit un atome de la transformée par :

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}} \psi\left(\frac{t-b}{a}\right) \quad a \in \mathcal{R}^{+*}, b \in \mathcal{R}$$

La famille  $\{\psi_{a,b}\}_{a,b}$  est la *famille d’ondelettes associées* à  $\psi$ . En prenant  $\psi$  d’énergie 1 ( $\|\psi\|_{\mathcal{L}^2} = 1$ ), toutes les fonctions  $\psi_{a,b}$  sont alors de norme 1. A gauche de la figure (4.1), on voit trois ondelettes  $\psi_{1,-9}$ ,  $\psi_{1,0}$  et  $\psi_{1,9}$ , obtenues par translation de l’ondelette mère, et à droite,  $\psi_{0.5,-9}$ ,  $\psi_{1,0}$  et  $\psi_{2,9}$ , obtenues par translation-dilatation de  $\psi$ .

La *transformée continue en ondelettes* de la fonction  $f$  d’énergie finie est la famille des coefficients  $C_f(a,b)$  définis par :

$$C_f(a,b) = \int_{\mathcal{R}} f(t) \overline{\psi_{a,b}(t)} dt = (f, \psi_{a,b})_{\mathcal{L}^2}$$

La transformation admet une inverse, sous une condition supplémentaire dite d’*admissibilité* et la formule de synthèse est :

$$f(t) = \frac{1}{K_\psi} \int_{]0, \infty[ * \mathcal{R}} C_f(a,b) \psi_{a,b}(t) \frac{da db}{a^2}, \quad t \in \mathcal{R}$$

D'une certaine manière,  $C_f(a, b)$ , le coefficient de  $f$  sur l'ondelette  $\psi_{a,b}$ , caractérise les *fluctuations* de la fonction  $f$  au voisinage de la *position*  $b$ , à l'*échelle*  $a$ . Supposons que  $\psi$  soit nulle en dehors de  $[-M, M]$ , alors  $\psi_{a,b}$  est nulle en dehors de l'intervalle  $[-Ma + b, Ma + b]$ . La valeur de  $C_f(a, b)$  dépend alors des seules valeurs de  $f$  dans ce voisinage de  $b$  de longueur proportionnelle à  $a$ .

La transformé  $C_f$  associée à une fonction  $f$  d'une variable réelle  $t$ , une infinité de coefficients doublement indicés par  $a \in \mathcal{R}^{+*}$  et  $b \in \mathcal{R}$ . D'un certain point de vue, la transformation en fait trop : elle est *redondante*. Il est parfois souhaitable d'éviter cette redondance. On introduit la transformée discrète qui dans certains cas permet d'atteindre cet objectif.

### 4.3 Bases orthonormées d'ondelettes

#### 4.3.1 Le passage à la transformée discrète en ondelettes

Il est légitime de se demander s'il est nécessaire de connaître la transformé  $C_f$  sur  $\mathcal{R}^{+*} \times \mathcal{R}$  tout entier pour construire  $f$ . Lorsque la réponse est négative, l'utilisation d'un sous-ensemble discret semble un objectif raisonnable. L'idée est la suivante : on considère des sous ensembles discrets de  $\mathcal{R}^{+*} \times \mathcal{R}$ . Fixons  $a_0 > 1$ ,  $b_0 > 0$  et prenons  $a \in \{a_0^p\}_{p \in \mathcal{Z}}$  et  $b \in \{na_0^p b_0\}_{p, n \in \mathcal{Z}}$ . Par conséquent, au lieu d'utiliser la famille d'ondelettes  $\psi_{a,b}(t)$ , on se sert, pour la transformée discrète de la famille dénombrable d'ondelettes :

$$\psi_{p,n}(t) = a_0^{-p/2} \psi(a_0^{-p}t - nb_0) \quad a_0 > 1, b_0 > 0 \text{ fixés et } p, n \in \mathcal{Z}$$

Pour  $f \in \mathcal{L}^2$ , on définit la *transformée discrète en ondelettes* de la fonction  $f$  par :

$$C_f(p, n) = \int_{\mathcal{R}} f(t) \overline{\psi_{p,n}(t)} dt = (f, \psi_{p,n})_{\mathcal{L}^2} \quad p, n \in \mathcal{Z}$$

Dans les deux formules précédentes et dans la suite, nous changeons de notations pour alléger, dans le cas discret, l'écriture des atomes et des coefficients.

Le choix usuel  $a_0 = 2$  et  $b_0 = 1$  est dicté par le théorème de Shannon (voir [Mal00]). Il est alors naturel d'aborder une question plus difficile : existe-il, et dans quelles conditions, une fonction  $\psi$  telle que la famille  $\{\psi_{j,k}\}_{(j,k) \in \mathcal{Z}^2}$  où  $\psi_{j,k}(t) = 2^{-j/2} \psi(2^{-j}t - k)$  soit une base orthonormée de  $\mathcal{L}^2(\mathcal{R})$  ? La réponse est intimement liée à la notion d'analyse multirésolution.

#### 4.3.2 Analyses multirésolutions et bases orthonormées d'ondelettes

Une analyse multirésolution de  $\mathcal{L}^2(\mathcal{R})$  est une famille  $M = \{V_j\}_{j \in \mathcal{Z}}$  de sous-espaces vectoriels emboîtés qui possède les propriétés 4.1 à 4.5 ci-dessous, que l'on peut regrouper en trois blocs :

–  $\{V_j\}_{j \in \mathcal{Z}}$  est une suite d'espaces d'approximations, c'est à dire que

$$V_j \text{ est un sous-espace fermé de } \mathcal{L}^2 \tag{4.1}$$

$$V_j \subset V_{j-1} \tag{4.2}$$

$$\bigcup_{j \in \mathcal{Z}} V_j = \mathcal{L}^2 \text{ et } \bigcap_{j \in \mathcal{Z}} V_j = \{0\} \tag{4.3}$$

La propriété 4.1 assure l'existence de la projection orthogonale de  $f$  sur chacun des espaces  $V_j$ , projection qui approche  $f$ ; 4.2 traduit l'emboîtement des espaces et l'amélioration de l'approximation lorsque  $j$  décroît; 4.3 assure que la suite  $V_j$  converge vers  $\mathcal{L}^2$  tout entier et donc que la suite de projections converge vers  $f$ ;

- tous les espaces  $V_j$  sont obtenus par la *dilatation* ou la *contraction* dyadique des fonctions d'un espace unique (par exemple  $V_0$ ) :

$$\forall j \in \mathcal{Z}, \quad v(t) \in V_j \Leftrightarrow v(2t) \in V_{j-1} \quad (4.4)$$

Cette propriété caractérise les aspects multirésolutions de la suite  $M$  et joue un rôle crucial dans la construction des bases d'ondelettes;

- une dernière propriété concerne la *translation* de fonctions. Elle suppose l'existence d'une fonction qui, par translation entière, permet de construire une base de  $V_0$  :

$$\exists g \in V_0 \text{ telle que } \{g(t - k)\}_{k \in \mathcal{Z}} \text{ est une base de Riesz de } V_0 \quad (4.5)$$

A partir de la famille  $M$ , on définit une deuxième famille de sous-espaces notés  $\{W_j\}$ , où  $W_j$  est le supplémentaire orthogonal de  $V_j$  dans  $V_{j-1}$  :

$$V_{j-1} = V_j \oplus W_j \quad \text{avec } W_j \perp V_j$$

Par opposition aux espaces  $\{V_j\}$  qui sont les espaces d'*approximations*, on dira que les espaces  $\{W_j\}$  sont les espaces de *détails*.

On obtient une série de propriétés sur les sous-espaces  $\{W_j\}_{j \in \mathcal{Z}}$  qui sont utiles pour la compréhension géométrique de construction :

$$w(t) \in W_j \Leftrightarrow w(2t) \in W_{j-1} \quad (4.6)$$

$$W_j \perp W_k \quad j \neq k \quad (4.7)$$

$$W_j \perp V_k \quad j \leq k \quad (4.8)$$

$$V_J = V_K \oplus W_K \oplus \dots \oplus W_{J+1} \quad J < K \quad (4.9)$$

$$V_J = \bigoplus_{j=J+1}^{\infty} W_j \quad (4.10)$$

$$\mathcal{L}^2(\mathcal{R}) = V_J \oplus \{\bigoplus_{j=-\infty}^J W_j\} \quad (4.11)$$

$$\mathcal{L}^2(\mathcal{R}) = \bigoplus_{j=-\infty}^{\infty} W_j \quad (4.12)$$

Commentons quelques-unes de ces propriétés. Par exemple, 4.11 indique qu'un élément de  $\mathcal{L}^2$  peut s'écrire sous la forme d'une somme orthogonale d'une approximation grossière et d'une infinité de détails plus fins. La propriété 4.12, quant à elle, exprime le fait que toute fonction de  $\mathcal{L}^2$  est une somme infinie de détails orthogonaux. Notons  $A^j = P_{V_j}f$  et  $D^j = P_{W_j}f$ , les projections orthogonales de  $f \in \mathcal{L}^2$  sur les espaces  $V_j$  et  $W_j$  respectivement. On a alors :

$$A^{j-1} = A^j + D^j \quad \text{avec } A^j \perp D^j.$$

Les espaces  $V_j$  sont des espaces d'approximations dans le sens suivant :  $A^j$  converge vers  $f$  dans  $\mathcal{L}^2(\mathcal{R})$  lorsque  $j$  tend vers  $-\infty$ ; de même, les espaces  $W_j$  sont les espaces de détails au sens



où, dans  $\mathcal{L}^2$ , on a d'une part  $D^j$  qui converge vers 0 quand  $j$  tend vers  $-\infty$  et d'autre part

$$f = A^J + \sum_{-\infty}^J D^j$$

Autrement dit, pour un niveau d'approximation  $J$  fixé, les  $D^j$  sont les corrections à ajouter à l'approximation pour retrouver  $f$ .

Énonçons maintenant le résultat fondamental associé aux analyses multirésolution, en notant

$$f_{j,k}(t) = 2^{-j/2} f(2^{-j}t - k),$$

pour  $f$  une fonction quelconque.

### **Théorème 2 (Bases orthonormées d'ondelettes)**

Soit  $M$  une analyse multirésolution de  $\mathcal{L}^2(\mathcal{R})$ . A partir de  $g$  (cf. 4.5), on peut construire une fonction d'échelle  $\varphi$  puis une ondelette  $\psi$  telles que :

$$\forall J \in \mathcal{Z}, \{ \{ \varphi_{J,k} \}_{k \in \mathcal{Z}}, \{ \psi_{j,k} \}_{j,k \in \mathcal{Z}, j \leq J} \} \text{ est une base orthonormée de } \mathcal{L}^2$$

et  $\{ \psi_{j,k} \}_{j,k \in \mathcal{Z}}$  est une base orthonormée d'ondelettes de  $\mathcal{L}^2$ .

La démonstration de ce théorème repose sur le résultat des deux propositions suivantes. Ces deux propositions établissent les liens entre les notions d'analyse multirésolution et d'ondelette orthogonale et proposent une manière de construire la seconde à partir de la première. Cette construction montre, au passage, le rôle fondamental joué par les équations aux deux échelles dans le domaine temporel et fréquentiel. Commençons par la construction de la fonction échelle  $\varphi$ .

### **Proposition 1 (Construction de la fonction échelle)**

Considérons la fonction échelle  $\varphi$  définie à l'aide de sa transformée de Fourier  $\widehat{\varphi}$  par :

$$\widehat{\varphi}(\omega) = \frac{\widehat{g}(\omega)}{(\sum_{k \in \mathcal{Z}} |\widehat{g}(\omega + k)|^2)^{1/2}}$$

Alors :

1.  $\varphi \in V_0$
2.  $\{ \varphi_{0,k} = \varphi(t - k) \}_{k \in \mathcal{Z}}$  est une base orthonormée de  $V_0$
3. équations aux deux échelles pour  $\varphi$  :

$$\begin{aligned} \exists! a = \{ a_k \}_{k \in \mathcal{Z}}, a \in l^2(\mathcal{Z}) \text{ tel que :} \\ \frac{1}{2} \varphi\left(\frac{t}{2}\right) = \sum_{k \in \mathcal{Z}} a_k \varphi(t - k) \text{ dans } \mathcal{L}^2 \end{aligned}$$

4.  $m_0(\omega) = \sum_{k \in \mathcal{Z}} a_k e^{-2i\pi k \omega}$  est périodique de période 1,  $m_0 \in \mathcal{L}^2(0, 1)$ , et vérifie :

$$\begin{aligned} \widehat{\varphi}(2\omega) &= m_0(\omega) \widehat{\varphi}(\omega), \quad \omega \in \mathcal{R} \\ |m_0(\omega)|^2 + |m_0(\omega + \frac{1}{2})|^2 &= 1, \quad \omega \in \mathcal{R} \end{aligned}$$

5. plus généralement,  $\forall j \in \mathcal{Z}, \{\varphi_{j,k} = 2^{-\frac{j}{2}} \varphi(2^{-j}t - k)\}_{k \in \mathcal{Z}}$  est une base orthonormée de  $V_j$

**Proposition 2 (Construction de l'ondelette)**

L'ondelette  $\psi$  est définie à l'aide de sa transformée de Fourier  $\widehat{\psi}$ . Soit  $\rho$  une fonction périodique de période  $1/2$ ,  $|\rho(\omega)| = 1$ ,  $\omega \in \mathcal{R}$ , posons  $m_1(\omega) = \rho(\omega)e^{-2i\pi\omega} \overline{m_0(\omega + \frac{1}{2})}$  et définissons :

$$\widehat{\psi}(\omega) = m_1\left(\frac{\omega}{2}\right)\widehat{\varphi}\left(\frac{\omega}{2}\right)$$

Alors :

1.  $\psi \in W_0$
2.  $\{\psi_{0,k} = \psi(t - k)\}_{k \in \mathcal{Z}}$  est une base orthonormée de  $W_0$
3. équations aux deux échelles pour  $\psi$  :

$$\exists! b = \{b_k\}_{k \in \mathcal{Z}}, b \in l^2(\mathcal{Z}) \text{ tel que : } m_1(\omega) = \sum_{k \in \mathcal{Z}} b_k e^{-2i\pi k \omega}, \text{ et :}$$

$$\frac{1}{2}\psi\left(\frac{t}{2}\right) = \sum_{k \in \mathcal{Z}} b_k \varphi(t - k) \text{ dans } \mathcal{L}^2$$

4.  $m_1$  est périodique de période 1,  $m_1 \in \mathcal{L}^2(0,1)$ , et vérifie :

$$|m_1(\omega)|^2 + |m_1(\omega + \frac{1}{2})|^2 = 1, \omega \in \mathcal{R}$$

$$m_0(\omega) \overline{m_1(\omega)} + m_0(\omega + \frac{1}{2}) \overline{m_1(\omega + \frac{1}{2})} = 0, \omega \in \mathcal{R}$$

5. plus généralement,  $\forall j \in \mathcal{Z}, \{\psi_{j,k} = 2^{-\frac{j}{2}} \psi(2^{-j}t - k)\}_{k \in \mathcal{Z}}$  est une base orthonormée de  $W_j$

Stéphane Mallat [Mal89] a proposé l'algorithme rapide de décomposition-reconstruction pour la transformée discrète en ondelettes dès la fin des années 1980. Il a ainsi établi le lien entre les bases orthonormées d'ondelettes, dont la mise au point mathématique était alors récente, et les bancs de filtres classiques en traitement du signal. Ce point de vue unificateur a rapproché les deux communautés, permis un développement accru des applications vers le signal ou l'image et suscité l'intérêt théorique. Par exemple, l'approche fructueuse par les filtres a ainsi débouché sur la synthèse d'ondelettes ou sur la compression. En outre, deux traits inattendus sont à relever : l'algorithme est remarquablement simple et sa complexité est seulement linéaire dans la taille des données, c'est à dire plus faible que celle de la transformée de Fourier rapide. Cet aspect est évidemment crucial pour les applications.

Dans la section suivante, nous présentons l'algorithme de la transformée discrète en ondelettes des signaux échantillonnés (abrégée *DWT* comme *Discrete Wavelet Transform*). Il s'agit d'un cadre purement discret, au sens où on décompose non pas une fonction mais une suite finie, au moyen de filtres à réponse impulsionnelle finie. Dans le langage du traitement du signal, il s'agit de l'implémentation d'un banc de filtres à deux canaux et à reconstruction parfaite.

## 4.4 Des bases orthonormées à l'algorithme de Mallat

Faire une décomposition, c'est calculer les coordonnées du signal sur les vecteurs de la base.

Soit  $M = \{V_j\}_{j \in \mathcal{Z}}$  une analyse multirésolution de  $\mathcal{L}^2(\mathcal{R})$ , c'est à dire que  $\{V_j\}$  est une famille de sous-espaces vectoriels fermés emboîtés qui permettent l'approximation de fonctions et possèdent des propriétés faisant intervenir la dilatation et la translation de fonction.

Pour  $j \in \mathcal{Z}$  fixé, l'espace  $W_j$  est défini comme le supplémentaire orthogonal de l'espace  $V_j$  dans l'espace  $V_{j-1}$ . Pour une fonction  $f \in \mathcal{L}^2$ , la décomposition en ondelettes consiste à calculer les coordonnées des projections orthogonales de  $f$  sur  $V_j$  et  $W_j$  respectivement, notées  $A^j = P_{V_j} f$  et  $D^j = P_{W_j} f$ . Ces coordonnées dans les espaces  $V_j$  et  $W_j$  (qui sont munis des bases orthonormées :  $\{\varphi_{j,k}\}_{k \in \mathcal{Z}}$  et  $\{\psi_{j,k}\}_{k \in \mathcal{Z}}$  où  $\varphi$  et  $\psi$  désignent la fonction d'échelle et l'ondelette) ne sont rien d'autre que les coefficients d'approximation et de détail définis par les relations suivantes :

$$A^j = \sum_{p \in \mathcal{Z}} a_p^j \varphi_{j,p} \quad \text{avec} \quad a_p^j = (A^j, \varphi_{j,p})_{\mathcal{L}^2} = (f, \varphi_{j,p})_{\mathcal{L}^2}$$

$$D^j = \sum_{p \in \mathcal{Z}} d_p^j \psi_{j,p} \quad \text{avec} \quad d_p^j = (D^j, \psi_{j,p})_{\mathcal{L}^2} = (f, \psi_{j,p})_{\mathcal{L}^2}$$

Les calculs des coordonnées du signal sur les vecteurs de la base d'ondelettes s'effectuent comme d'habitude, par l'évaluation de produits scalaires. Le coût de l'approximation numérique des intégrales correspondantes est élevé.

Heureusement, on peut dans le contexte des bases d'ondelettes exploiter trois propriétés :

- d'une part, il n'y a qu'un nombre fini de coordonnées à calculer si le signal est à support compact ;
- d'autre part, la base est organisée par niveau d'échelle ;
- enfin, les fonctions de base d'une échelle sont liées à celles de la suivante par la relation aux deux échelles.

La conséquence est que si les coordonnées  $a^{j-1}$  de l'approximation à une échelle sont connues, les coordonnées  $a^j$  et  $d^j$  sur l'échelle suivante s'en déduisent presque immédiatement, au moyen d'une transformation très simple : une convolution, ou filtrage linéaire, suivi d'un sous-échantillonnage dyadique.

Avant de justifier l'algorithme esquissé ci-dessus, présentons le en deux phases : les filtres et le calcul efficace des coefficients.

## 4.5 Les quatre filtres

Pour une ondelette orthogonale  $\psi$ , la fonction d'échelle associée  $\varphi$  satisfait une relation fondamentale qui est l'équation aux deux échelles suivante :

$$\frac{1}{2} \varphi\left(\frac{t}{2}\right) = \sum_{n \in \mathcal{Z}} a_n \varphi_{0,n} = \sum_{n \in \mathcal{Z}} a_n \varphi(t - n)$$

Les filtres intervenant dans la transformée en ondelettes discrète et dans la transformée inverse (notée  $IDWT$ ) sont intimement liés à la suite  $(a_n)_{n \in \mathcal{Z}}$ . Si  $\varphi$  (et par conséquent  $\psi$ ) n'a qu'un nombre fini d'éléments non nuls, on peut alors voir cette suite comme un filtre passe-bas. Ce filtre, noté  $w$ , laisse donc passer les basses fréquences et retient les hautes. Il est à réponse impulsionnelle finie ( $FIR$ ), de longueur notée  $K$ , de somme 1 et de norme  $\frac{1}{\sqrt{2}}$ .

A partir du filtre  $w$ , on définit quatre filtres à réponse impulsionnelle finie, de taille  $K$  et de norme 1. On note les filtres de décomposition (repérés par le  $D$  final) :  $LoD$  et  $HiD$ . Le premier est un passe-bas (repéré par  $Lo$  au début) et le second un passe-haut (repéré par  $Hi$  au début). Les deux filtres de reconstruction (repérés par le  $R$  final) sont notés  $LoR$  et  $HiR$ .

Les deux filtres de reconstructions sont liés par :

$$LoR = \frac{w}{\|w\|} \text{ et } HiR_k = (-1)^{k-1} LoR_{K+1-k} \text{ pour } k = 1 \dots K$$

Ce sont des filtres miroirs en quadrature. Les deux filtres de décomposition sont obtenus par image miroir des filtres de reconstruction :

$$LoD = LoR_{K+1-k} \text{ et } HiD_k = HoR_{K+1-k} \text{ pour } k = 1 \dots K$$

## 4.6 Calcul efficace des coefficients

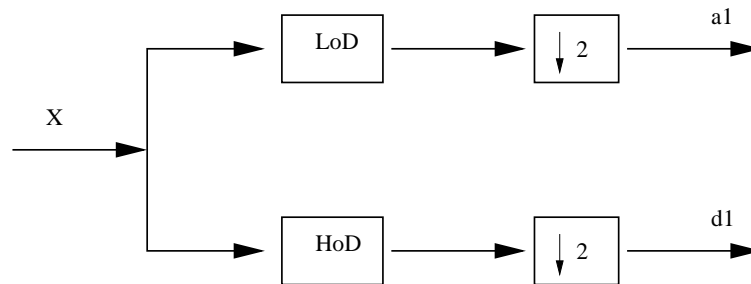


FIG. 4.2 – Premier pas de  $DWT$ , où  $\boxed{F}$  représente la convolution par  $F$  et  $\boxed{\downarrow 2}$  la décimation

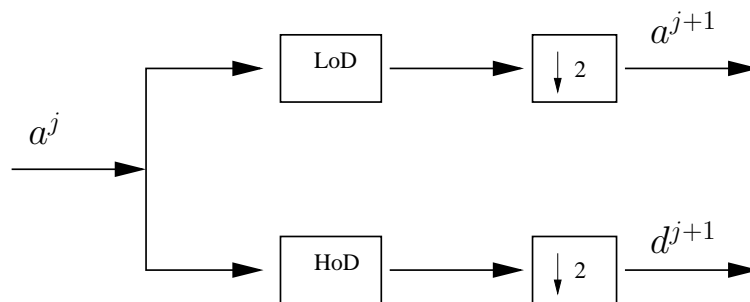


FIG. 4.3 – Représentation d'un pas de décomposition de la  $DWT$  (pour  $j = 0, a^j = X$ )

L'algorithme de la transformée discrète en ondelettes ( $DWT$ ) d'un signal  $X$  de longueur  $N$  consiste à effectuer plusieurs pas élémentaires de décomposition.

Partant du signal  $X$ , le premier pas produit deux vecteurs de coefficients : les coefficients d'approximation  $a^1$  et les coefficients de détail  $d^1$ . Ces vecteurs sont obtenus par une convolution du signal  $X$  avec le filtre passe-bas  $LoD$  pour l'approximation, et avec le filtre passe-haut  $HiD$  pour le détail, suivi dans les deux cas par une décimation dyadique<sup>10</sup>. Ils sont de longueur  $\frac{N}{2}$  environ.

Plus précisément, le premier pas de l'algorithme peut se représenter par la figure (4.2).

Les opérations mises en jeu conduisent de manière naturelle à envisager des décompositions d'ordre plus élevé. Le pas suivant consiste donc à décomposer les coefficients d'approximations  $a^1$  en deux, en remplaçant le signal  $X$  par  $a^1$  et en produisant  $a^2$  et  $d^2$ . L'algorithme se poursuit selon le même schéma. La décomposition en permettant le passage du niveau  $j$  au niveau  $j+1$  est illustrée par la figure (4.3). Il s'agit donc d'un algorithme très simple.

## 4.7 Complexité de l'algorithme

Supposons que les filtres  $LoD$  et  $HiD$  ont  $K$  coefficients non nuls et que le signal (assimilé à  $a_0$ ) est de longueur  $N$ . Alors les signaux  $a^j$  et  $d^j$  sont de longueur  $2^{-j}N$ , si l'on néglige les éventuels extra-coefficients suivant le schéma de calcul adopté. La phase de décomposition élémentaire permettant de passer des coefficients de l'approximation du niveau  $j-1$  aux coefficients de détails et d'approximations du niveau  $j$ , est donnée par :

$$\begin{cases} a^j &= dec(a^{j-1} * LoD) \\ d^j &= dec(a^{j-1} * HiD) \end{cases}$$

Chacune des deux lignes précédentes nécessite environ  $2^{-j+1}NK$  opérations. Cette phase de décomposition élémentaire requiert donc  $2^{-j+2}NK$  additions et multiplications. La décomposition au niveau  $J$  correspond à  $J$  décompositions élémentaires successives pour  $j=1$  à  $j=J$ . Elle requiert donc

$$\sum_{j=1}^J 2^{-j+2}NK = 2 \left( \sum_{j=0}^{J-1} 2^{-j} \right) NK \leq 4NK$$

additions et multiplications. Par conséquent, la complexité globale de la décomposition en ondelettes au niveau  $J$  d'un signal de longueur  $N$  par un banc de filtres comportant  $K$  coefficients non nuls est de l'ordre de  $4NK$ . En procédant semblablement, on montre que la phase de reconstruction requiert au plus  $4NK$  additions et multiplications. La transformée discrète en ondelettes a donc une complexité totale en  $O(N)$ , c'est à dire linéaire en la taille des données, avec une constante qui croît linéairement selon la longueur des filtres utilisés. C'est tout à fait remarquable puisque c'est une complexité plus faible que la transformée de Fourier rapide.

## 4.8 Transformée invariante par translation

La transformée discrète en ondelettes ( $DWT$ ) classique souffre d'un inconvénient connu, elle n'est pas une transformée invariante par translation dans le temps, contrairement à la transformation continue en ondelettes. Ceci signifie que, même dans le cas d'une extension périodique

<sup>10</sup>La décimation du signal  $X$  est définie par  $Y = dec(X)$  où  $Y_n = X_{2n}$

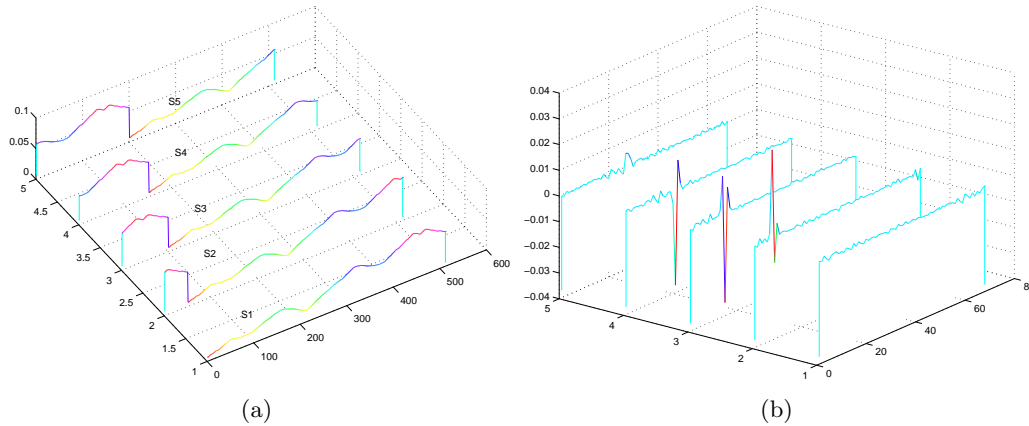


FIG. 4.4 – (a)  $S_1$  est le signal d'origine. Le signal  $S_i$  est un décalage circulaire de 50 points du signal  $S_{i-1}$ ,  $2 \leq i \leq 5$  (b) Les coefficients de  $DWT$  de  $S_i$ . On constate que les coefficients de  $DWT$  des 5 signaux sont différents.

d'un signal  $X$ , la  $DWT$  d'une version translatée de  $X$  n'est pas, en général, la version translatée de la  $DWT$  de  $X$ . Pour illustrer ce problème, nous avons pris 5 signaux  $(S_i)_{1 \leq i \leq 5}$  (voir figure 4.4(a)). Le signal  $(S_i)_{2 \leq i \leq 5}$  est un décalage circulaire (translation) de 50 points du signal  $(S_{i-1})_{2 \leq i \leq 5}$ . Nous avons appliqué  $DWT$  sur les signaux  $S_i$  sur trois niveaux en utilisant l'ondelette de Haar, les coefficients de détails de  $DWT$  des 5 signaux  $S_i$  sont montrés dans la figure 4.4(b). Sur cette figure on constate que les coefficients de  $DWT$  de chacun des signaux sont différents<sup>11</sup>. C'est la raison pour laquelle, la transformation  $DWT$  n'est pas souvent utilisée.

Pour restaurer l'invariance par translation, l'idée est d'effectuer la moyenne des  $DWT$  de tous les signaux translatés-périodisés issus de  $X$  (appelées  $DWT$   $\varepsilon$ -décimées), convenablement synchronisés. Ceci définit la transformée invariante par translation appelée  $SWT$  (en anglais *Stationary Wavelet Transform*).

Dans la suite, la  $SWT$  est définie pour des signaux dont la taille est divisible par  $2^n$ , où  $n$  est le niveau maximum de la décomposition.

#### 4.8.1 $DWT$ $\varepsilon$ -décimée

Il existe plusieurs manières, légèrement différentes, d'utiliser la  $DWT$ . Rappelons que la base de son calcul s'appuie sur une convolution suivie d'une décimation. Cette dernière ne conserve que les éléments d'indice pair. En fait, la décimation peut très bien s'effectuer en conservant les éléments d'indice impair au lieu des éléments d'indice pair et cette alternative se pose naturellement à chaque étape du processus de décomposition. Si le calcul systématique de toutes les décompositions possibles du signal d'origine est effectué, on obtient alors  $2^n$  décompositions au niveau  $n$ .

Pour repérer le choix fait à l'étape  $j$ , on définit la variable  $\varepsilon_j$ . On pose  $\varepsilon_j = 1$  (respectivement 0) si on décime, à l'étape  $j$ , les éléments d'indice impair (respectivement pair). Chaque décompo-

<sup>11</sup>Les coefficients d'approximations de  $DWT$  des 5 signaux sont aussi différents, mais nous ne les montrons pas sur la figure car l'impression ne montre pas nettement la différence

sition est ainsi identifiée par une séquence de 0 et 1 :  $\varepsilon = \varepsilon_1 \dots \varepsilon$  et appelée une *DWT*  $\varepsilon$ -décimée.

### 4.8.2 Calcul de la SWT

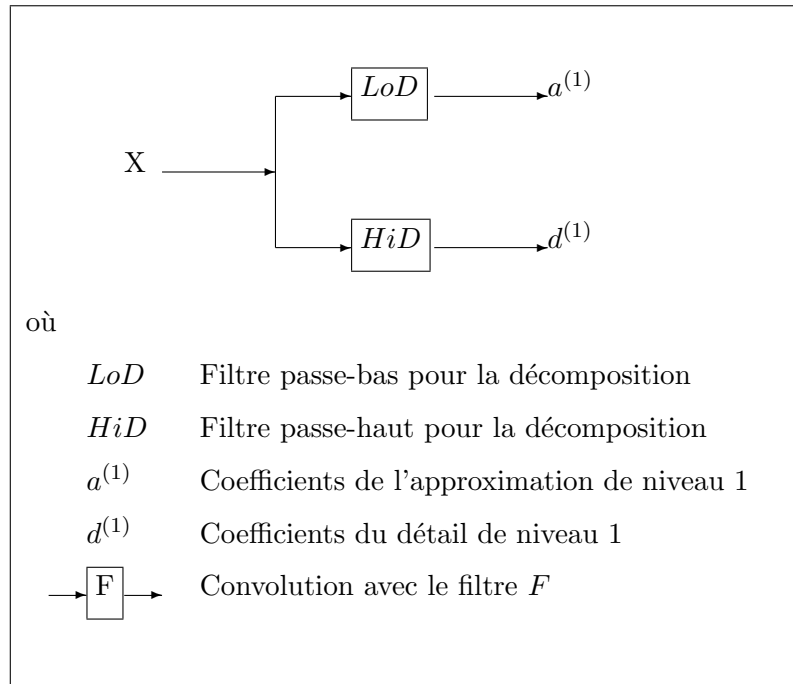


FIG. 4.5 – *SWT* 1D : décomposition au niveau 1

Il est possible d'obtenir toutes les *DWT*  $\varepsilon$ -décimées pour un signal donné de longueur  $N$ , en calculant les coefficients d'approximation et de détail pour chaque séquence  $\varepsilon$  possible. Cette opération pourrait s'effectuer en utilisant de manière itérative, une version légèrement modifiée de l'étape de base du calcul de la *DWT*. Il s'agit de la version classique pour laquelle  $\varepsilon = 0$ , mais pour  $\varepsilon = 1$  les éléments d'indice impair seront retenus lors de l'étape de décimation. Bien sûr, ce n'est pas la bonne manière de calculer la *DWT*  $\varepsilon$ -décimée car de nombreux calculs sont effectués plusieurs fois. Nous allons donc décrire une autre méthode : la transformée invariante par translation appelée *SWT*.

L'algorithme utilisé pour le calcul de la *SWT* est très semblable à celui de la *DWT*. Pour le niveau 1, toutes les *DWT*  $\varepsilon$ -décimées (deux seulement à ce niveau) pour un signal donné peuvent être obtenues en convoluant le signal avec les filtres appropriés, comme dans le cas de la *DWT* mais sans décimation. Dans ce cas, les coefficients de l'approximation et du détail au niveau 1, notés ici  $a^{(1)}$  et  $d^{(1)}$  (et non  $a^1$  et  $d^1$  comme pour la *DWT*) sont tous deux de taille  $N$ , la taille du signal d'origine. Cette opération peut être représentée par le schéma proposé par la figure (4.5).

Dans le cas général, les coefficients  $a^{(j)}$  de l'approximation de niveau  $j$  sont convolués avec une version sur-échantillonnée des deux filtres usuels, pour produire les coefficients  $a^{(j+1)}$  et  $d^{(j+1)}$  d'approximations et de détails au niveau  $j + 1$ . L'algorithme peut être représenté par le

schéma de la figure (4.6). Notons que les tailles de toutes les structures de coefficients obtenues par la décomposition (approximation et détails) sont toutes égales à la taille du signal d'origine  $X$ .

## 4.9 Propriété de la transformation en ondelettes

Supposons que les contours  $\Gamma = (x(t), y(t))$  sont paramétrés par la longueur d'arc  $t$ , en appliquant la transformation en ondelettes sur les contours, nous obtenons :

$$\begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} A^n x(t) \\ A^n y(t) \end{bmatrix} + \sum_{k=n_0}^n \begin{bmatrix} D^k x(t) \\ D^k y(t) \end{bmatrix} \quad (4.13)$$

avec

$$\begin{bmatrix} A^n x(t) \\ A^n y(t) \end{bmatrix} = \sum_{i \in \mathcal{Z}} \begin{bmatrix} a_i^n \varphi_{n,i}(t) \\ b_i^n \varphi_{n,i}(t) \end{bmatrix} \quad (4.14)$$

$$\begin{bmatrix} D^k x(t) \\ D^k y(t) \end{bmatrix} = \sum_{i \in \mathcal{Z}} \begin{bmatrix} c_i^k \psi_{k,i}(t) \\ d_i^k \psi_{k,i}(t) \end{bmatrix} \quad (4.15)$$

où  $A^n \square$  (resp.  $D^n \square$ ) représente l'approximation (resp. le détail) de  $\square$  à l'échelle  $n$ , avec  $n_0 \leq n \leq \log_2 N$  ( $N$  est le nombre d'échantillons du contour). L'échelle de décomposition  $k = n_0$  (resp.  $k = n$ ) représente l'échelle la plus fine (resp. l'échelle la plus grossière). Les coefficients  $a_i^n, b_i^n$  (resp.  $c_i^k, d_i^k$ ) sont des coefficients d'approximations (resp. coefficients de détails) de la décomposition en ondelettes. Ils sont définis par :

$$a_i^n = (x, \varphi_{n,i})_{\mathcal{L}^2}, \quad b_i^n = (y, \varphi_{n,i})_{\mathcal{L}^2} \quad (4.16)$$

$$c_i^k = (x, \psi_{k,i})_{\mathcal{L}^2}, \quad d_i^k = (y, \psi_{k,i})_{\mathcal{L}^2} \quad (4.17)$$

Si nous utilisons un sous ensemble de coefficients d'ondelettes, principalement les coefficients aux échelles grossières, nous obtenons une représentation multi-échelle (ou multi-résolution) du contour. D'une manière formelle, nous pouvons modifier l'équation 4.13 en équation :

$$\begin{bmatrix} \tilde{x}(n_1, t) \\ \tilde{y}(n_1, t) \end{bmatrix} = \begin{bmatrix} A^n x(t) \\ A^n y(t) \end{bmatrix} + \sum_{k=n_1}^n \begin{bmatrix} D^k x(t) \\ D^k y(t) \end{bmatrix} \quad (4.18)$$

avec  $n_0 \leq n_1 \leq n + 1$ . La suite des contours  $(\tilde{\Gamma}_{n_1} = (\tilde{x}(n_1, t), \tilde{y}(n_1, t)))_{n_0 \leq n_1 \leq n+1}$  représentent l'approximation multi-résolution du contour de départ  $\Gamma$ . La suite des contours d'approximations est caractérisé par deux résolutions (ou l'échelle) extrêmes, d'une part pour  $n_1 = n_0$  on obtient le contour de départ, d'autre part pour  $n_1 = n + 1$ , on obtient la représentation la plus grossière du contour. Pour donner un exemple, nous montrons le flocon de neige de Koch dans la figure 4.7 et les différentes approximations avec un sous ensemble des coefficients d'ondelette [CK96]. La figure 4.7(0) représente le contour de départ, échantillonné uniformément en 3072 points. Les figures 4.7(1-8) sont des approximations de la résolution la plus fine vers la résolution la plus grossière, ils correspondent à la décomposition en huit niveaux de décomposition.

Les détails de l'équation 4.15 (même chose pour l'approximation de l'équation 4.14) peuvent être représentés en utilisant les coordonnées polaires ie.

$$\begin{bmatrix} D^k x(t) \\ D^k y(t) \end{bmatrix} = \sum_{i \in \mathcal{Z}} \begin{bmatrix} c_i^k \psi_{k,i}(t) \\ d_i^k \psi_{k,i}(t) \end{bmatrix} = \sum_{i \in \mathcal{Z}} \begin{bmatrix} \cos \theta_i^k & -\sin \theta_i^k \\ \sin \theta_i^k & \cos \theta_i^k \end{bmatrix} \begin{bmatrix} r_i^k \\ 0 \end{bmatrix} \psi_{k,i}(t)$$



avec

$$\theta_i^k = \arctan\left(\frac{d_i^k}{c_i^k}\right) \quad \text{et} \quad r_i^k = \sqrt{(c_i^k)^2 + (d_i^k)^2}$$

Nous regardons maintenant l'effet de la transformation similitude sur l'ensemble des coefficients d'ondelettes.

### Changement d'échelle

Le changement d'échelle du contour par un facteur  $\beta$  peut être écrit comme :

$$\begin{bmatrix} \check{x}(t) \\ \check{y}(t) \end{bmatrix} = \beta \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \beta \begin{bmatrix} A^n x(t) \\ A^n y(t) \end{bmatrix} + \sum_{k=n_0}^n \beta \begin{bmatrix} D^k x(t) \\ D^k y(t) \end{bmatrix}$$

Donc en utilisant la linéarité de la transformation en ondelette, cela entraîne :

$$\begin{bmatrix} \check{a}_i^k \\ \check{b}_i^k \end{bmatrix} = \beta \begin{bmatrix} a_i^k \\ b_i^k \end{bmatrix} \quad \text{et} \quad \begin{bmatrix} \check{c}_i^k \\ \check{d}_i^k \end{bmatrix} = \beta \begin{bmatrix} c_i^k \\ d_i^k \end{bmatrix}$$

Si on utilise les coordonnées polaires, on a :

$$\check{\theta}_i^k = \arctan\left(\frac{\beta d_i^k}{\beta c_i^k}\right) = \arctan\left(\frac{d_i^k}{c_i^k}\right) = \theta_i^k \quad \check{r}_i^k = \sqrt{(\beta c_i^k)^2 + (\beta d_i^k)^2} = \beta r_i^k$$

Cela signifie que, l'effet du changement d'échelle du contour entraîne un changement d'amplitude des coefficients d'ondelettes, mais pas de la phase.

### Translation

Par translation, nous avons :  $(\check{x}(t), \check{y}(t))^t = (x(t) + \alpha, y(t) + \beta)^t$ . Pour alléger notre écriture, dans la suite nous établissons l'effet de la translation sur les coefficients d'ondelettes par rapport à  $x(t)$ . On obtient le même résultat en ce qui concerne l'effet de la translation par rapport à  $y(t)$  en effectuant le même raisonnement. Sachant que l'ondelette  $\psi$  est d'intégrale nulle (par définition), en supposant que la fonction échelle est d'intégrale égale à un, nous avons

$$\int_{\mathcal{R}} \psi(t) dt = 0, \quad \int_{\mathcal{R}} \phi(t) dt = 1$$

D'après l'équation ci-dessus, l'équation 4.16 et l'équation 4.17, nous avons :

$$\begin{aligned} \check{a}_i^n &= (x + \alpha, \varphi_{n,i})_{\mathcal{L}^2} \\ &= a_i^n + \alpha \int_{\mathcal{R}} \varphi_{n,i}(t) dt \\ &= a_i^n + \alpha \int_{\mathcal{R}} 2^{-n/2} \varphi(2^{-n}t - i) dt \\ &= a_i^n + \alpha \int_{\mathcal{R}} 2^{n/2} \varphi(t) dt \\ &= a_i^n + \alpha 2^{n/2} \end{aligned}$$

$$\begin{aligned}
 \check{c}_i^k &= (x + \alpha, \psi_{k,i})_{\mathcal{L}^2} \\
 &= c_i^n + \alpha \int_{\mathcal{R}} \psi_{k,i}(t) dt \\
 &= c_i^n + \alpha \int_{\mathcal{R}} 2^{-n/2} \psi(2^{-n}t - i) dt \\
 &= c_i^n + \alpha \int_{\mathcal{R}} 2^{n/2} \psi(t) dt \\
 &= c_i^n
 \end{aligned}$$

En résumé, la relation entre les coefficients d'ondelettes du contour avant et après la translation vérifie les équations suivantes :

$$\begin{bmatrix} \check{a}_i^k \\ \check{b}_i^k \end{bmatrix} = \begin{bmatrix} a_i^k + 2^{k/2}\alpha \\ b_i^k + 2^{k/2}\beta \end{bmatrix} \quad \text{et} \quad \begin{bmatrix} \check{c}_i^k \\ \check{d}_i^k \end{bmatrix} = \begin{bmatrix} c_i^k \\ d_i^k \end{bmatrix}$$

### Rotation

Si le contour est retourné, dans le sens inverse des aiguilles d'une montre, par rapport au centre de gravité d'angle  $\alpha$ , nous avons :

$$\begin{bmatrix} \check{a}_i^k \\ \check{b}_i^k \end{bmatrix} = \begin{bmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{bmatrix} \begin{bmatrix} a_i^k \\ b_i^k \end{bmatrix} \quad \text{et} \quad \begin{bmatrix} \check{c}_i^k \\ \check{d}_i^k \end{bmatrix} = \begin{bmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{bmatrix} \begin{bmatrix} c_i^k \\ d_i^k \end{bmatrix}$$

Si on veut exprimer la formule en utilisant les coordonnées polaires, nous avons :

$$\check{\theta}_i^k = \theta_i^k + \alpha \quad \text{et} \quad \check{r}_i^k = r_i^k$$

### Remarque 4

Il faut souligner que les propriétés ci-dessus sont valables dans les cas de la transformation en ondelettes discrètes avec décimation DWT (section 4.6) et sans décimation SWT (section 4.8.2).

## 4.10 Descripteur d'ondelettes invariant à la transformation similitude

D'après les propriétés des coefficients de la transformation en ondelettes qu'on vient de présenter, nous constatons qu'elles ne sont pas invariantes par rapport à la transformation similitude. Nous rappelons que le groupe de la *transformation similitude* est un groupe qui contient les transformations *translation*, *rotation* et *changement d'échelle*. Donc, nous devons normaliser ces coefficients pour qu'ils soient invariants si nous voulons les utiliser comme descripteurs. G. C.-H. Chuang et al. [CK96] ont proposé la procédure de normalisation décrite dans l'algorithme 4.

La dimension du vecteur descripteur obtenu par l'algorithme de normalisation 4 est relativement grande. De plus, si l'on regarde bien de très près, implicitement le vecteur descripteur obtenu ci-dessus n'est pas invariant par rapport à la transformation similitude si les contours avant et après la transformation n'ont pas le même point de départ. Q. M. Tieng et al. [TB97a] proposent un autre descripteur d'ondelettes. Ils calculent d'abord la signature du contour, cette signature est ensuite utilisée pour construire la représentation de passage par zéro (zero-crossing) de la transformation en ondelettes. Ils utilisent la transformation en ondelettes discrètes *sans*

---

**Algorithm 4** : Normalisation des coefficients d'ondelettes

---

**Require:** Les coefficients d'ondelettes d'un contour

**Ensure:** Descripteur de contour invariant à la transformation similitude  
utilisant la transformation en ondelettes

1: Normalisation à la translation

$$(a_i^n, b_i^n) \leftarrow (a_i^n, b_i^n) - (\alpha, \beta) \text{ avec}$$

$$(\alpha, \beta) = \frac{1}{N_1} \sum (a_i^n, b_i^n)$$

où  $N_1$  représente le nombre des coefficients de  $a_i^n$  (ou  $b_i^n$ )

2: Normalisation au changement échelle

Les coefficients de détails

$$r_i^k \leftarrow r_i^k / \bar{r} \text{ avec}$$

$$\bar{r} = \frac{1}{N_2} \sum_{k,i} r_i^k$$

où  $N_2$  représente le nombre des coefficients de  $c_i^k$  (ou  $d_i^k$ )

On procède au même traitement pour les coefficients d'approximation

3: Normalisation à la rotation

Les coefficients de détails

$$\theta_i^k \leftarrow \theta_i^k / \bar{\theta} \text{ avec}$$

$$\bar{\theta} = \frac{1}{N_2} \sum_{k,i} \theta_i^k$$

où  $N_2$  représente le nombre des coefficients de  $c_i^k$  (ou  $d_i^k$ )

On procède au même traitement pour les coefficients d'approximations

---

*décimation SWT* (section 4.8.2) qui est plus coûteuse en temps de calcul, et à la phase d'appariement entre la représentation par les passages par zéro, leur algorithme nécessite l'intégration de la procédure d'élimination des faux zero-crossing. De plus ils n'ont pas résolu le problème d'alignement du point de départ ; ils recourent à la technique de recherche exhaustive. H. S. Yang et al. [HSYL98] ont proposé un descripteur en ondelettes invariant à la transformation similitude et indépendant du choix du point de départ, à travers une analyse mathématique d'une manière élégante mais en même temps compliquée.

Pour remédier à certains défauts des méthodes mentionnées ci-dessus, nous proposons un *nouveau descripteur simple* basé sur la transformation *DWT* [KEh04a][KEh04b]. Nous appellerons ce nouveau descripteur par une abréviation *WD* dans la suite de ce rapport. Il est important de souligner les deux termes *nouveau* et *simple* ; nouveau car, comme nous allons le voir, il est différent des autres méthodes ; en plus, notre descripteur est très simple à calculer car il n'est rien d'autre que les coefficients de la transformation en ondelettes discrètes. L'esquisse des différentes étapes de la construction du vecteur descripteur invariant à la transformation similitude *WD* que nous proposons est décrite dans l'algorithme 5.

Les détails sur l'explication et l'implémentation des instructions de l'algorithme 5 sont déjà donnés dans les différents chapitres précédents, sauf l'explication de l'instruction de la ligne 7 sur le choix du niveau  $L$ . La connaissance de la valeur (niveau) de  $L$  est importante, elle nous permet de sélectionner automatiquement l'ensemble des coefficients pour représenter le contour. La question posée est : Quel est le niveau de décomposition  $L$  à utiliser ? La réponse à cette question est donnée dans le paragraphe suivant.

---

**Algorithm 5** : Extraction du *WD*

---

**Require**: Les coordonnées des points du contour

**Ensure**: Descripteur de contour invariant à la transformation similitude proposé

- 1: Paramétriser le contour par la longueur d'arc (section 2.3.1)
  - 2: Normalisation la longueur totale du contour à une constante (à un par exemple)
  - 3: Ré-échantillonner le contour uniformément en  $N = 2^n$  points
  - 4: Fixation du point de départ sur le contour en utilisant une des quatre méthodes de la section 2.4
  - 5: Calcule de la signature du contour (la distance au centre de gravité, section 2.2.4)
  - 6: Application de la transformation en ondelettes discrètes avec décimation *DWT* (section 4.6) sur la signature obtenue
  - 7: Le vecteur descripteur est un vecteur dont les éléments sont les coefficients d'approximation du contour au niveau  $L$ .
- 

**Le choix du niveau de décomposition  $L$**

Normalement, le vecteur descripteur doit contenir les éléments les plus significatifs dans le sens où à partir de ces éléments, on est capable de représenter le contour le plus fidèlement possible. Cela signifie aussi que s'il existe une transformation inverse de l'espace des descripteurs vers l'espace des contours, le signal (ou contour) obtenu par cette transformation (inverse), sur l'ensemble des éléments composants le vecteur descripteur, doit approcher au mieux le signal d'origine. Une indication globale pour savoir si deux signaux sont relativement proches est l'énergie du signal car on sait que si deux signaux sont similaires, alors il est clair que la différence d'énergie des deux signaux est faible. Par contre, l'inverse n'est pas vrai en général : si la différence d'énergie de deux signaux est faible cela n'implique pas que ces deux signaux soient similaires. Certains descripteurs de formes exploitent cette idée, par exemple, dans le cas de la transformée de Fourier, les coefficients le plus significatifs sont les coefficients dont les amplitudes sont les plus grandes. La raison est que, d'après l'équation 3.8 et l'équation 3.9, on peut montrer que ([CP00] page 60) :

$$\sum_{x=0}^{N-1} |f(x)|^2 = N \sum_{u=0}^{N-1} |F(u)|^2$$

où  $F(u)$ ,  $u = 0 \dots N - 1$  sont des coefficients de Fourier de  $f(x)$ ,  $x = 0 \dots N - 1$ . De l'équation ci-dessus, on peut déduire que l'énergie du signal se concentre principalement sur les coefficients de grande amplitude. Puisque les coefficients de Fourier de haute fréquence tendent vers zéro, cela explique pourquoi il est raisonnable de choisir les coefficients de faible fréquence comme éléments du vecteur descripteur.

Dans le cas de notre descripteur d'ondelette, nous pouvons envisager de procéder de la même manière, c'est à dire qu'on trie l'ensemble des coefficients d'ondelettes dans l'ordre décroissant par rapport à leur amplitude, et on sélectionne quelques dizaines des premiers coefficients comme éléments du vecteur descripteur de *WD*. L'explication à cela est que, comme dans le cas de la transformée de Fourier, les informations du signal se concentrent principalement sur les coefficients d'ondelettes de grande amplitude. Mais pour des raisons pratiques, plus particulièrement

le temps de recherche, ce critère de sélection n'est pas adapté. Le problème est que, lors de l'appariement, pour pouvoir comparer entre deux vecteurs descripteurs, on doit s'assurer ou vérifier que chaque élément d'un vecteur se trouve à la même position dans l'autre vecteur. Afin de résoudre ce problème, Tieng et al. [TB97b] proposent d'utiliser la distribution d'énergie à travers les niveaux de décomposition, le niveau où se concentre le plus d'énergie sera sélectionné temporairement. Et enfin, le niveau  $L$  est choisi si la plupart des contours de la base admettent la valeur  $L$  comme niveau temporaire. Cette stratégie est aussi adoptée par les auteurs de [HSYL98].

Ici, nous proposons une autre stratégie qui consiste à mesurer la perte d'énergie, en pourcentage par rapport à l'énergie du signal de départ, à travers les niveaux de décomposition [KEh04a][KEh04b]. Un niveau est sélectionné temporairement si à ce niveau cette perte est faible et au delà de ce niveau la perte devient très importante. L'objectif de ce choix c'est de préserver l'essentiel des informations du signal en s'autorisant la perte d'une petite quantité d'énergie qui correspond principalement au bruit qui entache le signal. Pour donner un exemple, nous montrons un signal et ses approximations dans la figure 4.8. L'explication se trouve dans la figure même. On constate que, plus le niveau de décomposition est élevé, plus la perte d'énergie augmente et par conséquent la qualité d'approximation devient de plus en plus mauvaise. On constate aussi que la perte d'énergie entre l'approximation au niveau trois et le signal d'origine est faible ( $< 0.50\%$ ) et la perte devient de plus en plus importante à partir du niveau quatre ( $\geq 2.80\%$ ), en parallèle la qualité d'approximation est excellente (resp. médiocre) pour les niveaux de décomposition inférieurs ou égaux à trois (resp. supérieurs ou égaux à quatre). Et donc, l'ensemble des coefficients d'approximations au niveau 3 est un choix raisonnable pour représenter le signal. Pour revenir au choix du niveau  $L$  à utiliser pour l'ensemble des contours de la base, nous pouvons procéder de deux manière différentes :

- **Par vote** : le niveau  $L$  est sélectionné s'il obtient le plus de votes favorables
- **Histogramme de la moyenne d'énergie** : le niveau  $L$  est sélectionné si la perte d'énergie est faible à ce niveau d'approximation (ou décomposition) et si la perte devient plus importante au delà de ce niveau

Pour la première technique, chaque contour choisit (vote) un niveau tel que l'approximation obtenue par la décomposition en ondelette à ce niveau représente au mieux le contour concerné, ensuite on comptabilise le nombre de votes pour chaque niveau. Le niveau  $L$  est élu s'il possède le plus de voix par rapport au nombre de votes des autres niveaux. L'approche utilisée dans cette technique ressemble à l'approche proposée dans [TB97b][HSYL98]. Le reproche qu'on peut faire à cette technique c'est que la décision est prise localement puis répercutée globalement.

Pour la deuxième technique, d'une manière formelle, si on suppose qu'on a une base de contours représentée par des signatures  $(S^j)_{1 \leq j \leq M}$ , avec  $S^j = (s_i^j)_{0 \leq i \leq N}$  où  $N = 2^n$  le nombre d'échantillons de chacun des signaux  $S^j$  et  $M$  est le nombre de contours de la base. Soient  $e_k(S^j)$ ,  $1 \leq k \leq n$  représente l'énergie du signal d'approximation de  $S^j$  au niveau  $k$  et soit :

$$\dot{e}_k = \frac{1}{M} \sum_{j=1}^M e_k(S^j), \quad k = 1 \dots n$$

Alors le niveau  $L$ ,  $1 \leq L \leq n$ , est sélectionné si les valeurs  $100 - \dot{e}_L$  et  $100 - \dot{e}_{L+1}$  sont respectivement faibles et grandes. Pour donner un exemple, prenons le cas de la sous-base B de la base des contours CE-1 MPEG-7 de la section 5.2, la répartition d'énergie des  $\dot{e}_k$ ,  $k = 1 \dots 7$  est donnée dans le tableau 4.1. L'histogramme correspondant au tableau 4.1 est donné dans la

figure 4.9. Selon cet histogramme (ou tableau), le choix raisonnable de la valeur est  $L = 3$  ou

Niveau	0	1	2	3	4	5	6	7
Energie moyenne	100	99.99	99.97	99.78	98.89	95.73	91.43	91.15

TAB. 4.1 – Le pourcentage en énergie moyenne des signatures de la base sur chaque niveau de décomposition : chaque élément du tableau représente la valeur de  $\dot{e}_k$ ,  $k = 0 \dots 7$

$L = 4$ . Mais selon notre expérimentation, la valeur  $L = 3$  donne un meilleur résultat.

Par rapport à la première technique, dans la deuxième technique la décision se fait globalement par rapport à tous les contours de la base. En plus, bien que les deux procédures soient à exécuter hors ligne pour déterminer la valeur de  $L$ , on peut dire que le temps d'exécution pour la première technique est beaucoup plus important que pour la deuxième.

### 4.11 Descripteur d'ondelettes invariant à la transformation affine

Dans le cas de la transformation affine, nous voulons prolonger directement la méthode que nous avons proposée dans la section précédente, avec quelques modifications, par exemple au lieu d'utiliser la longueur d'arc comme paramètre, on peut le remplacer par le paramètre surface fermée (enclosed-area parameter) etc. Mais le problème est que l'on ne peut pas l'appliquer directement sur les contours, les techniques pour fixer le point de départ comme dans le cas de la transformation similitude. La raison est qu'on n'a pas la même intuition dans le cas de la transformation similitude que dans le cas de la transformation affine : si un point  $P$  est considéré comme point de départ par l'une des quatre méthodes décrites dans la section 2.4, l'image de  $P$  par une transformation affine ne sera probablement pas sélectionné comme point par la même méthode. Pour illustrer cette idée, prenons un exemple simple dans le cas où la *longueur maximum* par rapport au centre de gravité, est utilisée pour fixer le point de départ, du contour de la figure 4.10.

$$O = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, P1 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}, P2 = \begin{bmatrix} 1 \\ 3 \end{bmatrix}, P3 = \begin{bmatrix} -2 \\ -2 \end{bmatrix}$$

En prenant la transformation affine suivante :

$$A = \underbrace{\begin{bmatrix} 1 & -2 \\ 0 & 1 \end{bmatrix}}_{\text{Etirement}} \underbrace{\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}}_{\text{Rotation}} = \begin{bmatrix} -2 & -1 \\ 1 & 0 \end{bmatrix}$$

Si  $(Q_i)_{1 \leq i \leq 3}$  sont les images des trois points  $(P_i)_{1 \leq i \leq 3}$  par la transformation affine  $A$ , on obtient :

$$Q1 = \begin{bmatrix} -1 \\ 1 \end{bmatrix}, Q2 = \begin{bmatrix} -5 \\ 1 \end{bmatrix}, Q3 = \begin{bmatrix} 6 \\ -2 \end{bmatrix}$$

On vérifie bien que le centre de gravité des deux contours obtenus par ces points se trouve bien au point  $O$  l'origine du repère. Si on utilise la longueur maximum pour fixer le point de départ les deux contours avant et après la transformation affine,  $P_2$  sera considéré comme point de

départ par cette méthode pour le contour engendré par les trois points  $(P_i)_{1 \leq i \leq 3}$ , alors que  $Q_2$  l'image de  $P_2$  n'est pas le point de départ pour le contour  $(Q_i)_{1 \leq i \leq 3}$  car  $|OQ_2| < |OQ_3|$ .

Par conséquent, nous devons utiliser un autre moyen pour arriver à proposer un nouveau descripteur affine. Nous rappelons que le groupe de la *transformation affine* est un groupe qui contient les transformations *similitudes et les étirements* et que la forme générale de la transformation d'étirement est la suivante :

$$\begin{bmatrix} a & s \\ 0 & 1/a \end{bmatrix}$$

Revenons sur les propriétés de la transformation en ondelettes de la section 4.9, si deux contours  $\check{\Gamma} = (\check{x}(t), \check{y}(t))$  et  $\Gamma = (x(t), y(t))$  sont reliés par une transformation affine, alors nous avons :

$$\begin{bmatrix} \check{x}(t) \\ \check{y}(t) \end{bmatrix} = \begin{bmatrix} a_1 & a_2 \\ b_1 & b_2 \end{bmatrix} \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} + \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \quad (4.19)$$

où la matrice  $\begin{bmatrix} a_1 & a_2 \\ b_1 & b_2 \end{bmatrix}$  et le vecteur  $\begin{bmatrix} \alpha \\ \beta \end{bmatrix}$  sont des composants de la transformation affine. D'après les propriétés de la section 4.9, nous avons :

$$\begin{aligned} \begin{bmatrix} A^l \check{x}_i \\ A^l \check{y}_i \end{bmatrix} &= \begin{bmatrix} a_1 & a_2 \\ b_1 & b_2 \end{bmatrix} \begin{bmatrix} A^l x_i \\ A^l y_i \end{bmatrix} + \begin{bmatrix} 2^{l/2} \alpha \\ 2^{l/2} \beta \end{bmatrix} \\ \begin{bmatrix} D^l \check{x}_i \\ D^l \check{y}_i \end{bmatrix} &= \begin{bmatrix} a_1 & a_2 \\ b_1 & b_2 \end{bmatrix} \begin{bmatrix} D^l x_i \\ D^l y_i \end{bmatrix} \end{aligned}$$

La présence du vecteur  $(2^{l/2} \alpha, 2^{l/2} \beta)^t$  est due au vecteur de translation  $(\alpha, \beta)$ . Pour éliminer ce vecteur de l'équation ci-dessus, on peut normaliser des coefficients de détails en retranchant la moyenne des coefficients (voir algorithme 4), on peut aussi normaliser en déplaçant le contour de sorte que l'origine du repère se retrouve au centre de gravité du contour. En adoptant ce déplacement, nous avons la relation suivante :

$$\begin{bmatrix} A^l \check{x}_i & D^l \check{x}_i \\ A^l \check{y}_i & D^l \check{y}_i \end{bmatrix} = \begin{bmatrix} a_1 & a_2 \\ b_1 & b_2 \end{bmatrix} \begin{bmatrix} A^l x_i & D^l x_i \\ A^l y_i & D^l y_i \end{bmatrix}$$

Si on pose :

$$M^l(i) = A^l x_i D^l y_i - A^l y_i D^l x_i$$

nous aurons :

$$\check{M}^l(i) = (a_1 b_2 - a_2 b_1) M^l(i)$$

Cela veut dire que  $M$  est un invariant relatif par rapport à la transformation affine. Pour obtenir l'invariant absolu, il suffit de remplacer les  $M^l(i)$  par les  $I^l(i)$  tel que :

$$I^l(i) = M^l(i) / M^k(j) \quad (4.20)$$

Ici,  $M^k(j)$  est une constante, son rôle dans l'équation ci-dessus est uniquement pour la normalisation. Afin de réduire l'effet du bruit sur la représentation, le niveau  $k$  et la position  $j$  sont choisis avec :

$$k = \arg \max_{1 \leq l \leq n} \sum_{j=1}^N |M^l(j)|^2 \quad \text{et} \quad j = \arg \max_{1 \leq i \leq N} |M^k(i)|$$

Donc, on peut utiliser le vecteur  $I^l(i)$  comme un vecteur descripteur affine car si les deux contours  $\check{\Gamma}$  et  $\Gamma$  sont reliés par une transformation affine, de l'équation 4.19 et l'équation 4.20, on peut montrer que :

$$\check{I}^l(i) = I^l(i) \quad (4.21)$$

et que, comme dans la remarque 4, cette relation est valable dans les deux cas *DWT* et *SWT* à condition que  $\check{\Gamma}$  et  $\Gamma$  aient le même point départ. Qu'est ce qui se passe si les deux contours n'ont pas le même point de départ ?

Nous avons vu dans l'exemple de la figure 4.4 que les coefficients de *DWT* ne sont pas invariants par rapport à la translation (point de départ), donc a fortiori les coefficients  $I^l(i)$ , qui sont obtenus à partir de la transformation *DWT*, ne sont pas invariants par rapport à la transformation affine. Autrement dit, dans le cas où les invariants  $I^l(i)$  sont obtenus à partir des coefficients de *DWT*, l'égalité de l'équation 4.21 est fautive, cela veut dire aussi qu'on ne peut utiliser le vecteur  $I^l$  comme descripteur affine. Par contre, dans le cas où les coefficients  $I^l(i)$  sont obtenus à partir des coefficients de la transformation *SWT*, on peut montrer qu'il existe une constante inconnue  $\tau$  telle que :

$$\check{I}^l(i) = I^l(i + \tau)$$

Il est possible de calculer la valeur  $\tau$  pendant l'exécution, lors de la phase d'appariement entre les vecteurs descripteurs en utilisant la recherche exhaustive très coûteuse, mais on ne peut pas appliquer directement sur les contours, une des quatre méthodes pour fixer le point de départ, pour déterminer la valeur de  $\tau$ , pour des raisons déjà évoquées au début de la section. En fait, si on aligne l'élément, dont l'amplitude est maximale, de chacun des deux vecteurs  $\check{I}^l$  et  $I^l$  ou bien si on utilise l'orientation d'axe naturel de l'ensemble ordonné  $\check{I}^l$  et  $I^{l12}$ , on peut aussi déterminer la valeur de  $\tau$ .

Il faut souligner que l'invariant de l'équation 4.21 est obtenu en utilisant à la fois les coefficients d'approximations et les coefficients de détails de la transformation *SWT*, et que cet invariant utilise uniquement un niveau dyadiques. M. I. Khalil et al. [KB01][KB02] proposent plusieurs autres invariants en utilisant uniquement les coefficients de détail de *SWT* sur deux, trois, quatre et six niveaux dyadiques. Mais comme la transformation *SWT* est une version sur-échantillonnée de la transformation *DWT*, d'où la *redondance* et par conséquent tous les invariants obtenus à partir des coefficients *SWT* sont aussi redondants. Afin de réduire la redondance, Q. M. Tieng et al. [TB97b] proposent un nouveau descripteur affine utilisant uniquement les points extrêmes, les maxima et les minima de l'équation 4.21. A nouveau, deux problèmes principaux se posent :

- En raison du bruit, on doit intégrer une procédure de suppression des faux points extrêmes pour pouvoir comparer deux descripteurs qui n'ont pas le même nombre de points extrêmes
- On doit trouver le correspondant entre chaque élément du premier descripteur avec le deuxième. Bien que le nombre de points extrêmes soit relativement faible, le coût lié au temps de recherche reste non négligeable surtout dans le cas où la taille de la base est importante.

Afin de résoudre ces problèmes, nous proposons un nouveau descripteur affine que nous appelons *AWT* dans la suite de ce rapport [KEh][KEh04c]. Les caractéristiques de l'*AWD* sont :

---

<sup>12</sup>On peut considérer une suite de valeur  $(I^l(i))_{1 \leq i \leq N}$  comme un vecteur ou bien comme un ensemble ordonné



- Descripteur complètement non-redondant
- Il permet un appariement simple, il n'y a pas de surcoût lié à la recherche du correspondant entre chaque élément des descripteurs
- La dimension du vecteur descripteur est constante

**Algorithm 6** : Extraction de l'AWD**Require:** Les coordonnées des points du contour**Ensure:** Descripteur de contour invariant à la transformation affine

- 1: Paramétriser le contour par la surface fermée (section 2.3.3)
- 2: Normaliser la surface fermée totale du contour à une constante (à un par exemple)
- 3: Ré-échantillonner le contour uniformément en  $N = 2^n$  points
- 4: Appliquer la transformation en ondelettes discrètes sans décimation *SWT* (section 4.8.2) sur le contour puis déduire la suite des valeurs  $I^l$  de l'équation 4.20
- 5: Décaler circulairement du vecteur  $I^l$  pour que son premier élément  $I^l(1)$  ait l'amplitude maximale
- 6: Supprimer les éléments redondants du vecteur  $I^l$
- 7: Le vecteur descripteur est un vecteur dont les éléments sont des valeurs  $I_i^l$  qui restent après la suppression de la redondance.

L'esquisse des différentes étapes pour construire le vecteur descripteur est résumée dans l'algorithme 6. Pour l'instruction 5 de cet algorithme, nous avons utilisé l'amplitude maximale pour placer le premier élément, mais à la place de l'amplitude maximale, on peut aussi utiliser l'orientation de l'axe naturel. En revanche on ne peut pas utiliser les deux autres méthodes, courbure maximale ou axe principal, car en fait il n'est pas possible de parler de la courbure ou de l'axe principal du vecteur  $I^l$ , car ça n'a pas de sens. Le rôle de cette instruction est simplement d'éviter de chercher le correspondant entre les éléments du premier vecteur descripteur et ceux du deuxième. La seule instruction que nous n'avons pas encore abordée dans l'algorithme est la suppression de la redondance. Comment peut-on procéder pour supprimer la redondance ? Dans la section suivante, nous allons expliquer la méthode que nous proposons pour répondre à cette question.

**Elimination de la redondance**

Dans cette section, nous allons présenter la méthode que nous proposons pour supprimer la redondance [KEh][KEh04c]. Cette méthode est assez générale ; par exemple, on peut l'utiliser pour supprimer la redondance des invariants proposés par M. I. Khalil et al. dans [KB01] et [KB02]. Du point de vue pratique et théorique, cela est très important car cela permet de réduire la quantité des données à gérer, le temps d'exécution et donc le temps de recherche ; par conséquent cela répond aux exigences de la norme MPEG-7 (contraintes numéro 2, 4 et 5). L'idée de base de notre proposition est inspirée de la relation entre les coefficients de *DWT*  $\varepsilon$ -décimés et les coefficients de *SWT* de la section 4.8 <sup>13</sup>. Cette relation, nous explique aussi comment extraire une *DWT*  $\varepsilon$ -décimés à partir de la structure contenant les coefficients d'approximations

<sup>13</sup>Au passage, nous rappelons que, au contraire de l'ensemble des coefficients de *SWT*, l'ensemble des coefficients de la transformation *DWT*  $\varepsilon$ -décimée est compact, ils ne sont pas redondants

et de détails de la *SWT*.

Pour mieux comprendre cela, reprenons un exemple donné par [MMOP03] page 106-107). Décomposons une suite de huit nombres au niveau  $n = 3$  en utilisant la *SWT* avec une ondelette orthogonale. Les calculs effectués à chaque étape par l'algorithme sont rassemblés dans les tableaux suivants. Dans ces derniers, on note  $a(j, \varepsilon_1, \dots, \varepsilon_j)$  ou  $d(j, \varepsilon_1, \dots, \varepsilon_j)$  un coefficient d'approximation ou détail au niveau  $j$  obtenu pour la *DWT*  $\varepsilon$ -décimé caractérisée par la séquence  $\varepsilon = [\varepsilon_1, \dots, \varepsilon_j]$ .

**Etape 0** (*Données d'origine*)

Au départ, le signal est assimilé à l'approximation de niveau 0, les huit coefficients sont tous notés  $a(0)$ .

$a(0)$	$a(0)$	$a(0)$	$a(0)$	$a(0)$	$a(0)$	$a(0)$	$a(0)$
--------	--------	--------	--------	--------	--------	--------	--------

**Etape 1**

Par filtrage, l'algorithme décompose le vecteur des  $a(0)$  en deux : la première ligne contient les deux détails obtenus pour  $\varepsilon_1 = 0$  et pour  $\varepsilon_1 = 1$  entrelacés et notés  $d(1, \varepsilon_1)$ . De même, on trouve les approximations dans la deuxième ligne.

$d(1, 0)$	<b><math>d(1,1)</math></b>	$d(1, 0)$	<b><math>d(1,1)</math></b>	$d(1, 0)$	<b><math>d(1,1)</math></b>	$d(1, 0)$	<b><math>d(1,1)</math></b>
$a(1, 0)$	<b><math>a(1,1)</math></b>	$a(1, 0)$	<b><math>a(1,1)</math></b>	$a(1, 0)$	<b><math>a(1,1)</math></b>	$a(1, 0)$	<b><math>a(1,1)</math></b>

**Etape 2**

L'algorithme ne touche plus à la première ligne du tableau contenant les détails de niveau 1. En revanche, les deux approximations de niveau 1 sont décomposées en 2, et on construit :

- les quatre détails de niveau 2, repérés par  $d(2, \varepsilon_1, \varepsilon_2)$  et figurés dans la deuxième ligne ;
- les quatre approximations de niveau 2, repérés semblablement.

$d(1, 0)$	<b><math>d(1,1)</math></b>	$d(1, 0)$	<b><math>d(1,1)</math></b>	$d(1, 0)$	<b><math>d(1,1)</math></b>	$d(1, 0)$	<b><math>d(1,1)</math></b>
$d(2, 00)$	<b><math>d(2,10)</math></b>	$d(2, 01)$	$d(2, 11)$	$d(2, 00)$	<b><math>d(2,10)</math></b>	$d(2, 01)$	$d(2, 11)$
$a(2, 00)$	<b><math>a(2,10)</math></b>	$a(2, 01)$	$a(2, 11)$	$a(2, 00)$	<b><math>a(2,10)</math></b>	$a(2, 01)$	$a(2, 11)$

**Etape 3**

$d(1, 0)$	<b><math>d(1,1)</math></b>	$d(1, 0)$	<b><math>d(1,1)</math></b>	$d(1, 0)$	<b><math>d(1,1)</math></b>	$d(1, 0)$	<b><math>d(1,1)</math></b>
$d(2, 00)$	<b><math>d(2,10)</math></b>	$d(2, 01)$	$d(2, 11)$	$d(2, 00)$	<b><math>d(2,10)</math></b>	$d(2, 01)$	$d(2, 11)$
$d(3,000)$	$d(3,100)$	$d(3,010)$	$d(3,110)$	$d(3,001)$	<b><math>d(3,101)</math></b>	$d(3,011)$	$d(3,111)$
$a(3,000)$	$a(3,100)$	$a(3,010)$	$a(3,110)$	$a(3,001)$	<b><math>a(3,101)</math></b>	$a(3,011)$	$a(3,111)$

Si  $\varepsilon = [\varepsilon_1, \dots, \varepsilon_n]$  avec  $\varepsilon_i = 0$  ou 1, on obtient alors  $2^n = 8$  *DWT*  $\varepsilon$ -décimé. Le choix d'une séquence  $\varepsilon$  permet d'extraire la *DWT*  $\varepsilon$ -décimé correspondante à partir du tableau représentant les valeurs de la *SWT*. Notons  $C_\varepsilon$  la structure de décomposition d'une *DWT*  $\varepsilon$ -décimé pour un  $\varepsilon$  donné. Alors, on peut extraire cette dernière de la structure de la décomposition par la *SWT* en sélectionnant les coefficients appropriés de la manière suivante :

$$C_\varepsilon = \begin{array}{|c|c|c|c|} \hline d(1, \varepsilon_1) & d(1, \varepsilon_1) & d(1, \varepsilon_1) & d(1, \varepsilon_1) \\ \hline d(2, \varepsilon_1\varepsilon_2) & d(2, \varepsilon_1\varepsilon_2) & & \\ \hline d(2, \varepsilon_1\varepsilon_2\varepsilon_3) & & & \\ \hline a(2, \varepsilon_1\varepsilon_2\varepsilon_3) & & & \\ \hline \end{array}$$

Par exemple, la  $DWT$   $\varepsilon$ -décimé correspondant à  $\varepsilon = [\varepsilon_1 \varepsilon_2 \varepsilon_3] = [101]$  est représentée en caractère gras dans la suite de tableaux présentés dans l'exemple précédent.

D'une manière formelle, pour une séquence  $\varepsilon = [\varepsilon_1, \dots, \varepsilon_n]$  avec  $\varepsilon = 0$  ou  $1$ , il est possible d'obtenir l'ensemble des coefficients de la  $DWT$   $\varepsilon$ -décimé par la formule suivante :

$$\begin{aligned} A_{DWT_\varepsilon}^l(x) &= [A_{DWT_\varepsilon}^l x_i]_{1 \leq i \leq 2^{n-l}} \\ &= [A_{SWT}^l(\alpha), A_{SWT}^l(\alpha + 2^l), \dots, A_{SWT}^l(\alpha + 2^l(2^{n-l} - 1))] \end{aligned} \quad (4.22)$$

avec

$$\alpha = 1 + \sum_{j=1}^l \varepsilon_j 2^{j-1}$$

La formule ci-dessus s'écrit avec les coefficients d'approximations, mais elle est également valable pour les coefficients de détails.

De l'équation 4.22 et l'équation 4.20, on peut en déduire notre descripteur en ondelettes invariant à la transformation affine *compact et non-redondant*, en prenant la séquence  $(\varepsilon_i = 0)_{1 \leq i \leq l}$ , suivant :

$$D^l = (D^l(j)_{1 \leq j \leq 2^{n-l}}) = [I^l(1), I^l(1 + 2^l), \dots, I^l(1 + 2^l(2^{n-l} - 1))] \quad (4.23)$$

## 4.12 Complexité du calcul des descripteurs

Dans cette section on s'intéresse à la complexité liée au nombre d'opérations utilisées pour extraire les deux descripteurs  $WD$  et  $AWD$  que nous proposons.

Pour le descripteur invariant à la transformation similitude  $WD$  (algorithme 5), en supposant que le nombre de points du contour est proportionnel à  $N = 2^n$ , avec  $N$  représente le nombre d'échantillons utilisés pour ré-échantillonner uniformément le contour. Selon l'équation 2.7, la paramétrisation du contour par la longueur d'arc (instruction numéro 1) nécessite  $O(N)$  opérations. Sans rentrer dans les détails, on peut dire que les instructions numéro 2 et 3 nécessitent également  $O(N)$  opérations. Pour l'instruction numéro 4, concernant la fixation du point de départ, bien que les quatre méthodes soient différentes, leur complexité reste la même  $O(N)$ . Le calcul de la signature et des coefficients de  $DWT$  de cette signature, instructions numéro 6 et 7, nécessite également  $O(N)$  opérations. En résumé, la complexité de notre algorithme pour calculer le vecteur descripteur invariant à la transformation similitude  $WD$  est de l'ordre  $O(N)$ .

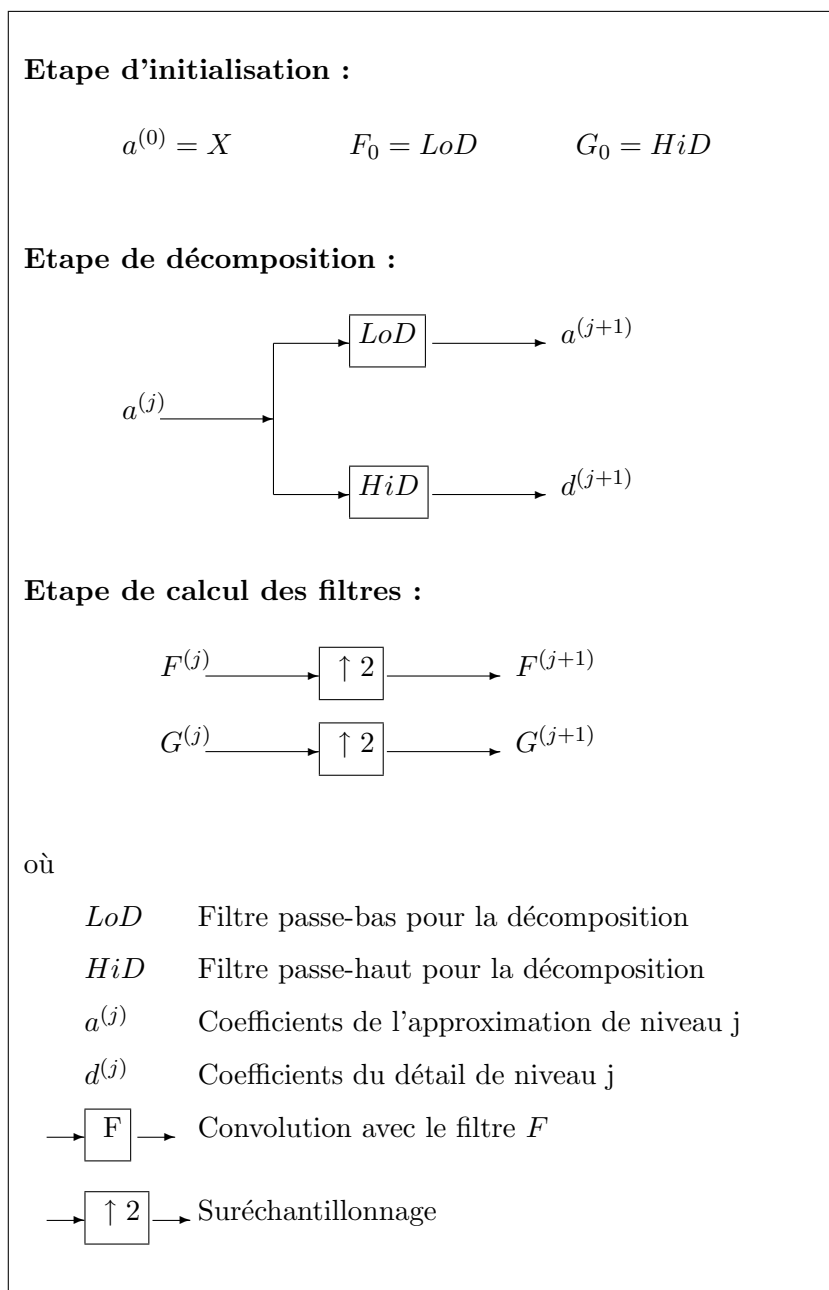
Dans le cas du descripteur invariant à la transformation affine  $AWD$ , algorithme 6, la différence principale entre cet algorithme et l'algorithme 5 est l'instruction numéro 1, 4 et 6. Or on sait, en utilisant l'équation 2.13, que le nombre d'opérations utilisées pour la paramétrisation par la surface fermée, instruction numéro 1, nécessite  $O(N)$  opérations. La complexité de la transformation  $SWT$ , instruction numéro 4, est de l'ordre  $O(N \log_2 N)$  et enfin d'après la formule 4.23, le nombre d'opérations nécessaires pour supprimer les éléments redondants, l'instruction numéro 6, est inférieur ou égal à  $O(N)$ . D'où, en procédant au même raisonnement que dans le cas de l'algorithme 5, au total la complexité l'algorithme 6 pour extraire le vecteur descripteur invariant à la transformation affine  $AWD$  est de l'ordre  $O(N \log_2 N)$ .

### 4.13 Conclusion

Dans ce chapitre, nous avons présenté nos travaux relatifs à la reconnaissance des formes utilisant la transformation en ondelettes. Nous avons présenté, dans la première partie de ce chapitre, la théorie liée à la transformation en ondelettes en la séparant en deux parties : le cas continu et le cas discret. La liaison des deux cas s'effectue à travers l'analyse multi-résolution. Dans le cas discret, nous avons présenté aussi la transformée invariante par translation.

Dans un second temps, nous avons présenté nos travaux relatifs à la définition de nos deux descripteurs d'ondelettes invariants respectivement à la transformation similitude  $WD$  et à la transformation affine  $AWD$ . Bien qu'il y ait déjà des travaux dans le domaine de la reconnaissance des formes utilisant la transformation en ondelettes, les descripteurs que nous avons proposés sont différents : nous avons proposé notamment une méthode simple et facile à mettre en oeuvre pour sélectionner le niveau de décomposition dans le cas de la  $WD$ , ainsi qu'une méthode pour supprimer la redondance dans le cas de  $AWD$ .

Nous présentons, dans le chapitre suivant, les expérimentations et les évaluations de nos deux descripteurs d'ondelettes dans les applications de reconnaissance des formes. Nous présenterons également les comparaisons de notre descripteur avec certains autres descripteurs que nous avons présentés dans le chapitre 3.

FIG. 4.6 – *SWT 1D* : algorithme de décomposition

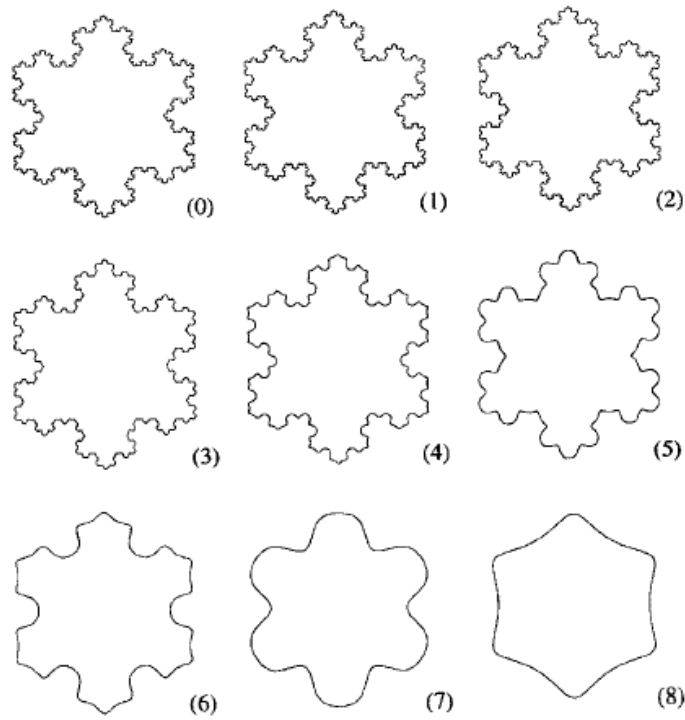


FIG. 4.7 – (0) Contour de départ (1-8) Les approximations du contour de départ de la résolution la plus fine vers la résolution la plus grossière [CK96]

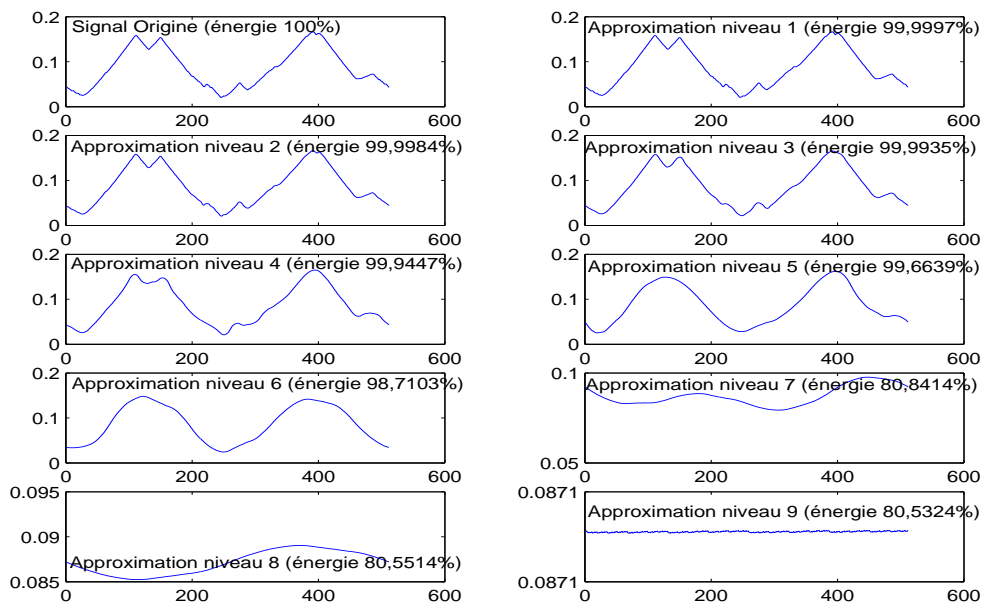


FIG. 4.8 – Energie du signal et de ses approximations sur plusieurs niveaux de décomposition

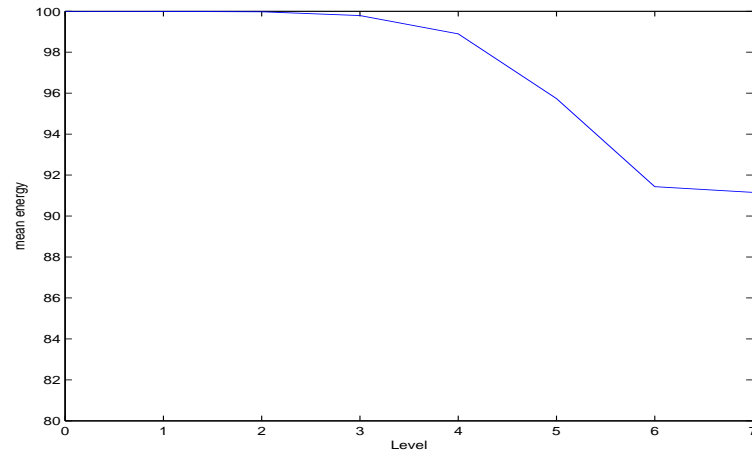


FIG. 4.9 – La représentation graphique de l'énergie en moyenne de la décomposition des signatures de la base par *DWT* sur sept niveaux

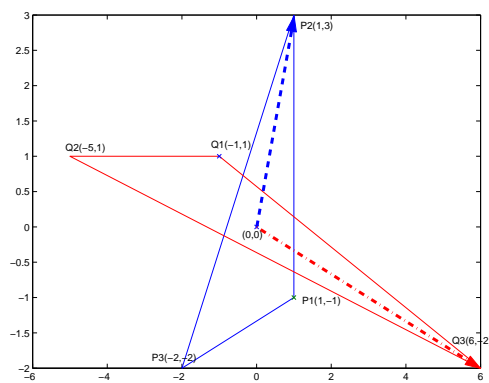


FIG. 4.10 – Contre-exemple : la relation d'ordre de la longueur des points du contour n'est pas préservée dans le cas de la transformation affine





# Chapitre 5

## Expérimentations, évaluations et comparaisons

### 5.1 Introduction

Dans ce chapitre, nous allons mener des expérimentations sur les deux descripteurs de formes que nous avons proposés par rapport à leur capacité de décrire et de reconnaître les contours par rapport aux différentes modifications que peuvent subir ces derniers. Ce chapitre commence d'abord par la présentation du contexte des expérimentations dans la section 5.2. Dans le cadre du descripteur de formes invariant à la transformation similitude (section 5.3), nous allons tester plusieurs atomes d'ondelettes, ainsi que la comparaison des quatre méthodes pour fixer le point de départ. Afin de rendre cette expérimentation et évaluation consistante, nous l'avons expérimenté et évalué sur la base des contours MPEG-7. Dans la section 5.4, nous allons mener une étude comparative des performances de plusieurs descripteurs de Fourier ( $FD$ ). Bien que cette étude concerne la plupart des  $FD$  elle n'est cependant pas exhaustive. L'objectif de cette étude est de répondre à la question : quel est le meilleur  $FD$  à retenir ? Une fois que le meilleur  $FD$  est connu, nous continuons notre présentation autour de l'étude comparative de la performance entre le meilleur  $FD$  et le descripteur de Fourier à fenêtre ( $SFD$ ), et enfin autour de l'étude comparative de la performance entre  $FD$  et le descripteur d'espace-échelle de courbure  $CSSD$ . Il faut souligner que,  $CSSD$  fait partie des descripteurs sélectionnés par le groupe MPEG-7 comme descripteur de formes basé sur le contour robuste répondant à la plupart des critères de la norme MPEG-7. On rappelle au passage que les six principes (critères) de la norme MPEG-7 sont : bonne précision de recherche, descripteur compact, application générale, faible complexité par rapport au temps de calcul utilisé pour extraire et comparer les descripteurs, faible dégradation des performances quand on augmente la taille de la base de données et représentation hiérarchique (ou multi-échelle ou multirésolution). Le résultat des comparaisons des trois descripteurs  $FD$ ,  $SFD$  et  $CSSD$  nous permet de savoir quel est le meilleur descripteur. La réponse à cette question est très importante pour nous et nous intéresse car nous voulons comparer la performance des descripteurs que nous avons proposés avec celui qui est considéré comme le meilleur afin de savoir où se situe notre travail. Ces études comparatives sont effectuées en utilisant les mêmes critères de comparaison, les six principes de la norme MPEG-7, et évaluées sur la même base de contours.

Avant de terminer le chapitre, dans la section 5.5, nous présentons l'étude expérimentale sur l'évaluation de notre descripteur de formes invariant à la transformation affine. Enfin, nous

concluons notre chapitre à la section 5.6.

## 5.2 Contexte des expérimentations

Nous avons implémenté sous MATLAB l'algorithme de reconnaissance des formes basé sur le contour en utilisant les descripteurs que nous avons proposés : le descripteur en ondelettes invariant à la transformation similitude et le descripteur en ondelettes invariant à la transformation affine. Ces algorithmes ont été exécuté sur un PC sous Windows 2000 Professionnel, disposant d'un processeur Intel Pentium 4 cadencé à 1.70 GHz, de 261 Mo RAM et de 9.04 Go espace disque dur.

Pour conduire nos expérimentations, nous avons utilisé la base des contours MPEG-7 CE-1. Cette base a été obtenue à partir d'objets réels et d'objets déformés dûs à la prise de vue et à la segmentation. Cette base a été créée dans le but de tester le comportement des descripteurs par rapport à la déformation des contours. La base CE-1 contient 3 sous bases [LJLE00] :

**La base A** contient deux sous bases, la base A1 et la base A2 ; chacune de ces bases contient 420 contours qui sont regroupés en 70 classes et chaque classe contient 6 contours similaires. La base A1 est utilisée pour tester l'invariance par rapport aux changements d'échelle, et la base A2 est utilisée pour tester l'invariance par rapport à la rotation.

**La base B** est la base la plus importante et la plus intéressante ; elle contient 1400 contours qui sont regroupés également en 70 classes. Chaque classe dans cette base contient 20 contours similaires. Cette base est utilisée pour tester la robustesse des descripteurs par rapport à la transformation similitude et la robustesse des descripteurs par rapport à diverses déformations arbitraires de forme, y compris la rotation, le changement d'échelle, l'étirement, la défection, l'indentation et d'autres variations.

**La base C** se compose de 200 contours d'un poisson en mouvement pris dans un court extrait d'une vidéo et d'une base de 1100 autres contours de poissons. Il faut préciser que cette base n'est pas classée. Cette base est utilisée pour tester la robustesse des descripteurs de forme par rapport à une transformation non-rigide.

Nous n'avons pas, en notre possession, la base A et les 200 contours du même poisson en mouvement extrait d'une vidéo. Donc pour nos expérimentations, évaluations et comparaisons, nous n'utilisons que la base B et une sous base de la base C de 1100 contours. Des exemples de quelques contours de la base B sont présentés dans la figure 5.1. La figure 5.1(a) montre 70 contours, dont chacun est extrait des différentes classes de la sous-base B de la grande-base CE-1 MPEG7, pour montrer que cette base contient des objets assez divers. La figure 5.1(b) donne un exemple de 20 contours qui appartiennent à la même classe. Il montre aussi que des objets de la même classe ont des caractéristiques et natures assez diverses les uns par rapport aux autres. La figure 5.2 et la figure 5.3 montrent les 280 premiers contours de la sous-base C de la grande-base CE-1 MPEG-7.

Dans notre implémentation, puisque l'on doit prendre en compte la symétrie entre les contours, nous utilisons les deux parcours, selon la direction des aiguilles d'une montre et la direction inverse. Pour mesurer la similarité entre les descripteurs, nous avons utilisé la distance city block pour des raisons déjà expliquées dans la section 2.9. En ce qui concerne l'indexation, nous utilisons la stratégie de recherche séquentielle. Pour évaluer et comparer les performances entre les différents descripteurs, nous utilisons la mesure de Précision et Rappel ( $P/R$ ). Pour chaque requête, nous calculons les valeurs de Précision correspondantes aux valeurs de Rappel. La valeur

Précision finale pour une valeur de Rappel est la moyenne des valeurs de Précision de toutes les requêtes.

### 5.3 Evaluation du descripteur *WD*

Dans cette section nous effectuons une évaluation sur la capacité de notre descripteur invariant à la transformation similitude *WD*. Nous commençons d'abord par donner quelques résultats concrets, ensuite nous comparons les performances des descripteurs obtenus avec différents atomes d'ondelettes. A l'issue de cette comparaison, nous serons en mesure de savoir, parmi les atomes d'ondelettes utilisés, quel atome est le meilleur. Nous comparons également les performances des descripteurs obtenus en utilisant les différentes méthodes de la section 2.4 pour fixer le point de départ. L'objectif de cette deuxième comparaison est de déterminer, parmi les quatre techniques pour fixer le point de départ, laquelle donne les meilleures performances en terme de Précision et Rappel. Toutes les expérimentations et comparaisons sont effectuées sur la sous-base B de la base CE-1 MPEG-7 et toutes les valeurs de précisions retranscrites dans ces comparaisons sont des valeurs moyennes des Précisions calculées sur 1400 requêtes.

#### 5.3.1 Résultat concret

Tout de suite pour commencer l'expérimentation, nous donnons quelques résultats concrets. Prenons les deux premiers contours de la figure 5.1(a) et les deux premiers contours de la figure 5.1(b) comme requête. Dans cet exemple, nous avons utilisé l'ondelette *rbio5.5* (voir la section suivante) et la distance maximum au centre de gravité pour fixer le point de départ. Les 20 premières réponses à ces quatre requêtes sont présentées dans la figure 5.4. Pour la première et la deuxième requête, notre algorithme a trouvé respectivement les quinze contours et dix huit contours de la même classe (figure 5.4(a) et (b)), alors que pour la troisième et la quatrième requête notre algorithme a trouvé respectivement treize et onze contours de la même classe (figure 5.4(c) et (d)). On constate que le nombre de résultats corrects dépend de la classe à laquelle la requête appartient car la déformation présente dans chaque classe est assez diverse et n'est pas de même nature d'une classe à l'autre.

#### 5.3.2 L'effet du nombre de moments nuls

Rappel	10	20	30	40	50	60	70	80	90	100	mean
db1	92.4	80.0	70.0	62.1	55.1	48.2	42.1	35.3	27.0	16.9	52.9
db2	92.3	80.3	71.2	63.2	55.8	48.8	42.7	35.9	27.7	17.2	53.5
db3	92.4	80.8	71.1	63.4	56.6	49.3	43.1	36.6	27.9	17.3	53.8
db6	92.6	80.8	71.3	63.7	56.8	49.5	43.4	36.7	28.1	17.4	54.0
db15	92.7	80.5	71.4	63.8	56.5	49.8	43.6	36.7	28.2	17.7	55.0

TAB. 5.1 – Tableau de comparaison des performances en moyenne de la recherche en utilisant les atomes d'ondelettes de la même famille mais avec le nombre de moments nuls différents. Les éléments de ce tableau sont des valeurs moyennes de Précision (P) calculées en utilisant 1400 requêtes.

Dans cette section, nous allons discuter l'importance du nombre de moments nuls, sur la représentation du contour par le *WD*. Les trois propriétés : le nombre de moments nuls, la

longueur du support et la régularité de l'ondelette sont liés : les ondelettes qui ont des nombres de moments nuls faibles ont des petits supports et leurs courbes sont moins régulières. Plus le nombre de moments nuls augmente, plus leurs supports deviennent larges et plus leurs courbes sont régulières. Par définition, l'ondelette  $\psi$  possède  $m$  moments nuls si  $\psi$  satisfait l'équation :

$$\int_{\mathcal{R}} t^l \psi(t) dt = 0, \quad \text{pour } l = 0 \dots m - 1$$

L'ondelette que nous utilisons dans cette expérimentation est celle issue de la famille  $dbN$  ( $db$  abréviation de Daubechie et  $N$  est le nombre de moments nuls). Le tableau 5.1 résume le résultat de cette expérimentation. Ce tableau 5.1 montre que la performance globale de l'ondelette Daubechie d'ordre 15 est meilleure que la performance de l'ondelette Daubechie d'ordre 1, 2, 3 et 6. En fait, selon notre expérimentation pour l'ondelette issue de la même famille, on constate que plus le nombre de moments nuls est élevé, plus la performance globale en terme de P/R est meilleure en dépit du temps de calcul. La raison est que l'ondelette possédant le nombre de moments nuls élevés tend à emballer plus d'informations dans les coefficients d'approximation de la transformation en ondelettes. En rejetant les coefficients de détails, le  $WD$  obtenu à partir de l'ondelette possédant le nombre de moments nuls relativement faibles ne contient pas suffisamment d'informations sur le contour qu'il représente. Cela se traduit en termes de performance par une faible dégradation par rapport aux performances de  $WD$  obtenu à partir de l'ondelette possédant un nombre de moments nuls plus élevé.

### 5.3.3 Comparaison des $WDs$ basée sur différents atomes d'ondelettes

En introduction, nous avons noté l'avantage de la transformation en ondelettes par rapport à d'autres transformations est que les utilisateurs sont libres de choisir ou de construire son propre noyau (ou atome ou base) d'ondelette. Dans ce qui suit, nous allons appliquer plusieurs noyaux d'ondelettes les plus couramment utilisés, pour construire le descripteur  $WD$ , ensuite les descripteurs issus des différents noyaux seront évalués et comparés. Nous rappelons que chaque atome d'ondelettes a ses propres caractéristiques [MMOP03] : le nombre de moments nuls, le support compact, la symétrie, la régularité, orthogonale ou bi-orthogonale, discrète ou continue, etc. Le tableau 5.2 résume quelques propriétés de certaines familles d'ondelettes <sup>14</sup>.

Donc, afin de rendre la comparaison entre le  $WD$  issu des différents atomes possibles, nous

Atome	Support	Moment	Symétrie	Longueur de filtre
dbN	2N-1	N	non	2N
SymN	2N-1	N	presque	2N
CoifN	6N-1	2N	presque	6N
BiorNr.Nd	2Nd+1	Nr	possible	$\max(2Nr, 2Nd)+2$
RbioNd.Nr	2Nr+1	Nd	possible	$\max(2Nr, 2Nd)+2$

TAB. 5.2 – Propriétés des familles d'ondelettes

devons imposer des critères communs à tous ces atomes. Les critères que nous adoptons sont : la transformation doit être discrète et la longueur du support doit être inférieure ou égale à une constante. Ces deux critères nous conduisent à choisir les atomes suivants (dont la longueur

---

<sup>14</sup>L'ondelette  $db1$  ou  $sym1$  est simplement l'ondelette de Haar

Rappel	10	20	30	40	50	60	70	80	90	100	mean
db6	92.6	80.8	71.3	63.7	56.8	49.5	43.4	36.7	28.1	17.4	54.0
sym6	92.5	80.8	71.4	63.8	56.6	49.5	43.3	36.7	28.0	17.4	54.0
coif2	92.7	80.8	71.5	63.6	56.7	49.4	43.2	36.6	27.8	17.45	54.0
bior1.5	92.9	81.0	71.9	63.8	56.8	49.9	44.1	37.1	28.4	18.0	54.4
bior3.5	93.1	82.1	73.6	65.8	58.8	52.1	46.5	40.0	30.4	19.6	56.2
bior5.5	91.6	78.8	69.1	60.8	53.1	46.1	40.0	33.8	25.5	15.7	51.4
rbio1.5	92.4	80.0	70.0	62.1	55.1	48.2	42.1	35.3	27.0	16.9	52.9
rbio3.5	91.5	77.9	68.1	59.9	52.7	45.7	39.6	33.3	25.3	15.8	51.0
rbio5.5	92.9	82.2	73.3	66.1	59.0	52.2	46.4	39.6	30.3	19.7	56.2
dmey	90.5	76.5	65.7	57.7	50.9	44.4	38.9	33.0	25.2	14.9	49.8

TAB. 5.3 – Tableau de comparaison des performances en moyenne de la recherche en utilisant différents atomes d’ondelettes et en utilisant la distance maximum au centre de gravité pour fixer le point de départ. Les éléments de ce tableau sont des valeurs moyennes de Précision (P) calculées en utilisant 1400 requêtes.

du support est inférieure ou égale à 12 sauf dans le cas de l’approximation discrète de l’ondelette Meyer qui est de longueur 62) : l’ondelette de Daubechie (*db6*), Coiflets (*coif2*), Symlets (*sym6*), l’approximation discrète de l’ondelette Meyer (*dmey*), l’ondelette discrète Bi-orthogonale (*bior1.5*, *bior3.5* et *bior5.5*) et inverse bi-orthogonale (*rbio1.5*, *rbio3.5* et *rbio5.5*). La figure 5.5 et la figure 5.6 montre respectivement les fonctions d’échelle de ces ondelettes. L’ondelette inverse bi-orthogonale, comme son nom l’indique, correspond à l’inverse de l’ondelette bi-orthogonale, c’est-à-dire qu’on inverse le rôle des filtres de décomposition en filtres de reconstruction et vice-versa. Dans cette comparaison, nous avons utilisé la distance maximum au centre de gravité pour fixer le point de départ.

Les trois sous-graphes du graphe 5.7 montrent le résultat de performance de ces ondelettes. Le tableau 5.3 résume le résultat comparatif des performances des trois graphes de la figure 5.7. Chaque élément de ce tableau donne la valeur de Précision correspondant à la valeur de Rappel et à l’ondelette respectivement de la première ligne et de la première colonne.

### Interprétation

Selon ces trois graphes de ce tableau, on constate que les performances moyennes des cinq atomes *db6*, *sym6*, *coif2*, *bior1.5* et *rbio1.5* sont globalement les mêmes, les trois graphes correspondants aux trois atomes *db6*, *sym6* et *coif2* se chevauchent dans la figure 5.7(a). Nous constatons aussi que les deux atomes *bior3.5* et *rbio5.5* ont les mêmes performances et ils sont les meilleurs par rapport aux autres atomes. En revanche, la performance de *dmey* est la plus faible de tous les autres atomes utilisés dans notre expérimentation. Les performances des deux atomes *bior5.5* et *rbio3.5* sont également faibles mais elles sont supérieures à celle de *dmey*. En fait l’ondelette Meyer est une ondelette orthogonale indéfiniment dérivable mais elle n’est pas à support compact. L’approximation de l’ondelette Meyer par *dmey* nécessite un processus de troncature afin d’obtenir un nombre fini de coefficients. Cette troncature introduit des erreurs et elle crée une représentation sensible au bruit et aux déformations. Par conséquent, la performance du *WD* obtenue à partir de l’ondelette *dmey* est très faible par rapport aux autres *WD* utilisant un atome d’ondelette différent.

Sur la base de ces résultats, nous utilisons le noyau d'ondelette *rbio5.5* dans toutes les expériences qui suivent.

### 5.3.4 Comparaison des méthodes de fixation du point de départ

Rappel	10	20	30	40	50	60	70	80	90	100	mean
Max. distance	92.9	82.2	73.3	66.1	59.0	52.2	46.4	39.6	30.3	19.7	56.2
Max. courbure	91.7	78.7	68.7	59.3	50.9	44.3	36.8	29.4	20.3	11.4	49.2
Axe Principal	88.1	71.5	58.3	50.3	43.6	38.9	34.3	28.6	22.6	14.3	45.0
Axe Naturel	83.0	63.2	50.2	41.8	35.9	30.0	24.6	18.3	13.9	8.71	36.9

TAB. 5.4 – Tableau de comparaison de la performance en moyenne des quatre techniques pour fixer le point de départ. Les éléments de ce tableau sont des valeurs moyennes de Précision (P) calculées en utilisant 1400 requêtes

Le but de cette troisième expérience est d'étudier l'impact des techniques pour fixer le point de départ sur la performance du descripteur de forme. Nous rappelons que ces quatre techniques sont : la distance maximum au centre de gravité, la courbure maximum, l'axe principal et l'axe naturel. Ici dans notre implémentation, nous avons utilisé deux points de départ :

- Dans le cas de la première technique, on trie les points dans l'ordre décroissant par rapport à leur distance au centre de gravité. Les deux premiers points sont utilisés comme points de départ. Dans le cas de la courbure maximum, de manière similaire, on prend les deux premiers points comme point de départ si la courbure (en valeur absolue) en ces deux points est maximum.
- Dans le cas d'axe principal, c'est les deux points d'intersection du contour avec l'axe principal qui sont utilisés.
- Dans le cas d'axe naturel, si  $\theta$  est l'orientation d'axe naturel, c'est les deux points d'intersection entre le vecteur dirigé selon la direction d'angle  $\theta$  et  $-\theta$  avec le contour qui sont considérés comme point de départ.

Le résultat de cette comparaison est présenté dans la figure 5.8 et le tableau 5.4.

### Interprétation

Selon la figure 5.8 et le tableau 5.4, la technique utilisant la distance maximum au centre de gravité est plus robuste que les trois autres méthodes. Les explications sont : même en présence du bruit, si la puissance du bruit est relativement faible par rapport à la puissance du signal, ce bruit n'a pas vraiment d'impact sur le point où la distance est maximale par rapport au centre de gravité. En revanche, si la puissance du bruit devient plus importante par rapport à la puissance du signal, ce bruit peut entraîner éventuellement un changement de l'ordre des points entre le deuxième et le premier et cela n'a pas non plus d'impact sur le résultat de la reconnaissance car nous avons pris en considération les deux premiers points, où la distance au centre de gravité est maximale, comme point de départ.

Dans le cas de la courbure, pour les mêmes raisons que la distance maximum, cette technique est relativement robuste par rapport aux deux autres techniques, l'axe principal et l'axe naturel,

mais leur performance reste quand même inférieure par rapport à la performance de la technique utilisant le maximum distance car, on sait très bien que leur calcul est très sensible au bruit (dérivée seconde). La convolution avec le noyau Gaussien permet de réduire cette sensibilité, mais cependant cela reste insuffisant.

Enfin, selon la figure 5.8 et le tableau 5.4, on constate que la technique utilisant l'axe principal est plus robuste que la technique utilisant l'axe naturel. Il est difficile d'expliquer ce résultat de façon plus rigoureuse, mais d'une manière simpliste, si on regarde de très près les équations 2.16, 2.21 et 2.22 on constate que les composants du bruit sont beaucoup plus amortis dans les deux premières équations et un peu moins dans la dernière.

Une autre façon d'expliquer le résultat de la performance de la distance maximum au centre de gravité ou de la courbure maximum, pour fixer le point de départ par rapport aux deux autres techniques, est de raisonner en termes du nombre de points qui contribuent au calcul de la position du point de départ. De ce point de vue, on constate que dans le cas de la distance maximum ou courbure maximum, seul un point détermine la position du point départ alors que dans les deux autres techniques, l'axe principal et l'axe naturel, tous les points sont utilisés pour déterminer la position du point de départ. Cela veut dire que la moindre perturbation sur un point du contour se répercutera sûrement (instabilité) sur la position du point de départ.

Sur la base de ces résultats, pour fixer le point de départ, nous utilisons la distance maximum au centre de gravité dans toutes les expériences qui suivent.

## 5.4 Evaluations et comparaisons des descripteurs invariants à la transformation similitude

Dans le but de comparer le descripteur que nous avons proposé avec les autres descripteurs, nous allons d'abord comparer et évaluer trois descripteurs invariants à la transformation similitude. Ces trois descripteurs sont le descripteur de Fourier (noté  $FD$ ), le descripteur de la transformation de Fourier à fenêtre (noté  $SFD$ ) et le descripteur d'espace-échelle de courbure (noté  $CSSD$ ). Nous commencerons par comparer plusieurs  $FD$ , ensuite nous comparerons le descripteur de Fourier le plus robuste obtenu lors de la première comparaison avec  $SFD$  et  $CSSD$ . Pour finir la section, nous présenterons enfin l'étude comparative des performances de notre descripteur  $WD$  et le  $FD$ .

### 5.4.1 Comparaison des $FDs$

Dans la littérature de la reconnaissance de formes basée sur le contour, il existe de nombreux  $FDs$ , la majorité d'entre eux diffèrent principalement sur la représentation du contour. Dans cette section, nous développons et nous comparons quatre  $FDs$ , le  $FD_1$  basé sur la représentation du contour par la distance au centre de gravité (section 2.2.4), le  $FD_2$  basé sur la représentation du contour par les nombre complexes (section 2.2.1), le  $FD_3$  basé sur la représentation du contour par la courbure (section 2.2.6) et le  $FD_4$  basé sur la représentation du contour par la surface fermée (section ). Les techniques employées pour obtenir ces descripteurs sont présentées dans la section 3.3.4, hormis le cas de la représentation complexe, on ne peut pas normaliser par le coefficient DC  $a_0$  car il provoquera une division par zéro. Pour éviter cela, on remplace le

coefficient DC  $a_0$  de l'équation 3.13 par la distance moyenne au centre de gravité  $\hat{r}$  :

$$\hat{r} = \sum_{i=1}^n \sqrt{(x_k - x_c)^2 + (y_k - y_c)^2}$$

avec  $x_c, y_c$  sont définis dans l'équation 2.2. Les vecteurs descripteurs de  $(FD_i)_{1 \leq i \leq 4}$  sont de dimension 15, il s'agit des  $(b_j)_{1 \leq j \leq 15}$  définis par l'équation 3.13. Les résultats de la recherche, en utilisant la troisième sous-figure de la figure 5.1 comme requête, et de la comparaison des quatre descripteurs sont montrés dans la figure 5.9. Dans cette figure, les mesures de performance P/R sont calculées de la même manière que les comparaisons dans les sections 5.3.3 et 5.3.4. Pour chaque requête, on calcule les valeurs de P qui correspondent aux valeurs de R de 10% à 100%. Les valeurs de P dans la figure 5.9 sont des valeurs moyennes des valeurs de P sur 1400 requêtes.

### Interprétation

Le résultat des comparaisons ci-dessus nous amène à la même conclusion que celle obtenue par D. Zhang et al. [ZL02][Zha02] : le descripteur obtenu à partir de la représentation du contour par la distance au centre de gravité  $FD_1$  est plus robuste que les autres descripteurs de Fourier. Pour comprendre ce résultat, nous tentons de donner l'explication à travers l'analyse de chacune des représentations des contours.

Nous commençons d'abord par la représentation du contour par la courbure. Nous savons que pour calculer la courbure, nous devons calculer la dérivée seconde et dans le cas discret, cette dérivée seconde peut être calculée à partir de trois échantillons consécutifs. Dans le cas de la représentation du contour par la surface fermée, d'après l'équation 2.6, le calcul de cette représentation à chaque position nécessite la connaissance a priori des coordonnées de deux points consécutifs. Par contre, le calcul de la représentation complexe et de la représentation par la distance au centre de gravité à chaque position nécessitent la connaissance des coordonnées d'un seul point. Donc, cela signifie qu'il est normal que la sensibilité aux variations locales ou au bruit soit beaucoup plus importantes dans le cas de la représentation par la courbure que dans le cas de la représentation par la surface fermée et aussi importante dans le cas de la surface fermée que dans le cas de la distance au centre de gravité et de la représentation complexe. De plus, entre la représentation complexe et la représentation par la distance au centre de gravité, il est facile de voir que la représentation complexe est plus sensible que la représentation par la distance au centre de gravité car si on change la valeur de  $x$  en  $y$  et de  $y$  en  $x$  ou bien si on change la valeur  $x$  en  $-x$  ou  $y$  en  $-y$ , on constate que le nombre complexe  $\pm x \pm iy$  et  $\pm y \pm ix$  n'est pas, en général, le même, par contre on a l'égalité  $|\pm x \pm iy| = |\pm y \pm ix| = \sqrt{x^2 + y^2}$ .

D'autre part, selon l'équation 3.11, tous les coefficients de Fourier dépendent des valeurs du signal en entrée. Toutes les perturbations sur les échantillons du signal en entrée entraînent également des perturbations sur les coefficients de Fourier. Tout cela explique pourquoi il n'est pas surprenant qu'on obtienne le résultat observé dans la figure 5.9, c'est à dire que le descripteur de Fourier  $FD_1$  obtenu à partir de la représentation par la distance au centre de gravité est plus robuste que les autres descripteurs de Fourier.

Il faut préciser que pour ce résultat, nous n'avons pas effectué la paramétrisation, ni de ré-échantillonnage uniforme des contours comme c'est le cas dans les deux descripteurs basés sur la transformation en ondelettes  $WD$  et  $AWD$  que nous avons proposée. Lorsque nous avons intégré



ces deux étapes intermédiaires dans le processus de calcul des descripteurs, nous avons constaté que le gain en performance est globalement très faible, moins de 1%, par rapport aux efforts en termes de temps de calcul, même en faisant varier le nombre d'échantillons.

### 5.4.2 Comparaison des descripteurs *FD*, *SFD* et *CSSD*

Dans cette section, nous reportons le résultat (figure 5.10) d'évaluations et de comparaisons entre le *FD* et les deux autres descripteurs *SFD* (section 3.4), *CSSD* (section 3.5.3) et *CSSD*<sup>+</sup> (section 3.5.4), de D. Zhang et al dans ses publications [ZL01], [ZL03a] et [Zha02]. Le dispositif utilisé pour évaluer et comparer ces descripteurs est le même que celui que nous avons utilisé dans la section précédente.

La figure 5.10(a) montre que le *FD* est plus performant que le *SFD*. Ceci s'explique par le fait que le *SFD* est une transformation de Fourier appliquée sur une partie du contour (la fenêtre), d'où le *SFD* peut capturer les caractéristiques locales du contour d'une manière plus précise, cependant il ne capture pas les caractéristiques globales du contour qui sont importantes pour le représenter. D'autre part, la distortion locale affecte tous les coefficients de Fourier, mais les effets les plus significatifs sont principalement localisés sur les coefficients de hautes fréquences. L'avantage de *SFD* par rapport au *FD* est sa potentielle application dans la recherche partielle des formes ou d'objets occultés ce qui est un sujet au delà de la portée de notre investigation.

La comparaison entre le *FD* et le *CSSD*<sup>+</sup> a été déjà réalisé par S. Abbasi et al. dans l'article [SAK00], cependant cette comparaison a plusieurs limitations :

1. Seule la mesure de Précision est calculée
2. La base utilisée n'est pas classée
3. L'évaluation n'est pas impartiale
4. Le *FD* utilisé n'est pas le *FD* le plus robuste

En tenant compte de ces limites, D. Zhang et al., [ZL03a], proposent une nouvelle comparaison beaucoup plus rigoureuse par rapport à celle qui a été faite dans [SAK00]. La figure 5.10(b) illustre une partie des résultats de cette comparaison. Contrairement à ce qui a été présenté dans [SAK00], cette figure montre que le *FD* est plus robuste que le *CSSD*<sup>+</sup>.

En résumé et en se basant sur le graphique de la figure 5.10, on peut dire que le *FD* est plus robuste que le *SFD*, le *CSSD* et le *CSSD*<sup>+</sup>. Puisque le *CSSD*<sup>+</sup> a été sélectionné par le groupe MPEG7 comme descripteur robuste basé sur le contour, cela montre très bien que le *FD* est un candidat potentiel pour faire partie de l'ensemble des descripteurs robustes de la norme MPEG7.

### 5.4.3 Comparaison du *WD* et du *FD*

Nous avons vu dans la section précédente que le *FD* est plus robuste que le *SFD* et le *CSSD*<sup>+</sup>. Maintenant, il est important de comparer la performance entre le *WD* que nous avons proposé et le *FD*. Pour effectuer cette comparaison, nous utilisons l'atome d'ondelette et la technique de fixation du point de départ donnant la meilleur résultat en terme de performance pour le *WD*, c'est à dire qu'on utilise l'atome *rbio5.5* et la distance maximum au centre de gravité pour

Rappel	10	20	30	40	50	60	70	80	90	100	mean
FD	93.0	81	69.5	64.3	57.8	53.4	47.4	42.9	33.8	24.5	56.7
WD	92.9	82.2	73.3	66.1	59.0	52.2	46.4	39.6	30.3	19.7	56.2

TAB. 5.5 – Tableau de comparaison de la performance en moyenne de notre méthode et de la méthode de Fourier. Les éléments de ce tableau sont des valeurs moyennes de Précision (P) calculées en utilisant 1400 requêtes

fixer le point de départ. De la même manière, pour être objectif dans la comparaison, on utilise le *FD* qui donne également la meilleure performance en terme de P/R, plus précisément, on utilise le *FD* obtenu à partir de la représentation du contour par la distance au centre de gravité.

Le tableau 5.5 et la figure 5.11 montrent le résultat de cette comparaison. Selon le tableau 5.5 et la figure 5.11, les deux descripteurs ont le même performance, de plus ils ont de nombreux points communs :

- Les deux descripteurs sont robustes par rapport au bruit. Le *FD* utilise les coefficients de faibles fréquences qui représentent essentiellement les structures globales qui sont robustes à l'irrégularité du contour. Les bruits sont éliminés à travers les coefficients de hautes fréquences. Pour le *WD*, les bruits sont filtrés lors de décomposition au niveau un et deux
- Les deux descripteurs sont indépendants à l'application, ils ne nécessitent pas de connaissance a priori sur le type de contour
- Les deux descripteurs sont compacts et leurs dimensions sont constantes, quinze pour le *FD* et seize pour le *WD*
- La complexité liée au temps de calcul de chacun des deux descripteurs est la même, elle est de l'ordre  $O(N)$ . En fait, la complexité de la transformée de Fourier rapide est de l'ordre  $O(N \log N)$ , mais puisque qu'on s'intéresse uniquement aux quinze premiers coefficients, on peut appliquer directement la formule de l'équation 3.11 pour calculer le *FD*, c'est pour cette raison que la complexité de *FD* est de l'ordre  $O(N)$ .

Cependant, les deux descripteurs sont différents sur les points suivants :

- Domaine du descripteur : Le *WD* représente les informations à la fois dans le domaine spatial et dans le domaine fréquentiel, alors que *FD* représente les informations uniquement dans le domaine fréquentiel
- Atome d'analyse. : L'avantage de *WD* par rapport à *FD* est qu'il est possible de changer d'atome d'ondelette pour calculer le *WD*, ce qui n'est pas le cas pour le *FD*. De plus, pour certaines familles d'ondelettes, il est possible de choisir le nombre de moments nuls associés à l'ondelette  $\psi$ . La flexibilité du choix des atomes et du nombre de moments nuls permet, dans certains cas, de contrôler l'efficacité du résultat de classification.

## 5.5 Evaluation du descripteur affine

Dans cette section, nous passons maintenant aux expérimentations concernant notre descripteur affine *AWD*. L'extraction de *AWD* est présentée dans l'algorithme 6. Pour pouvoir appliquer cet algorithme, nous utilisons le ré-échantillonnage en  $N = 512$  points et nous calculons le vecteur *AWD* au niveau  $l = 6$ , cela signifie que les vecteurs *AWD* sont de dimension  $2^{9-6} = 8$ . Pour supprimer la redondance, nous utilisons la séquence  $(\varepsilon_i = 0)_{1 \leq i \leq 6}$ . En ce qui concerne la base

des contours, puisque la sous-base C de la base CE-1 MPEG7 n'est pas classée, nous créons trois nouvelles bases  $DB_1$ ,  $DB_2$  et  $DB_3$  qui sont des transformations affines des 500 contours de la base C. Ces transformations affines sont composées de la rotation d'angle  $\theta = \frac{k\pi}{9}$ ,  $k = 1, 2, \dots, 9$ , de l'étirement de facteurs  $a$ ,  $s$ . Nous rappelons que la matrice d'étirement a la forme générale suivante :

$$\begin{bmatrix} a & s \\ 0 & 1/a \end{bmatrix}$$

Dans toutes nos expérimentations, nous avons fixé  $a = 1$  pour l'étirement. De plus pour la base  $DB_1$  nous fixons la valeur de  $s$  à 1, puis à 2 et à 3 pour la seconde et troisième base  $DB_2$  et  $DB_3$ . Donc pour chacune de ces trois bases, nous avons 5000 contours ( $5000 = 500 + 9 \cdot 500$ ) et les trois bases contiennent en tout 15000 contours.

Avant de commencer l'étude de la performance de *AWD*, nous montrons quelques exemples concrets en prenant les contours de la figure 5.12(a) comme requête. Dans cette figure, le contour  $P_1$  est le contour d'origine, le contour  $P_2$  est une rotation d'angle  $\pi/4$  de  $P_1$  suivi d'un étirement de facteur  $s = 0.5$ . Le contour  $P_3$  est une rotation d'angle  $\pi/2$ , suivi d'un étirement de facteur  $s = 1.5$  et d'un changement d'échelle non homogène de 0.5 et 1.5 respectivement selon l'axe  $X$  et l'axe  $Y$  de  $P_1$ . Et enfin, pour  $P_4$ , nous avons ajouté artificiellement des bruit uniformes sur l'intervalle  $[-1, 1]$  au contour d'origine  $P_1$ . En passant  $P_1$  comme requête au système de reconnaissance en utilisant le descripteur *AWD*, les neuf premiers résultats de la recherche dans la base  $DB_1$  sont présentés dans la figure 5.12(b). Ces résultats prouvent que le *AWD* donne le meilleur résultat en terme de reconnaissance des formes et il est robuste à la transformation affine.

Nous avons relancé le programme de recherche avec la même requête, mais cette fois ci avec la base  $DB_2$  et la base  $DB_3$ , nous constatons que le système de reconnaissance, que nous avons développé, nous renvoie les mêmes résultats que ceux obtenus dans le cas de la base  $DB_1$ . La même expérimentation a été conduite pour la requête  $P_2$  et  $P_3$  sur chacune des bases  $(DB_i)_{1 \leq i \leq 3}$ , les résultats obtenus à ces deux requêtes  $P_2$  et  $P_3$  restent toujours les mêmes que dans le cas de la requête  $P_1$ . Cela signifie que, la transformation affine n'a pas d'impact sur le *AWD* ou autrement dit, *AWD* est invariant par rapport à la transformation affine. Nous avons continué l'expérimentation en utilisant  $P_4$  comme requête, les neuf premiers résultats à  $P_4$  sont donnés dans la figure 5.12(c). En fait, ces résultats ne correspondent pas exactement à la version affine de  $P_1$ , mais les neuf résultats suivants sont bien des versions affine de  $P_1$ . La présence du bruit sur le contour introduit des perturbations sur le vecteur descripteur *AWD*. Ces perturbations dépendent de la puissance du bruit : plus la quantité du bruit est importante, plus le vecteur descripteur *AWD* est corrompu, et par conséquent, il est tout à fait normal qu'au delà d'un certain seuil, le vecteur descripteur *AWD* ne représente plus le contour non bruité et c'est la raison pour laquelle, les neuf premiers résultats de la requête  $P_4$  ne correspondent pas à la version affine de  $P_1$ .

Intéressons nous maintenant à l'étude des performances de l'*AWD*. D'abord, nous utilisons le taux de succès pour évaluer cette étude. Ce taux est défini par :

$$\text{le taux de succès d'une requête} = \frac{m}{\min(n, m_{max})} \times 100$$

avec  $n$  le nombre de résultat utilisés pour évaluer la performance du système,  $m$  est le nombre de résultats qui correspondent exactement à la transformation affine de la requête ( $m \leq n$ ) et  $m_{max}$  la valeur maximum possible de  $m$  dans la base, c'est à dire le nombre de contours de la

base qui sont des transformations affines de la requête.

La figure 5.13(a) montre que le taux de succès évalué en utilisant la base  $DB_1$  pour les valeurs de  $n$  variant de  $n = 1$  à  $n = 40$ <sup>15</sup>. Ce taux est calculé en utilisant la moyenne des taux de succès de 500 requêtes obtenues à partir de la transformation affine, générées aléatoirement, sur les 500 contours de la base C qui sont utilisés pour créer les trois bases  $(DB_i)_{1 \leq i \leq 3}$ .

Une deuxième étude d'évaluation de  $AWD$  concerne la sous-base B de la base CE-1 MPEG7, nous utilisons la mesure de performance  $P/R$  comme ce que nous avons déjà fait, c'est à dire que pour chaque valeur de  $R$ , on calcule la valeur de  $P$  correspondante. La valeur de  $P$  reportée sur le graphique correspond à la valeur moyenne de  $P$  sur 1400 requêtes. La figure 5.13(b) montre le résultat de cette évaluation. Il faut distinguer la figure 5.13(b) de la figure 5.13(a), la figure 5.13(b) utilise la mesure de performance  $P/R$ , alors que la figure 5.13(a) utilise le taux de succès pour mesurer la performance. De plus l'évaluation de la figure 5.13(a) se fait sur la base de synthèse générée en appliquant la transformation affine sur les 500 contours des poissons de la sous base C de la base CE-1 MPEG7, alors que l'évaluation de la figure 5.13(b) se fait sur la base d'objets réels.

## 5.6 Conclusion

Dans ce chapitre nous avons mené une évaluation du descripteur  $WD$  basé sur la transformation en ondelettes que nous avons proposé. Cette évaluation montre que la performance de  $WD$  varie selon l'atome d'ondelette utilisé, selon le nombre de moments nuls de l'ondelette et aussi la technique utilisée pour fixer le point de départ. Il montre également que plus le nombre de moments nuls augmente meilleur sera la performance, mais au dépend du temps de calcul. Ce résultat montre aussi que parmi les quatre techniques pour fixer le point de départ : la distance maximum au centre de gravité, la courbure maximum, l'axe principal et l'axe naturel, la distance maximum au centre de gravité est plus robuste que les trois autres techniques.

Nous avons également étudié la performance du descripteur de Fourier  $FD$  issu des différentes représentations. Le résultat de cette étude montre que le  $FD$  obtenu à partir de la représentation utilisant la distance au centre de gravité est plus robuste que le  $FD$  obtenu à partir d'autres représentations. Ce résultat confirme la conclusion d'expérimentation effectuée par Zhang et al. [ZL02][Zha02]. Nous avons également étudié la comparaison entre le  $WD$  avec le  $FD$ , le descripteur basé sur la transformation de Fourier à fenêtre  $SFD$  et le descripteur basé sur la courbure multi-échelle  $CSSD$ . Le critère de cette évaluation comparative est basé sur la mesure de performance Précision/Rappel ( $P/R$ ) et la base utilisée pour l'évaluation est la sous-base B de la base CE-1 MPEG7. Bien que le  $CSSD$  soit considéré comme un descripteur robuste par le groupe MPEG7, cette comparaison montre que les performances de  $WD$  et  $FD$  sont les mêmes et sont meilleurs que celle de  $CSSD$  qui sont elles mêmes meilleures que celles de  $SFD$ .

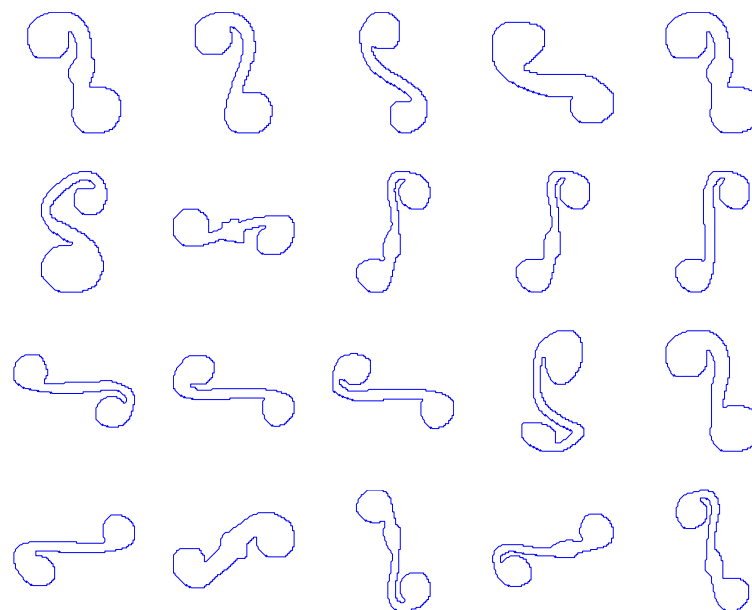
Dans le cas du descripteur en ondelette invariant à la transformation affine  $AWD$  que nous avons proposé, les expérimentations montrent que, pour la base de synthèse obtenue en appliquant la transformation affine sur la sous base C de la base CE-1 MPEG7, le taux de succès est excellent (environ 99%), par contre la performance en terme de Précision/Rappel pour la base réelle (la sous base B de la base CE-1 MPEG7) est plutôt modeste.

---

<sup>15</sup>Nous obtenons le même résultat pour la base  $DB_2$  et  $DB_3$

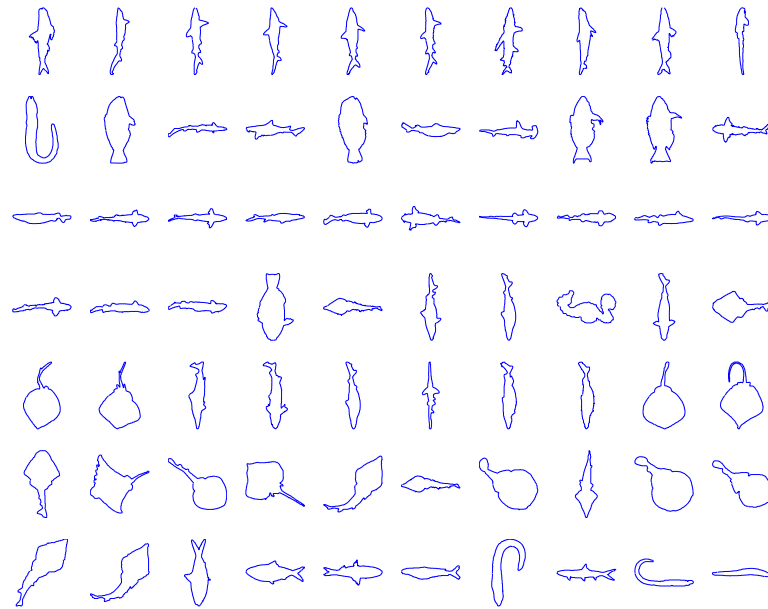


(a)

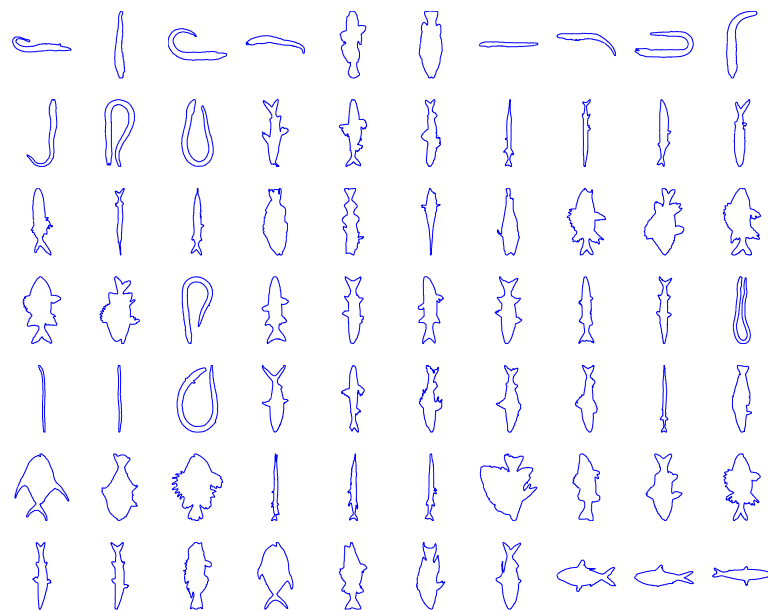


(b)

FIG. 5.1 – (a) 70 contours qui représentent 70 classes (b) 20 contours de la même classe. Tous ces contours font partie de la sous-base B de la grande base CE-1 MPEG-7

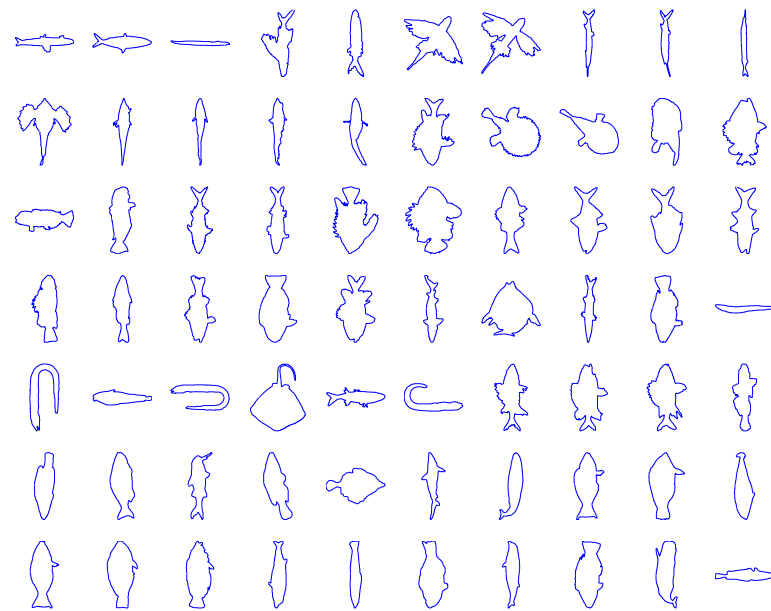


(a)

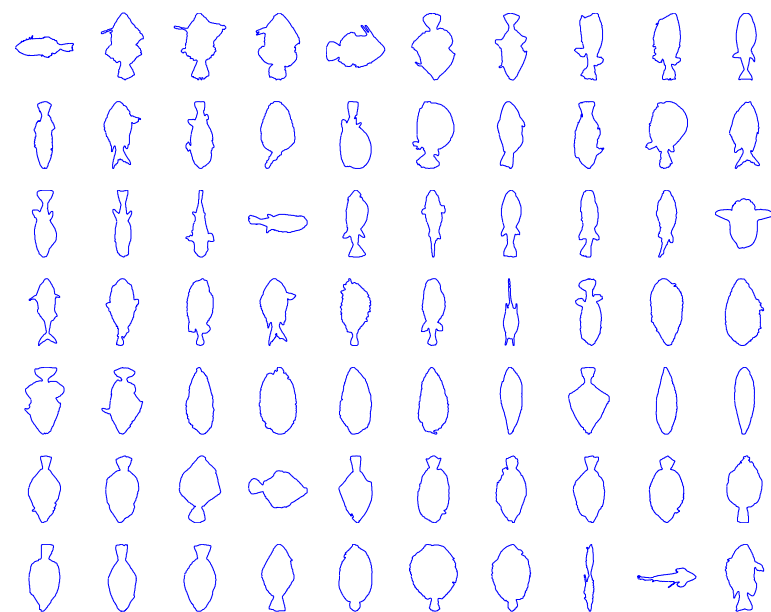


(b)

FIG. 5.2 – Les 140 premiers contours de la sous-base C de la grande base CE-1 MPEG7



(a)



(b)

FIG. 5.3 – Les 140 contours suivants de la sous-base C de la grande base CE-1 MPEG7

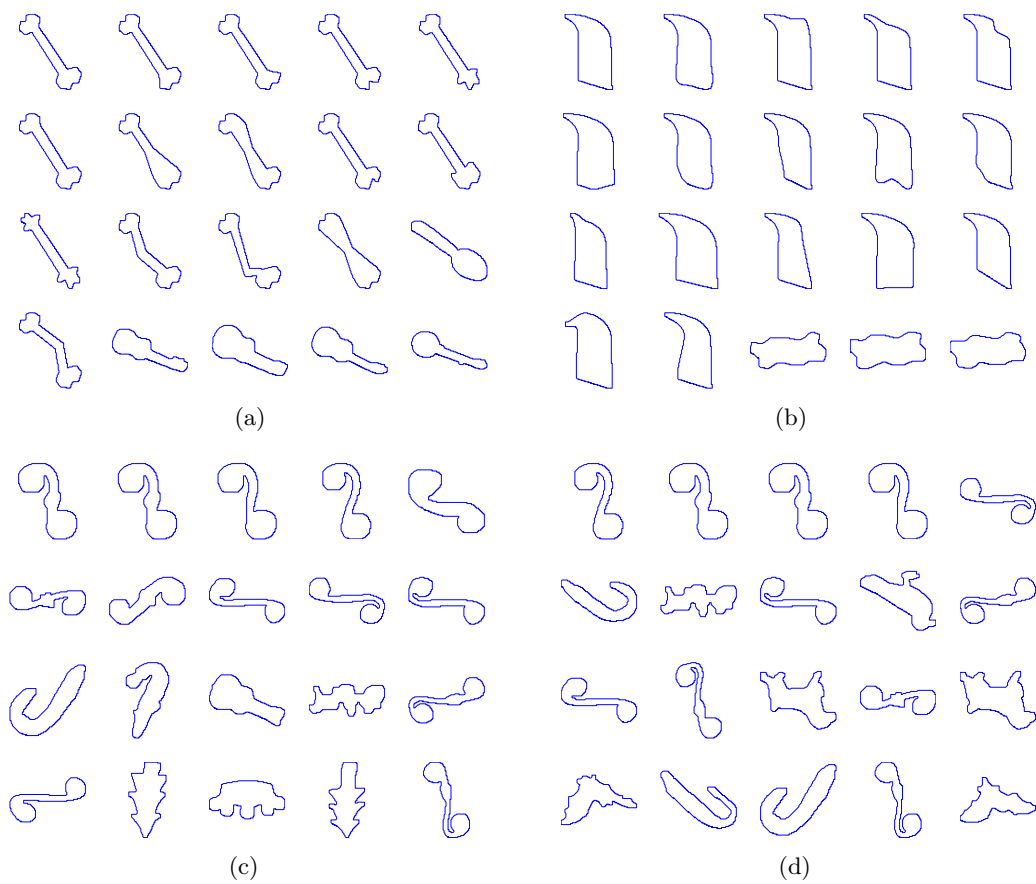


FIG. 5.4 – Les figures (a) et (b) représentent les vingt premières réponses aux deux premières requêtes de la figure 5.1a. Les figure (c) et (d) représentent les vingt premières réponses aux deux premières requêtes de la figure 5.1b



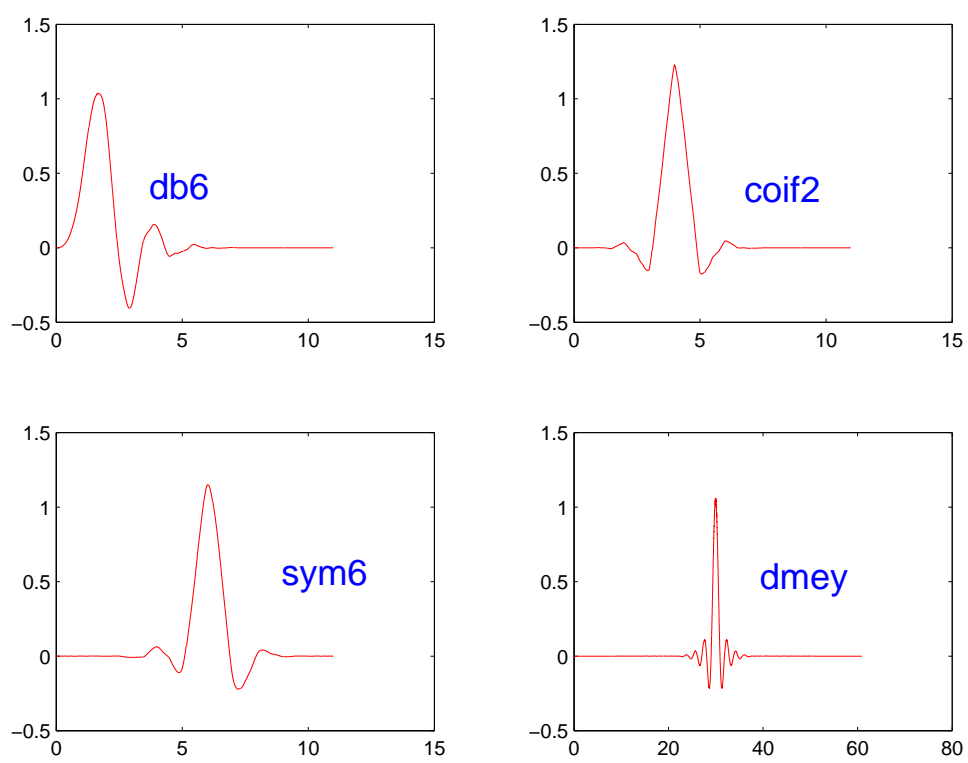


FIG. 5.5 – La fonction d'échelle d'ondelette db6, coif2, sym6 et dmey

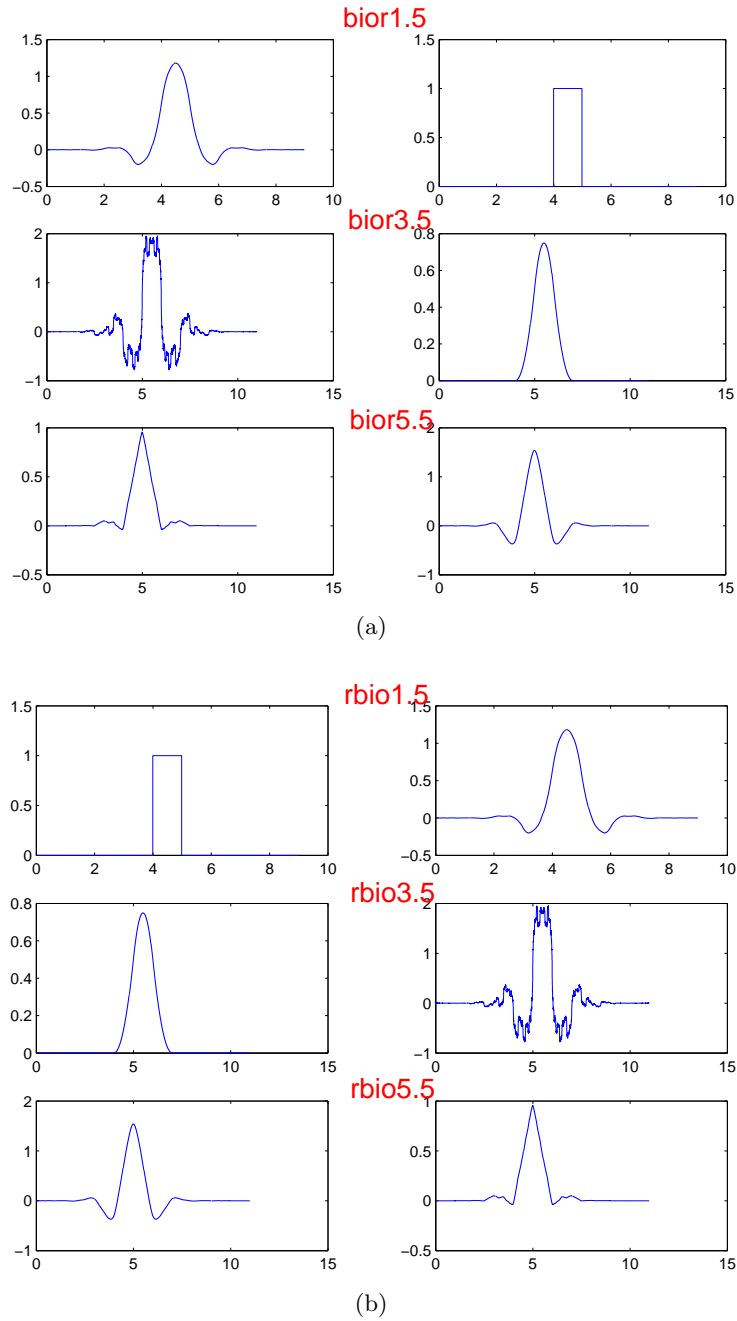


FIG. 5.6 – La fonction d'échelle de décomposition (à gauche) et de reconstruction (à droite) d'ondelette (a) bior1.5, bior3.5 et bior5.5 (b) rbio1.5, rbio3.5 et rbio5.5

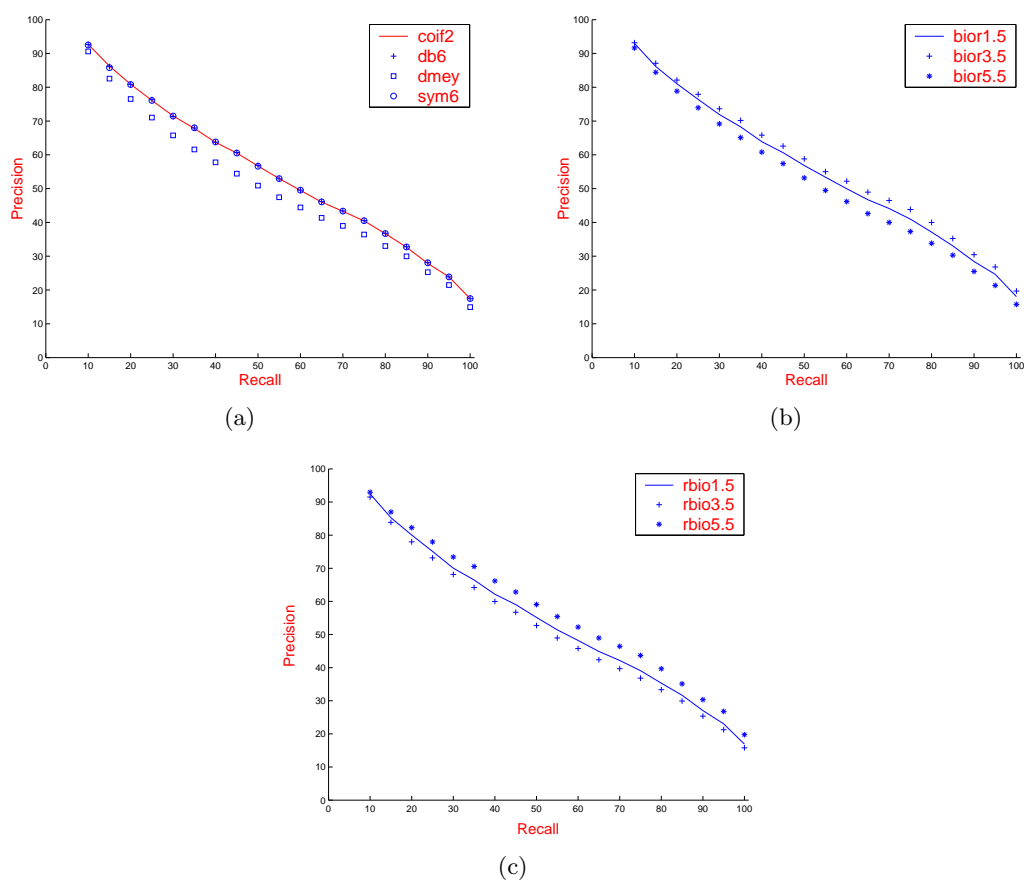


FIG. 5.7 – La comparaison de la performance en moyenne de la recherche en utilisant différents atomes d'ondelettes (a) Coif2, db6, dmey, sym6 (b) bior1.5, bior3.5 et bior5.5 (c) rbio1.5, rbio3.5 et rbio5.5

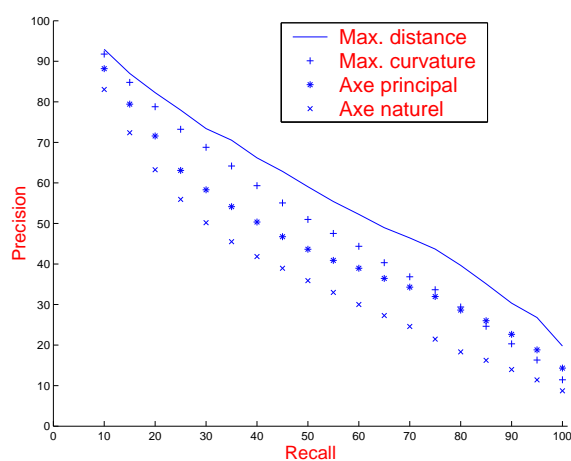


FIG. 5.8 – La comparaison de la performance de la recherche entre les quatre techniques pour fixer le point de départ

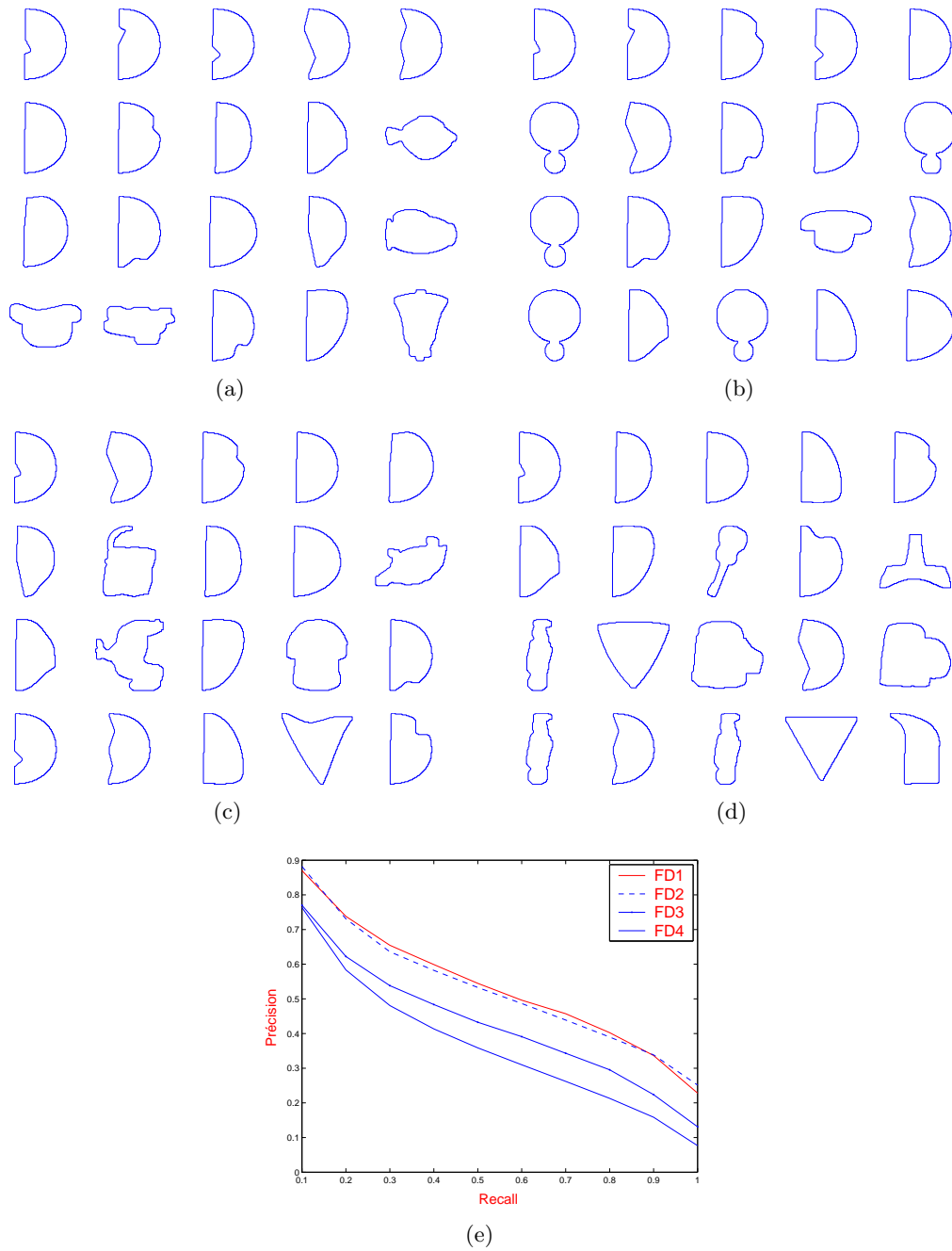
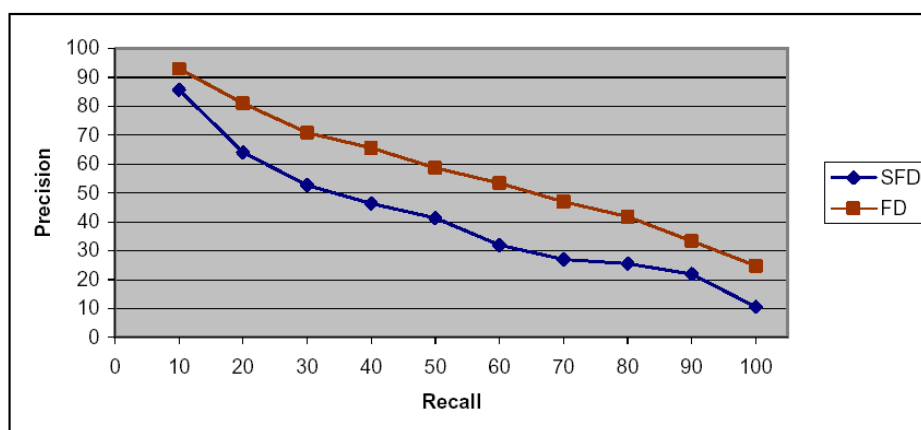
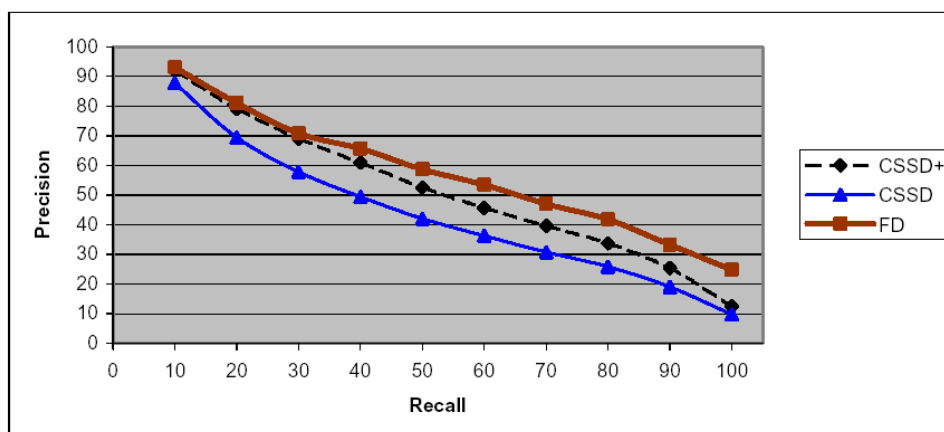


FIG. 5.9 – Les vingt premières réponses à la requête de la quatrième sous-figure de la figure 5.1a en utilisant (a)  $FD_1$  (b)  $FD_2$  (c)  $FD_3$  (d)  $FD_4$ . La figure (e) donne le résultat de comparaison des quatre descripteurs de Fourier  $FD_1$ ,  $FD_2$ ,  $FD_3$  et  $FD_4$



(a)



(b)

FIG. 5.10 – Les valeurs moyennes de Précision et Rappel de 1400 requêtes utilisant (a) Le descripteur de Fourier et le descripteur de Fourier à fenêtre glissante (b) Le descripteur de Fourier et le descripteur d'espace-échelle de courbure sur la sous-base B de la base CE-1 MPEG7 [Zha02]

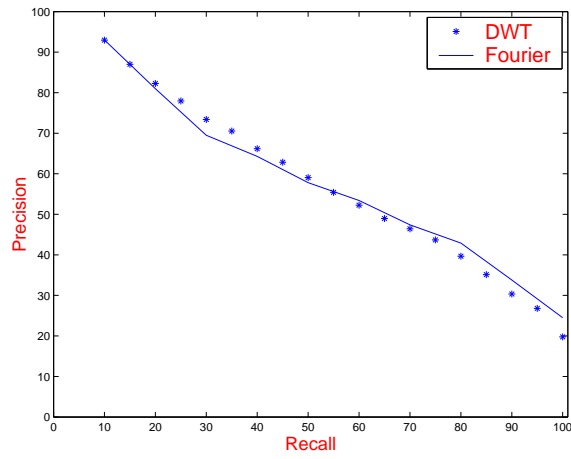
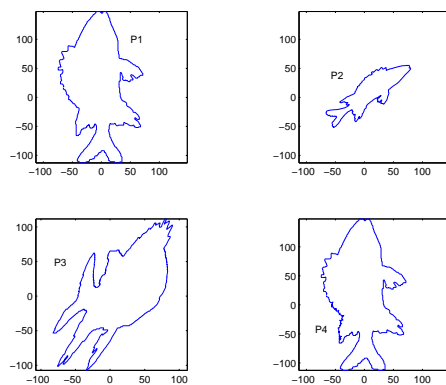
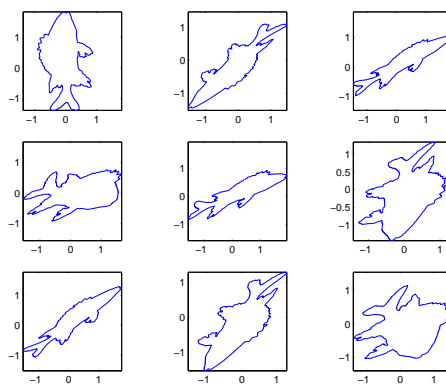


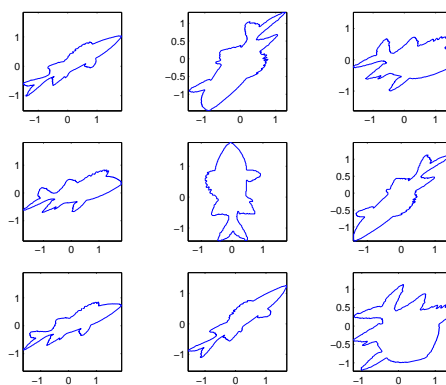
FIG. 5.11 – La comparaison de la performance de la recherche entre  $WD$  et  $FD$



(a)



(b)



(c)

FIG. 5.12 – (a) Les contours de requête,  $P_1$  est le contour origine,  $P_2$  et  $P_3$  sont des transformations affines de  $P_1$  et  $P_4$  est le contour  $P_1$  avec un bruit uniforme entre  $[-1,1]$  (b) les résultats de la requête  $(P_i)_{1 \leq i \leq 3}$  (c) les résultats de la requête  $P_4$

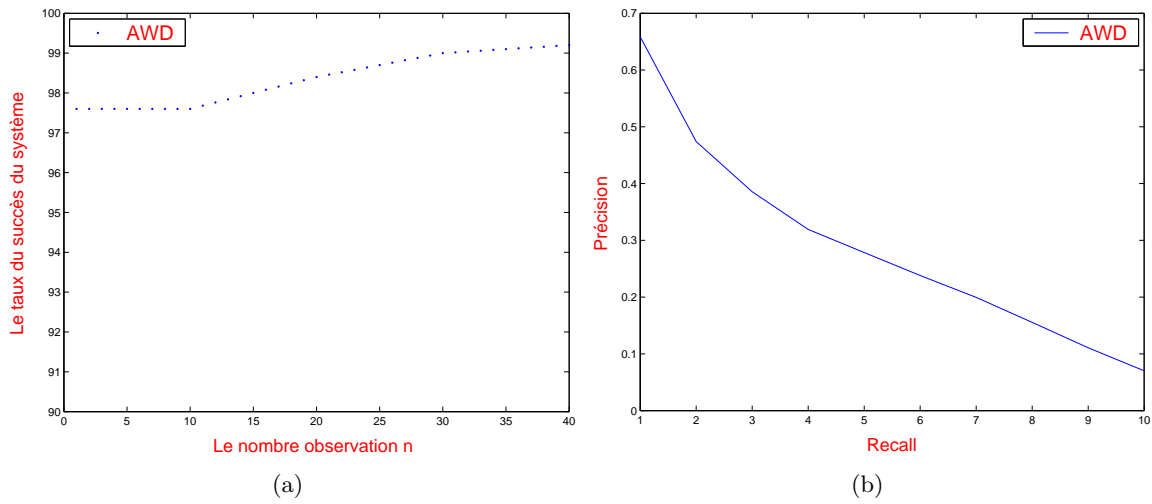


FIG. 5.13 – (a) Le taux de succès de *AWD* pour la base  $(DB_i)_{1 \leq i \leq 3}$  (b) La mesure de performance P/R sur la sous-base B de la base CE-1 MPEG7



# Chapitre 6

## Conclusion Générale

Dans ce travail de thèse, nous nous sommes intéressés principalement à la description des formes basée sur le contour dans le contexte des applications de recherche d'images par le contenu. Dans ce contexte, nous avons proposé, développé puis évalué de nouveaux descripteurs. Nous avons pris en compte les aspects liés à l'efficacité de la recherche en utilisant la mesure de performance Précision/Rappel. Cette mesure nous permet de comparer la performance de nos descripteurs avec certains descripteurs existants.

Dans cette conclusion générale, nous présentons tout d'abord une synthèse des études et des contributions proposées, puis nous esquissons plusieurs perspectives dans le prolongement de cette étude.

### 6.1 Synthèse des travaux effectués et apport de la thèse

Dans cette thèse, nous avons essayé d'apporter notre contribution à la problématique de la reconnaissance des formes basée sur le contour. La première partie de nos travaux concerne l'étude des descripteurs de forme existants (chapitre 3) et des principaux outils mathématiques associés permettant de calculer ces descripteurs (chapitre 2). Nous nous sommes plus particulièrement intéressés aux descripteurs *FD* et *CSSD*.

La seconde partie de ce travail, chapitre 4, a concerné l'étude de la théorie liée à la transformation en ondelettes. En dehors du désir d'explorer de nouveaux outils ou de nouveaux matériels comme celui de la transformation en ondelettes, les motivations qui nous amènent de la première partie vers cette deuxième partie sont multiples :

- Possibilité de changer le noyau d'ondelette
- Possibilité d'adapter le nombre de moments nuls associés à l'ondelette d'analyse
- La transformation en ondelettes est une transformation permettant d'analyser le signal localement et/ou globalement
- En fonction des objectifs du traitement par ondelettes, on peut préférer la transformée continue à la transformée discrète si la redondance peut être mise à profit pour analyser le signal. On fait le choix contraire si une compression du signal est souhaitée

Cette étude nous a permis de proposer des contributions suivantes :

- Nous avons proposé un *nouveau descripteur simple*, le *WD*, invariant à la transformation similitude en utilisant la transformation en ondelettes discrètes
- Nous avons proposé d’exploiter l’histogramme de la moyenne d’énergie des signatures de la base pour sélectionner automatiquement le niveau de décomposition qui est indispensable à la transformation en ondelettes discrètes. La connaissance du niveau de décomposition est importante, elle nous permet de choisir l’ensemble des coefficients à retenir pour représenter les contours
- Nous avons également proposé un *nouveau descripteur*, le *AWD*, invariant à la transformation affine. Contrairement aux descripteurs affines proposés dans [KB01][KB02], le descripteur que nous proposons n’est pas redondant

Dans la dernière partie, le chapitre 5, nous avons enfin présenté l’étude expérimentale pour évaluer les descripteurs que nous avons proposés et pour comparer la performance de notre descripteur *WD* à celle de certains descripteurs existants. Le résultat de cette comparaison montre que le *WD* a les mêmes performances que celles de *FD* mais meilleures que celles du *CSSD* qui est pourtant considéré, par le groupe MPEG7, comme descripteur de forme basé sur le contour le plus robuste.

## 6.2 Perspectives

L’approche que nous avons développée doit être considérée comme une première réalisation. De nombreuses extensions sont possibles. Nous souhaitons axer nos recherches futures sur différents points intervenant dans les trois parties dans cette thèse.

Le premier point concerne l’étude et l’analyse d’autres façons de représenter les contours par les coefficients d’ondelettes. En d’autres termes, il est possible de représenter les contours par d’autres types d’invariants obtenus à partir des coefficients d’ondelettes et dans ce cas une étude comparative devra être menée afin de savoir s’il est meilleur que ce que nous proposons actuellement.

Une autre extension possible à ce travail peut être menée pour exploiter la propriété multi-échelle de la transformation en ondelettes. L’idée est de donner plus d’importance aux coefficients d’ondelettes à la résolution grossière. Plus la résolution augmente, plus l’importance diminue. Il faut souligner qu’une fois qu’on est capable de bien maîtriser la façon d’accorder le degré d’importance à tous les coefficients d’ondelettes, il sera possible d’étendre cette idée au cas du descripteur multi-échelle ou multi-résolution en générale. Par exemple, on peut étendre cette idée dans le cas du descripteur *FD* ou *CSSD*.

Nous pouvons aussi envisager de continuer notre travail dans le cas de la recherche d’image par le contenu mais dans le domaine compressé car dans ce domaine, les ondelettes constituent une méthode très compétitive. La nouvelle norme de compression d’images JPEG 2000 est basée sur l’utilisation des ondelette. Elles permettent, grâce à des représentations généralement très creuses, de réduire considérablement le nombre d’informations à coder.

La dernière perspective est de continuer notre recherche dans le domaine de l’indexation. En fait, un système de recherche d’images par le contenu (SRIC) efficace nécessite également des techniques d’indexation multidimensionnelle efficaces. Or, il faut constater que peu de SRICs réels font appel aux techniques d’indexation multidimensionnelles car la dimension élevée des

descripteurs pose de sérieux problèmes de performances aux algorithmes d'indexation et de recherche. Brièvement, les propriétés particulières des espaces de grande dimension et le problème liée à la malédiction de la dimension font que leurs performances se dégradent exponentiellement lorsque la dimension augmente. De plus, bien qu'il existe déjà des algorithmes d'indexation et de recherche développés en bases de données, ils sont souvent inadaptés aux descripteurs d'images car un bon nombre de ces algorithmes reposent sur des hypothèses irréalistes concernant la distribution des descripteurs. En particulier, ils supposent que les descripteurs suivent une distribution particulière, notamment uniforme ou que leur dimension puissent être facilement réduite par des méthodes d'analyse des données.



# Bibliographie

- [AMP84] Y.S. Abu-Mostafa and D. Psaltis. Recognitive aspects of moment invariants. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, pages 698–706, Nov. 1984.
- [ASBH90] Klaus Arbter, Wesley E. Syder, Hans Burkhardt, and Gerd Hirzinger. Application of affine-invariant fourier descriptors to recognition of 3-d objects. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 12(7) :640–647, July 1990.
- [Bim99] A. Del Bimbo. Visual information retrieval. *Morgan Kaufmann Publishers, Inc. San Francisco, USA*, 1999.
- [CK96] Gene C.H. Chuang and C.C Jay Kuo. Wavelet descriptor of planar curves : Theory and applications. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 5(1) :56–70, 1996.
- [CP00] Gasquet Claude and Witomski Patrick. *Analyse de Fourier et applications*. 2000.
- [CV91] D. Cyganski and R.F. Vaz. A linear signal decomposition approach to affine invariant contour identification. *SPIE : Intelligent Robots and Computer Vision X*, 1607 :98–109, 1991.
- [Den99] Y. Deng. *A region Based Representation for Image and Video Retrieval*. PhD thesis, University of California, 1999.
- [dFCJ01] Luciano da Fontoura Costa and Roberto Marcondes Cesar Jr. *Shape Analysis and Classification Theory and Praticce*. Crc press llc edition, 2001.
- [ea90] G. Eichmann et al. Shape representation by gabor expansion. *SPIE, Hybrid Image and signal Processing II*, 1297 :86–94, 1990.
- [Fre61] H. Freeman. On the encoding of arbitrary geometric configurations. *IRE Trans. on Electronic Computers*, EC-10 :260–268, 1961.
- [FS78] H. Freeman and A. Saghri. Generalized chain codes for planar curves. *In Proceedings of the 4<sup>th</sup> International Joint Conference on Pattern Recognition, Kyoto, Japan*, EC-10 :701–703, November 1978.
- [GW93] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Addison-wesley edition, 1993.
- [HSYL98] Sang Uk Lee Hee Soo Yang and Kyoung Mu Lee. Recognition of 2d object contours using starting-point-independent wavelet coefficient matching. *Journal of Visual Communication and Image Representation*, 9(2) :171–191, 1998.
- [Hu62] M. Hu. Visual pattern recognition by moment invariants. *IRE Trans. Inform. Theory IT*, 8 :179–187, 1962.
- [IRR94] Herbert Süsse Irene Rothe, Klauss Voss and Jörg Rothe. A general method to determine invariants. Technical Report TR503, 1994.

- [IRV96] Herbert Süsse Irene Rothe and Klaus Voss. The method of normalization to determine invariants. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 18(4) :366–376, 1996.
- [IV96] J. Iivarinen and A. Visa. Shape recognition of irregular objects. In *D. P. Casasent (Ed.), Intelligent Robots and Computer Vision XV : Algorithms, Techniques, Active Vision, and Materials Handling*, Proc. SPIE(2904) :25–32, 1996.
- [JdFC95] R.M. Cesar Jr. and L. da F. Costa. Piecewise linear segmentation of digital contours in  $o(n \log(n))$  through a technique based on effective digital curvature estimation. *Real-Time Imaging*, 1 :409–417, 1995.
- [JdFC96] R.M. Cesar Jr. and L. da F. Costa. Towards effective planar shape representation with multiscale digital curvature analysis based on signal processing techniques. *Pattern Recognition*, 29(9) :1559–1569, 1996.
- [KB01] M. Khalil and M. Bayoumi. A dyadic wavelet affine invariant function for 2d shape recognition. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 23(10) :1152–1163, 2001.
- [KB02] M. Khalil and M. Bayoumi. Affine invariant for objet recognition using the wavelet transform. *Pattern Recognition Letters*, 23 :57–72, 2002.
- [KEh] KITH Kimcheng and ZAHZAH El-hadi. 2d affine-based recognition using discrete wavelet. *International Conference on Computer Vision and Graphics, September 22-24, 2004 Warsaw, Poland*.
- [KEh04a] KITH Kimcheng and ZAHZAH El-hadi. 2d shape recognition using discrete wavelet descriptor under similitude transform. *Proceedings Combinatorial Image Analysis : 10th International Workshop, IWCIA 2004, Auckland, New Zealand., 3322/2004 :679–689, December 1-3 2004*.
- [KEh04b] KITH Kimcheng and ZAHZAH El-hadi. Discrete wavelet descriptors for 2d shape recognition. *Proceedings Third International Conference on Image and Graphics, Hong Kong, China, ICIG 2004 :76–79, Dec.18-20 2004*.
- [KEh04c] KITH Kimcheng and ZAHZAH El-hadi. A dwt for 2d shape recognition under affine transform. *THE INTERNATIONAL WORKSHOP ON MULTIDISCIPLINARY IMAGE, VIDEO, AND AUDIO RETRIEVAL AND MINING Sherbrooke, Canada, October 25-26 2004*.
- [KK00] Hae-Kwang Kim and Jong-Deuk Kim. Region-based shape descriptor invariant to rotation, scale and translation. *Signal Processing : Image Communication*, 16 :87–93, 2000.
- [Lin94] T. Linderberg. *Scale-Space Theory in Computer Vision*. Kluwer Academic Publishers, 1994.
- [Lin96] T. Linderberg. Scale-space : A framework for handling image structures at multiple scales. In *Proc. CERN School of Computing, Egmond aan Zee, The Netherlands*, pages 8–21, September 1996.
- [LJLE00] R. Lakämper L. J. Latecki and U. Echhardt. Shape descriptors for non-rigid shapes with a single closed contour. *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 424–429, 2000.
- [LS98] G. Lu and A. Sajjanhar. On performance measurement of multimedia information retrieval systems. In *Proc. of International Conference on Computational Intelligence and Multimedia applications*, Australia(Feb. 9-11) :781–787, 1998.

- 
- [MA02] Farzin Mokhtarian and Sadegh Abbasi. Shape similarity retrieval under affine transforms. *Pattern Recognition*, 35 :31–41, 2002.
- [Mal89] Stephan Mallat. A theory for multiresolution signal decomposition : the wavelet representation. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 11 :674–693, 1989.
- [Mal00] Stéphane Mallat. *Une exploration des signaux en ondelettes*. Les éditions de l’Ecole Polytechnique, Novembre 2000.
- [MM86] Farzin Mokhtarian and Alan Mackworth. Scale-based description and recognition of planar curves and two-dimensional shapes. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 18(1) :34–43, 1986.
- [MM92] Farzin Mokhtarian and Alan Mackworth. A theory of multiscale, curvature-based shape representation for planar curves. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 14(8) :789–805, 1992.
- [MMOP03] Michel Misiti, Yves Misiti, Georges Oppenheim, and Jean-Michel Poggi. *Les ondelettes et leurs applications*. hermes Sciences, 2003.
- [Mok95] Farzin Mokhtarian. Silhouette-based isolated object recognition through curvature scale space. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 17(5) :539–544, 1995.
- [ODA03] Ahmed El Oirrak, M. Daoudi, and D. Aboutajdine. Affine invariant descriptors using fourier series. *Pattern Recognition Letters*, 23 :1109–1118, 2003.
- [PH74] T. Pavlidis and S.L. Horowitz. Segmentation of plane curves. *IEEE Transactions on Computers*, C23(8) :860–870, 1974.
- [Ram72] U. Ramer. An iterative procedure for the polygonal approximation of plane curves. *Computer Graphics and Image Processing*, 1 :244–256, 1972.
- [Rub] Y. Rubner. *Perceptual Metrics for Image Database Navigation*. PhD thesis, Stanford University.
- [SAK00] F. Mokhtarian S. Abbasi and J. Kittler. Enhancing css-based shape retrieval for objects with shallow concavities. *Image and Vision Computing*, 18 :199–211, 2000.
- [SB91] M. J. Swain and D. H. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1) :11–32, 1991.
- [ShWI99] Dinggang Shen, Wai him Wong, and Horace H.S. Ip. Affine-invariant image retrieval by correspondence matching of shapes. *Image and Vision Computing*, 17 :489–499, 1999.
- [Smi97] J.R. Smith. *Integrated Spatial and Feature Image System : Retrieval, Analysis and Compression*. PhD thesis, Columbia University, 1997.
- [TB97a] Q. M. Tieng and W. W. Boles. Recognition of 2d object contours using the wavelet transform zero-crossing representation. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 19(8) :910–916, 1997.
- [TB97b] Q. M. Tieng and W. W. Boles. Wavelet-based affine invariant representation : A tool for recognizing planar objects in 3d space. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 19(8) :846–857, 1997.
- [Wit83] Witkin. Scale-space filtering. In *Proc. 8th International Joint Conference on Artificial Intelligence (IJCAI’83), Karlsruhe, Germany*, 2 :1019–1022, 1983.

- [Zha02] Dengsheng Zhang. *Image Retrieval Based on Shape*. PhD thesis, Faculty of Information Technologie, March 2002.
- [ZL01] Dengsheng Zhang and Guojun Lu. A comparaisn of shape retrieval using fourier descriptors and short-time fourier descriptors. *In Proc. of The Second IEEE Pacific-Rim Conference on Multimedia PCM01*, pages 855–860, October 2001.
- [ZL02] Dengsheng Zhang and Guojun Lu. A comparative study of fourier descriptors for shape representation and retrieval. *ACCV2002 : The 5th Asian Conference on Computer Vision, 23-25 January, Melbourne, Australia*, pages 646–651, January 2002.
- [ZL03a] Dengsheng Zhang and Guojun Lu. A comparative study of curvature scale-space and fourier descriptors for shape-based image retrieval. *Journal of Visual Communication & Image Representation*, (14) :41–60, 2003.
- [ZL03b] Dengsheng Zhang and Guojun Lu. Evaluation of similarity measurement for image retrieval. *In Proc. of IEEE International Conference on Neural Networks and Signal Processing (ICNNSP03), Nanjing, China*, pages 14–17, December 2003.
- [ZL04] Dengsheng Zhang and Guojun Lu. Review of shape representation and description techniques. *Pattern Recognition*, 37(1) :1–19, 2004.
- [ZR72] Charles T. Zahn and Ralph Z. Roskies. Fourier descriptors for plane closed curves. *IEEE Transactions on Computers*, C-21(3) :269–281, March 1972.



## Résumé

Cette thèse s'intéresse aux systèmes de reconnaissance des objets 2D par le contour, le but est de rechercher les objets (contours) les plus similaires à un objet (contour) requête fourni en exemple. Pour cela, il est d'abord nécessaire de décrire les contours d'objets stockés dans la base. L'objectif est de traduire la similarité visuelle entre les contours en une simple notion de proximité spatiale entre descripteurs. Ainsi, pour retrouver les contours similaires au contour requête, il suffit de retrouver les descripteurs les plus proches du descripteur du contour requête.

Un des problèmes fondamentaux du système de reconnaissance des formes réside dans le choix de la description invariante de contour. En effet, on estime que les objets retrouvés par le système correspondent rarement exactement à la requête, des transformations similitudes ou affines peuvent les séparer. C'est pourquoi la description du contour doit posséder de bonnes propriétés d'invariance afin de conférer de la robustesse au système de reconnaissance.

La première partie de cette thèse concerne l'étude des outils mathématiques les plus utilisés dans le cadre du système de reconnaissance des formes tels que les descripteurs par les Moments, les descripteurs de Fourier, les descripteurs utilisant la courbure multi-échelle etc. La seconde partie de ce travail est consacrée à l'étude de la théorie de la transformation en ondelettes qui nous a permis de proposer deux nouveaux descripteurs qui ont été évalués et comparés en utilisant la base CE-1 MPEG7, les résultats de ces évaluations et comparaisons ont montré que ces deux descripteurs sont robustes au même titre que les meilleurs descripteurs actuels avec des avantages au niveau de la complexité et du choix des noyaux d'ondelettes en fonction des applications.

**Mots-clés:** Reconnaissance de forme, La transformation en Ondelettes, Similitude, Affine, Descripteur.

## Abstract

This thesis concerns the 2D object recognition systems using contours. The aim is to retrieve the most similar objects in a database to an object query. To do that, it is necessary to describe the objects of the database and express the dissimilarity measure among objects.

One of the fundamental problems of the recognition and retrieval systems is the choice of the invariant description. Retrieved objects obtained by actual systems rarely match queries. Similitude or affine transforms may occur among them. To be robust, a recognition system must have a good invariance properties.

In the first part of this thesis, some mathematical tools are studied in the frame of pattern recognition as the Moments, the Fourier and the multi-scale descriptors. The second part is dedicate to the wavelet transform theory which enables us to propose two new descriptors. These descriptors were compared and assessed to the best actual descriptors. The experimentations were performed on the MPEG7 CE-1 data base and the assessment showed that these two descriptors are real concurrents to the best actual descriptor (Fourier descriptor) and better than the Multi-scale descriptor chosen by the MPEG7 norm. Moreover, the proposed descriptors are better in term of complexity and free-choice of kernels depending on applications.

**Keywords:** Pattern Recognition, Wavelet Transform, Descriptor, Similitude, Affine.

