

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE MOULOUDMAMMERI, TIZI-OUZOU



FACULTE DE GENIE ELECTRIQUE ET D'INFORMATIQUE
DEPARTEMENT AUTOMATIQUE

MEMOIRE DE MAGISTER

En Automatique

Option : **Traitement d'Images et Reconnaissance de Formes.**

Présenté par

FERTAS FOUAD

Comparaison et classification de formes basées sur un système d'inférences.

Devant le jury d'examen composé de :

- | | | | |
|------------------------------|------------|------------|-------------|
| - HAMMOUCHE Kamal | Professeur | à l'UMMTO. | Président. |
| - AHMED-OUAMER Rachid | Professeur | à l'UMMTO. | Rapporteur. |
| - HADDAB Salah | M.C.A. | à l'UMMTO. | Examineur. |
| - ALKAMA Sadia | M.C.A. | à l'UMMTO. | Examineur. |

REMERCIEMENTS

*Ce travail de recherche a été réalisé à l'université de mouloud mammeri de tizi ousou , sous la direction de mon Directeur de thèse monsieur **Ahmed-ouamer Rachid** professeur au département d'informatique , à qui je présente l'expression de ma profonde gratitude pour son soutien continu, sa disponibilité et ses précieux conseils et qui m'a laissé une grande liberté dans le choix et l'orientation de mes travaux tout en sachant me guider. Je tiens à souligner la qualité de ses observations pertinentes, sa grande rigueur scientifique et son investissement dans cette étude.*

*Je tiens à remercier, le Professeur **Hammouche kamel**, qui m'a fait l'honneur d'accepter de présider le jury.*

Je tiens à exprimer mes remerciements à :

*Monsieur **HADDAB Salah**, Maître de Conférences classe A, à l'université de tizi ousou, pour avoir accepté de faire partie des membres de jury et d'examiner ce travail.*

*Madame **ALKAMA Sadia**, Maître de Conférences classe A, à l'université de tizi ousou, pour avoir accepté d'examiner cette thèse en tant que membre du jury.*

Je remercie toute ma famille; une pensée spéciale va à mes parents et ma femme qui m'ont soutenu et conseillé durant ces trois années (et depuis bien longtemps...).

Enfin, je remercie tous mes amis, et tout l'encadrement en automatique de l'université de tizi ousou qui m'ont aidé de près ou de loin.

TABLE DES MATIERES

ABRÉVIATIONS.....	iii
INTRODUCTION GÉNÉRALE.....	1
CHAPITRE I: SEGMENTATION.....	5
1. INTRODUCTION.....	5
2. DEFINITION DE LA SEGMENTATION.....	8
2.1. Approche globale de l'image (pixellaire).....	8
2.2. Segmentation par région	8
2.3. Segmentation par détection de contours	9
3. IMPORTANCE DE LA FORME PERCEPTUELLE EN 2D.....	10
4. METHODE DE DETECTION DE CPP.....	12
4.1. Propriétés Géométriques des Formes de Contour : dx, dy.....	12
4.2. Détection de CPP Fort.....	13
4.3. Séquence Préservant l'Ordre de Boîte Arc-tangente.....	15
4.4. Détection de CPP Faible.....	20
5. CONCLUSION.....	26
CHAPITRE II: CLASSIFICATION ET RECONNAISSANCE STRUCTURELLE DE FORMES	28
1. INTRODUCTION ET NOTION ÉLÉMENTAIRES.....	28
1.1. Aperçu historique.....	28
1.2. Notion de structure.....	28
1.3. Notion de syntaxe.....	29
1.4. Le système d'inférence.....	30
1.5. Règles d'inférences	30
1.6. Dédution logique.....	31
1.7. Choix des primitives.....	32
1.8. Classification.....	32
2. CLASSIFICATION DE GET.....	33
2.1. Ligne Droite.....	34
2.2. Courbe.....	34
3. REPRESENTATION DES GET.....	36
3.1. Les modèles de GETs.....	36
3.2. Classification perceptuelle.....	40
4. LA RECONNAISSANCE	40

4.1. Classification des ellipses	43
5. CONCLUSION	46
CHAPITRE III: RESULTATS EXPERIMENTAUX ET INTERPRETATIONS	48
1. FORME D'UN CERCLE	48
1.1. Segmentation.....	51
1.2. Classification.....	54
2. FORME GEOMETRIQUE	63
2.1. Segmentation	64
2.2. Classification.....	67
3. COMPARAISON ENTRE LA METHODE PROPOSEE ET LA METHODE DU GRADIENT LOCALE.....	83
4. ÉVALUATION DE LA METHODE	84
4.1. Cercle	84
4.2. forme géométrique.....	84
4.3. Feuille d'arbre (Figure III-14).....	84
4.4. Le dinosaure herbivore (Figure III-29).....	84
5. CONCLUSION	85
CONCLUSIONS GÉNÉRALES	86
BIBLIOGRAPHIE.....	89

LISTE DES FIGURES

Figure I-1 : une illustration des types GET et du CPP.	11
Figure I-2 : le fonctionnement de l'approche de détection de caractéristique.	11
Figure I-3 : le calcul de dx et dy pour chaque pixel de contour.	13
Figure I-4 : schéma de dx et dy changeant de signe pour la détection de CPP forts.	14
Figure I-5 : changements de CPP faibles ne peuvent pas être découverts selon un schéma Traversant zéro. .	15
Figure I-6 : les traces de contour et leurs séquences Arc-tangentes correspondantes.	17
Figure I-7 : Séquence de barre de colonne à 3D pour la séquence des boîtes pour le contour de la figure 5 ..	19
Figure I-8 : deux critères de segmentation de boîtes.	24
Figure I-9 : l'algorithme de la détection des CPP.	25
Figure II-1 : scène schématique (à gauche) et une description hiérarchique des surfaces présentes (à droite)	28
Figure II-2 : courbe discrétisée et sa représentation sous la forme d'une chaîne de chiffres (à gauche) conforme au codage défini (à droite).	29
Figure II-3 : ensemble de primitives (à gauche) permet de représenter chaque graphique (à droite).	32
Figure II-4 : une structure de classification hiérarchique des GET (classification de premier niveau).	34
Figure II-5: classification des segments de droite. (a) Segment de droite, (b) vue 3D de la séquence de boîte, (c) vue 2D de la séquence de boîte	35
Figure II-6 : classification d'un arc. (Détails dans la section 1.2 de chapitre 4).	35
Figure II-7: (a) des exemples de partition de la courbe. (b) les huit catégories des segments génériques. (c) les huit catégories des points de partition de la courbe.	37
Figure II-8 : AC est caractérisé par deux groupes de fonctions qui partitionnent la courbe en deux segments des arcs, AB qui est le CS2 et BC qui est le CS1 (Fig. II-7 (b)).	39
Figure II-9 : le concept hiérarchie de GET.	41
Figure II-10 : un GS.	41
Figure II-11 : un CPP.	42
Figure II-12 : exemple de traces.	43
Figure II-13 : un modèle d'ellipse.	44
Figure II-14 : la représentation graphique.	45
Figure II-15 : quelques exemples possibles de modèles d'ellipses incomplètes.	45
Figure III-1 : la forme originale.	48
Figure III-2 : l'étape de binarisation.	48
Figure III-3 : l'histogramme de l'image.	49
Figure III-4 : le contour de la forme	49
Figure III-5 : les quatre points CPP forts en rouge	52

Figure III-6 : séquence de boites correspond au segment entre les deux premiers points CPP forts.....	56
Figure III-7 : séquence de boite correspond au segment entre les deux derniers points CPP forts	60
Figure III-8 : la forme de la figure III-1 segmentée en arcs.	63
Figure III-9 : une forme géométrique, les points en rouge sont les CPP forts et les points en bleu sont les CPP faibles type sommet-vallée-sommet	63
Figure III-10 : séquence de boite correspond au segment 1 et 2, la flèche indique la position de point CPP faible.....	67
Figure III-11: séquence de boite correspond au segment 1 entre les deux points CPP forts.....	67
Figure III-12 : séquence de boite correspond au segment 2 entre les deux points CPP forts	68
Figure III-13 : la forme géométrique de la Figure III-9 segmentée en ligne droites.....	74
Figure III-14 : une feuille d’arbre segmentée en arcs et en segments de droites.	75
Figure III-15 : forme géométrique.....	76
Figure III-16 : le chapeau de willington.....	77
Figure III-17 : les points CPP forts en rouge	77
Figure III-18 : les points CPP faibles en bleu type (Sommet-vallée-sommet).....	77
Figure III-19 : les points CPP faibles en jeune type (Changement de la Tendance de Catégories Arc-tangente).....	77
Figure III-20 : les points CPP faible en bleu type (Sommet-vallée-sommet)	78
Figure III-21: l’étape de binarisation initiale	78
Figure III-22 : le contour de la forme	78
Figure III-23 : les points CPP forts	78
Figure III-24 : Tous les points CPP, Les points CPP forts en rouge, les points CPP faibles type sommet-vallée-sommet en bleu et les points CPP faibles type changement de la tendance des catégories en jaune.	79
Figure III-25 : les lignes droites en bleu et le bruit type ligne droite en blanc, les arcs en verts	79
Figure III-26 : feuille d’arbre.....	80
Figure III-27 : Tous les points CPP, Les points CPP forts en rouge, les points CPP faibles type sommet-vallée-sommet en bleu et les points CPP faibles type changement de la tendance des catégories en jaune.	80
Figure III-28 : les lignes droites en bleu et le bruit type ligne droite en blanc, les arcs en ver	81
Figure III-29 : dinosaure herbivore, Les points CPP forts en rouge, les points CPP faibles type sommet-vallée-sommet en bleu et les points CPP faibles type changement de la tendance des catégories en jaune.	82
Figure III-30 : comparaison des résultats obtenus de la méthode proposée et celle du gradient local	83

LISTE DES TABLEAUX

Tableau I -1 : Catégories de valeurs de degré Arc-tangente	18
Tableau I -2 : les degrés d'Arc-tangente et leurs catégories de l'exemple de trace dans la Figure I -5...	18
Tableau I -3 : une séquence de boîte après agrégation basée sur la Table I -2.....	18
Tableau II-1 : la définition de <i>GSs</i>	38
Tableau II-2 : la définition de CPPs	42
Tableau II-3 : la définition d'ellipse	44

Abréviations

CBIR: content-based image retrieval.

CPP: curve partitioning points.

GET: generic edge tokens.

GS: generic segments.

OPABS: order preserving arctangent bin sequence.

PO: perceptual organization.

PCPG: perceptual curve partitioning and grouping.

SIFT: scale invariant feature transformation.

TSW: tangent sliding window.

INTRODUCTION GÉNÉRALE

Introduction générale

Les dernières décennies sont marquées par l'évolution rapide de la technologie sur tous les plans de notre vie, l'homme n'a pas cessé de moderniser et de développer de nouvelles machines de plus en plus performantes, son objectif est de fabriquer des machines dotées de capacités qualifiées d'intelligence.

Aujourd'hui les recherches intenses portant sur le traitement de l'information, la théorie de la décision et les méthodes de reconnaissance de formes d'une part, et sur la disponibilité des échantillons d'image ou bases de données ainsi que les développements continus des moyens de stockage de calculs et de traitement d'une autre part, ont permis la conception des machines et des systèmes qui facilitent le travail de l'homme et le remplace dans les tâches répétitives.

La reconnaissance de formes est une science de définition algorithmique permettant de classer des images ou bien des signaux qui caractérisent des objets dont l'espace a varié par rapport à un objet type appelé prototype, en d'autres termes elle permet le passage de la perception des objets à leur reconnaissance, généralement un problème de reconnaissance se présente comme :

- La définition précise des classes entre lesquelles on va décider.
- Le choix du vecteur d'attributs extrait à partir du signal ou bien de l'image qui caractérise l'objet.
- Le choix d'une méthode de décision.
- L'exploitation.

La reconnaissance de formes à travers leurs contours n'est pas une discipline récente. Les recherches faites dans ce sens, voient chaque jour de nouvelles approches, pourtant l'ordinateur, outil essentiel pour ce domaine ne reconnaît pas de formes avec un taux de confiance égale à celui de l'homme.

Dans notre travail, nous proposons une méthode pour la classification de formes basée sur la comparaison des descriptions structurelles, l'étude consiste à décrire la forme avec des primitives simple (arcs, segments de droites, points partageants la courbe), comparer deux formes revient à comparer leurs descriptions structurelles.

Le travail réalisé comporte trois chapitres.

Le premier chapitre sera consacré à l'étape de segmentation de contour de la forme, nous élaborons une méthode statistique non paramétrique (traversant zéro) pour la détection des CPP forts, et le schéma des OPABS pour la détection des points CPP faibles.

Le chapitre deux fera l'objet d'une étude détaillée pour les techniques de classifications proposées pour les primitives (la classification premier niveau puis la classification deuxième niveau) et une technique de classification pour les points CPP à l'aide d'un système d'inférences.

Les résultats obtenus et leurs interprétations feront l'objet du dernier chapitre.

Nous terminerons notre étude par une conclusion générale du travail élaboré et des axes principaux autour desquels nous pourrons élargir cette étude.

CHAPITRE I:

SEGMENTATION

Introduction

Une image est une collection d'informations qui se présentait sur un support photographique qui permettait le traitement d'un phénomène temporaire. Une analyse fine des phénomènes enregistrés sur l'archivage et l'illustration.

Le traitement d'images est né de l'idée de la nécessité de remplacer l'observateur humain par la machine. L'image ou les signaux provenant des capteurs ont été numérisés pour pouvoir être traités par l'ordinateur. L'image a été codée et mémorisée sur différents supports (magnétique, magnéto-optique, etc.).

Une image numérique est constituée d'un certain nombre d'objets, pour les mettre en évidence, il faut utiliser des algorithmes adéquats selon le type de l'image et ses caractéristiques intrinsèques.

En effet, les méthodes de détection de ces objets appelées méthodes de segmentation existent en nombre considérable, le problème réside dans le choix de telle ou telle méthode. Il n'existe pas de méthode spécifique à un type d'image, l'appréciation du spécialiste (agronome, météorologue, médecin,...) reste le moyen par excellence de décision.

En effet, le choix d'une technique est lié :

A la nature de l'image :

- Eclairage non homogène, reflets,
- Présence de bruit, de zones texturées.
- Contours flous, en partie occultés.

Aux opérations situées en aval de la segmentation :

- Localisation, mesure, calcul 3D.
- Reconnaissance de formes, interprétation.
- Diagnostique, contrôle qualité.

Aux primitives à extraire :

- Contours, segments de droite, arcs, angles,...
- Régions, formes.

- Texture.

Aux contraintes d'exploitation :

- Complexité algorithmique, fonctionnement en temps réel.
- Taille de la mémoire disponible en machine.

Un problème fondamental pour la compréhension de l'image et de la vidéo est comment efficacement choisir la saillie visuelle efficace parmi de nombreuses caractéristiques de niveau inférieur et construire une représentation appropriée pour l'informatique sémantique. Ce n'est pas chaque stimulus visuel relié à la saillie visuelle qui attire l'attention visuelle humaine, c'est seulement celui dont les attributs visuels locaux diffèrent de façon significative des attributs environnants qui déclenchera les signaux de neurones biologiques dans le système du cerveau. Pour imiter le mécanisme visuel humain, nous devons extraire des caractéristiques en saillie perpétuelles des images, qui peuvent efficacement capturer l'attention visuelle aux objets cibles. Les renseignements sur la couleur sont relativement faciles à obtenir et donc, sont généralement utilisés dans beaucoup de systèmes de vision. Évidemment, les méthodes à base de couleur échoueront pour les images dont les couleurs ne sont pas disponibles. Les caractéristiques de texture fournissent plus de renseignements spatiaux ou relationnels que les couleurs. Tamura et al. [1] ont proposé un ensemble de dessins de texture qui contiennent six caractéristiques choisies par les expériences psychologiques: la grossièreté, le contraste, la direction, la ressemblance de ligne, la régularité et l'âpreté (rugosité). Les restrictions des caractéristiques de la texture c'est le fait qu'elles ne sont pas génériques pour différentes applications et quelques méthodes impliquent des coûts informatiques élevés ainsi qu'une complexité d'implémentation [2].

La perception visuelle humaine est largement basée sur les formes [3], c'est-à-dire elle peut percevoir une scène basée sur les contours d'objets seuls sans utiliser des renseignements de texture et de couleur. La forme de contour est une des caractéristiques fondamentales d'objets qui peuvent être discriminées sans effort par la vision humaine. L'extraction efficace de caractéristiques de formes est essentielle pour l'informatique sémantique visuelle. Les découvertes de la recherche en neuroscience nous fournissent des éclaircissements utiles sur les critères de la vision humaine génériques. Se basant sur l'Organisation Perceptuelle (PO) - la Partition et le Groupement de la Courbe Perceptuelle.

(PCPG) [4] est un modèle de perception de formes dans lequel chaque contour est composé des Jetons de contour Génériques (GET) raccordés à des Points Partageant la Courbe (CPP). Ces caractéristiques perceptuelles (les GET et les CPP) sont liées directement aux entités de saillie visuelles et leur représentation correspondante fournit les cartes de représentation de saillie pour la compréhension visuelle à haut niveau. Zheng, et al. [5] ont présenté une méthode qui convertit une image en une carte de saillie de contour faite à partir des Jetons de Contour Génériques (GET). Dans cette approche, les caractéristiques GET encodent tant contour que contenu de texture d'une image selon la distribution de différents types de GET. Même avec ce schéma trop simplifié, la représentation de GET à base de saillie est capable de manipuler des tâches d'analyse d'images stimulantes comme CBIR [6], l'instrument de visualisation [6] et la segmentation d'images [7].

Pour fournir des cartes de saillie de meilleur niveau de caractéristiques pour les tâches d'interprétation de haut niveau, dans ce chapitre, l'un de nos buts est celui d'améliorer l'exactitude de la détection de CPP en utilisant une méthode statistique non- paramétrique. L'approche inclut : 1) Au lieu d'utiliser les gradients, introduire un espace arc tangent pour rendre la preuve CPP plus mesurable. 2) Utiliser l'ordre de pixel séquentiel le long d'une courbe comme heuristique pour améliorer l'exactitude de la détection de CPP. 3) Concevoir une nouvelle méthode d'évaluation de CPP en incluant un schéma à Traversant zéro pour localiser les CPP forts et un schéma de Séquence de Boîte Préservant l'Ordre d'Arc tangente (OPABS) pour détecter les CPP faibles.

Le schéma OPABS proposé est une méthode statistique non- paramétrique [8] utilisant les mesures tant locales que globales des données séquentielles de la courbe. Les représentations d'images basées sur la saillie visuelle, comme SIFT [9] [10], ont été largement reconnus dans la communauté de vision informatique. Les descripteurs d'images basés sur les GET/ CPP (les cartes de saillie) sont capables de construire un pont sur le gap sémantique.

2. Définition de la segmentation

La segmentation d'image est le processus qui permet de partitionner celle-ci en zones d'intérêt correspondant à des objets de la scène d'où elle est issue.

La segmentation est un traitement de bas niveau qui consiste à créer une partition de l'image E en sous-ensembles E_i , appelés régions telle qu'aucune région ne soit vide, l'intersection entre deux régions soit vide et l'ensemble des régions recouvre toute l'image. Une région est un ensemble de pixels connexes ayant des propriétés communes qui les différencient des pixels des régions voisines. [11].

Les méthodes de segmentation peuvent être classées en trois groupes :

- Une approche globale de l'image,
- Les méthodes associées aux régions qui permettent une partition de l'image,
- Les méthodes associées aux contours qui permettent un découpage de l'image.

2.1. Approche globale de l'image (pixellaire)

Le principe consiste à regrouper les pixels selon leurs attributs (intensité, contraste, vecteur moment) sans tenir compte de leurs localisation au sein de l'image. Cela permet de construire des classes de pixels, les pixels adjacents, appartenant à une même classe, forment alors des régions. Parmi les approches pixellaires, on distingue les méthodes par seuillage, basées uniquement sur un seul attribut (souvent l'intensité du pixel), des méthodes permettant une classification multi attributs (par exemple l'intensité du pixel et un calcul de distances). L'objectif consiste à affecter à chaque pixel de l'image une étiquette e_i indiquant la classe (région) i la plus proche (mesure de distance entre le pixel et le centre de gravité de la classe).

2. 2. Segmentation par régions

L'objectif de ces méthodes consiste à partitionner l'image traitée en régions homogènes, chaque objet de l'image pouvant être ainsi constitué d'un ensemble de régions. Dans le but de produire des régions volumineuses et afin d'éviter une division parcellaire des régions, un critère de proximité géographique peut être ajouté au critère d'homogénéité. Au final, chaque pixel de l'image reçoit une étiquette lui indiquant son appartenance à telle ou telle région. On obtient ainsi une carte des régions de l'image.

Quatre approches principales de la segmentation par régions sont disponibles:

Méthodes par séparation (Split).

Méthodes par fusion (Merge).

Méthodes par séparation-fusion (Split & Merge).

Méthodes par croissance de régions.

2. 3. Segmentation par détection de contours

Le contour est la représentation dans l'image des frontières des objets de celle-ci, la notion de contour englobe souvent les points de discontinuité de la fonction de gris $F(x, y)$, la recherche des points contours revient à la recherche des variations locales et significatives de cette fonction de gris, ces variations sont détectées par la notion de dérivée.

Le principe repose sur la définition de différentiation appliquée sur la fonction de gris, cette différentiation est définie à partir de deux opérateurs très importants, le gradient et le laplacien, un autre opérateur est aussi utilisé, la seconde dérivée directionnelle prise dans la direction f du gradient.

Un contour au point (x, y) d'orientation q sera repéré par un extremum de la dérivée première, ou encore par un passage par zéro de la dérivée seconde de la fonction $F(x, y)$ dans la direction normale au contour.

➤ Détection du contour par les techniques du gradient

Le gradient est défini à partir des dérivées partielles $\frac{\partial}{\partial x}$ et $\frac{\partial}{\partial y}$ traduisant la variation selon x et y : $G = (\frac{\partial}{\partial x}, \frac{\partial}{\partial y})$.

Généralement on préfère utiliser sa norme $|G|$ au lieu d'exploiter directement ses coordonnées.

Dès les années soixante, beaucoup de détecteurs localisent les contours par estimation du gradient ou du laplacien par la fonction de gris $F(x, y)$ ont été utilisés, ces détecteurs correspondent à des opérateurs bidimensionnels de différentiation discrète, les plus utilisés sont : l'opérateur de Roberts, le masque de Prewit [12], le masque de Sobel [13].

Il existe également d'autres gradients (masques) tels que les opérateurs de Kirsch [14].

➤ Détection du contour par les techniques du laplacien

Une deuxième façon de quantifier une transition de niveau de gris est le calcul du laplacien noté Δ il est défini à partir des dérivées partielles $\frac{\partial^2}{\partial x^2}, \frac{\partial^2}{\partial y^2}, \frac{\partial^2}{\partial x \partial y}$

$$\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial x \partial y}$$

Notons que le critère de Shen et Castan [15][16][17], exprime uniquement un rapport signal/bruit, et de ce fait, il est équivalent au critère de détection de Canny [18], nous sommes intéressés à l'approche contour, le filtre de canny est utilisé car il donne un faible taux d'erreur dans la signalisation des contours, et minimise les distances entre les contours détectés et les contours réels.

3. Importance de la forme perceptuelle en 2D

Le modèle de Courbe Perceptuelle Partageante et Groupant (PCPG) [4] a été proposé pour imiter la fonction de perception humaine dans le fait de partager et le fait de grouper des contours d'objets des images. Dans ce modèle de PCPG, les courbes en 2D sont partagées dans un ensemble minimal de Jetons de Contours Génériques de type (GET) (Figure I-1 (a)) qui sont raccordés aux Points Partageant la Courbe (CPP) (Figure I-1(b)). Chaque type de GET satisfait une combinaison unique d'augmentation ou de diminution monotone dans la géométrie x-y.

Les CPP sont les points de saillie sur les courbes de contour. Certains d'entre eux peuvent être trouvés là où les propriétés de monotonie le long d'un contour sont cassées. Ils sont appelés les CPP Forts et leurs propriétés de géométrie x-y et valeurs de tangente sont acquises de leurs voisins (Figure I-1 (b) - (1~ 3)). Dans d'autres cas, certains points critiques de la forme ne peuvent pas être découverts seulement en vérifiant les conditions de monotonie. Par exemple, dans la Figure I-1 (b) - (4~ 6), les points marqués sont visuellement spéciaux à la perception de la vision humaine [19], mais les changements le long de ces traces ne violent pas les propriétés monotoniques. Pour distinguer les différences, nous appelons les points partagés dans la Figure I-1 (b) - (4-6) les CPP faibles, qui restent pour la perception des points critiques, mais sans casser les continuités des propriétés monotoniques.

Dans le module précédent de détection de CPP [19], les CPP forts sont découverts en vérifiant les données de gradient et les CPP faibles sont vérifiés par une méthode de courbure à base de gradient. La méthode de courbure locale est sensible au bruit de contour et au manque de vue globale pour prendre des décisions justes, le détecteur à base de courbure local souvent ne prend pas en compte les CPP faibles et apporte certains faux positifs en raison des impacts des données locales maximums et bruyantes, et cela provoque donc une mauvaise classification sur les types de GET.

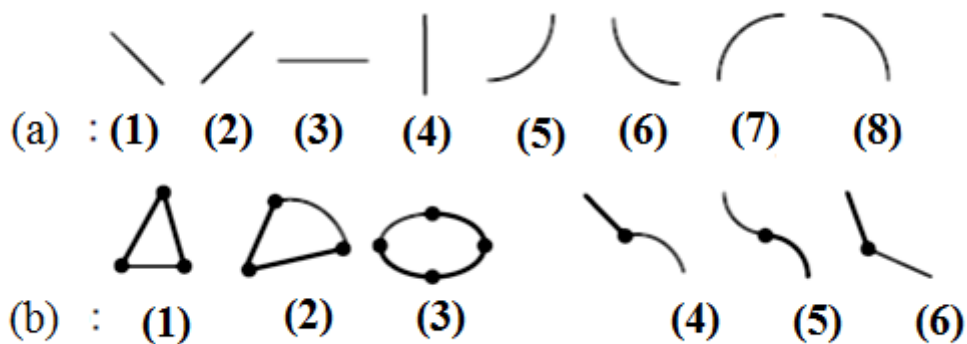


Figure I-1 : une illustration des types GET et du CPP.

(a) Les 8 types de GET, (b) différents types du CPP, La plupart des CPP sont invariables de vue et de Rotation sauf le type 3 qui est flottant si la courbe est tournée, le type (1) et (3) sont les CPP forts, (4) et (6) sont les CPP faibles.

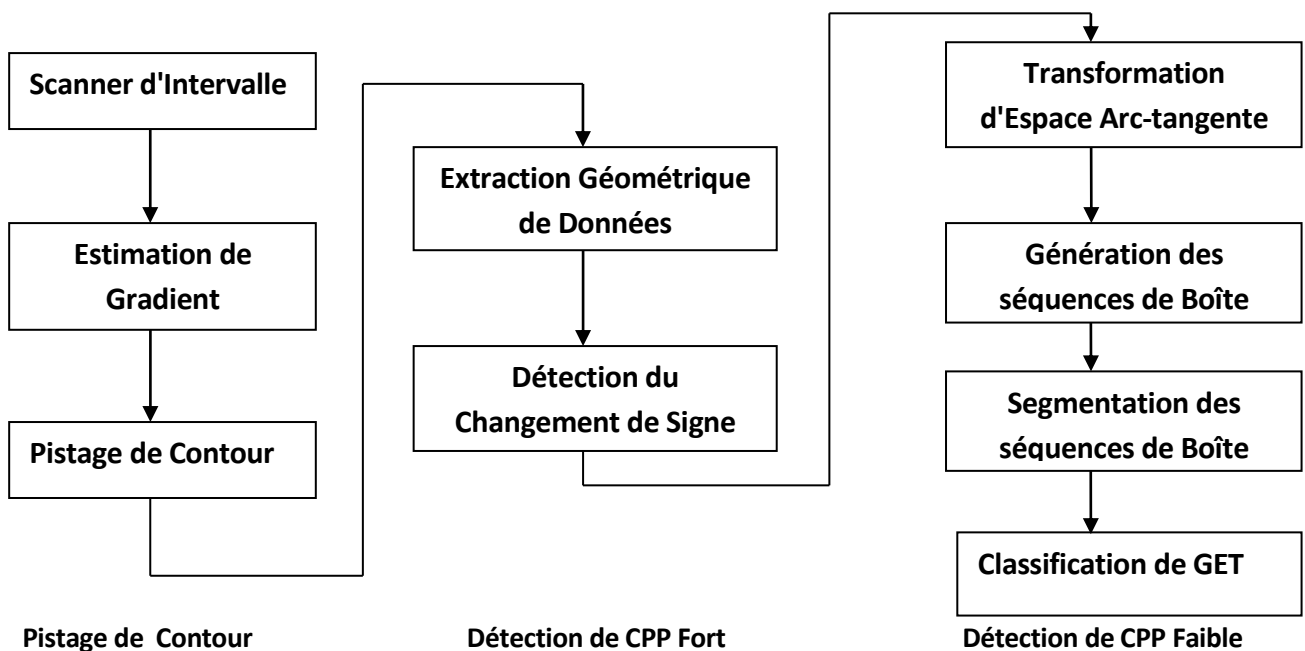


Figure I-2 : le fonctionnement de l'approche de détection de caractéristiques.

4. méthode de détection de CPP

Dans cette section, nous présentons une méthode statistique non-paramétrique pour la détection de CPP [20]. La Figure I-2 illustre le fonctionnement du système. L'approche profite des propriétés géométriques de la trace de contour pour extraire les valeurs dx et dy de chaque pixel de contour. En utilisant une méthode locale pour vérifier les traversées zéro de dx et de dy, les CPP forts sont identifiés et ensuite une trace de contour est partagée en segments bruts. Selon les demandes de l'application, les segments bruts peuvent être davantage partagés en segments plus petits en appliquant une méthode de détection de CPP faible. Une approche globale proposée, le schéma (OPABS) Séquence de Boîte Préservant l'Ordre d'Arc tangent, est utilisée pour la détection de CPP faible. Le schéma d'OPABS est une méthode sans distribution qui ne compte pas sur les hypothèses que les données sont tirées d'une distribution probabiliste donnée. C'est la méthode basée sur une statistique non-paramétrique.

4. 1. Propriétés Géométriques des Formes de Contour : dx, dy

les propriétés géométriques, dx et dy de chaque pixel de contour sont dérivées pour la détection de CPP et la classification de GET, Une Fenêtre Faisant glisser la Tangente (fenêtre glissante des tangents) (TSW) est utilisée pour déterminer les dx et dy de chaque pixel. La TSW est une boîte rectangulaire minimale pour entourer plusieurs pixels de contour qui sont les voisins du pixel (dans le milieu) pour lequel dx et dy sont calculés. La taille de la fenêtre de TSW est le nombre des pixels dans le TSW et doit être une valeur impaire. La taille minimale est 3. Ici nous utilisons 5 comme taille, car il donne empiriquement la meilleure performance. Dès qu'un TSW est mis, dx et dy d'un pixel peuvent être obtenus comme étant les différences le long de x et de y dans le TSW. Spécialement, prenez deux pixels de fin du TSW, dx est la différence entre les deux points finals dans X, et dy est la différence entre les deux dans Y. Dans la Figure I-3, la taille TSW est 5, deux points finals du petit segment dans le TSW sont N1 et N2, dx et dy de P sont :

$$dx = N2.x - N1.x. \quad (1)$$

$$dy = N2.y - N1.y. \quad (2)$$

Les valeurs de dx et dy indiquent la pente du pixel le long de la trace, les changements de dx et de dy le long de la courbe reflètent la propriété de forme de la courbe.

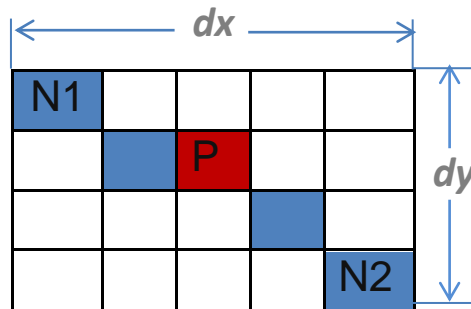


Figure I-3 : le calcul de dx et dy pour chaque pixel de contour.

N1 et N2 sont 2 points finals du segment entouré dans le TSW dont la taille est 5. Le dx de P =4, le dy de P =3.

4. 2. Détection de CPP Fort

Un CPP Fort correspond toujours à une jonction GET le long de la courbe où un critère générique de type GET est terminé et la valeur de son dx ou dy doit traverser un point zéro. Dans la Figure I-4, la trace de contour a deux lignes de suite découpées à un CPP fort qui est marqué par un cercle rouge. La grille montre des pixels dans le cercle rouge dans une vue élargie. Chaque cellule de la grille est un pixel. La table de côté juste dans la Figure I-4 montre les valeurs de dx et dy correspondant aux pixels de contour dans la grille. P1 se positionne à l'extrémité de départ, alors que P15 est à la fin. Comme nous pouvons voir dans la table de P1 à P15, le changement de signe des valeurs de dx se produit entre P8 et P9. Systématiquement, P8 et P9 sont visuellement les points monotoniques changeants de la trace de contour originale. Ainsi, un CPP fort peut être détecté en observant le schéma de dx et dy changeant de signe (traversant zéro).

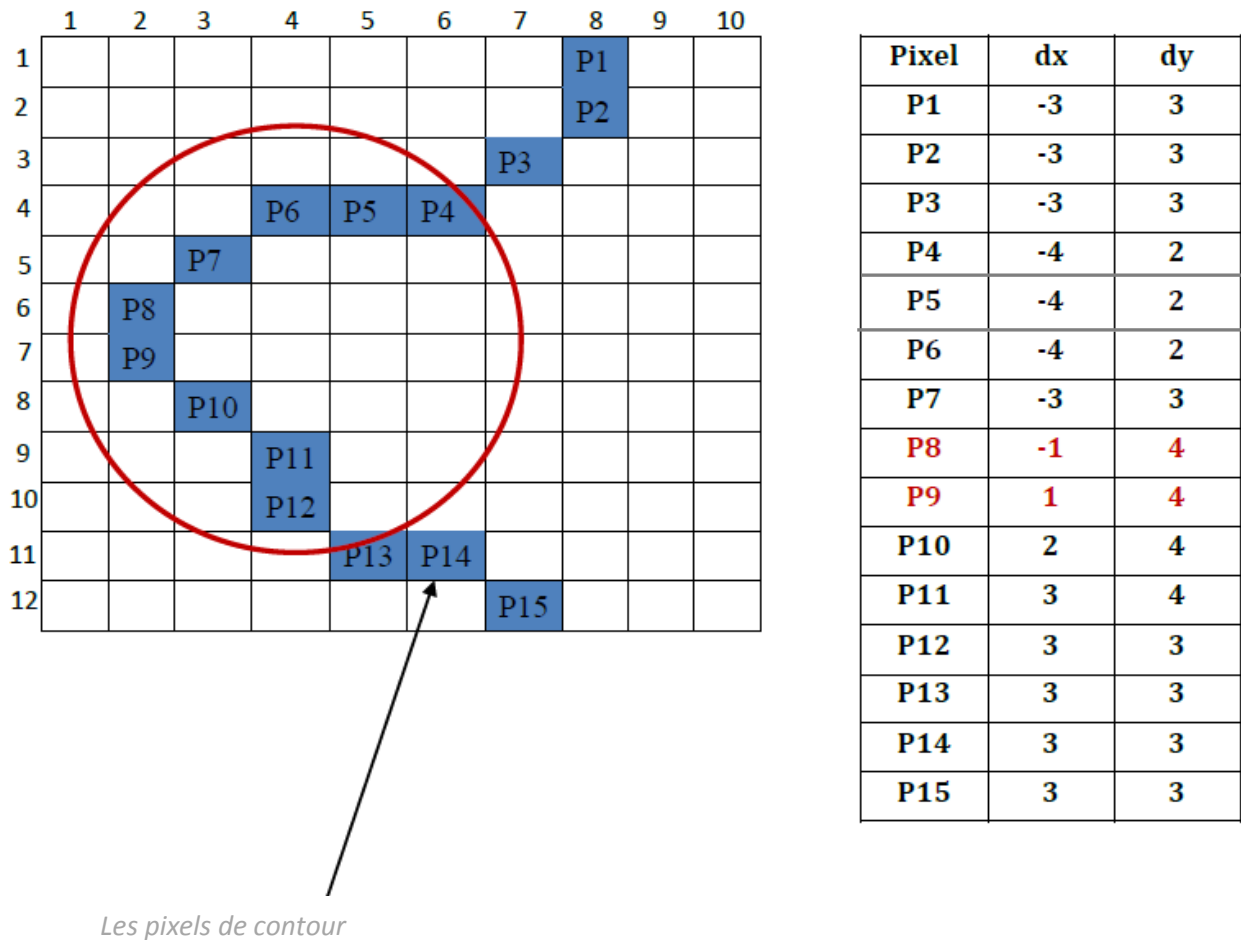


Figure I-4 : le schéma de dx et dy changeant de signe (traversant zéro) pour la détection de CPP forts.

Un CPP Faible est aussi un point d'une forme en saillie qui a une trace de contour. Même si les différences des deux côtés du CPP faible ne sont pas assez grandes pour être traitées comme un changement de monotonie, la saillie est visible et la continuité le long de la trace est cassée. Mais, les changements de CPP faibles ne peuvent pas être découverts selon un schéma Traversant zéro. Dans la Figure I-5, nous voyons que P9 ou P10 est un point perceptuel de saillie, mais les signes des valeurs de leurs dx et dy le long de la trace de contour sont inchangés. La méthode de détection utilisée pour les CPP forts ne fonctionne pas pour les CPP faibles.

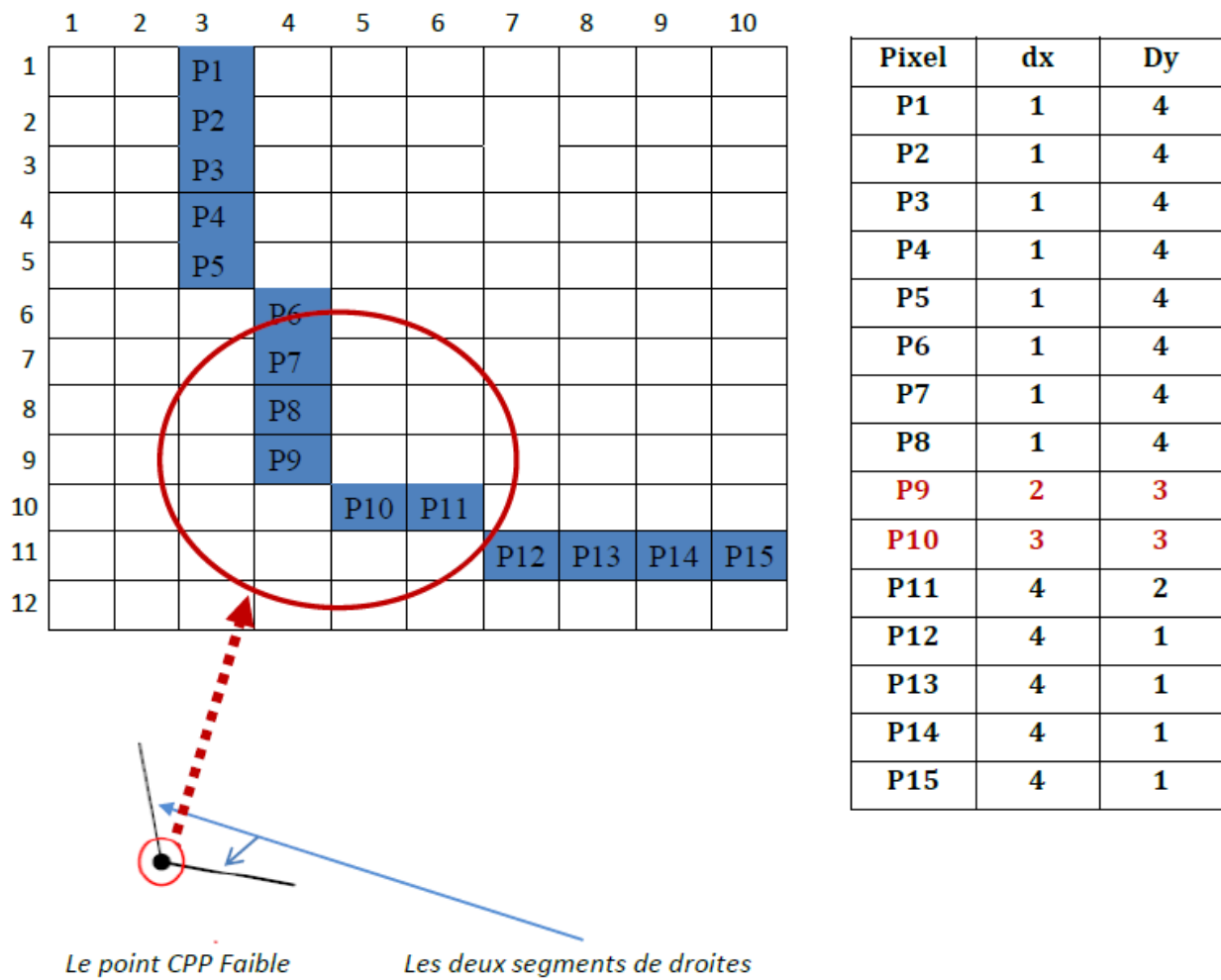


Figure I-5 : les changements de CPP faibles ne peuvent pas être découverts selon un schéma Traversant zéro.

4. 3. Séquence de Boîte Préservant l'Ordre Arc-tangente

Pour détecter les CPP faibles, nous présentons une nouvelle représentation d'histogramme qui utilise des Arc-tangentes dérivés de dx et de dy et préserve l'ordre de pixels sur la courbe. Cette représentation peut réarranger les données pour signifier la preuve des CPP faibles. La formule convertissant dx et dy dans une valeur d'Arc-tangente est :

$$\text{degré_Arctg} = \begin{cases} \arctg\left(\frac{dy}{dx}\right) \times \frac{180}{\pi}, & dx \neq 0 \\ 90 & dx = 0 \text{ } pre_arctg \geq 0 \\ -90 & dx = 0 \text{ } pre_arctg < 0 \end{cases} \quad (3)$$

Où *pre_arctg* représente la valeur du degré Arc-tangente pour le pixel d'ordre précédent dans la séquence.

En appliquant cette formule (Eq. 3), les pixels de contour le long d'une courbe sont convertis en séquence de degré Arc-tangente. Par exemple, la Figure I-6 (a) et (b) montre un polygone et respectivement sa correspondante séquence de degré Arc-tangente. L'axe Y dans la Figure I-6 (b) représente la valeur du degré Arc-tangente d'un pixel et l'axe X est l'ordre de séquence de pixels de contour. Les nombres nommés sur la trace de contour sont les points en saillie que la vision humaine peut percevoir et leurs positions correspondantes sur le côté droit de la séquence Arc-tangente sont marquées en conséquence. Comme nous pouvons voir, les valeurs de degré Arc-tangente entre chaque 2 points en saillie marqués le long de la séquence chute dans une gamme limitée autour d'une valeur particulière de degré parce que leurs pixels de contour correspondants ont des pentes semblables. Pour le contour de cercle dans la Figure I-6 (c), la majorité des valeurs de degré Arc-tangente dans la Figure I-6 (d) entre 2 points en saillie sont présentées comme une ligne penchée selon que les pentes de leurs pixels de contour correspondants changent progressivement dans une certaine tendance (la diminution).

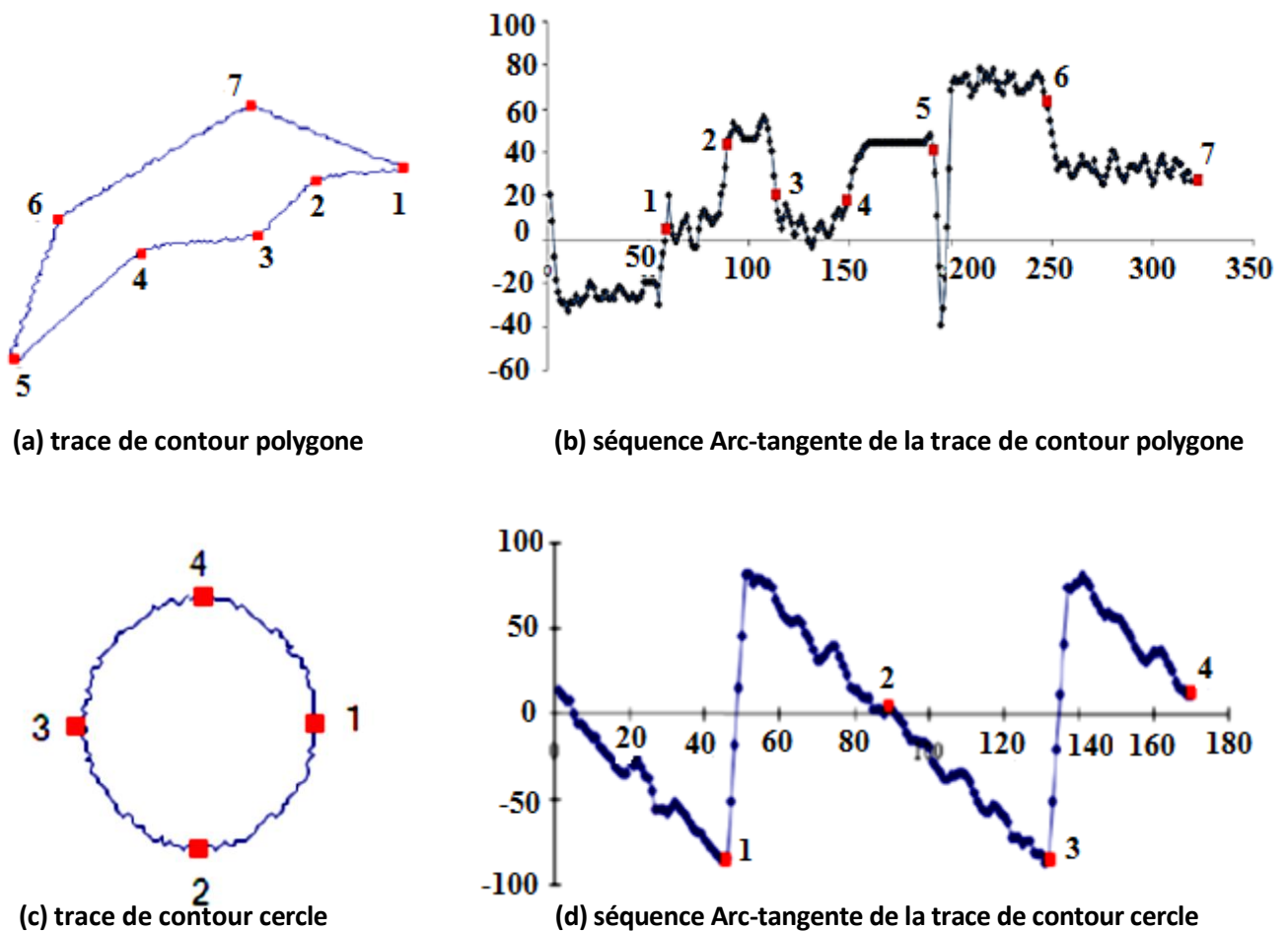


Figure I-6 : les traces de contour et leurs séquences Arc-tangentes correspondantes.

Il est important de noter que, sur la trace de contour du polygone (la Figure I-6 (a)), les points 2, 3, 4 et 6 sont des CPP faibles qui ont la preuve aussi évidente que les CPP forts ont dans l'espace Arc-tangente. L'ordre de degré Arc-tangente reflète bien les propriétés de forme et donne une différente vue moins d'ambiguïté pour la détection de CPP.

Sur la base des propriétés de la séquence de degré Arc-tangente, nous proposons l'histogramme d'une Séquence de Boîte Préservant l'Ordre d'Arc-tangente (OPABS) comme une représentation pour une meilleure détection de CPP. Les valeurs de degré d'Arc-tangente sont uniformément classifiées dans huit catégories séparées dont chacune couvre une gamme de pentes (la Table 1).

Table 1 : Catégories de valeurs de degré Arc-tangente.

Catégories	Cat-0	Cat-1	Cat-2	Cat-3	Cat-4	Cat-5	Cat-6	Cat-7
gamme de pentes	-90, -67.5	-67.5, -45	-45, -22.5	-22.5, 0	0, 22.5	22.5, 45	45, 67.5	67.5, 90

Pour n'importe quelle trace de contour, les degrés Arc-tangente sont calculés pour tous les pixels et sont ensuite cartographiés par catégories de boîte correspondantes selon la Table 1, et enfin l'histogramme OPABS est agrégé en conséquence. Par exemple, la trace de contour dans la Figure I-5 contient un CPP faible. La cartographie et les résultats d'agrégation sont illustrés dans la Table 2 et la Table 3. Il y a trois types de renseignements dans la Table 3 : l'ordre de boîte, la catégorie d'Arc-tangente et la taille de boîte.

Table 2 : les degrés d'Arc-tangente et leurs catégories de l'exemple de trace dans la Figure I-5.

Pixel	dx	dy	Les degrés d'Arc-tangente	Catégories d'Arc-tangente
P1	1	4	75.96389	Cat-7
P2	1	4	75.96389	Cat-7
P3	1	4	75.96389	Cat-7
P4	1	4	75.96389	Cat-7
P5	1	4	75.96389	Cat-7
P6	1	4	75.96389	Cat-7
P7	1	4	75.96389	Cat-7
P8	1	4	75.96389	Cat-7
P9	2	3	56.30995	Cat-6
P10	3	3	45	Cat-6
P11	4	2	26.5651	Cat-5
P12	4	1	14.03626	Cat-4
P13	4	1	14.03626	Cat-4
P14	4	1	14.03626	Cat-4
P15	4	1	14.03626	Cat-4

Table 3 : une séquence de boîte après agrégation basée sur la Table 2.

ordre	Catégorie	Pixels	Taille de la boîte
1	Cat-7	P1-P8	8
2	Cat-6	P9-P10	2
3	Cat-5	P11	1
4	Cat-4	P12-P15	4

Il y a 4 boîtes dans la Table 3 et leurs catégories Arc-tangente sont les Cat-7, Cat-6, Cat-5 et Cat-4 respectivement. La première et la dernière boîte contiennent plus de pixels, alors que la deuxième et la troisième boîte ont seulement 2 et 1 pixel respectivement. Dans la Figure I-7 (a), c'est la boîte de séquence à 3D dérivée de la Table 3. L'axe X est l'ordre de boîte et l'axe-Y représente les catégories d'Arc-tangente de cat-0 à cat-7. L'axe Z est la taille de la boîte. La forme le long de la séquence de boîte ressemble à une vallée, avec 2 sommets (pics) sur les deux extrémités. Si nous projetons cet histogramme en 3D dans une vue d'ordre de taille (x-z) en 2D, nous aurions une image plus claire de cette forme de vallée de la séquence de boîte en respectant la taille de la boîte (la Figure I-7 (b)). Pendant la construction d'OPABS, la distribution de la densité de pixel est déterminée seulement par les données Arc-tangente.

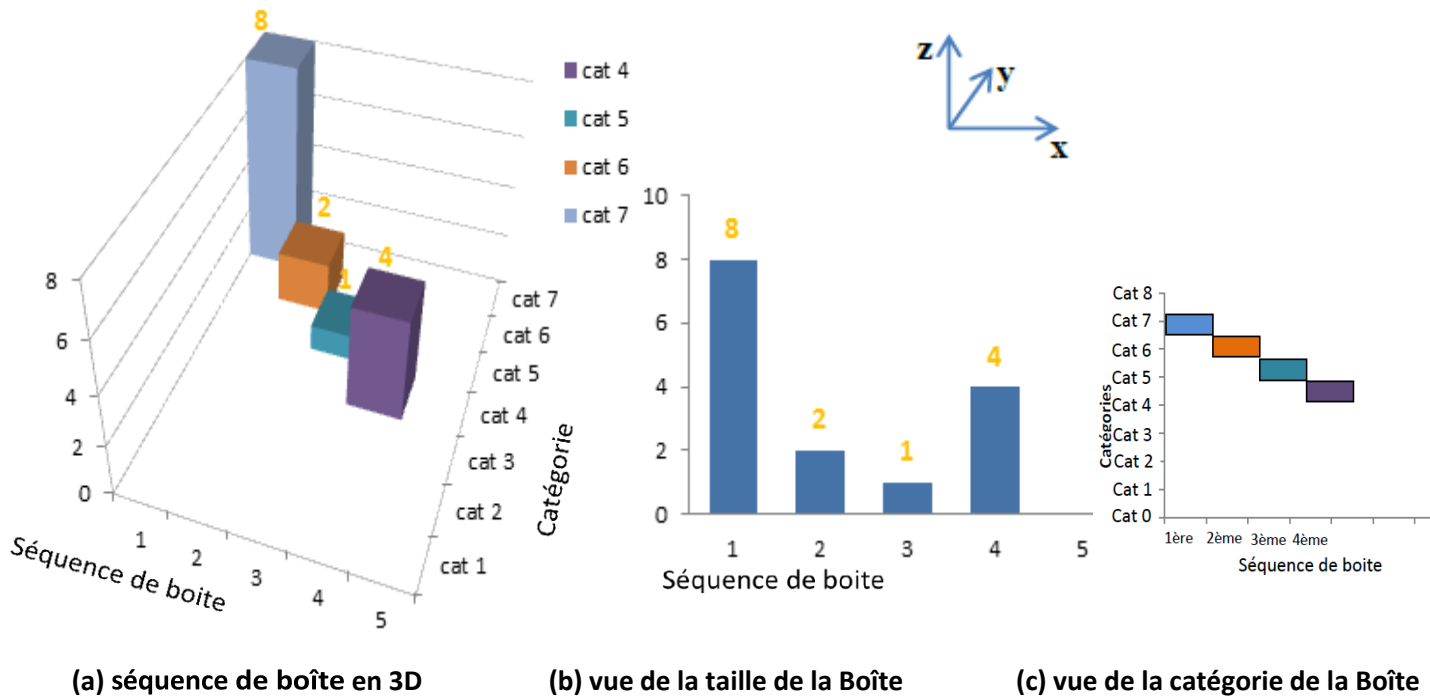


Figure I-7 : séquence de barre de colonne à 3D pour la séquence des boîtes et ses vues projetées en 2D pour le contour de la figure I-5.

Dans la Figure I-7 (b), les 2ème et 3ème boîtes avec de plus petites tailles sont en fait la région de transition de la trace de contour. Projetez ce graphique en 3D (la Figure I-7 (a)) dans une vue d'ordre de catégorie (x-y) en 2D, la tendance changeante d'Arc-tangente est clairement présentée dans la Figure I-7 (c), les valeurs d'Arc-tangente sont diminuées de Cat-7 à Cat-4 le long de l'ordre de boîte. Cette représentation d'histogramme OPABS a les propriétés suivantes :

- Les points de données dans une boîte partagent la même catégorie inclinée.
- Seulement les pixels consécutifs partageant la même catégorie inclinée seront groupés dans une boîte et la séquence d'ordre de pixel est toujours préservée dans une boîte.
- La taille et la catégorie d'Arc-tangente le long de la séquence de boîte reflètent les propriétés de la forme de contour.

4. 4. Détection de CPP Faible

Notre but est de découvrir efficacement les CPP faibles, qui sont les points d'interruption de la propriété de continuité de la forme. D'après les exemples précédents de séquence de boîte, nous savons que chaque boîte est un segment d'une trace qui contient des données partageant une même propriété, c'est-à-dire une catégorie de degré Arc-tangente. Donc chaque endroit entre 2 boîtes voisines est la place de la transition entre 2 ensembles de pixels de contour avec différentes propriétés et c'est la position possible de CPP. Il est important de noter que pas chaque boîte est un segment valide selon notre définition du GET parce que les différences entre les boîtes adjacentes ne signifient pas que la discontinuité arrive. Il y a 2 types de continuité sur la séquence de boîte. Le premier est les tailles des boîtes le long d'une séquence de boîte. Si quelques tailles de boîte sont relativement semblables, ou, continuent à augmenter ou à diminuer progressivement, cela signifie qu'il n'y a aucun changement, ou, aucun changement soudain en termes de tailles. L'autre type de continuité concerne les catégories d'Arc-tangente. Etant donné que les catégories Arc-tangente représentent les pentes de la courbe de contour, la tendance changeante des catégories Arc-tangente reflète les changements inclinés d'une forme de contour. Si la tendance changeante de catégories demeure consistante, que ce soit diminution ou augmentation, ou inchangée, la continuité est préservée. Autrement, le contour devrait être partagé à cause de la discontinuité.

Ayant extrait une séquence de boîte d'une trace de contour, nous découvrons les CPP faibles suivant les propriétés d'une séquence de boîte. Il y a deux critères pour trouver les CPP.

Sommet - vallée – Sommet

Le sommet (Pic) et la Vallée sont les termes pour décrire les tailles relatives des boîtes adjacentes. S'il y a une forme sommet-vallée-sommet formée sur la séquence de boîte, le CPP pourrait se positionner dans la boîte de vallée et le pixel du milieu de la boîte de vallée est le CPP. La forme sommet-vallée-sommet signifie que, au commencement, beaucoup de valeurs de degré Arc-tangente de pixels de contour appartiennent à une catégorie, ainsi un côté du sommet est formé. plutard, les propriétés (les pentes) du plus petit nombre de pixels suivants ont changé en d'autres catégories (boîtes). Comme le nombre de pixels dans cette (ces) catégorie(s) est beaucoup plus petit, une falaise est formée. Alors les pentes de pixels suivants sont changées de nouveau et le tout tombe dans une autre boîte avec une grande population de pixel. Ainsi, le long de la séquence de boîte, un autre sommet apparaît jusqu'à former une forme Sommet-Vallée-Sommet complète. Les pixels dans la/les boîte(s) de vallée mènent le changement. Un ratio métrique Sommet-Vallée est utilisé pour la détection:

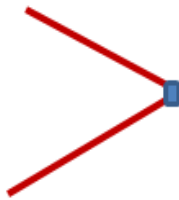
$$\text{Sommet – vallée – ratio} = \frac{\text{la taille de la boîte de sommet}}{\text{la taille de la de boîte de vallée}} \quad (4)$$

Où la taille de la boîte de sommet et la taille de la de boîte de vallée sont les nombres de pixel de boîte. Un seuil entraîné est utilisé pour déterminer la décision de segmentation. Le ratio Sommet-Vallée devrait être utilisé pour vérifier les deux côtés d'une vallée. Une fois la forme Sommet-Vallée-Sommet a été confirmée, le CPP doit être quelque part dans la/les boîte(s) de vallée. Etant donné que dans la plupart du temps, la région de vallée a peu de pixels, peu importe quel pixel dans la boîte de vallée est choisi comme CPP il n'affectera pas la performance de segmentation générale. Juste pour la simplicité, nous prenons le pixel du milieu des boîtes de vallée comme CPP. La Figure I-8 (a-d) montre un exemple.

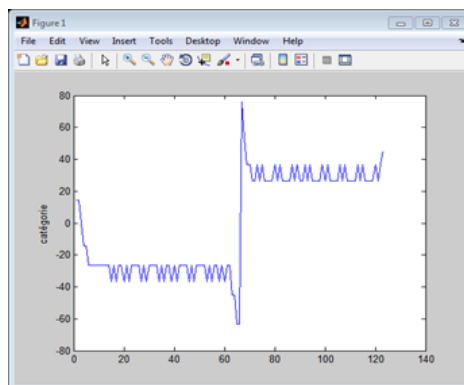
Le Changement de la Tendence de Catégories Arc-tangente

En plus du changement de la taille de boîte, la tendance changeante de la catégorie Arc-tangente est utilisée pour la détection de CPP faibles. Si la tendance changeante de la catégorie Arc-tangente reste inchangée, la continuité est préservée. Autrement, la continuité est interrompue. Plus précisément, si la tendance changeante de la catégorie Arc-tangente augmente (de Cat-1 à Cat-7), ou diminue (de Cat-7 à Cat-1), la continuité est préservée, autrement, la continuité est cassée et un CPP peut être découvert dans la boîte critique ou ses boîtes voisines. Voir l'exemple dans la Figure I-8 (e-h), une vue d'ordre de taille en 3D de la séquence de boîte, de la

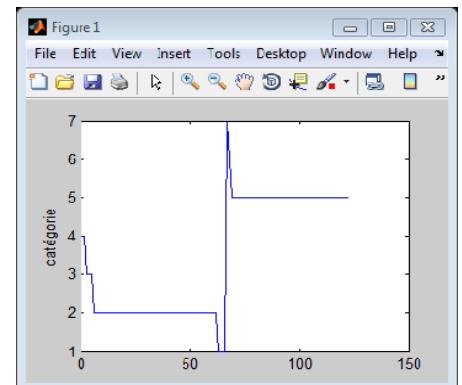
gauche vers la droite, en tout la taille continue à diminuer jusqu'à l'atteinte de la 7ème boîte. Alors la taille de la boîte a un grand saut en hauteur sur le 7ème endroit de la boîte. Mais la forme de la boîte n'est pas un Sommet-Vallée-Sommet car la taille changeante sur le côté gauche de la 6ème boîte diminue progressivement et le ratio de taille (Eq. 4) est probablement moins qu'un seuil défini. Ainsi, il ne remplit pas la condition du schéma Sommet-Vallée-Sommet. Pourtant la discontinuité peut être trouvée sur l'autre vue de catégorie Arc-tangente en 2D. Les catégories Arc-tangente le long de la séquence de boîte sont Cat-3, Cat-2, Cat-1, Cat-0, Cat-1, Cat-2, Cat-3, Cat-4 de la vue de boîte de catégorie en 2D dans la Figure I-8 (h). La tendance changeante est cassée aux 4ème et 5ème endroits de boîte. Les catégories de boîte Arc-tangente diminuent du Cat-3 au Cat-0 et commencent à augmenter du Cat-0 au Cat-4 et ensuite les pixels majoritaires chutent dans la 7ème boîte avec Cat-3. La continuité de la tendance changeante des valeurs de degré Arc-tangente est cassée aux 4ème et 5ème boîtes. Donc le CPP devrait être trouvé entre la 4ème et la 5ème boîte. Etant donné que l'ordre original de pixel est persévéré dans chaque boîte, le dernier pixel de la 4ème boîte, ou le premier pixel de la 5ème boîte, serait l'exact CPP dans ce cas-là.

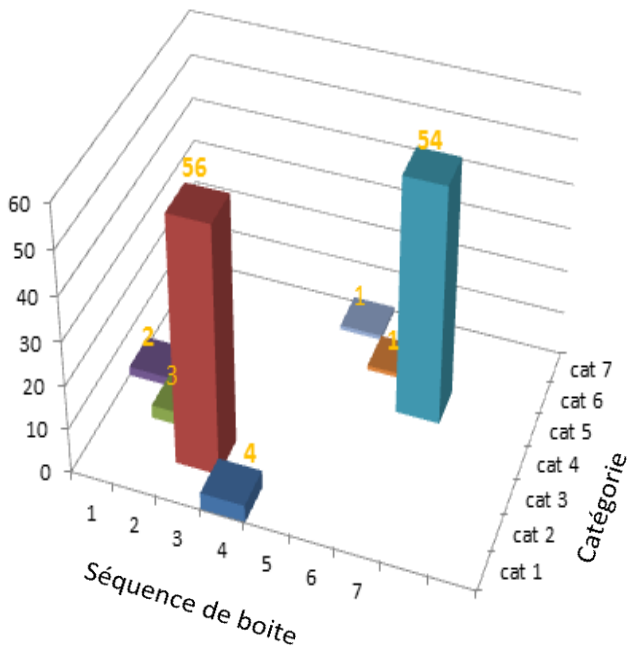


(a)

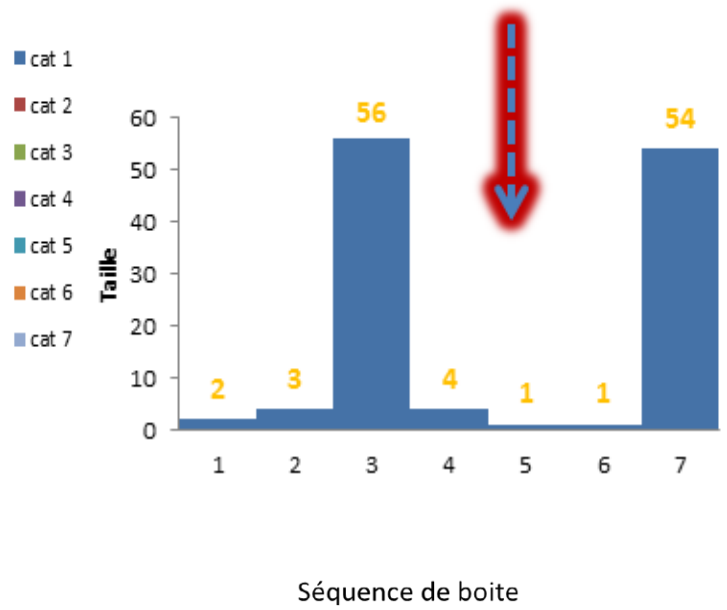


(b)





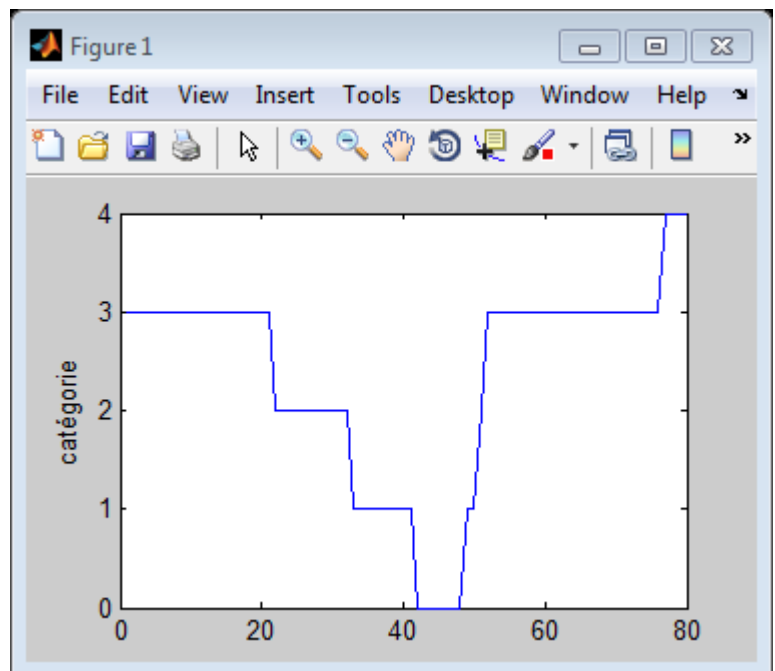
(c)



(d)



(e)



(f)

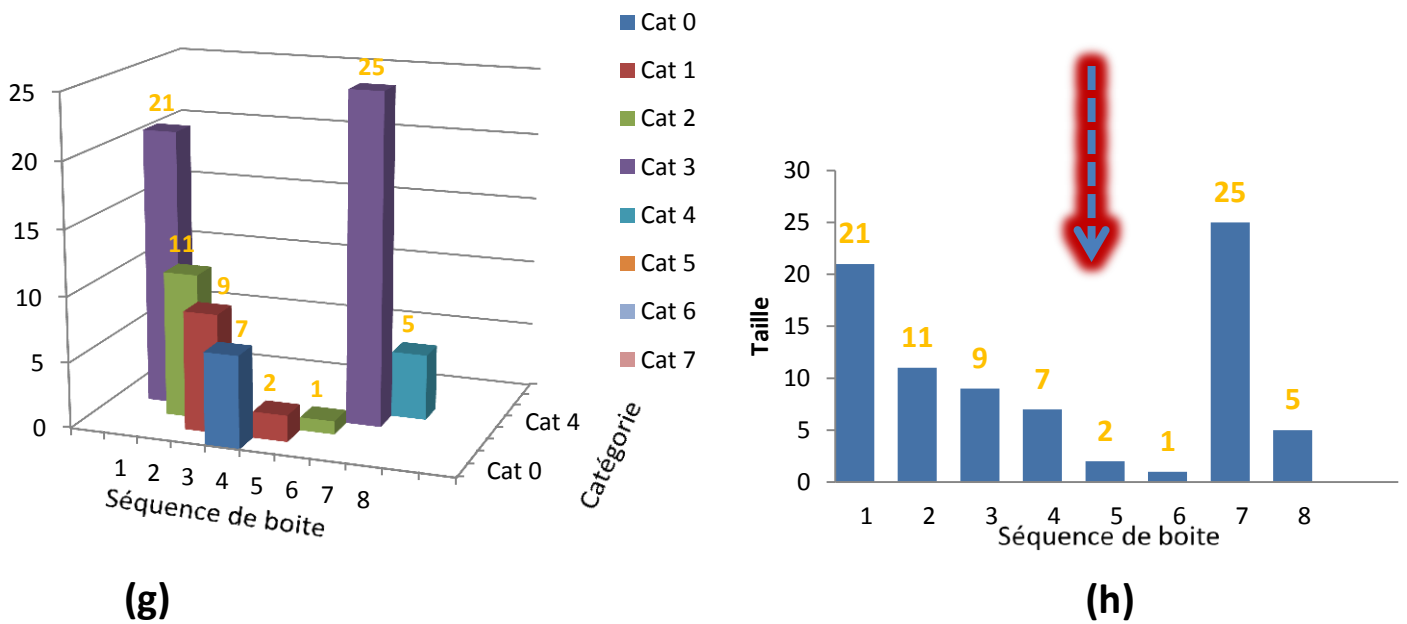


Figure I-8 : deux critères de segmentation de boîtes.

De ce schéma OPABS, l'endroit de CPP faible peut être déterminé sur la base des statistiques globales le long de la courbe en analysant ses vues projetées en 2D. La méthode de détection profite des informations sur l'ordre séquentiel de pixel, et elle utilise les tendances tant des tailles de boîtes que des catégories Arc-tangente et utilise les vues tant globales que locales. Ainsi, elles sont plus génériques pour les différents types d'images et robustes au bruit. Voici l'algorithme de cette méthode de détection.

Algorithme 1. Détection de CPP et partitionnement de la courbe	
1.	Scanner l'image par intervalle et mettez tous les candidats de pixels de contour dans un tableau_candidat,
2.	Pour chaque candidat de pixel de contour dans le tableau_candidat Faire
3.	Trouver un tracé de contour, et enregistrez les pixels de contour dans un tableau de tracé de pixels.
4.	Pour tous P(i) dans le tableau de tracé de pixels Faire
5.	mettre en place les signes initiaux de dx et dy
6.	vérifier les signes de dx et dy pour chaque P(i)
7.	Si le signe change alors
8.	Considérer ce point comme un CPP fort, mettre dans un tableau_CPP
9.	construire les OPABS pour pixels entre 2 CPP forts
10.	vérifier l'espace de l'ordre_taille des OPABS
11.	Si il existe une vallée d'une certaine taille le long de l'ordre de boîte alors
12.	partitionner la séquence des boites, enregistrer dans un tableau_segment_boite
13.	Fin Si
14.	Pour chaque segment_boite[j] dans tableau_segment_boite Faire
15.	Vérifier l'ordre_catégorie de la séquence de boites (segmentée)
16.	Si la tendance de la catégorie est changée alors
17.	segment_boite[i] est encore partitionné
18.	Fin Si
19.	Classification pour chaque sous séquence de boites partitionnée
20.	Fin Pour
21.	Fin Si
22.	Fin Pour
23.	Fin Pour

Figure I-9 :l'algorithme de la détection des CPP.

Conclusion

Extraire le niveau de caractéristique de la saillie visuelle est essentiel pour construire une représentation attirant l'attention visuelle. Dans ce chapitre, nous avons d'abord présenté une méthode statistique non-paramétrique afin d'exécuter et d'améliorer le modèle de partition de courbe basé sur un PO pour l'extraction de caractéristiques de saillie à base de forme en 2D. Dans ce modèle de PCPG, les critères de perception génériques de la partition et du regroupement de courbe sont simulés en utilisant les modèles de pixels de courbe consécutivement ordonnés et la statistique de boîte de la propriété inclinée. Les segments partitionnés sont facilement classifiés en ligne droite ou en jetons de courbe en utilisant les mêmes critères génériques produits pendant la partition de courbe (détails dans le chapitre suivant). Les expériences montrent sa force pour manipuler efficacement et avec efficacité des images complexes. Les types de GET et CPP sont des éléments de forme utiles pour aider à l'informatisation sémantique de l'image et peuvent être prolongés davantage par le groupement des GET et des CPP pour construire une langue d'image complète pour l'encodage (codification) de contenu basé(e) sur la forme, pour la reconnaissance de modèles et pour l'indexation.

Dans le chapitre suivant nous allons procéder à la classification des formes issues du processus de segmentation, et pour se faire on doit d'abord attribuer chaque primitive (segment de droite et arcs) à sa propre classe, avec un tel agencement de ces primitives on construit une description structurelle de chaque forme, sur laquelle le système va s'appuyer pour reconnaître et classifier les formes.

CHAPITRE II

Classification et reconnaissance structurelle de formes

1. introduction et notions élémentaires

1.1. Aperçu historique

Dans le domaine de la reconnaissance des formes, les méthodes dites structurelles ou syntaxiques se sont développées à partir de 1965 environ. A l'origine, la reconnaissance structurelle a été introduite d'une part par les linguistes pour la compréhension des langues naturelle [21], et d'autre part par les informaticiens préoccupés par les langages de programmation et leurs compilations [22]. Aussi, le formalisme de la terminologie utilisé aujourd'hui en reconnaissance des formes s'inspire encore largement de cette époque.

L'analyse des structures, à l'origine linguistique, s'est alors rapidement étendue à la reconnaissance de structures plus générales notamment dans les domaines du traitement de l'image et de la parole [23]. Un des plus anciens exemples, et aussi un des plus connus, est celui de la classification automatique de chromosomes.

D'autres domaines d'application ont bénéficié de ces nouvelles méthodes, tels que la reconnaissance de caractères, de textures, des formules mathématiques, mais aussi l'analyse des signaux sismiques, d'électrocardiogrammes, ou encore l'étude des photos de chambres à bulles.

1. 2. Notion de structure

La notion de structure recouvre des concepts variés. Une structure doit décrire la composition d'un objet complexe en termes d'objets plus simples ainsi que les liens entre eux. Cette description est faite selon un modèle particulier, adapté aux objets considérés.

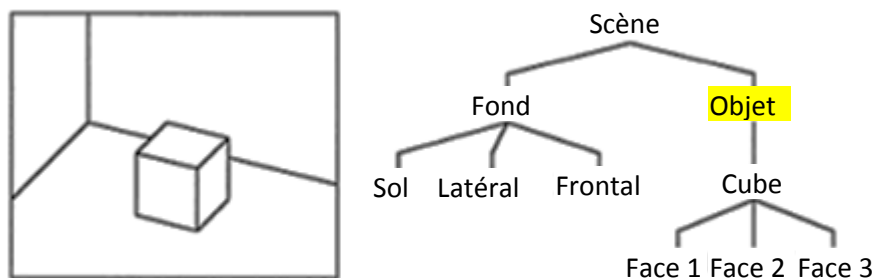


Figure II-1 : scène schématique (à gauche) et une description hiérarchique des surfaces présentes (à droite).

Le modèle hiérarchique qui décrit de manière récursive la décomposition d'un objet complexe en objets plus simples est très répandu en pratique. La principale raison en est probablement que les arbres qu'il engendre constituent une structure de données particulièrement facile à manipuler.

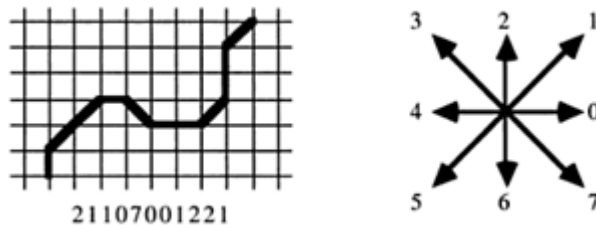


Figure II-2 : courbe discrétisée et sa représentation sous la forme d'une chaîne de chiffres (à gauche) conforme au codage défini (à droite).

Le modèle de structure le plus simple est la chaîne dont le principe consiste à présenter une forme comme une suite d'objets élémentaires. La présence ou l'absence d'un de ces objets, ainsi que leur ordre d'apparition caractérisent la forme globale.

Les chaînes jouent un rôle prépondérant dans l'étude théorique de la reconnaissance structurelle. En effet toute structure peut être représentée à l'aide d'une chaîne moyennant une convention de codage.

1. 3. Notion de syntaxe

La syntaxe est, à l'origine, la branche de la linguistique qui étudie la façon dont les mots se combinent pour former des phrases ou des énoncés dans une langue.

On distingue la syntaxe qui concerne les expressions (les mots), de la sémantique qui concerne ce qui est visé par les expressions (le sens, la signification).

Le terme de syntaxe est aussi utilisé en informatique, où sa définition est similaire. Ainsi, la syntaxe est le respect, de la grammaire formelle d'un langage, c'est-à-dire des règles d'agencement des lexèmes (en informatique, ce sont des entités lexicales d'un langage informatique) en des termes plus complexes, souvent des programmes. Dans la théorie des langages formels, ce qui joue le rôle de lexème est en général appelé lettre ou symbole, et les termes produits sont appelés mots.

1. 4. Le système d'inférence

L'inférence est un mouvement de la pensée allant des principes à la conclusion.

C'est une opération qui permet de passer d'une ou plusieurs assertions, des énoncés ou propositions affirmés comme vrais, appelés prémisses, à une nouvelle assertion qui en est la conclusion.

L'inférence peut prendre une forme probabiliste (inférence bayésienne) - il s'agit d'une opération logique portant sur des propositions tenues pour vraies (les prémisses) et concluant à la vérité d'une nouvelle proposition en vertu de sa liaison avec les premières. C'est pourquoi l'inférence est souvent réduite à la déduction nécessaire dans laquelle la vérité des prémisses assure totalement la vérité de la conclusion [24].

Cependant, pour les sémiologues, tels que Charles Peirce, la pensée n'opère pas sur des propositions, mais sur des signes. Ils proposent ainsi un modèle structuraliste, centré sur les relations formelles du langage, qui s'inspire de la linguistique [25].

1. 5. Règles d'inférence

Dans un système logique, les règles d'inférence sont les règles qui fondent le processus de déduction, de dérivation ou de démonstration. L'application des règles sur les axiomes du système permet d'en démontrer les théorèmes. Une règle d'inférence est une fonction qui prend un n -uplet de formules et rend une formule. Ses arguments sont appelés « les prémisses » et sa valeur la « conclusion ». Les règles d'inférence peuvent également être vues comme des relations liant prémisses et conclusions par lesquelles une conclusion est dite « déductible » ou « dérivable » des prémisses. Si l'ensemble des prémisses est vide, alors la conclusion est appelée un « théorème » ou un « axiome » de la logique.

Les règles d'inférences sont en général données dans la forme standard suivante :

Prémisse n° 1

Prémisse n° 2

...

Prémisse n° n

Conclusion

Cette expression dit que si on se trouve au milieu d'une dérivation logique, où les prémisses ont été déjà obtenues (c'est-à-dire dérivées logiquement des axiomes), alors on peut affirmer que la conclusion est obtenue. Le langage formel utilisé pour décrire les prémisses et la conclusion dépend du système formel ou logique où l'on s'est placé. Dans le cas le plus simple, les formules sont tout simplement des expressions logiques, c'est ainsi le cas pour le modus ponens :

$$\frac{A \rightarrow B \quad A}{\therefore B}$$

Les règles d'inférence peuvent aussi être formulées de cette manière :

1. un certain nombre de prémisses (peut-être aucune).
2. un symbole de dérivation \vdash signifiant « infère », « démontre » ou « conclut ».
3. une conclusion.

Cette formulation représente habituellement la vision relationnelle (par opposition à fonctionnelle) d'une règle d'inférence, où le symbole de dérivation représente une relation de démontrabilité existant entre prémisses et conclusion [26].

1. 6. Dédution logique

La déduction logique est un type de relation que l'on rencontre en logique mathématique. Elle relie des propositions dites *prémisses* à une proposition dite *conclusion* et préserve la vérité. Prémisses et conclusion qui sont ainsi reliées par une règle de déduction, assurent que si la règle est valide et si les prémisses sont vraies, la conclusion est elle aussi vraie. On dit alors que la conclusion est une conséquence des prémisses, ou parfois que la conclusion vient des prémisses. Une déduction logique est définie de manière à être à la fois nécessaire et formelle et est explicitée dans des domaines comme la théorie des modèles, qui permet de trouver des univers mathématiques dans lesquels la relation est utile et fournit un sens aux formules, et la théorie de la démonstration, qui fournit un cadre théorique pour sa définition de manière syntaxique. Une formule est une conséquence d'un ensemble d'autres formules, dans un langage, si et seulement si, en utilisant la logique elle-même (c'est-à-dire sans chercher à donner un sens aux formules) la formule doit être vraie si toutes les formules de l'ensemble des prémisses sont elles aussi vraies [27] [28].

1. 7. Choix des primitives

La première étape dans la modélisation d'un problème de reconnaissance est la détermination d'un ensemble de primitives qui permettront de décrire les formes complexes à reconnaître. Il n'existe à ce jour aucun modèle universel pour la sélection de primitives, qui sont généralement choisies en fonction de deux critères :

Elles doivent favoriser la représentation des formes complexes en termes de relation structurelle.

Elles doivent être faciles à déterminer ou à classer, leurs extractions n'utilise en règle générale aucune information relative à la description structurelle.

Des exemples ont été déjà donnés dans le paragraphe 3.1, des surfaces uniformes pour l'analyse de scène, des segments unitaires pour la représentation de courbes. Des segments de courbes caractérisés par leurs longueurs et leurs courbures sont parfois utilisés pour représenter des courbures fermées.

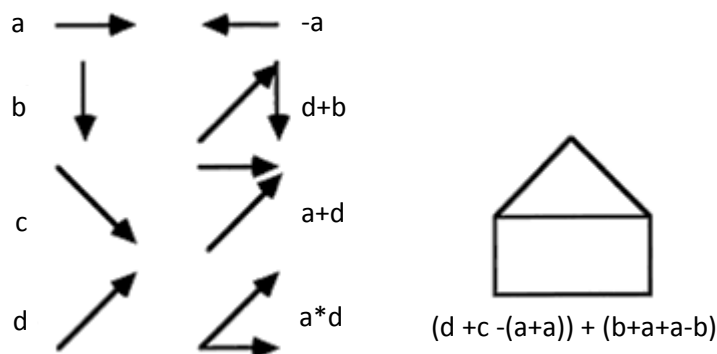


Figure II-3 : ensemble de primitives (à gauche) permet de représenter chaque graphique (à droite).

La figure II-3 représente un ensemble de primitives (à gauche) pour la reconnaissance de graphes planaires, utilisé en conjonction avec les opérateurs de connexion (au milieu), ce modèle permet de représenter chaque graphique (à droite) par une chaîne de caractères.

1. 8. Classification

Le problème de la classification est celui de l'assignation d'une forme à une classe. Dans l'approche structurelle, une classe est décrite par une grammaire. Une grammaire est composée d'un ensemble de règles syntaxiques qui déterminent l'ensemble des formes admissibles dans

cette classe et présentant en principe des caractéristiques structurelles communes. Cet ensemble de formes est appelé langage engendré par la grammaire.

Etant donné une forme particulière et un ensemble de grammaires, l'objectif de la classification est de déterminer si l'une des grammaires permet d'engendrer cette forme. L'algorithme utilisé pour répondre à cette question peut être décrit par un automate, et l'intérêt réside dans le fait qu'il existe des méthodes permettant de construire cet automate directement à partir de la grammaire.

Lorsqu'aucune grammaire ne permet d'engendrer la forme examinée, celle-ci est considérée comme inclassable. Lorsque plusieurs grammaires peuvent l'engendrer, nous sommes en présence d'une ambiguïté qui peut être levée par différentes techniques. Il est parfois possible de définir un ordre de domination sur l'ensemble des classes, dans ce cas la forme sera attribuée à la classe dominante. Mais on utilise aussi des méthodes probabilistes plus sophistiquées basées sur un modèle de grammaires stochastiques.

2. Classification de GET

La classification de Jeton de Contour Générique (GET) est un autre but de l'approche d'OPABS. Après qu'une trace de contour a été partagée en segments génériques, nous devons identifier leurs types. Selon le modèle de PCPG, il y a huit types de GET qui peuvent être généralisés davantage (classification de premier niveau) dans 2 classes perceptuelles: GET en arc et GET en segment de droite. Donc nous pouvons avoir une structure de classification hiérarchique des GET (la Figure II-4). Nous nous concentrons sur la classification de premier niveau, c'est-à-dire identifier les GET en courbe et les GET en ligne droite. Une fois les types de GET en courbe et en ligne droite ont été classifiés, la classification plus avancée de GET pourrait être facilement faite en vérifiant les propriétés topologiques d'un petit nombre de pixels.

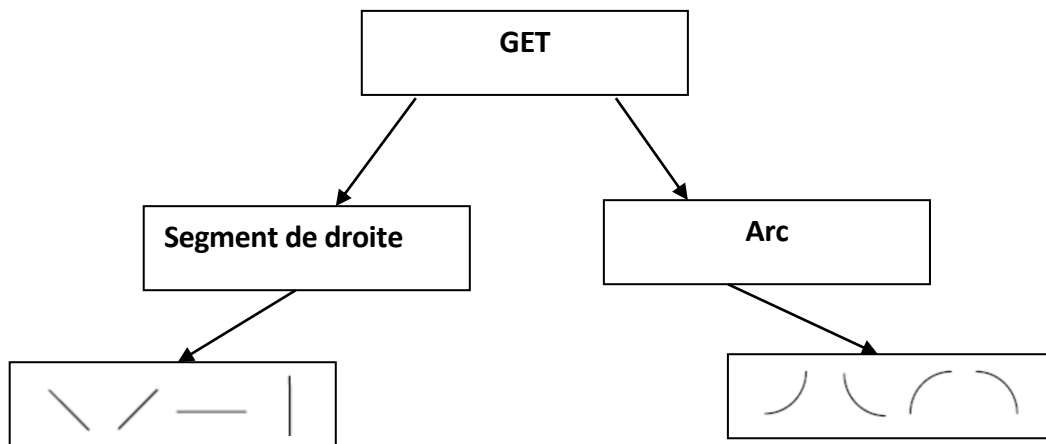


Figure II-4: une structure de classification hiérarchique des GET (classification de premier niveau).

2. 1. Ligne Droite

Pour le GET en ligne droite, ses pixels ont la pente semblable et appartiennent à une même catégorie Arc-tangente. Ainsi pour sa séquence de boîte, une boîte doit avoir une grande taille et le reste d'entre elles est avec beaucoup moins de population de pixels, que l'on traite comme des données bruyantes (la Figure II-5).

2. 2. Courbe

Pour le GET en courbe, les pentes de pixels soit elles augmentent ou elles diminuent monotonement et les changements de taille de ses boîtes de degré Arc-tangente s'effectuent doucement. Par exemple, dans la Figure II-6, l'importante tendance changeante de la catégorie des boîtes est Cat-7, Cat-6, Cat-5 et Cat-4, dans une tendance diminuante (d'abord la boîte de catégorie 3 est ignorée comme sa taille est relativement petite, elle est traitée comme bruit). Il n'y a aucune boîte Sommet-Vallée-Sommet sur la séquence. Donc c'est une courbe.

Après la segmentation, il y a seulement 2 types de segments de séquence de boîtes, l'un est avec un sommet sur sa séquence de boîte, l'autre est sans boîte de sommet, mais il a les boîtes avec la même tendance changeante de catégories Arc-tangente le long de la séquence de boîte. Ainsi, la prise de décision pour la classification courbe /ligne droite est en fait fondée

simplement sur la validation du sommet dans une séquence de boîte segmentée. S'il y a un sommet dans une séquence de boîte, c'est une ligne droite, autrement, c'est une courbe.

la classification plus avancée de types courbe et ligne droite est détaillée dans la section 3.1

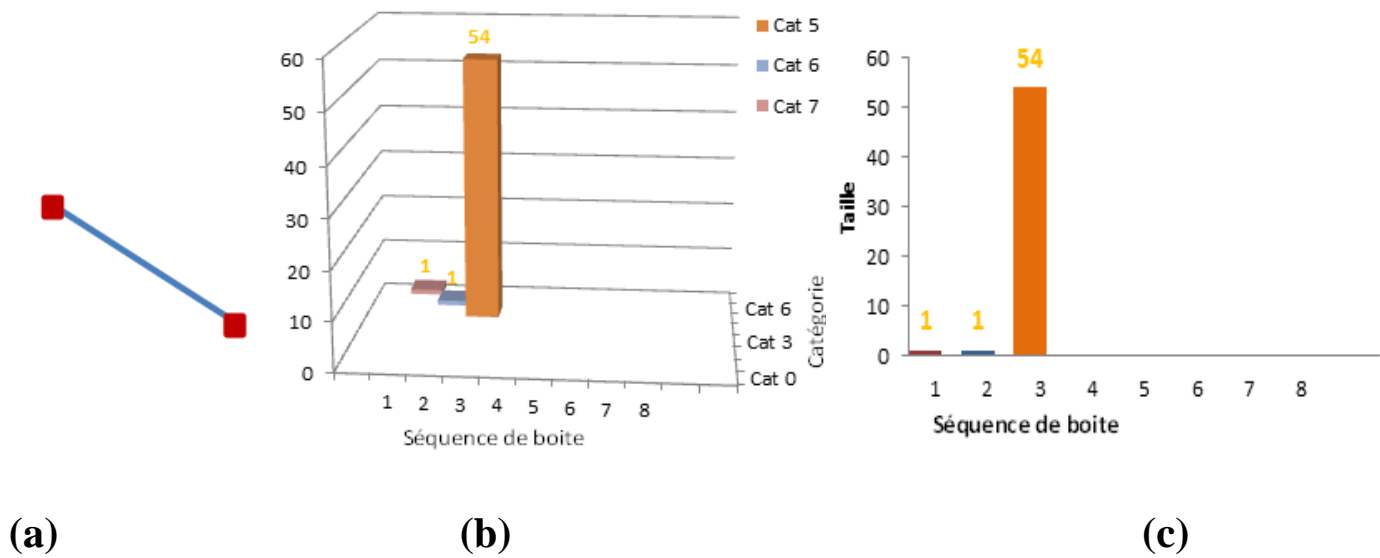


Figure II-5 : classification des segments de droite. (a) Segment de droite, (b) vue 3D de la séquence de boîte, (c) vue 2D de la séquence de boîte.

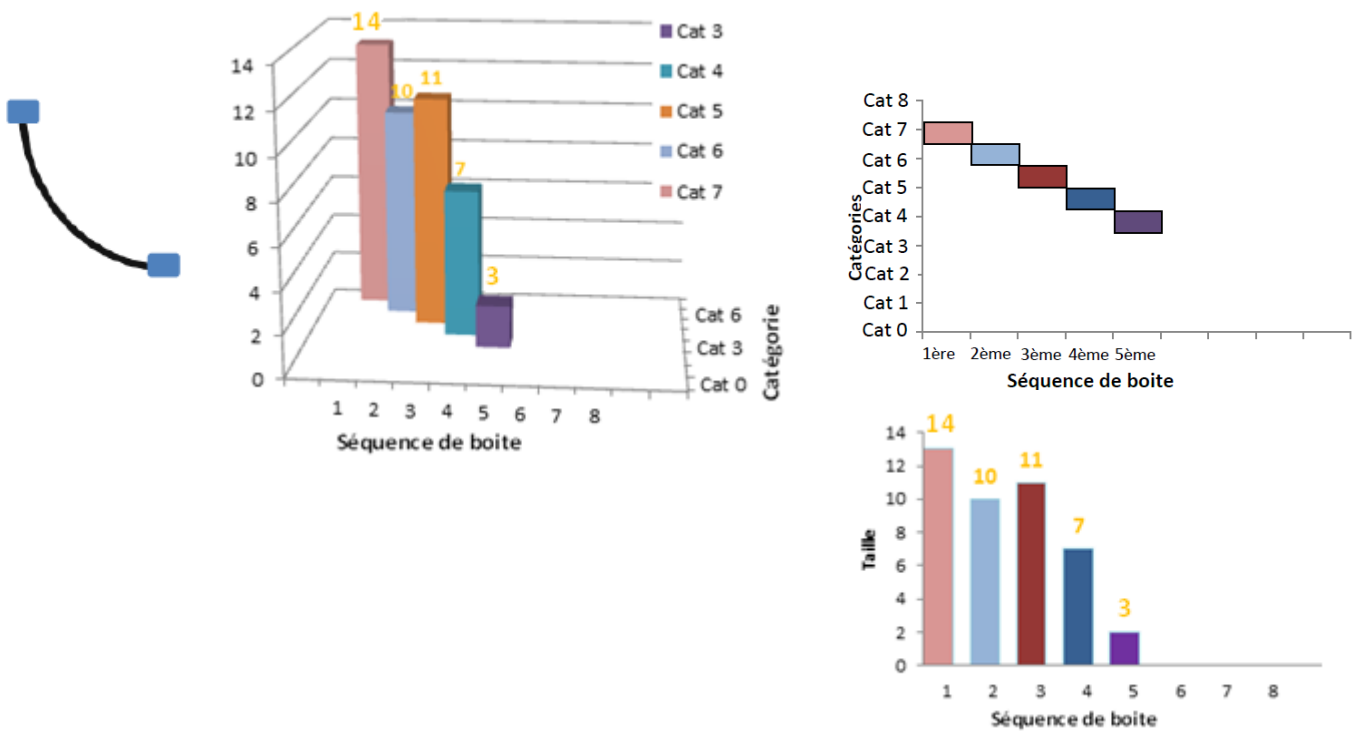


Figure II-6 : classification d'un arc. (Détails dans la section 1.2 de chapitre 3).

Dans ce qui suit on présente une méthode à base d'organisation perceptuelle pour la classification de deuxième niveau des caractéristiques du contour générique, appelés Jetons de Contour Générique (GET). Un GET est une primitive d'image perceptuelle signifiante qui représente une classe d'éléments de structure qualitativement équivalente. Un ensemble complet de GETs contient en même temps les classes de segments linéaires et non-linéaires et les jonctions des segments. Les tracés de contour qui sont extraits de l'image, en termes de géométrie de l'image descriptive et la force du contour, sont segmentés en GETs.

Dans la classification des formes en 2D, un objet est décrit par ses GETs créés et la relation entre les GETs. Un exemple de classification d'ellipse est présenté comme illustration de la méthode.

La classification perceptuelle, comme son prétraitement, est l'étude de la manière selon laquelle sont classées les caractéristiques en groupes perceptuels significatifs pour la reconnaissance de formes. La tâche semble très simple pour les systèmes biologiques, mais actuellement il est difficile de formaliser précisément et de décrire cette capacité naturelle. D'un point de vue computationnel (Calcul Informatique), la classification est un problème extrêmement difficile, et c'est certainement l'un des problèmes les moins compris dans la vision [29] [30]. Nous présentons dans ce qui suit une approche générique pour la représentation et la classification perceptuelle des formes en 2D. L'efficacité et la robustesse de la méthode sont montrées dans les résultats expérimentaux.

3. Représentation des GETs

3. 1. Les modèles de GETs

Les Jetons de Contour Générique (*GETs*) sont les primitives perceptuelles signifiantes d'images qui représentent une classe d'éléments de structure qualitativement équivalente. Un ensemble complet de GETs comprend à la fois les classes de segments linéaires et non-linéaires et les jonctions des segments.

Les segments génériques (*GSs*) ont été utilisés comme des jetons perceptuels. Les *GSs* peuvent être pistés en utilisant les tangentes et les tracés de courbes de contour d'image.

Les pixels initiaux de contour pour le pistage des contours sont déterminés en fonction de la dimension minimale des objets de l'application, puis les pixels de contour le long des tracés sont suivis selon les définitions des GSs. La description détaillée de l'algorithme peut être trouvée en [31]. Un modèle de calcul quantitatif des GSs peut être exprimé comme suit:

$$GS = \{p|\rho(p)\}$$

Par exemple **GS** est un ensemble de points satisfaisant quelque propriété ρ . Comme illustré à la Figure II-5(b), les segments génériques sont classifiés en huit catégories suivant la fonction tangente de **GS** $y=f(x)$ et sa fonction inverse $x=\phi(y)$. Les définitions pour ces segments génériques sont données dans le Tableau II-1, Nous allons donner leurs descriptions détaillées dans ce qui suit.

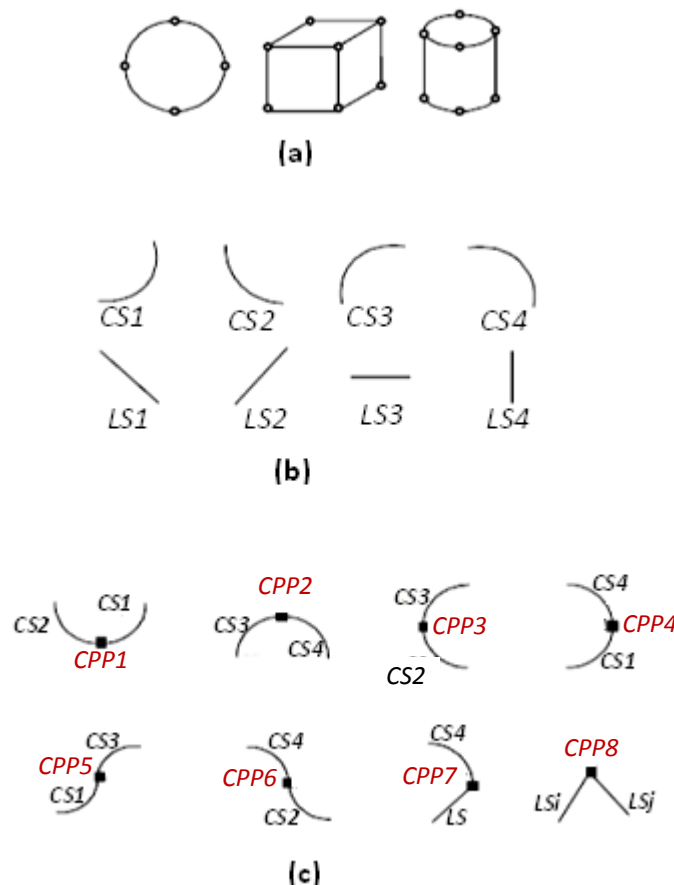


Figure II-7 : (a) des exemples de partition de la courbe. (b) les huit catégories des segments génériques. (c) les huit catégories des points de partition de la courbe.

Un Point partageant la Courbe (CPP) est un point perceptuel significatif dans lequel une transition de monotonie se déroule. La monotonie est évaluée le long d'un seul segment en terme de croissance monotone (désignée par « M + ») ou décroissance monotone (désignée par « M - »).

Les GSs sont définies qualitativement par l'ensemble de fonction de la tangente FT et sont classées en huit catégories à base des caractéristiques de la monotonie de l'ensemble de fonction de la tangente FT (Figure II-7 et Tableau II-1).

Une courbe en plan (X, Y) a la forme générale :

$$Y=f(x)$$

Sa fonction inverse peut s'exprimer par :

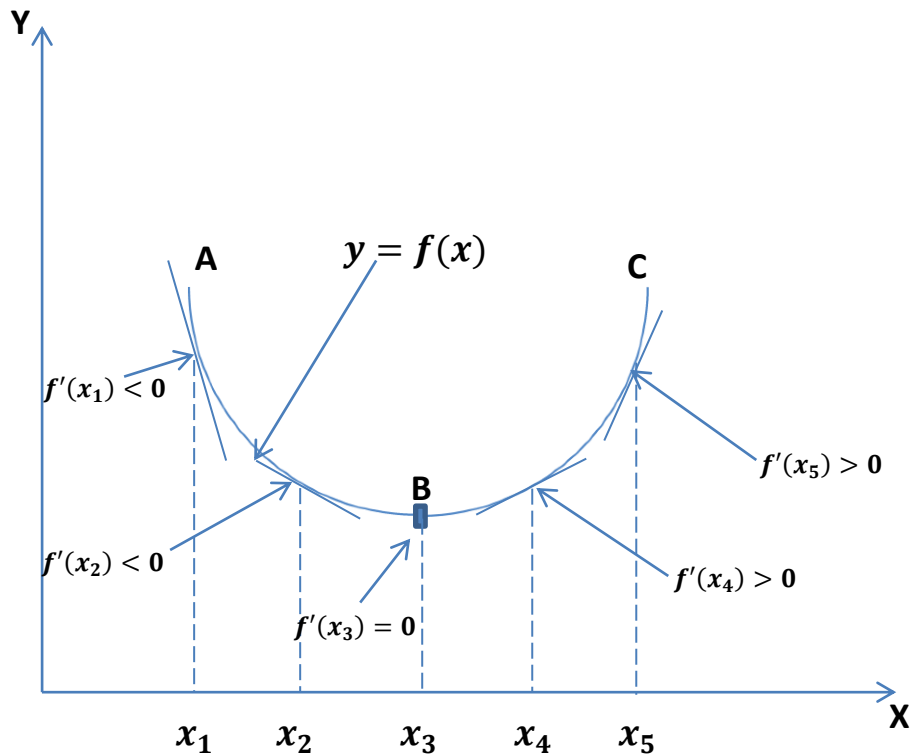
$$X=\phi(y)$$

Leurs dérivées premières sont représentées par $f'(x)$, $\phi'(y)$ respectivement, donc on a un ensemble de fonctions qui caractérise le tangent au point (x, y) dans la courbe, on l'appelle l'ensemble de fonction de la tangente FT.

$$FT = \{ f(x), \phi(y), f'(x), \phi'(y) \}.$$

<i>GS</i>	<i>f(x)</i>	$\Phi(y)$	<i>f'(x)</i>	$\Phi'(y)$
<i>CS1</i>	<i>M+</i>	<i>M+</i>	<i>M+</i>	<i>M-</i>
<i>CS2</i>	<i>M-</i>	<i>M+</i>	<i>M+</i>	<i>M-</i>
<i>CS3</i>	<i>M+</i>	<i>M+</i>	<i>M-</i>	<i>M+</i>
<i>CS4</i>	<i>M-</i>	<i>M-</i>	<i>M-</i>	<i>M+</i>
<i>LS1</i>	<i>M-</i>	<i>M+</i>	<i>C</i>	<i>C</i>
<i>LS2</i>	<i>M+</i>	<i>M+</i>	<i>C</i>	<i>C</i>
<i>LS3</i>	<i>C</i>	n/a	<i>O</i>	∞
<i>LS4</i>	n/a	<i>C</i>	∞	<i>O</i>

Tableau II-1 : la définition de GSs, M+ et M- signifie les propriétés de la croissance monotone et décroissance monotone respectivement, C représente une valeur constante et n/a signifie pas une fonction. [32][31]



$$\widehat{AB}: x_1 < x_2 < x_3$$

$$f(x_1) > f(x_2) > f(x_3)$$

$$f'(x_1) < f'(x_2) < f'(x_3)$$

$$\widehat{BC}: x_3 < x_4 < x_5$$

$$f(x_3) < f(x_4) < f(x_5)$$

$$f'(x_3) < f'(x_4) < f'(x_5)$$

B est le point CPP de jonction

Figure II-8 : AC est caractérisé par deux groupes de fonctions qui partitionnent la courbe en deux segments des arcs, AB qui est le CS2 et BC qui est le CS1 (Fig. II-7 (b)).

Dans notre classification de deuxième niveau les deux fonctions $f(x)$ et $f'(x)$ du Tableau II-1 sont suffisantes et discriminantes pour la distinction utilisant un système d'inférence : on peut attribuer chaque primitif à sa propre classe (classification niveau 2).

Un CPP est un point de jonction où deux groupes adjacents de GSs sont classés en une structure qui correspond à une perception stable. Comme le montre la Figure II-7 (c), les CPP sont aussi classés en huit types généraux. Leurs définitions sont données dans le Tableau II-2

3. 2. Classification perceptuelle

Comme **Lowe** [33] a déclaré : "Il y a une tendance pour les courbes à être complétées de manière à former des zones fermées". Pour former un objet en 2D, la classification perceptuelle se déroule après que les GETs et les CPP soient détectés dans une image.

Le processus de classification réorganise les données fournies par l'élimination des données non pertinentes et tri du reste en groupes, chacun correspondant à un objet particulier [34]. La classification implique la dérivation des structures à partir des images.

4. La reconnaissance

Les GETs sont les collections de classes de caractéristiques générales GSs et CPPs. La Figure II-9 est la structure de la hiérarchie des jetons de contour générique. Ici, TCPP signifie le réel CPP et VCPP signifie le CPP virtuel. Le TCPP est un CPP à vue invariante. Le VCPP est sensible aux changements de rotation, mais il est utile pour la classification des CSs en sections coniques. Une description générale d'un segment est développée sur le modèle perceptuel de: $GET [p_b, p_m, p_e]$, où p_b représente le point de début pour ce segment, p_m pour le point du milieu et p_e pour le point finale, c est l'indicateur qui montre à quelle catégorie appartient ce segment. Dans cette description (en Figure II-10), les positions du point de début p_b et du point finale p_e de ce segment sont échangeables.

Pour le Segment de Courbe (CS) (les arcs), l'information de la courbure de cette courbe est caractérisée par son point du milieu p_m . Alors que pour les segments (CS) de ligne droite, trois points p_b , p_e et p_m sont alignés dans la même ligne droite, donc la ligne peut être déterminée seulement par ses points de début p_b et le point finale p_e , le point milieu est redondant.

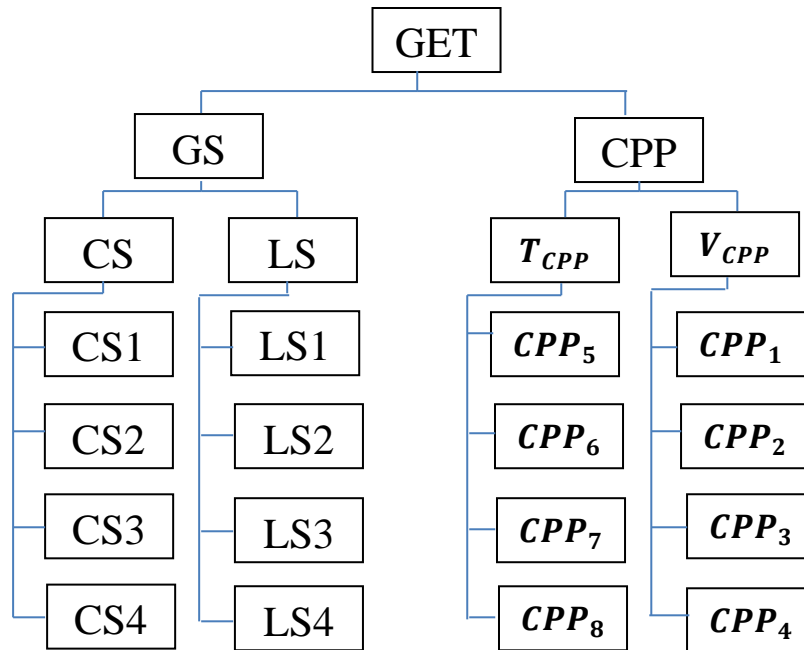


Figure II-9 : le concept hiérarchie de GET.

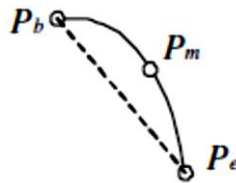


Figure II-10 : un GS

une valeur nulle est donc attribué au point du milieu p_m dans les segments de ligne droite. Les descriptions du segment de courbe et du segment de ligne droite sont respectivement :

$$CSc[p_b, p_m, p_e]$$

$$LSc[p_b, nulle, p_e]$$

La description générale d'un CPP est représentée comme suit:

$$CPPc[P, GS_i, GS_j], \quad i \neq j$$

Où P désigne le point de jonction, GS_i et GS_j représentent différents segments génériques, et c désigne le type de ce CPP. Dans cette description (Figure II-11), GS_i et GS_j partagent un même point finale P, i.e., $GS_i [p, p_{mi}, p_{ei}]$, $GS_j [p, p_{mj}, p_{ej}]$.

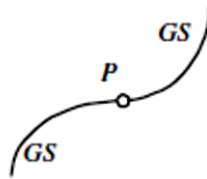


Figure II-11 : un CPP.

<i>Règle</i>	<i>Définitions</i>
<i>G1</i>	$(cpp_1, CS1, CS2)$
<i>G2</i>	$(cpp_2, CS3, CS4)$
<i>G3</i>	$(cpp_3, CS3, CS2)$
<i>G4</i>	$(cpp_4, CS4, CS1)$
<i>G5</i>	$(cpp_5, CS1, CS3)$
<i>G6</i>	$(cpp_6, CS2, CS4)$
<i>G7</i>	(cpp_7, CSi, LSj)
<i>G8</i>	(cpp_8, LSi, LSj)

Tableau II-2: la définition de CPPs

Les règles de segmentation et les règles de classification de la courbe générale sont basées sur ces jetons perceptuels génériques. Tous ces jetons sont perceptuellement distinguables.

Pour décrire un tracé de contour détecté qui sera utilisé dans la reconnaissance des formes, on utilise une chaîne de GETs et de CPPs. La représentation syntaxique d'un tracé est définie comme suit:

$$T\{GS_{i_1}, cpp_{j_1}, GS_{i_2}, \dots, cpp_{j_n}, \dots\}$$

Les structures composées en 2D de (a), (b) et (c) dans la Figure II-12 sont décrites par leurs représentations syntaxiques respectivement :

$$T_a\{LS3, cpp_7, CS2, cpp_5, CS3, cpp_2, CS4, cpp_4, CS1, cpp_7, LS3\}$$

$$T_b\{LS2, cpp_8, LS3, cpp_8, LS1, cpp_8, LS2, cpp_8\}$$

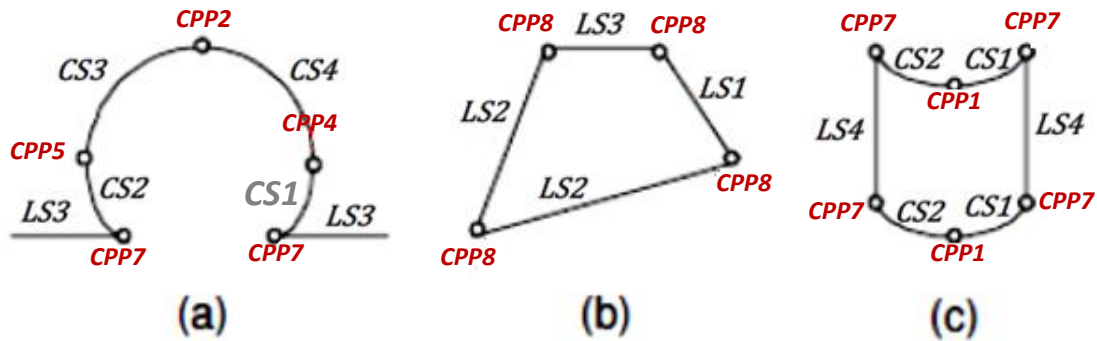
$$T_c\{LS4, cpp_7, CS2, cpp_1, CS1, cpp_7, LS4, cpp_7, CS1, cpp_1, CS2, cpp_7\}$$


Figure II-12 : exemple de traces.

4. 1. Classification des ellipses

Une ellipse est une courbe plane fermée qui peut être décrite qualitativement en utilisant les GETs. Une telle représentation de l'ellipse est donnée dans le Tableau II-3 qui correspond aux concepts illustrés dans la Figure II-13 et la Figure II-14. Dans la Figure II-14 une ellipse est représentée par un graphe dans lequel chaque sommet désigne un CPP et chaque arête (bord) représente un CS, comme il est illustré dans la Figure II-13 et la Figure II-14, pour regrouper les GET dans une ellipse intacte, quatre types de segments de courbe et quatre types de CPPs doivent être identifiés, pour un alignement incomplet, s'il existe au moins trois segments CSs ou trois CPPs dans une structure composée de GET.

$$\begin{aligned}
 & \textit{Ellipse} \{ \\
 & \quad \textit{CS1}[p_1, p_{m_1}, p_4] \\
 & \quad \textit{cpp}_1[p_1, \textit{CS1}, \textit{CS2}] \\
 & \quad \textit{CS2}[p_3, p_{m_2}, p_1] \\
 & \quad \textit{cpp}_3[p_3, \textit{CS2}, \textit{CS3}] \\
 & \quad \textit{CS3}[p_2, p_{m_3}, p_3] \\
 & \quad \textit{cpp}_2[p_2, \textit{CS3}, \textit{CS4}] \\
 & \quad \textit{CS4}[p_4, p_{m_4}, p_2] \\
 & \quad \textit{cpp}_4[p_4, \textit{CS4}, \textit{CS1}] \\
 & \quad \}
 \end{aligned}$$

Tableau II-3 : la définition d'ellipse.

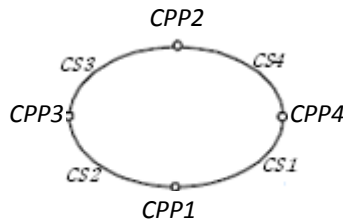


Figure II-13 : un modèle d'ellipse.

Donc, pour une forme détectée, nous mesurons sa confiance comme suit:

$$\text{Confiance} = P_{\text{forme}} \left(\frac{\sum \textit{GET}_{\text{image}}}{\sum \textit{GET}_{\text{modèle}}} \right)$$

Où P_{forme} signifie la certitude que la forme regroupée pourrait être, $\sum \textit{GET}_{\text{image}}$ est la somme des GETs satisfaisant la forme de la structure à partir de l'image, $\sum \textit{GET}_{\text{modèle}}$ est le total des GETs dans le modèle de classification. Les conditions de certaines concordances possibles sont illustrées dans la Figure II-15, par exemple, un contour qui a au moins trois CPPs structurellement cohérentes ou a au moins trois CSs structurellement cohérentes est une forte indication que c'est une ellipse. C'est pourquoi on peut déduire que nous trouvons une ellipse avec la mesure de confiance supérieure à 75%.

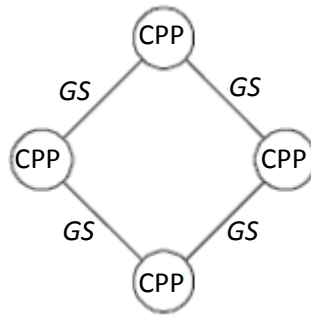


Figure II-14 : la représentation graphique.

Cette tolérance est également utile pour la reconnaissance d'objets partiellement occlus.

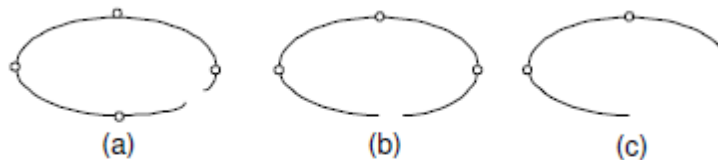


Figure II-15 : quelques exemples possibles de modèles d'ellipses incomplètes.

La procédure de classification d'ellipses est donnée dans l'algorithme suivant.

Algorithme

1. classifier tous les CPPs en huit catégories, et choisir les catégories CPP1, CPP2, CPP3 et CPP4 comme étant les groupes candidats.

2. démarrer à partir de n'importe quel CPP CPPb1 [P_{b_1}, CS_i, CS_j] donné dans les groupes candidats, selon les règles de classification, trouver le candidat possible CPPs dans les autres groupes de candidats :

(a) le long de la direction du sens des aiguilles d'une montre, trouver la séquence CPPs adjacente possible : CPPb2 [P_{b_2}, CS_j, CS_k], CPPb3 [P_{b_3}, CS_k, CS_l] et CPPb4 [P_{b_4}, CS_k, CS_i];

(b) le long de la direction du sens inverse des aiguilles d'une montre, trouver la séquence CPPs adjacente possible : CPPb4 [P_{b_4}, CS_k, CS_i], CPPb3 [P_{b_3}, CS_k, CS_l] et CPPb2 [P_{b_2}, CS_j, CS_k].

Ce traitement sera arrêté si n'importe quel CPP était déjà détecté dans la recherche dans le sens des aiguilles d'une montre.

(c) Selon la mesure de confiance décider s'il faut l'enregistrer comme une ellipse.

Les CPPs fournissent des preuves solides pour classer des objets significatifs.

En utilisant ce modèle générique, les algorithmes basés sur la reconnaissance à partir des CPPs sont aussi faisables et la tâche est de trouver les contours fermés avec quatre types de CPPs au lieu de trouver quatre types de CSs. Cette flexibilité est très utile pour la reconnaissance des contours occlus.

Conclusion

Les GETs sont des caractéristiques de contours perceptuellement significatifs qui sont descriptifs dans la représentation et peuvent donc être manipulés qualitativement dans le processus de la classification perceptuelle. Les GETs pourraient aussi être un ensemble de vocabulaires de base de la langue d'organisation perceptuelle pour la classification et la représentation des formes en 2D. Les classifications de bas niveau des GETs sont purement axées sur les données, mais ne dépendent pas des données de contour parfait, afin qu'ils soient très robustes. La classification de formes de haut niveau d'ellipse est de nature descriptive.

Les calculs principaux pour les deux niveaux de processus sont qualitatifs. La mise en œuvre de la classification de formes présentée dans ce chapitre est basée sur les données des GETs extraites à partir d'un tracé de contour. Des extensions peuvent être prévues pour inclure le modèle en 3D.

CHAPITRE III

**Résultats expérimentaux et
interprétations**

Dans notre étude nous avons utilisé le logiciel matlab version 2012a avec un PC doté d'un microprocesseur i7 troisième génération de fréquence 2.9 GHZ, et d'une RAM de 8 G byte. Le filtre de canny est utilisé pour l'extraction du contour de la forme.

1. Forme d'un cercle :

On prend par exemple une forme d'un cercle, on introduit la forme dans l'environnement matlab comme il est illustré dans la figure ci-dessous.

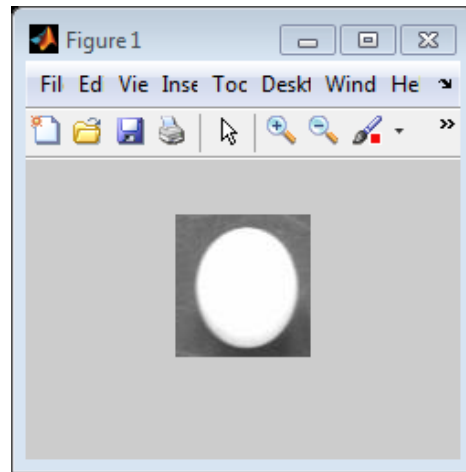


Figure III-1 : la forme originale.

La figure ci-dessous présente l'étape de binarisation pour la forme de l'image ci-dessus avec un seuil de 0.5 pour le noir et de même pour le blanc.

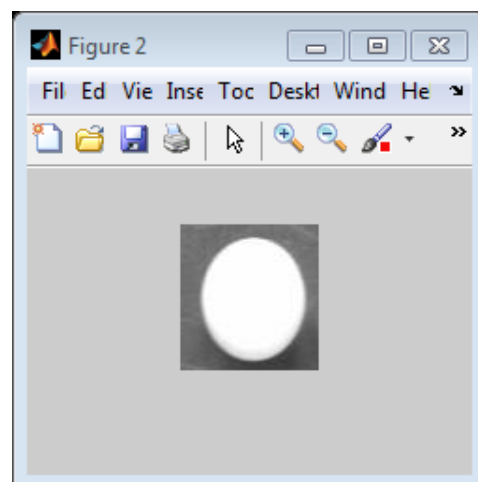


Figure III-2 : l'étape de binarisation.

La figure ci-dessous présente l'histogramme de la forme.

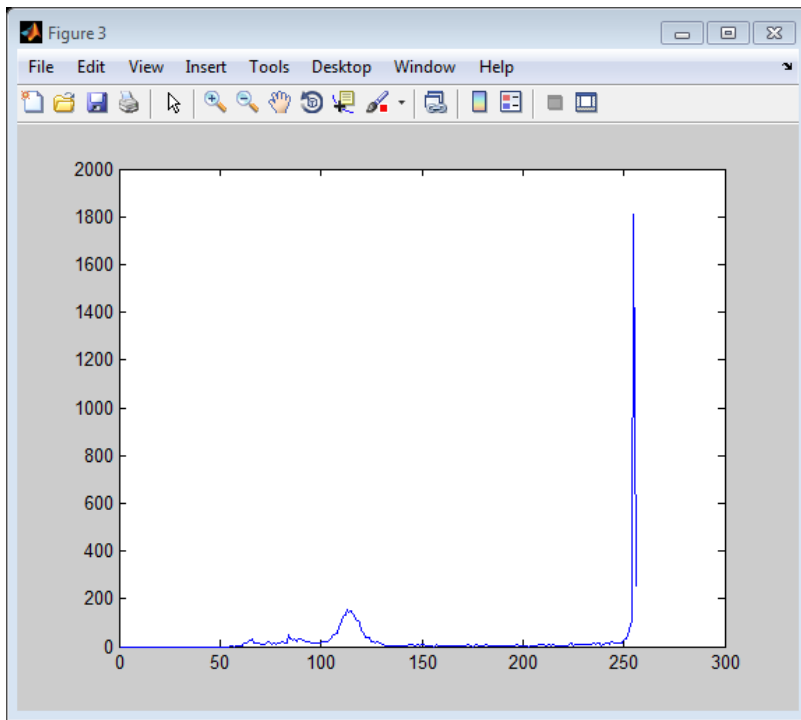


Figure III-3 : l’histogramme de l’image.

L’étape suivante consiste à extraire le contour de la forme avec le filtre de canny, et de placer les coordonnées des pixels de contour dans un tableau de tracé de pixels (ligne 3 de l’algorithme de la figure I-9 chapitre I).

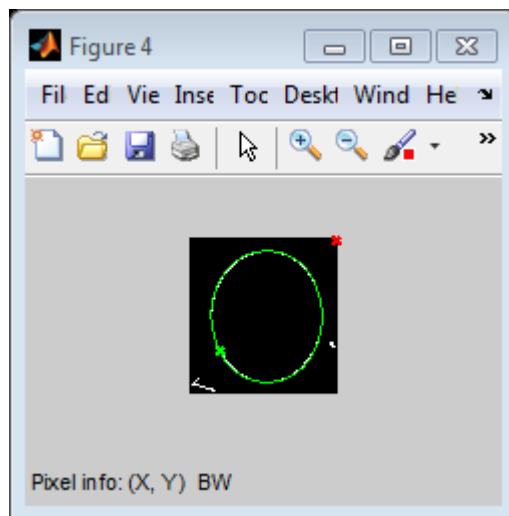


Figure III-4 : le contour de la forme.

X= [16 16 17 17 18 18 19 20 21 22 23 24
 25 26 27 28 29 30 31 32 33 34 35 36

37	38	39	40	41	42	43	44	45	46	47	48	
49	50	51	52	53	54	55	56	57	58	58	59	
60	61	61	62	62	63	63	64	64	65	65	65	
65	66	66	66	66	66	67	67	67	67	67	67	
67	67	67	66	66	66	66	66	65	65	65	64	
64	64	63	63	63	62	61	61	60	59	59	58	
57	56	55	54	53	52	51	50	49	48	47	46	
45	44	43	42	41	40	39	38	37	36	35	34	
33	32	31	30	29	28	27	26	25	24	23	22	
21	20	19	18	18	17	16	16	15	15	14	14	
14	13	13	13	12	12	12	12	12	12	11	11	
11	11	11	12	12	12	12	12	12	13	13	13	
13	14	14	14	15	15	16],						
Y=[57	58	59	60	61	62	63	64	65	66	67	68
	68	69	70	70	71	71	71	72	72	72	72	73
	73	73	73	73	73	73	72	72	72	72	72	71
	71	70	70	69	69	68	67	66	66	65	64	63
	62	61	60	59	58	57	56	55	54	53	52	51
	50	49	48	47	46	45	44	43	42	41	40	39
	38	37	36	35	34	33	32	31	30	29	28	27
	26	25	24	23	22	21	20	19	18	17	16	15
	14	13	13	12	11	11	10	10	9	9	8	8
	8	8	8	7	7	7	7	7	7	8	8	8
	8	8	9	9	9	10	10	11	12	12	13	14
	15	16	17	18	19	20	21	22	23	24	25	26
	27	28	29	30	31	32	33	34	35	36	37	38
	39	40	41	42	43	44	45	46	47	48	49	50
	51	52	53	54	55	56	57],					

Les vecteurs sont tous des vecteurs lignes, le contour est parcouru dans le sens antihoraire.

Les deux vecteurs X et Y représentent les coordonnées cartésiennes de tous les pixels de contours candidats (ligne 3, figure I-9, chapitre I).

1.1 Segmentation

L'étape suivante consiste à mettre en place les signes initiaux de dx et dy (ligne 5 de la figure I-9, chapitre I).

Dx=[

2	2	2	2	2	3	3	4	4	4	4	4	4
4	4	4	4	4	4	4	4	4	4	4	4	4
4	4	4	4	4	4	4	4	4	4	4	4	4
4	4	4	4	4	4	4	4	3	3	3	3	3
3	2	2	2	2	2	2	2	2	1	1	1	1
1	1	0	1	1	1	1	1	0	0	0	0	0
-1	-1	-1	-1	0	-1	-1	-1	-1	-2	-1	-1	-2
-1	-1	-2	-2	-2	-3	-3	-2	-3	-3	-3	-3	-4
-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4
-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4
-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4
-4	-3	-3	-3	-2	-3	-2	-2	-2	-2	-1	-2	-1
-1	-2	-1	-1	-1	0	0	-1	-1	-1	-1	-1	0
1	1	1	1	0	0	1	1	1	1	1	1	1
1	2	1	2	2	2],							

Dy=[

4	4	4	4	4	4	4	4	4	4	4	3	3
3	2	3	2	1	2	1	1	1	1	1	1	1
1	0	0	0	-1	-1	-1	-1	0	-1	-1	-1	-2
-2	-2	-2	-2	-3	-3	-3	-3	-3	-3	-3	-4	-4
-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4
-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4
-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4
-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4
-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4
-3	-3	-3	-2	-3	-2	-2	-2	-2	-2	-2	-1	-1
0	-1	-1	-1	-1	0	0	1	1	1	1	1	0

1	1	1	2	1	2	3	2	3	3	3	4
4	4	4	4	4	4	4	4	4	4	4	4
4	4	4	4	4	4	4	4	4	4	4	4
4	4	4	4	4	4	4	4	4	4	4	4
4	4	4	4	4	4	4	4	4	4	4	4

Le changement de signe au niveau de dx et dy, les éléments en rouges dans les vecteurs dx et dy indiquent le changement de signe qui signifie la présence d'un point partageant la courbe CPP fort (ligne 6 de l'algorithme de la figure I-9, chapitre I).

D'après l'approche présentée dans le chapitre I (section 4.2) quatre points partageant la courbes sont repérés (ligne 8 de l'algorithme de la figure I-9 chapitre I), leurs coordonnées respectivement sont :

$$X = [41 \quad 67 \quad 38 \quad 11],$$

$$Y = [73 \quad 37 \quad 7 \quad 40],$$

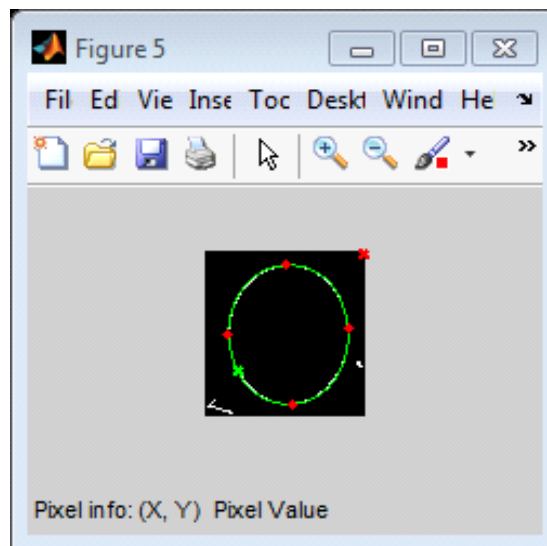


Figure III-5 : les quatre points CPP forts en rouge.

Après avoir calculé les coordonnées des points partageants la courbe, nous construisons des nouveaux vecteurs de coordonnées de pixels de contour de façon que le premier élément dans le vecteur est le premier point de contour qui suit le point partageant la courbe, le but

est de mettre en ordre les pixels de contour entre deux points CPP forts (ligne 9 de l'algorithme de la figure I-9, chapitre I) comme suit :

```
X22=[ 42  43  44  45  46  47  48  49  50  51  52  53
      54  55  56  57  58  58  59  60  61  61  62  62
      63  63  64  64  65  65  65  65  66  66  66  66
      66  67  67  67  67  67  67  67  67  67  66  66
      66  66  66  65  65  65  64  64  64  63  63  63
      62  61  61  60  59  59  58  57  56  55  54  53
      52  51  50  49  48  47  46  45  44  43  42  41
      40  39  38  37  36  35  34  33  32  31  30  29
      28  27  26  25  24  23  22  21  20  19  18  18
      17  16  16  15  15  14  14  14  13  13  13  12
      12  12  12  12  12  11  11  11  11  11  12  12
      12  12  12  12  13  13  13  13  14  14  14  15
      15  16  16  16  17  17  18  18  19  20  21  22
      23  24  25  26  27  28  29  30  31  32  33  34
      35  36  37  38  39  40  41],
```

```
Y22=[ 73  72  72  72  72  72  71  71  70  70  69  69
      68  67  66  66  65  64  63  62  61  60  59  58
      57  56  55  54  53  52  51  50  49  48  47  46
      45  44  43  42  41  40  39  38  37  36  35  34
      33  32  31  30  29  28  27  26  25  24  23  22
      21  20  19  18  17  16  15  14  13  13  12  11
      11  10  10  9  9  8  8  8  8  8  7  7
      7  7  7  7  8  8  8  8  8  9  9  9
      10  10  11  12  12  13  14  15  16  17  18  19
      20  21  22  23  24  25  26  27  28  29  30  31
      32  33  34  35  36  37  38  39  40  41  42  43
      44  45  46  47  48  49  50  51  52  53  54  55
```

56	57	57	58	59	60	61	62	63	64	65	66
67	68	68	69	70	70	71	71	71	72	72	72
72	73	73	73	73	73	73	73],				

Les éléments marqués en rouge sont les coordonnées des points CPP forts.

Dans la Figure III-5, le contour de la forme est segmenté en quatre segments, entre chaque deux points rouges on a un segment (arc) significatif ressemblant au type de GET illustrés dans la figure I-1 (a) chapitre I, donc on n'a pas besoin de chercher les points CPP faibles (section 4.3, chapitre I).

1.2 Classification

Pour attribuer les segments issus du processus de segmentation chacun à sa propre classe, on doit établir le schéma des OPABS (section 4.3 chapitre I).

En appliquant la formule (Eq. 3) de chapitre I, Les valeurs de degré d'Arc-tangente sont uniformément classifiées dans huit catégories (figure I-9, chapitre I), comme il est illustré dans le vecteur suivant :

Cat=[6	6	6	6	6	6	6	5	5	5	5	5
	5	5	5	5	4	5	4	4	4	4	4	4
	4	3	3	3	3	3	3	3	3	3	3	2
	2	2	2	2	2	2	2	2	1	1	1	1
	1	1	1	1	1	1	1	1	1	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0
	0	7	7	7	7	7	7	7	7	6	7	7
	6	7	7	6	6	6	6	6	6	6	6	6
	5	5	5	5	5	5	5	5	5	5	4	4
	3	4	4	4	4	3	3	3	3	3	3	3
	3	3	3	2	3	2	2	2	2	2	2	1
	1	1	1	1	1	1	1	1	1	1	0	1
	0	0	1	0	0	0	0	7	0	0	0	0

0 7 7 7 7 7 7 7 7 7 7 7
 7 7 6 7 6 6 6 4],

Le vecteur des catégories compatible avec X22 et Y22 est :

Cat22=[3 3 3 3 3 3 2 2 2 2 2 2
 2 2 2 1 1 1 1 1 1 1 1 1
 1 1 1 1 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 7 7 7 7
 7 7 7 7 6 7 7 6 7 7 6 6
 6 6 6 6 6 6 6 5 5 5 5 5
 5 5 5 5 5 4 4 3 4 4 4 4
 3 3 3 3 3 3 3 3 3 3 2 3
 2 2 2 2 2 2 1 1 1 1 1 1
 1 1 1 1 1 0 1 0 0 1 0 0
 0 0 7 0 0 0 0 0 7 7 7 7
 7 7 7 7 7 7 7 7 7 7 7 6
 6 6 6 6 6 6 6 6 6 5 5 5
 5 5 5 5 5 5 5 5 4 4 4 4
 4 4 4 3 3 3 3].

Les éléments de vecteur cat22 indiquent les catégories arc tangente, par exemple 0 indique catégorie 0 et 1 indique catégorie 1 ... etc.

On prend un sous vecteur entre le premier élément de vecteur cat22 et le premier élément en rouge de ce dernier, le sous vecteur représente l'ensemble des catégories d'arc tangent des pixels de contour entre deux points CPP forts.

Cat22_1=[3 3 3 3 3 3 3 2 2 2 2 2
 2 2 2 2 1 1 1 1 1 1 1 1
 1 1 1 1 1 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0],

Le sous vecteur cat22_1 contient 6 éléments de catégorie 3, 9 éléments de catégorie 2, 13 éléments de catégorie 1 et 16 éléments de catégorie 0.

Il est à noter que les boîtes ont des tailles quasiment similaires (presque la même taille), le changement de taille s'effectue doucement dans le sens croissant, il n'y a pas de boîte de sommet, et on ne peut pas appliquer le sommet-vallée-ratio, (section 2.2, chapitre II).

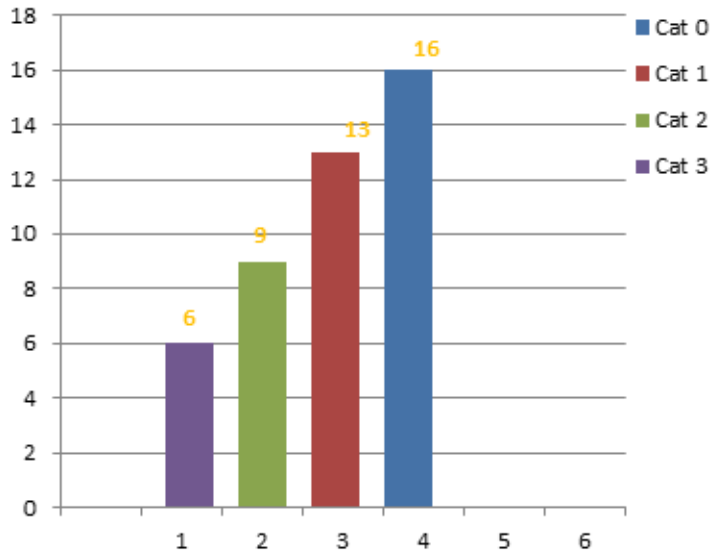


Figure III-6: séquence de boîtes correspond au segment entre les deux premiers points CPP forts.

Dans la Figure III-6 le changement de taille entre les catégories est léger et on n'a pas de boîte de sommet, donc le segment est un arc, (classification de premier niveau, section 2 chapitre II),

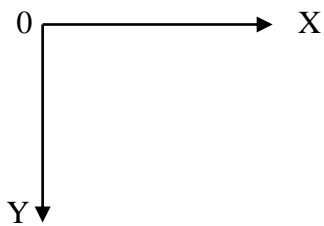
Reste à déterminer la classe des arcs à laquelle appartient le segment, entre les quatre classes CS1, CS2, CS3, CS4.

Après la classification du premier niveau, (deux classes perceptuelles: GET en arc et GET en segment de droite, section 2 chapitre II), Les deux fonctions f et f' sont suffisantes et discriminantes pour la classification de deuxième niveau (Tableau II-1 chapitre II).

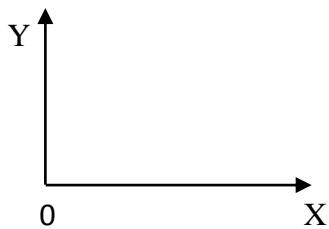
```
X22_1=[ 42  43  44  45  46  47  48  49  50  51  52  53
        54  55  56  57  58  58  59  60  61  61  62  62
```

	63	63	64	64	65	65	65	65	66	66	66	66
	66	66	66	66	67	67	67	67],				
Y22_1=[73	72	72	72	72	72	71	71	70	70	69	69
	68	67	66	66	65	64	63	62	61	60	59	58
	57	56	55	54	53	52	51	50	49	48	47	46
	45	44	43	42	41	40	39	38],				

Nous devons faire un redressement de vecteur Y22_1, car l'image est traitée dans le logiciel matlab d'une façon que l'axe des ordonnées soit orienté vers le bas, comme suit :



Il faut établir un nouveau plan de façon à ce que l'axe des ordonnées soit orienté vers le haut comme il est illustré dans le schéma suivant:



Après le changement de plan, le vecteur Y22_1 sera comme suit :

Y22_11=	[5	6	6	6	6	6	7	7	8	8	9	9
	10	11	12	12	13	14	15	16	17	18	19	20
	21	22	23	24	25	26	27	28	29	30	31	32
	33	34	35	36	37	38	39	40],				

Donc on aura les deux vecteurs suivants :

Le vecteur des abscisses X22_1, et celui des ordonnées Y22_11,

Pour le vecteur des abscisses X22_1,

X22_1=[42 43 44 45 46 47 48 49 50 51 52 53
 54 55 56 57 58 58 59 60 61 61 62 62
 63 63 64 64 65 65 65 65 66 66 66 66
 66 66 66 66 67 67 67 67],

Les éléments de vecteur X22_1 évoluent dans le sens croissant :

$X=\Phi(y) : x_1 \leq x_2 \leq x_3 \leq x_4 \leq x_5 \leq x_6 \leq \dots \leq x_{44}$, donc la fonction $X=\Phi(y)$ est monotone croissante $M+$ (1)

- Pour le vecteur des ordonnées Y22_11,

Y22_11= [5 6 6 6 6 6 7 7 8 8 9 9
 10 11 12 12 13 14 15 16 17 18 19 20
 21 22 23 24 25 26 27 28 29 30 31 32
 33 34 35 36 37 38 39 40],

On constate que les éléments de vecteur Y22_11 évoluent dans un sens croissant :

$F(x)=y : y_1 \leq y_2 \leq y_3 \leq y_4 \leq y_5 \leq y_6 \leq \dots \leq y_{44}$, donc la fonction $Y=f(x)$ est monotone croissante $M+$ (2)

- Pour la fonction $f'(x)$ la dérivée de la fonction $f(x)$,

$$F'(x) = \frac{\Delta y}{\Delta x}$$

On prend par exemple un pas de dérivée égale à 3,

$$\frac{\Delta Y_1}{\Delta X_1} = \frac{Y_4 - Y_1}{X_4 - X_1} = \frac{6 - 5}{45 - 42} = 0.33$$

$$\frac{\Delta Y_2}{\Delta X_2} = \frac{Y_7 - Y_4}{X_7 - X_4} = \frac{7 - 6}{48 - 45} = 0.33$$

$$\frac{\Delta Y_3}{\Delta X_3} = \frac{Y_{10} - Y_7}{X_{10} - X_7} = \frac{8 - 7}{51 - 48} = 0.33$$

$$\frac{\Delta Y_4}{\Delta X_4} = \frac{Y_{13} - Y_{10}}{X_{13} - X_{10}} = \frac{10 - 8}{54 - 51} = 0.66$$

$$\frac{\Delta Y_5}{\Delta X_5} = \frac{Y_{16} - Y_{13}}{X_{16} - X_{13}} = \frac{12 - 10}{57 - 54} = 0.66$$

$$\frac{\Delta Y_6}{\Delta X_6} = \frac{Y_{19} - Y_{16}}{X_{19} - X_{16}} = \frac{15 - 12}{59 - 57} = 1$$

$$\frac{\Delta Y_7}{\Delta X_7} = \frac{Y_{22} - Y_{19}}{X_{22} - X_{19}} = \frac{18 - 15}{61 - 59} = 1.5$$

.

.

$$\frac{\Delta Y_{14}}{\Delta X_{14}} = \frac{Y_{43} - Y_{40}}{X_{43} - X_{40}} = \frac{39 - 36}{67 - 66} = 3$$

Nous constatons que :

$$\frac{\Delta Y_1}{\Delta X_1} \leq \frac{\Delta Y_2}{\Delta X_2} \leq \frac{\Delta Y_3}{\Delta X_3} \leq \frac{\Delta Y_4}{\Delta X_4} \leq \frac{\Delta Y_5}{\Delta X_5} \leq \frac{\Delta Y_6}{\Delta X_6} \leq \frac{\Delta Y_7}{\Delta X_7} \leq \dots \leq \frac{\Delta Y_n}{\Delta X_n},$$

la fonction des dérivées est monotone croissante $M+$ (3).

On utilise un système d'inférence pour comparer les données du Tableau II-1 de chapitre II avec (1), (2) et (3) afin de conclure que le segment est un arc CS1 (Figure II-7 -b chapitre II).

On prend un sous vecteur entre le dernier élément en rouge de vecteur cat22 et son prédécesseur directe en rouge qui représente l'ensemble des catégories d'arc tangent des pixels de contour entre deux points CPP forts.

```
Cat22_2= [ 7 7 7 7 7 7 7 7 7 7 7 7
           7 7 7 6 6 6 6 6 6 6 6 6
           6 5 5 5 5 5 5 5 5 5 5 5
           4 4 4 4 4 4 4 3 3 3],
```

Le vecteur cat22_2 (figure II-6 chapitre II) contient 14 éléments de catégorie 7, 10 éléments de catégorie 6, 11 éléments de catégorie 5, 7 éléments de catégorie 4 et 3 éléments de catégorie 3, on constate que les boîtes ont des tailles presque similaires et le changement de taille s'effectue doucement dans le sens descendant, on n'a pas de boîte de sommet et on ne peut pas appliquer le sommet-vallée-ratio, (section 2.2, chapitre II).

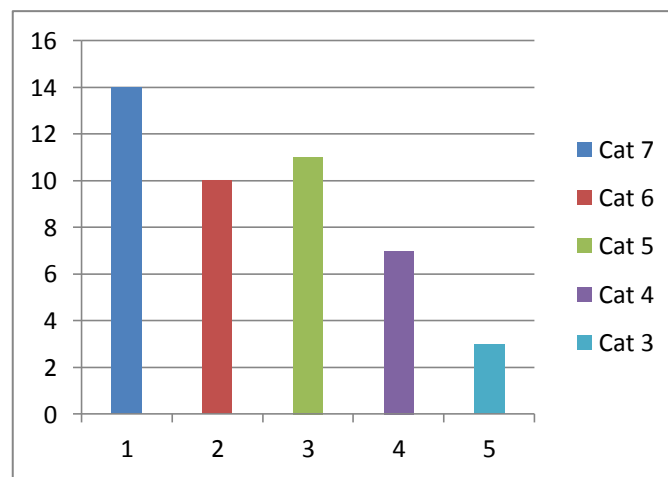


Figure III-7 : séquence de boîte correspond au segment entre les deux derniers points CPP forts.

Dans la Figure III-7 le passage entre les catégories se fait légèrement, et on n'a pas une boîte de sommet, donc le segment est un arc.

Pour la classification du deuxième niveau on a :

X22_2=[11 12 12 12 12 12 12 13 13 13 13 14
 14 14 15 15 16 16 16 17 17 18 18 19
 20 21 22 23 24 25 26 27 28 29 30 31
 32 33 34 35 36 37 38 39 40],

Y22_2=[41 42 43 44 45 46 47 48 49 50 51 52
 53 54 55 56 57 57 58 59 60 61 62 63
 64 65 66 67 68 68 69 70 70 71 71 71
 72 72 72 72 73 73 73 73 73],

Après le redressement, le vecteur Y22_2 sera comme suit :

Y22_21=[37 36 35 34 33 32 31 30 29 28 27
 26 25 24 23 22 21 21 20 19 18 17 16
 15 14 13 12 11 10 10 9 8 8 7 7
 7 6 6 6 6 5 5 5 5 5],

Dans le vecteur des abscisses X22_2 ces éléments évoluent dans le sens croissant : $x_1 \leq x_2 \leq x_3 \leq x_4 \leq x_5 \leq x_6 \leq \dots \leq x_{45}$, donc la fonction $X=\Phi(y)$ est monotone croissante **M+**..... (4)

Dans le vecteur des ordonnées Y22_21 ces éléments évoluent dans le sens décroissant : $F(x)=y : y_1 \geq y_2 \geq y_3 \geq y_4 \geq y_5 \geq y_6 \geq \dots \geq y_{45}$, donc la fonction $Y=f(x)$ est monotone décroissante **M-**..... (5)

Pour la fonction $f'(x)$ la dérivée de la fonction $f(x)$,

$$F'(x) = \frac{\Delta y}{\Delta x}$$

On prend par exemple un pas de dérivé 5,

$$\frac{\Delta Y_1}{\Delta X_1} = \frac{Y_6 - Y_1}{X_6 - X_1} = \frac{32 - 37}{12 - 11} = -5$$

$$\frac{\Delta Y_2}{\Delta X_2} = \frac{Y_{11} - Y_6}{X_{11} - X_6} = \frac{27 - 32}{13 - 12} = -5$$

$$\frac{\Delta Y_3}{\Delta X_3} = \frac{Y_{16} - Y_{11}}{X_{16} - X_{11}} = \frac{22 - 27}{15 - 13} = -2.5$$

$$\frac{\Delta Y_4}{\Delta X_4} = \frac{Y_{21} - Y_{16}}{X_{21} - X_{16}} = \frac{18 - 22}{17 - 15} = -2$$

$$\frac{\Delta Y_5}{\Delta X_5} = \frac{Y_{26} - Y_{21}}{X_{26} - X_{21}} = \frac{13 - 18}{21 - 17} = -1.25$$

$$\frac{\Delta Y_6}{\Delta X_6} = \frac{Y_{31} - Y_{26}}{X_{31} - X_{26}} = \frac{09 - 13}{26 - 21} = -0.8$$

$$\frac{\Delta Y_7}{\Delta X_7} = \frac{Y_{36} - Y_{31}}{X_{36} - X_{31}} = \frac{07 - 09}{31 - 26} = -0.4$$

.

.

.

On a: $\frac{\Delta Y_1}{\Delta X_1} \leq \frac{\Delta Y_2}{\Delta X_2} \leq \frac{\Delta Y_3}{\Delta X_3} \leq \frac{\Delta Y_4}{\Delta X_4} \leq \frac{\Delta Y_5}{\Delta X_5} \leq \frac{\Delta Y_6}{\Delta X_6} \leq \frac{\Delta Y_7}{\Delta X_7} \leq \dots \leq \frac{\Delta Y_n}{\Delta X_n}$, On constate que la fonction des dérivées est monotone croissante **M+**..... (6).

On utilise notre système d'inférence qui compare les données du Tableau II-1 du chapitre II avec (4), (5) et (6) pour conclure que le segment est un arc CS2 (figure II-7(b) chapitre II).

Car le contour est fermé et il est parcouru dans le sens antihoraire, le premier pixel de CS1 et le dernier pixel de CS2 sont adjacents, le système d'inférence déduit le point partageant la courbe CPP1 entre CS1 et CS2 sur la base du tableau 5 du chapitre II comme

il est illustré dans la Figure II-7(c). De la même façon on continue à classer les autres segments et les points partageant la courbe jusqu'à avoir construit la description structurale pour cette forme, comme suit :

$\{CS1 \ CPP4 \ CS4 \ CPP2 \ CS3 \ CPP3 \ CS2 \ CPP1\}$.

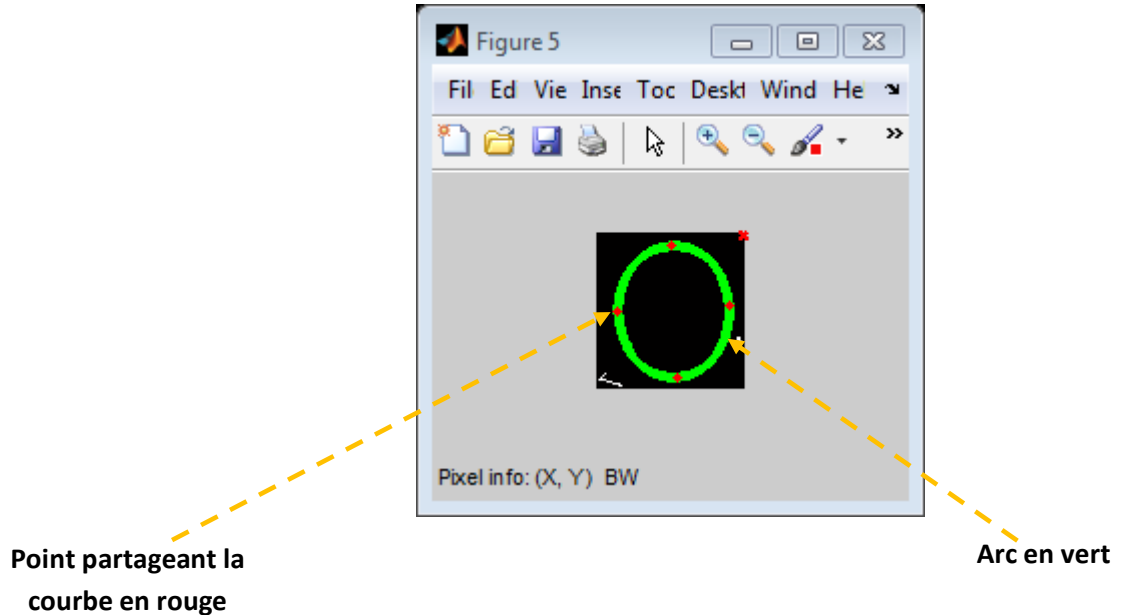


Figure III-8 : la forme de la figure III-1 segmentée en arcs.

2. Forme géométrique

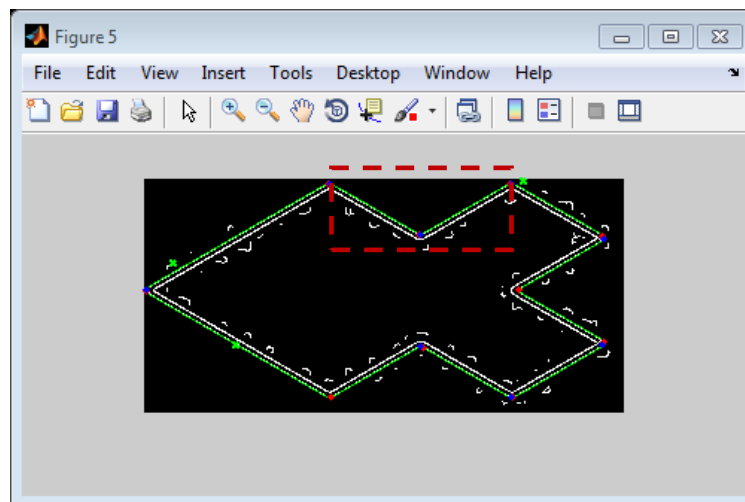


Figure III-9 : une forme géométrique, les points en rouge sont les CPP forts et les points en bleu sont les CPP faibles type sommet-vallée-sommet.

2.1 Segmentation

Pour la détection des points CPP forts Nous suivons les mêmes étapes illustrées dans la section 1.1 ci-dessus.

Notons que quelques points CPP faibles se confondent avec les points CPP forts ou dans leur voisinage, ce qui prouve que les deux méthodes vont dans le même sens, elles font l'objet de la détection des points partageants la courbe,

$X_{cpp} = [126 \ 188 \ 248 \ 309 \ 252 \ 308 \ 246 \ 185 \ 124 \ 2],$

$Y_{cpp} = [147 \ 114 \ 147 \ 110 \ 75 \ 39 \ 4 \ 38 \ 4 \ 76],$

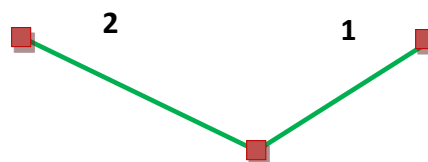
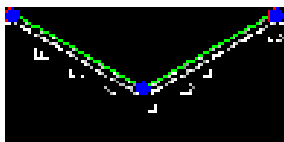
Les éléments des deux vecteurs X_{cpp} et Y_{cpp} sont les coordonnées cartésiennes des points CPP forts,

$X_{cpp_f} = [186 \ 247 \ 308 \ 309 \ 247 \ 186 \ 125 \ 2],$

$Y_{cpp_f} = [113 \ 147 \ 112 \ 41 \ 4 \ 38 \ 4 \ 75],$

Les éléments des deux vecteurs X_{cpp_f} et Y_{cpp_f} sont les coordonnées cartésiennes des points CPP faibles de type sommet-vallée-sommet,

La taille des deux vecteurs des coordonnées cartésiennes de cette forme géométrique est très grande, nous prenons le segment situé à l'intérieur du cadre en rouge de la Figure III-7 pour simplifier l'étude,



Les coordonnées cartésiennes de ce segment sont :

$X_{22} = [\text{246} \ 245 \ 244 \ 243 \ 242 \ 241 \ 240 \ 239 \ 238 \ 237 \ 236 \ 235$
 $234 \ 233 \ 232 \ 231 \ 230 \ 229 \ 228 \ 227 \ 226 \ 225 \ 224 \ 223$

222	221	220	219	218	217	216	215	214	213	212	211
210	209	208	207	206	205	204	203	202	201	200	199
198	197	196	195	194	193	192	191	190	189	188	187
186	185	184	183	182	181	180	179	178	177	176	175
174	173	172	171	170	169	168	167	166	165	164	163
162	161	160	159	158	157	156	155	154	153	152	151
150	149	148	147	146	145	144	143	142	141	140	139
138	137	136	135	134	133	132	131	130	129	128	127
126	125	124],									

Y22=[

4	4	5	5	6	6	7	8	8	9	9	10
10	11	12	12	13	13	14	15	15	16	16	17
17	18	19	19	20	20	21	21	22	23	23	24
24	25	25	26	27	27	28	28	29	30	30	31
31	32	32	33	34	34	35	35	36	36	37	38
38	38	37	36	36	35	35	34	34	33	32	32
31	31	30	30	29	28	28	27	27	26	26	25
24	24	23	23	22	22	21	20	20	19	19	18
17	17	16	16	15	15	14	13	13	12	12	11
11	10	9	9	8	8	7	7	6	6	5	5
4	4	4],									

Le vecteur des catégories correspondant est :

Cat22=[

3	3	2	2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2	2	2
3	5	5	5	5	5	5	5	5	5	5	5
5	5	5	5	5	5	5	5	5	5	5	5
5	5	5	5	5	5	5	5	5	5	5	5

5	5	5	5	5	5	5	5	5	5	5	5
5	5	5	5	5	5	5	5	5	5	5	5
4	4	3],								

Les éléments marqués en rouge sont les coordonnées des points CPP, le segment **1** situé entre les deux premiers éléments en rouge dans le vecteur des catégories et le reste sont les éléments de segment **2**,

Dans le vecteur cat22 on a : une boîte de **2** éléments de catégories 3 et une boîte de 58 éléments de catégorie 2 et une boîte d'un seul élément de catégorie 3, donc beaucoup de valeurs de degré Arc-tangente de pixels de contour appartiennent à la boîte de catégorie 2, ainsi un côté du sommet est formé, après on trouve une boîte de catégorie 3, donc une falaise est formée, les pentes de pixels suivants sont changées de nouveau et le tout tombe dans la boîte de catégorie 5 avec une grande population de pixels (59 éléments). Ainsi, le long de la séquence de boîte, un autre sommet apparaît, ce qui signifie que l'autre côté du sommet est formé, donc on a la forme d'un Sommet-Vallée-Sommet complète, on fixe un seuil égale à 7 pour la validation de Sommet – vallé – ratio.

Sommet – vallé – ratio = $\frac{\text{la taille de la boîte de sommet}}{\text{la taille de la de boîte de vallée}} = \frac{58}{1}$ pour le coté de segment 1.

Sommet – vallé – ratio = $\frac{\text{la taille de la boîte de sommet}}{\text{la taille de la de boîte de vallée}} = \frac{59}{1}$ pour le coté de segment 2.

Les deux valeurs de **sommet – vallé – ratio** sont supérieures par rapport au seuil choisit, donc le sommet-vallée-sommet est validé, et le point CPP faible est situé dans la vallée, les éléments entourés en rouge dans les vecteurs X22 et Y22 sont ces coordonnées cartésiennes.

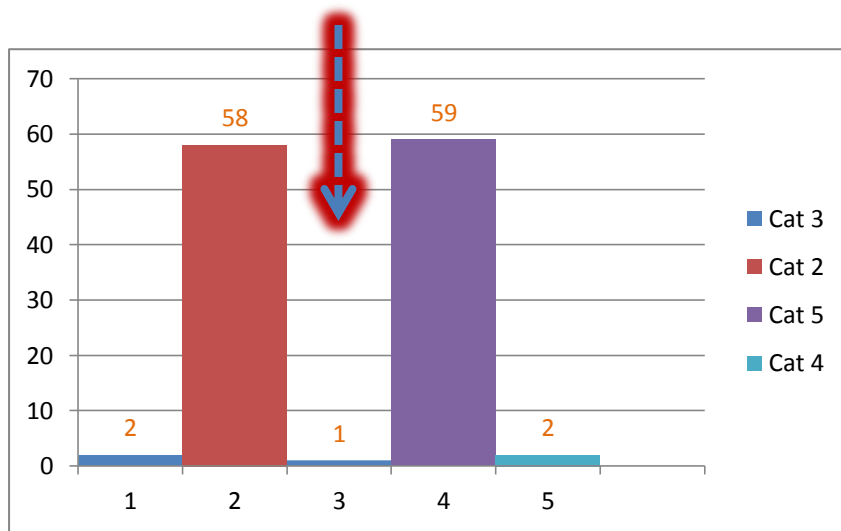


Figure III-10 : séquence de boîte correspond au segment 1et 2, la flèche indique la position de point CPP faible.

2.2 Classification

- Dans le segment 1 on a : 2 éléments de catégories 3, 58 éléments de catégorie 2 et un élément de catégorie 3, les éléments de catégorie 3 seront traité comme bruit,
- Dans le segment 2 on a : 59 éléments de catégories 5 et 2 éléments de catégorie 4.

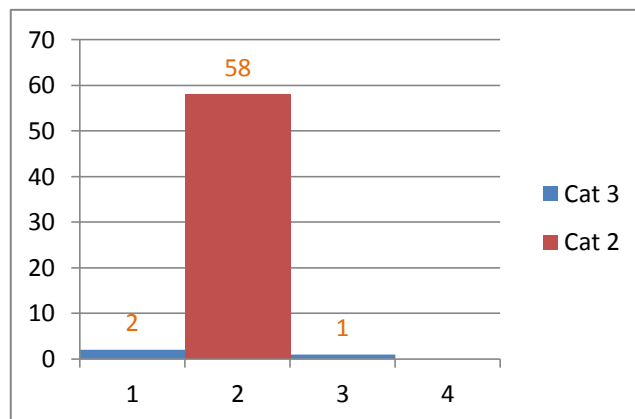


Figure III-11 : séquence de boîte correspond au segment 1 entre les deux points CPP forts.

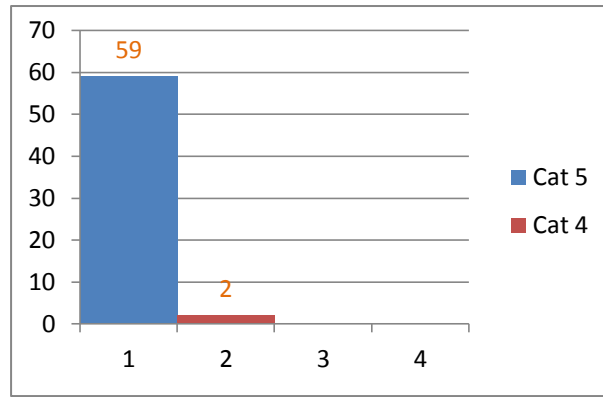


Figure III-12 : séquence de boîte correspond au segment 2 entre les deux points CPP forts.

Les deux segments 1 et 2 ont un sommet de taille sur leurs séquences de boîte, et le changement de taille entre les boîtes dans le sens croissant ou décroissant se fait brusquement, donc les deux segments sont lignes droites (section 2.1 et 2.2 de chapitre II).

Reste à déterminer la catégorie à laquelle chacun de ces deux segments appartient, LS1, LS2, LS3 ou LS4.

- Pour le segment 1 on a : le vecteur des abscisses

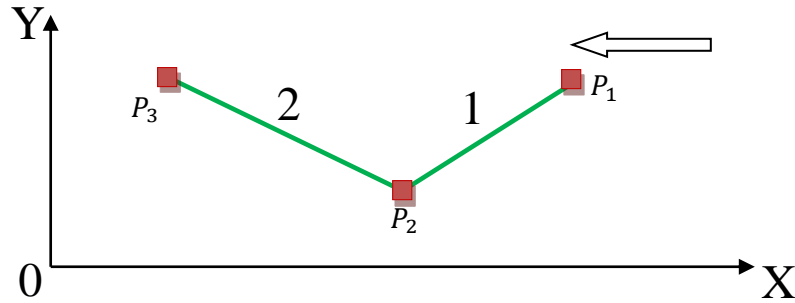
X=[245 244 243 242 241 240 239 238 237 236 235 234
 233 232 231 230 229 228 227 226 225 224 223 222
 221 220 219 218 217 216 215 214 213 212 211 210
 209 208 207 206 205 204 203 202 201 200 199 198
 197 196 195 194 193 192 191 190 189 188 187 186]

et le vecteur des ordonnées :

Y=[4 5 5 6 6 7 8 8 9 9 10 10
 11 12 12 13 13 14 15 15 16 16 17 17
 18 19 19 20 20 21 21 22 23 23 24 24

25	25	26	27	27	28	28	29	30	30	31	31
32	32	33	34	34	35	35	36	36	37	38	38]

Le contour de la forme géométrique est parcouru dans le sens antihoraire et par conséquent le segment dans le cadre rouge est parcouru dans ce sens comme il est illustré dans le schéma suivant :



Les coordonnées cartésiennes ci-dessus pour le segment 1 sont dans le sens de la flèche où le système commence par le pixel P_1 vers le pixel P_2 et pour le segment 2 : P_2 vers P_3 .

Pour se conformer au Tableau II-1 de chapitre II : pour le segment 1, le premier pixel est P_2 et le dernier est P_1 , de même avec le segment 2 : le premier pixel est P_3 et le dernier pixel est P_2 .

Donc les deux vecteurs des coordonnées cartésiennes pour les pixels de segment 1 sont comme suit :

X1_1=[186	187	188	189	190	191	192	193	194	195	196	197
198	199	200	201	202	203	204	205	206	207	208	209
210	211	212	213	214	215	216	217	218	219	220	221
222	223	224	225	226	227	228	229	230	231	232	233
234	235	236	237	238	239	240	241	242	243	244	245]

Y1_1= [119	119	120	121	121	122	122	123	123	124	125	125
126	126	127	127	128	129	129	130	130	131	132	132
133	133	134	134	135	136	136	137	137	138	138	139

140	140	141	141	142	142	143	144	144	145	145	146
147	147	148	148	149	149	150	151	151	152	152	153]

Dans le vecteur des abscisses X_{1-1} ces éléments évoluent dans le sens croissant :
 $x_1 \leq x_2 \leq x_3 \leq x_4 \leq x_5 \leq x_6 \leq \dots \leq x_{60}$, donc la fonction $X=\Phi(y)$ est monotone croissante $M+$ (7)

Dans le vecteur des ordonnées Y_{1-1} ces éléments évoluent dans le sens croissant :
 $F(x)=y : y_1 \leq y_2 \leq y_3 \leq y_4 \leq y_5 \leq y_6 \leq \dots \leq y_{60}$, donc la fonction $Y=f(x)$ est monotone croissante $M+$ (8)

Pour la fonction $f'(x)$ la dérivée de la fonction $f(x)$,

$$F'(X) = \frac{\Delta y}{\Delta x}$$

On prend par exemple un pas de dérivé égale à 3,

$$\frac{\Delta Y_1}{\Delta X_1} = \frac{Y_4 - Y_1}{X_4 - X_1} = \frac{121 - 119}{189 - 186} = 0.66$$

$$\frac{\Delta Y_2}{\Delta X_2} = \frac{Y_7 - Y_4}{X_7 - X_4} = \frac{122 - 121}{192 - 189} = 0.33$$

$$\frac{\Delta Y_3}{\Delta X_3} = \frac{Y_{10} - Y_7}{X_{10} - X_7} = \frac{124 - 122}{195 - 192} = 0.66$$

$$\frac{\Delta Y_4}{\Delta X_4} = \frac{Y_{13} - Y_{10}}{X_{13} - X_{10}} = \frac{126 - 124}{198 - 195} = 0.66$$

$$\frac{\Delta Y_5}{\Delta X_5} = \frac{Y_{16}-Y_{13}}{X_{16}-X_{13}} = \frac{127-126}{201-198} = 0.33$$

$$\frac{\Delta Y_6}{\Delta X_6} = \frac{Y_{19}-Y_{16}}{X_{19}-X_{16}} = \frac{129-127}{204-201} = 0.66$$

$$\frac{\Delta Y_7}{\Delta X_7} = \frac{Y_{22}-Y_{19}}{X_{22}-X_{19}} = \frac{131-129}{207-204} = 0.66$$

.

.

$$\frac{\Delta Y_7}{\Delta X_7} = \frac{Y_{58}-Y_{55}}{X_{58}-X_{55}} = \frac{152-150}{243-240} = 0.66$$

.

On a: $\frac{\Delta Y_1}{\Delta X_1} = \frac{\Delta Y_2}{\Delta X_2} = \frac{\Delta Y_3}{\Delta X_3} = \frac{\Delta Y_4}{\Delta X_4} = \frac{\Delta Y_5}{\Delta X_5} = \frac{\Delta Y_6}{\Delta X_6} = \frac{\Delta Y_7}{\Delta X_7} = \dots = \frac{\Delta Y_n}{\Delta X_n}$, On constate que la

fonction des dérivées est généralement constante C,..... (9)

Le système d'inférence compare les données du Tableau II-1 de chapitre II avec (7), (8) et (9) pour conclure que le segment est une ligne droite LS2 (Figure II-5(b) chapitre II), les deux vecteurs des coordonnées cartésiennes pour les pixels de segment 2 sont comme suit :

X2_1=[125 126 127 128 129 130 131 132 133 134 135 136
 137 138 139 140 141 142 143 144 145 146 147 148
 149 150 151 152 153 154 155 156 157 158 159 160
 161 162 163 164 165 166 167 168 169 170 171 172
 173 174 175 176 177 178 179 180 181 182 183
 184],

Y2_1=[153 153 152 152 151 151 150 150 149 149 148 148
 147 146 146 145 145 144 144 143 142 142 141 141
 140 140 139 138 138 137 137 136 135 135 134 134
 133 133 132 131 131 130 130 129 129 128 127 127

126 126 125 125 124 123 123 122 122 121 121
120],

Dans le vecteur des abscisses X_{2_1} ces éléments évoluent dans le sens croissant : $x_1 \leq x_2 \leq x_3 \leq x_4 \leq x_5 \leq x_6 \leq \dots \leq x_{60}$, donc la fonction $X=\Phi(y)$ est monotone croissante $M+$ (10)

Dans le vecteur des ordonnées Y_{2_1} ces éléments évoluent dans le sens décroissant : $F(x)=y : y_1 \geq y_2 \geq y_3 \geq y_4 \geq y_5 \geq y_6 \geq \dots \geq y_{60}$, donc la fonction $Y=f(x)$ est monotone décroissante $M-$ (11)

Pour la fonction $f'(x)$ la dérivée de la fonction $f(x)$,

$$F'(x) = \frac{\Delta y}{\Delta x}$$

On prend par exemple un pas de dérivée égale à 3,

$$\frac{\Delta Y_1}{\Delta X_1} = \frac{Y_4 - Y_1}{X_4 - X_1} = \frac{152 - 153}{128 - 125} = -0.33$$

$$\frac{\Delta Y_2}{\Delta X_2} = \frac{Y_7 - Y_4}{X_7 - X_4} = \frac{150 - 152}{131 - 128} = -0.66$$

$$\frac{\Delta Y_3}{\Delta X_3} = \frac{Y_{10} - Y_7}{X_{10} - X_7} = \frac{149 - 150}{134 - 131} = -0.33$$

$$\frac{\Delta Y_4}{\Delta X_4} = \frac{Y_{13} - Y_{10}}{X_{13} - X_{10}} = \frac{147 - 149}{137 - 134} = -0.66$$

$$\frac{\Delta Y_5}{\Delta X_5} = \frac{Y_{16} - Y_{13}}{X_{16} - X_{13}} = \frac{145 - 147}{140 - 137} = -0.66$$

$$\frac{\Delta Y_6}{\Delta X_6} = \frac{Y_{19}-Y_{16}}{X_{19}-X_{16}} = \frac{144-145}{143-140} = -0.33$$

$$\frac{\Delta Y_7}{\Delta X_7} = \frac{Y_{22}-Y_{19}}{X_{22}-X_{19}} = \frac{142-144}{146-143} = - 0.66$$

.
.

$$\frac{\Delta Y_7}{\Delta X_7} = \frac{Y_{58}-Y_{55}}{X_{58}-X_{55}} = \frac{121-123}{182-179} = - 0.66$$

.

On a: $\frac{\Delta Y_1}{\Delta X_1} = \frac{\Delta Y_2}{\Delta X_2} = \frac{\Delta Y_3}{\Delta X_3} = \frac{\Delta Y_4}{\Delta X_4} = \frac{\Delta Y_5}{\Delta X_5} = \frac{\Delta Y_6}{\Delta X_6} = \frac{\Delta Y_7}{\Delta X_7} = \dots = \frac{\Delta Y_n}{\Delta X_n} = 0.66,$ On

constate que l'évolution de la fonction des dérivées est généralement constante C,..... (12).

Le système d'inférence compare les données du Tableau II-1 de chapitre II avec (10), (11) et (12) pour conclure que le segment est une ligne droite LS1 (Figure II-7 b Chapitre II), et il déduit le point de jonction CPP8 entre les deux segments LS1 et LS2 sur la base des règles du tableau 5 de chapitre II, donc la description structurelle de la forme incluse dans la fenêtre rouge de Figure III-9 est :

$$\{LS2 \ CPP8 \ LS1\}.$$

Et de même nous concluons la description structurelle de la forme géométrique de la Figure III-9 :

$$\{ LS1 \ CPP8 \ LS1 \ CPP8 \ LS1 \ CPP8 \ LS2 \ CPP8 \ LS1 \ CPP8 \ LS2 \ CPP8 \ LS1 \ CPP8 \ LS2 \ CPP8 \ LS1 \ CPP8 \ LS1 \ CPP8 \}.$$

La chaîne est composée avec des lignes droites et des points partageants la courbe, elle présente la description structurelle de la forme géométrique de la Figure III-9, cette description sera utilisée dans l'étape de reconnaissance,

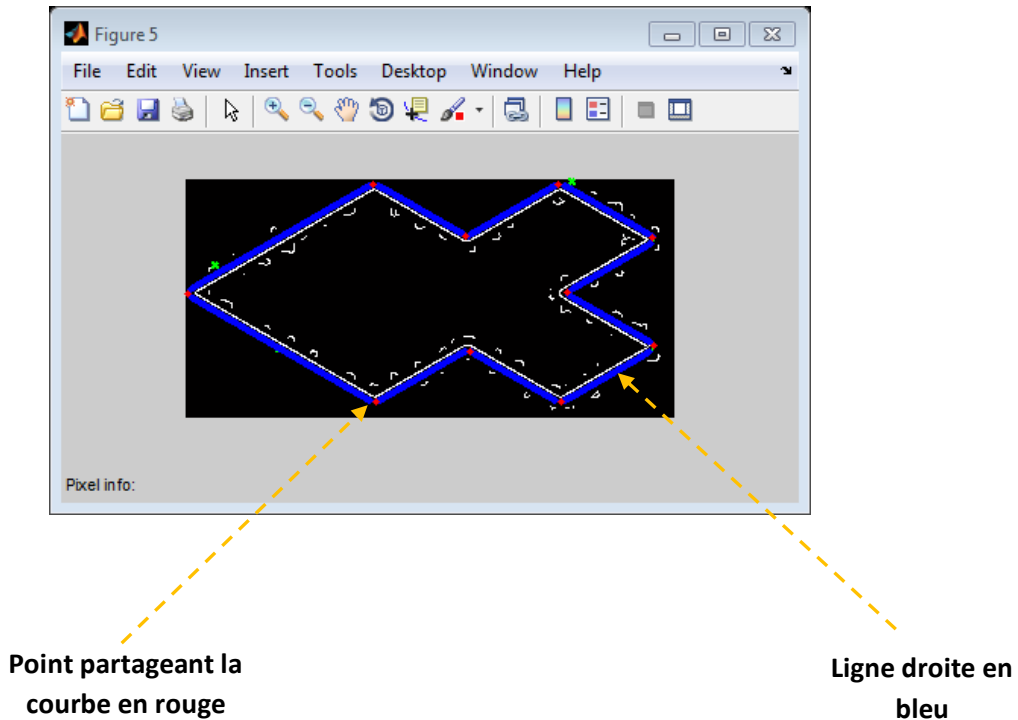
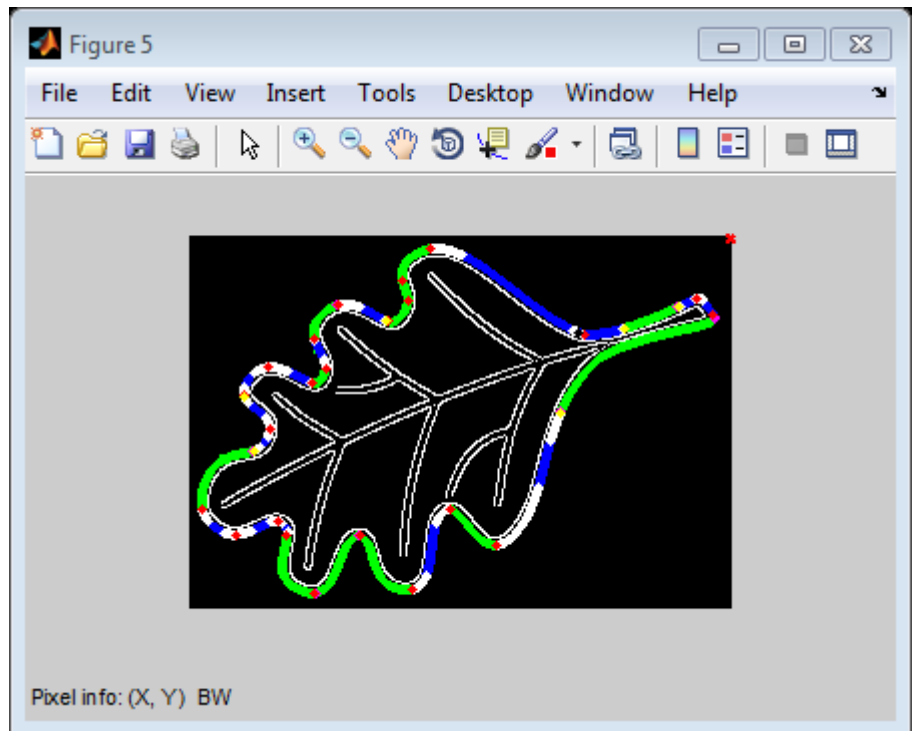


Figure III-13 : la forme géométrique de la Figure III-9 segmentée en ligne droites.

Dans l'étape de reconnaissance on utilise un système d'inférence afin de comparer les chaînes de description structurelle point par point tout en respectant l'ordre d'agencement des éléments (les arcs, les lignes droites, les CPP) dans la chaîne pour reconnaître les formes.

Figure III-14 : une feuille d'arbre segmentée en arcs et en segments de droites.



- Les éléments marqués en vert sont des arcs,
- Les éléments marqués en bleu sont des lignes droites,
- Les éléments marqués en blancs représentent le bruit lié au segment de ligne droite,
- Les points en rouge sont les points CPP forts,
- Les points en cyan sont des point CPP faibles (Sommet - vallée – Sommet),
- Les points en jaune sont les points CPP faibles type Changement de la
Tendance de Catégories Arc-tangente.

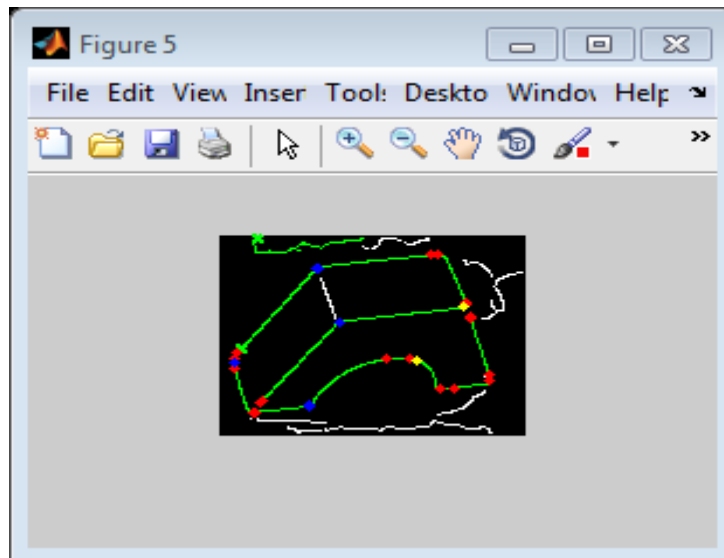


Figure III-15 : forme géométrique,

- Les points en rouge sont les points CPP forts,
- Les points en bleu sont des points CPP faibles, Sommet - vallée – Sommet
- Les points en jaune sont les points CPP faibles Changement de la Tendence de Catégories Arc-tangente.

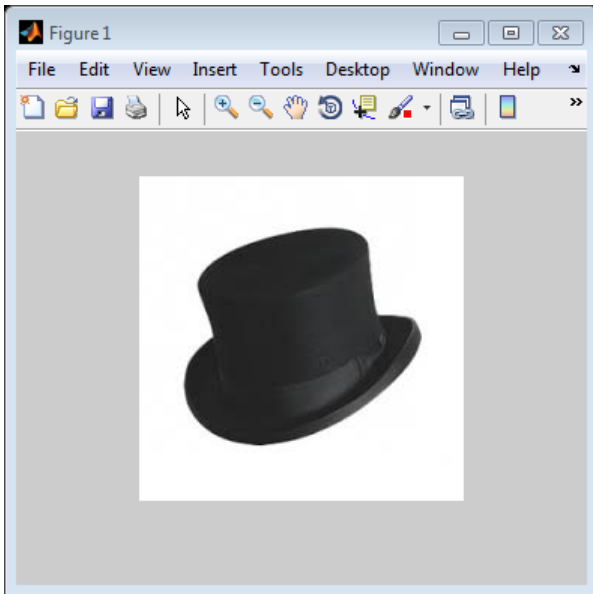


Figure III-16 : le chapeau de Wellington.

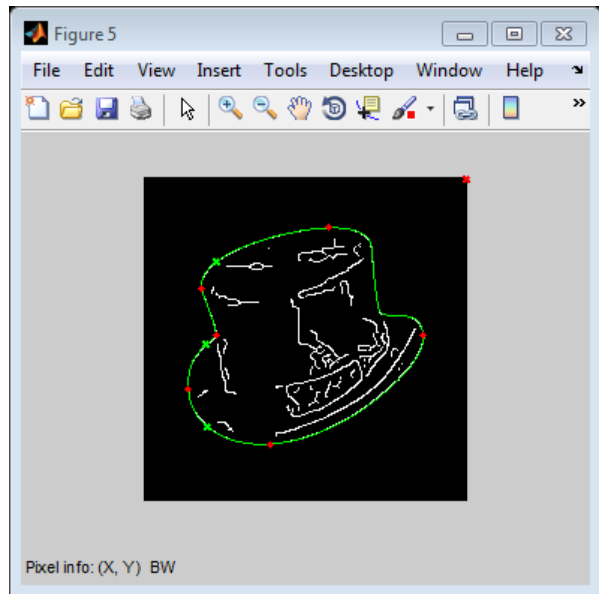


Figure III-17 : les points CPP forts en rouge

Les points en rouge dans la Figure III-15 sont les points CPP forts, les segments entre les points CPP forts sont composés (brute) et ont besoins d'être partitionnés de plus (chapitre I section 4.2).

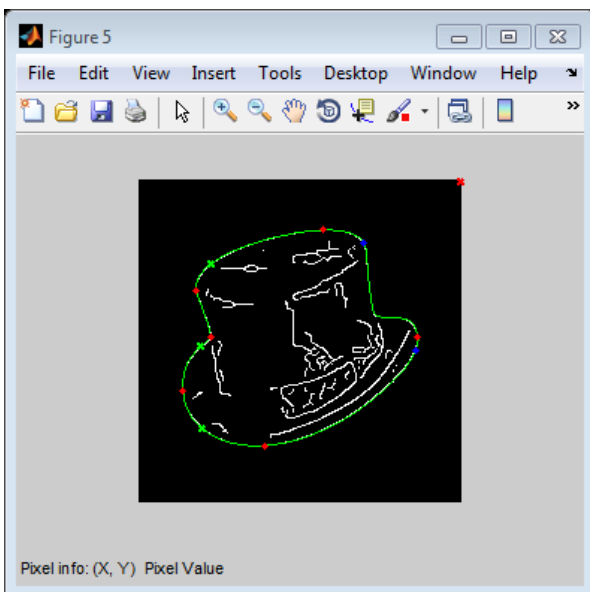


Figure III-18 : les points CPP faibles en bleu type (Sommet-vallée-sommet).

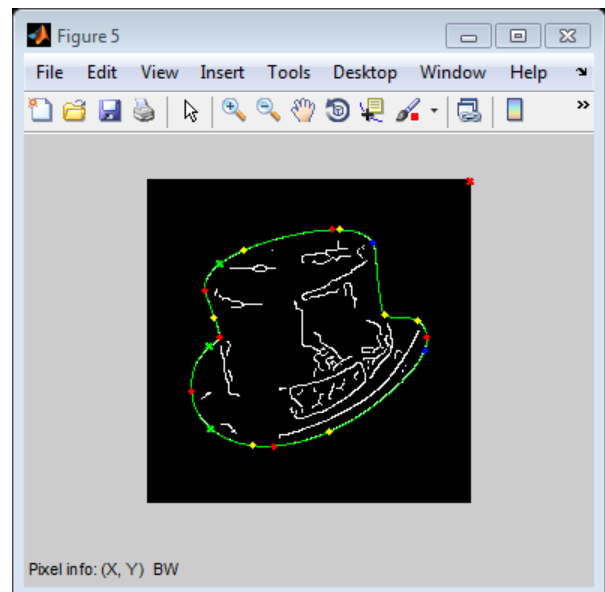


Figure III-19 : les points CPP faibles en jaune type (Changement de la Tendance de Catégories Arc-tangente).

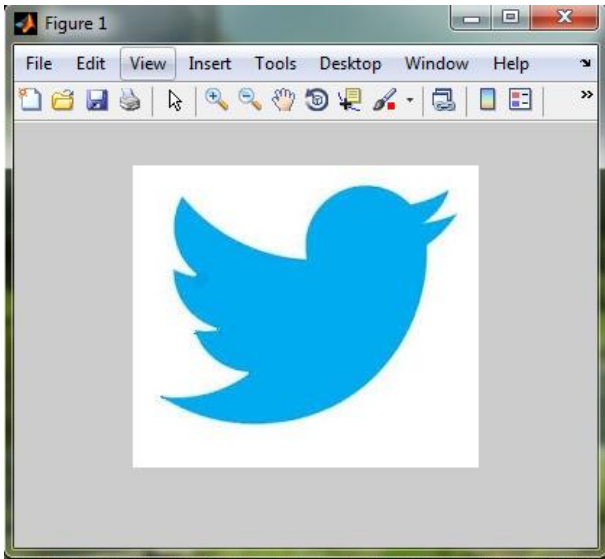


Figure III-20 : l'image initiale.

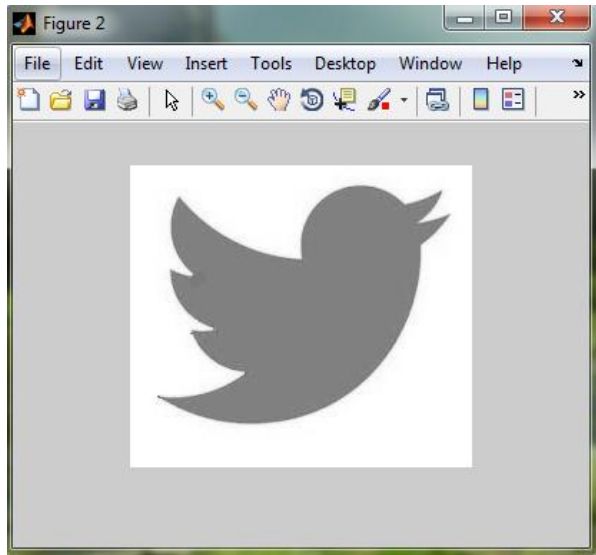


Figure III-21 : l'étape de binarisation initiale.

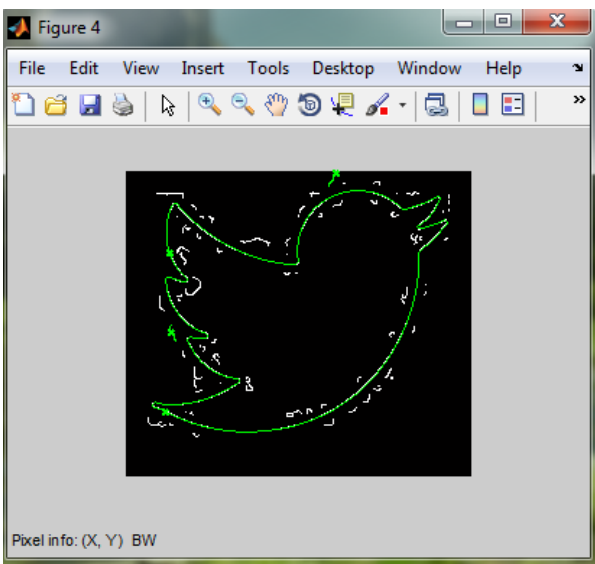


Figure III-22 : le contour de la forme.

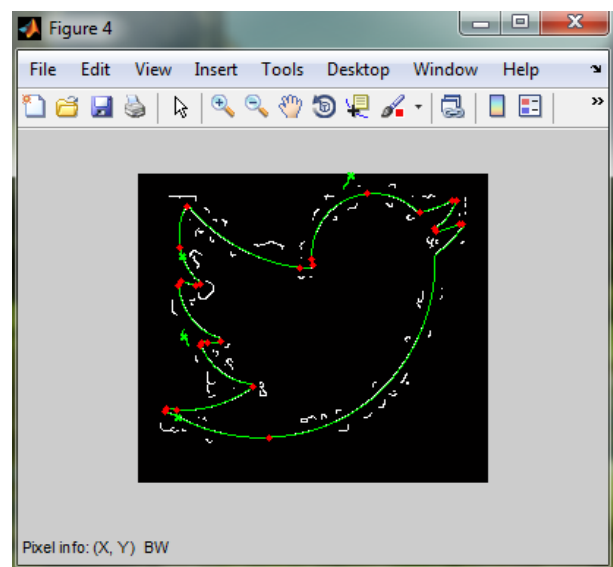


Figure III-23 : les points CPP forts.

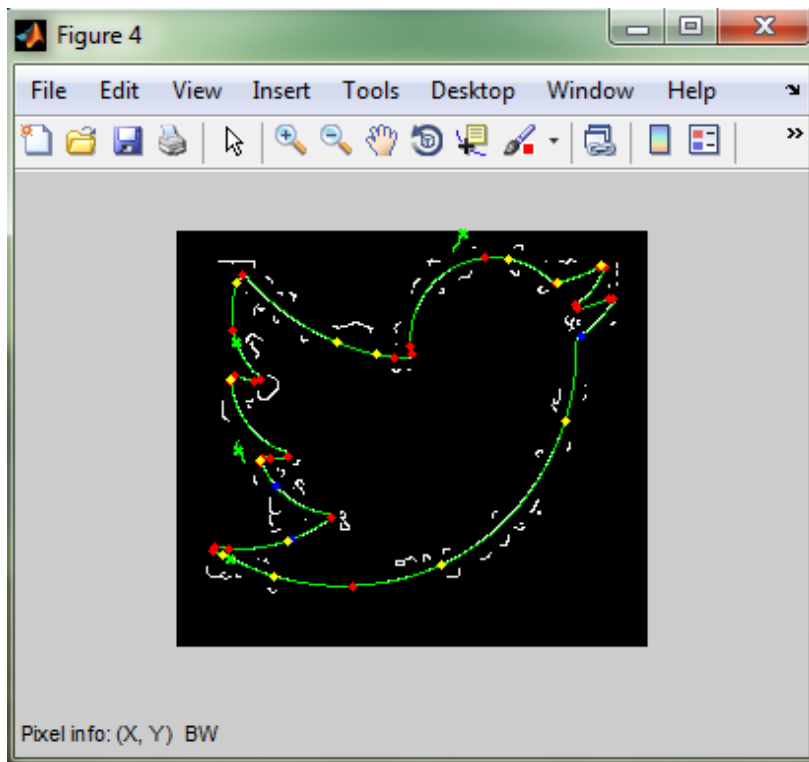


Figure III-24 : Tous les points CPP, Les points CPP forts en rouge, les points CPP faibles type sommet-vallée-sommet en bleu et les points CPP faibles type changement de la tendance des catégories en jaune.

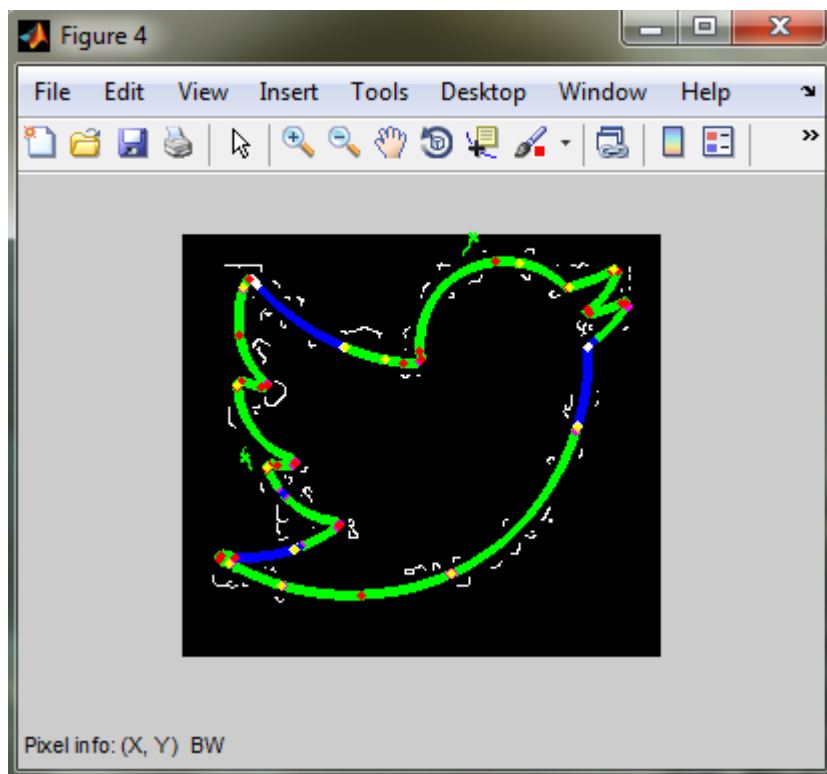


Figure III-25 : les lignes droites en bleu et le bruit type ligne droite en blanc, les arcs en verts.

Sa description structurale est :{CS1 CPP7 LS4 CPP7 CS1 CPP5 CS3 CPP2 CS4
 CPP4 CS1 CPP5 CS3 CPP2 CS4 CPP2 CS3 CPP7 LS1 CPP7 CS4 CPP7 LS3
 CPP7 CS3 CPP3 CS2 CPP6 CS4 CPP2 CS3 CPP3 CS2 CPP7 LS3 CPP7 CS4
 CPP6 CS2 CPP7 LS3 CPP8 LS3 CPP7 CS4 CPP6 CS2 CPP1},

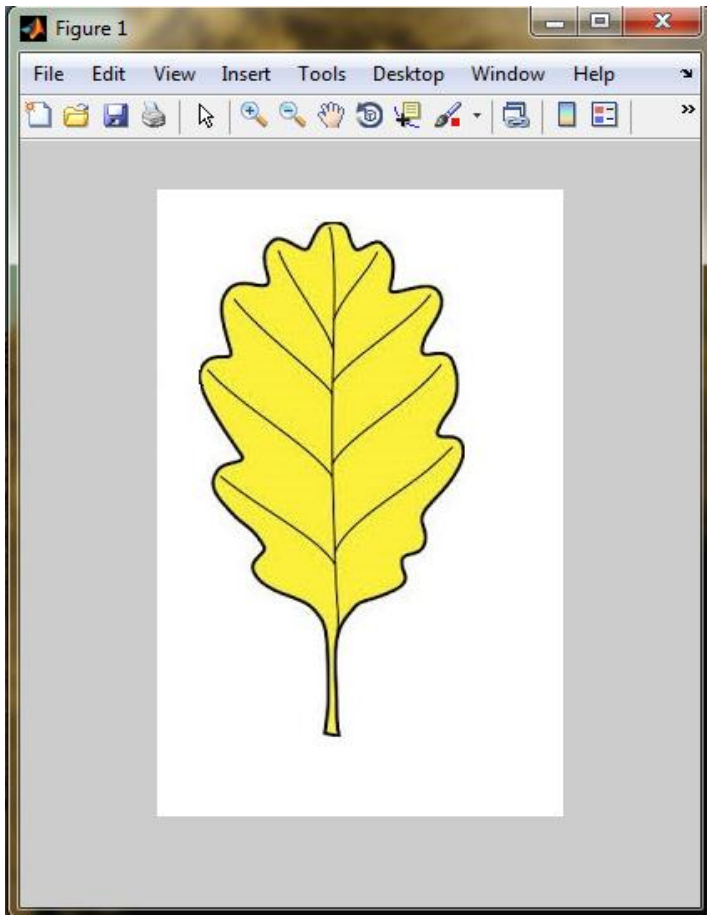


Figure III-26 : feuille d'arbre.

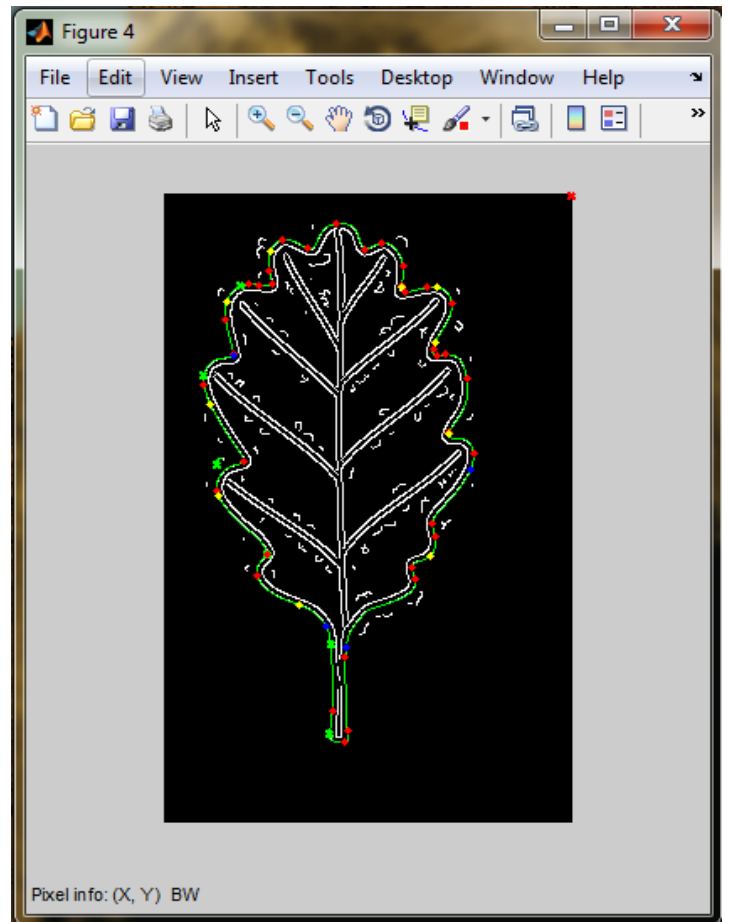


Figure III-27 : Tous les points CPP, Les points CPP forts en rouge, les points CPP faibles type sommet-vallée-sommet en bleu et les points CPP faibles type changement de la tendance des catégories en jaune.

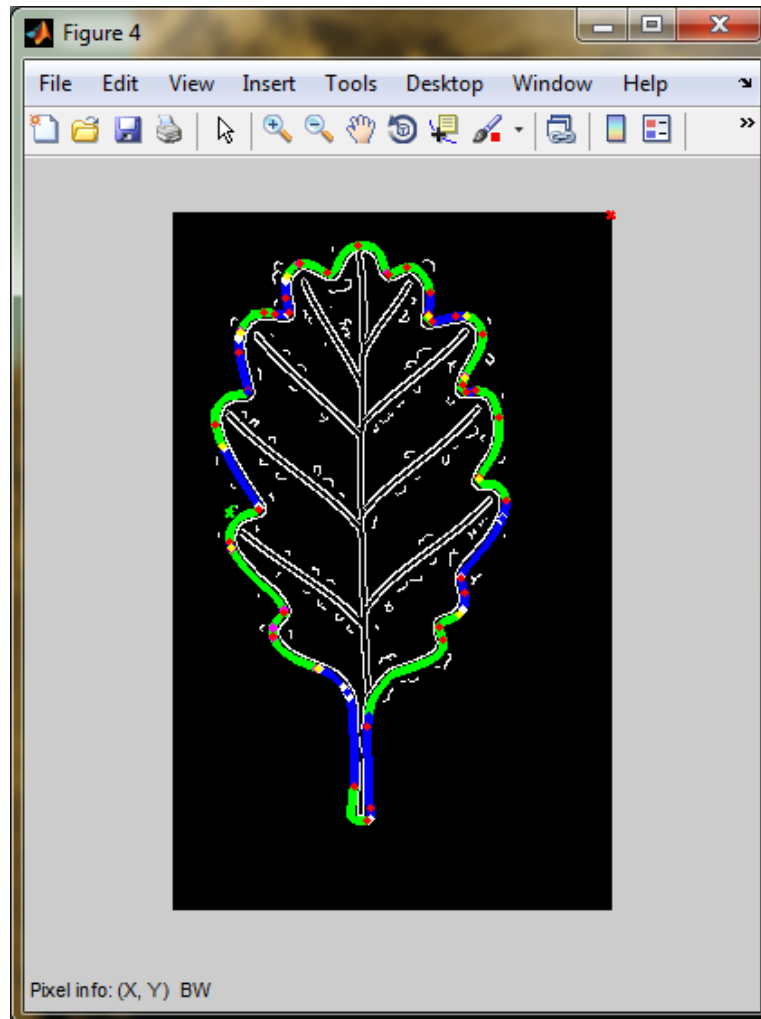


Figure III-28 : les lignes droites en bleu et le bruit type ligne droite en blanc, les arcs en verts.

Sa description structurelle est :

{LS4 CPP8 LS4 CPP8 LS4 CPP7 CS1
 CPP4 CS4 CPP4 CS1 CPP7 LS4 CPP8 LS4 CPP8 LS2 CPP8 LS4
 CPP7 CS4 CPP4 CS1 CPP4 CS4 CPP7 LS3 CPP7 CS4 CPP4 CS1
 CPP4 CS4 CPP7 LS3 CPP8 LS3 CPP7 CS4 CPP7 LS4 CPP7
 CS4 CPP2 CS3 CPP2 CS4 CPP2 CS3 CPP2 CS4 CPP2 CS3
 CPP7 LS4 CPP8 LS4 CPP8 LS3 CPP7 CS4 CPP2 CS3 CPP7 LS4
 CPP8 LS4 CPP7 CS3 CPP3 CS2 CPP7 LS1 CPP7 CS3 CPP7 LS1 CPP7
 CS2 CPP3 CS3 CPP3 CS2 CPP7 LS1 CPP8 LS4 CPP7 CS3 CPP7}.

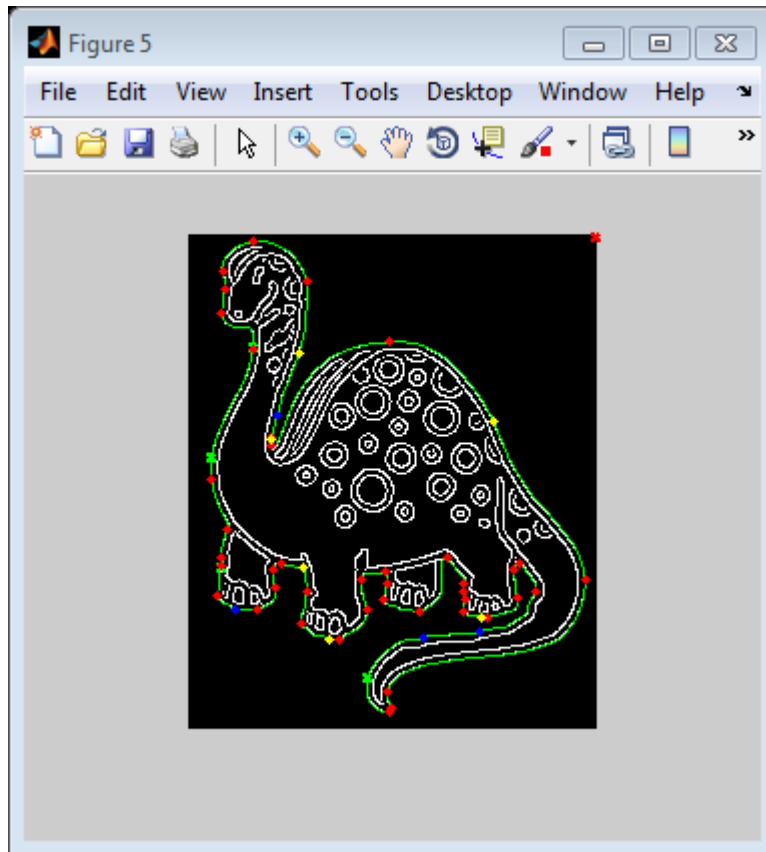


Figure III-29 : dinosaure herbivore, Les points CPP forts en rouge, les points CPP faibles type sommet-vallée-sommet en bleu et les points CPP faibles type changement de la tendance des catégories en jaune.

3. Comparaison entre la méthode proposée et la méthode du gradient locale :

La figure III-29 fournit de brefs résultats de comparaison. Le plus haut est l'image originale, la figure III-29 (a) montre les résultats de détection de CPP à partir de la méthode du gradient local à base de courbure [19] qui apporte un peu de bruit (entouré). La figure III-29 (b) vient de la méthode proposée qui n'est pas seulement capable de découvrir les CPP tant forts que faibles, mais de supprimer aussi les CPP bruyant. La méthode de détection de CPP proposée fournit des résultats prometteurs.

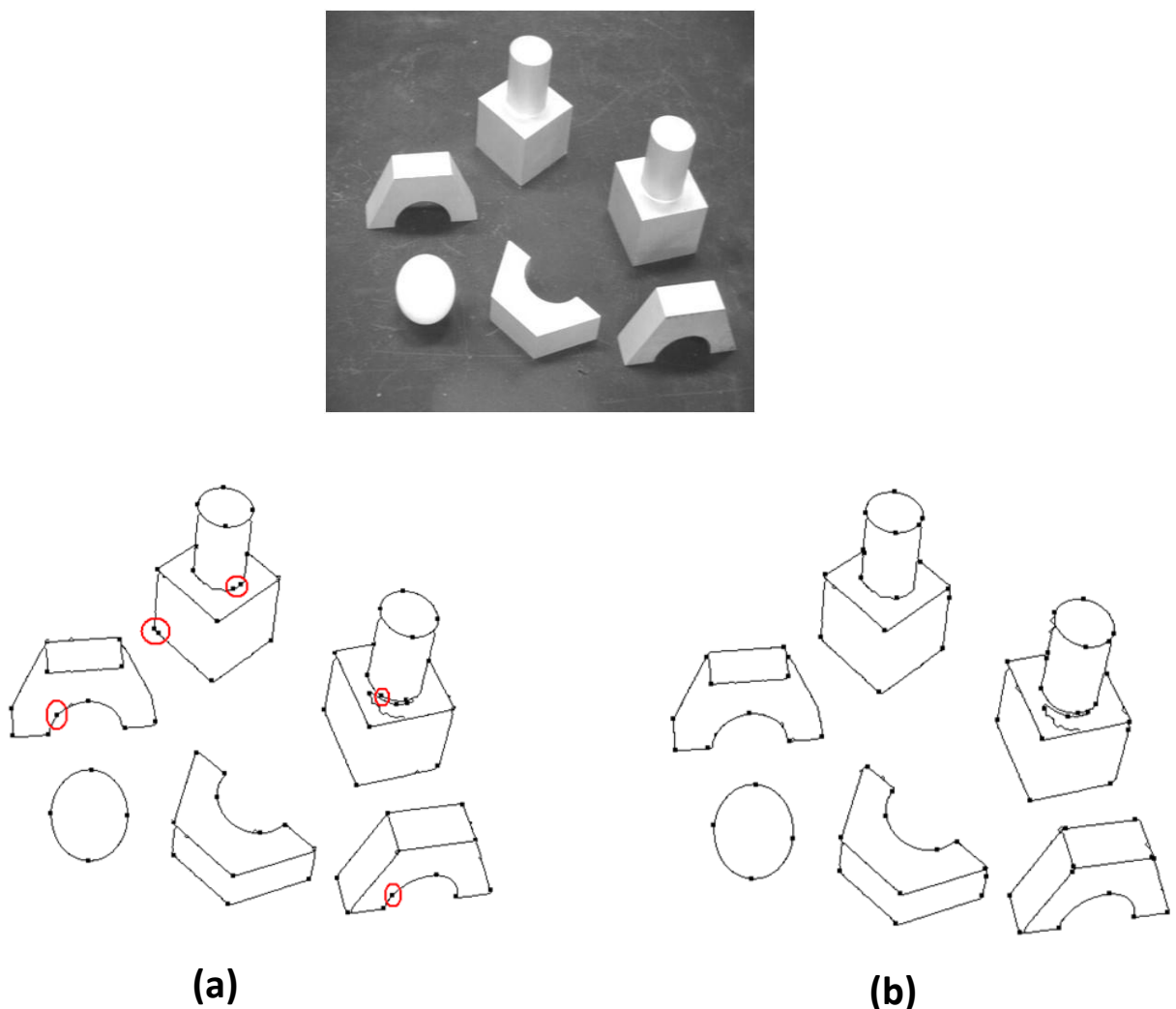


Figure III-30 : comparaison des résultats obtenus de la méthode proposée et celle du gradient local.

4. Evaluation de la méthode

L'évaluation de la méthode est basée sur deux critères principaux, le premier est le rapport entre l'erreur dans la position des points CPP et le nombre total des points détectés, le deuxième est le rapport entre les points manquant par rapport au nombre total des points détectés

4.1. Cercle

Nombre de points détectés : 04

Nombre de points manquants : 00

L'erreur = 00

Critère 1 = $((\text{Nb de points détectés} - \text{L'erreur}) / \text{Nb de points détectés}) * 100\% = 100\%$

Critère 2 = $((\text{Nb de points détectés} - \text{Nb de points manquants}) / \text{Nb de points détectés}) * 100\% = 100\%$

4.2. Forme géométrique

Nombre de points détectés : 11

Nombre de points manquants : 00

L'erreur = 00

Critère 1 = $((\text{Nb de points détectés} - \text{L'erreur}) / \text{Nb de points détectés}) * 100\% = 100\%$

Critère 2 = $((\text{Nb de points détectés} - \text{Nb de points manquants}) / \text{Nb de points détectés}) * 100\% = 100\%$

4.3. Feuille d'arbre (Figure III-14)

Nombre de points détectés : 30

Nombre de points manquants : 00

L'erreur = 01

Critère 1 = $(\text{Nb de points détectés} - \text{L'erreur}) / \text{Nb de points détectés} * 100\% = 96.66\%$

Critère 2 = $(\text{Nb de points détectés} - \text{Nb de points manquants}) / \text{Nb de points détectés} * 100\% = 100\%$

4.4. Le dinosaure herbivore (Figure III-29)

Nombre de points détectés : 51

Nombre de points manquants : 02

L'erreur = 01

Critère 1 = $(\text{Nb de points détectés} - \text{L'erreur}) / \text{Nb de points détectés} * 100\% = 98\%$

Critère 2 = $(\text{Nb de points détectés} - \text{Nb de points manquants}) / \text{Nb de points détectés} * 100\% = 96.1\%$

Conclusion :

Dans ce chapitre nous avons présenté en détails les résultats obtenus, dans l'étape de segmentation nous avons effectué :

- L'extraction du contour de la forme en utilisant le filtre de canny,
- l'enregistrement des pixels de contour dans un tableau de tracé de pixels,
- l'application du théorème des traversant zéro afin de trouver les points CPP forts, et l'élaboration du schéma des OPABS pour trouver les points CPP faibles.

Pour l'étape de classification, nous avons d'abord procédé par une classification du premier niveau tout en distinguant deux grandes classes, les lignes droites et les arcs, puis nous avons fait la distinction entre les quatre lignes droites et les quatre arcs dans la classification du deuxième niveau tout en attribuant chaque segment à sa propre classe, ensuite, les points partageants la courbe sont déterminés et classés.

Après avoir construit les huit classes des segments, et les huit classes des points partageants la courbe, nous construisons une chaîne de description structurelle de ces éléments tout en respectant l'ordre d'agencement de tous les éléments dans la chaîne, c'est l'étape de reconnaissance.

CONCLUSION GÉNÉRALE

Conclusion générale

Dans ce mémoire, nous avons présenté une nouvelle approche de reconnaissance de formes en utilisant leurs descriptions structurales.

Cette étude a consisté à créer en premier lieu une base de données, à partir des images prises pour différentes formes (feuilles d'arbres, formes géométriques, animaux, etc.), à partir de cette base nous avons extrait le contour de la forme en utilisant le filtre de JF canny, les coordonnées cartésiennes des pixels de contour sont ensuite enregistrés dans un tableau de pixels candidats.

La segmentation de contour en primitives simples (arc, segment de droite) consiste à trouver les points limitant chaque segment, les segments adjacents partagent un même point appelé point partageant la courbe CPP.

La localisation des points partageants la courbe passe par deux phases :

La première consiste à trouver les points CPP forts par l'application du théorème des traversant zéro.

La seconde consiste à localiser les points CPP faibles, pour cela nous avons utilisé le schéma des OPABS qui permet de caractériser les points ayant enregistré une forte variation dans les propriétés de l'ordre de séquences des boîtes.

La classification des primitives de contour segmenté s'effectue à deux niveaux :

Dans le premier on a procédé par la distinction entre les arcs et les segments de droites tout en attribuant chacune de ces primitives à sa propre classe par l'utilisation d'un système d'inférences. À la fin de cette étape on aura deux classes (la classe des arcs et la classe des lignes droites).

Dans le deuxième niveau, chaque classe est partagée à son tour en quatre classes différentes, par un système d'inférences qui utilise les propriétés géométriques des primitives, à la fin, nous aurons huit classes différentes.

Les points partageants la courbe CPP sont ensuite classés en huit classes, par un système d'inférences en utilisant l'information des deux segments partageants le même point CPP.

À la fin de l'étape de classification on aura huit classes de primitives (arcs et segments de droites) et huit classes des points CPP. Ces résultats serviront comme données de base pour la construction de la chaîne de description structurelle. Cette dernière est utilisée dans la reconnaissance de formes.

Cette étude que nous jugeons satisfaisante et prometteuse est d'une importance pratique majeure et nécessite d'être exploitée dans des systèmes de reconnaissance de formes de grandes performances.

Mots-clés : reconnaissance de formes, reconnaissance structurelle, système d'inférences, contours.

Bibliographie

- [1] H. Tamura, S. Mori, T. Yamawaki, Texture features corresponding to visual perception, IEEE Transactions on Systems, Man and Cybernetics, 8: 460-473, 1978.
- [2] M. Amadasun, R. King, Textural features corresponding to textural properties, IEEE Trans. On Systems, Man and Cybernetics, 19: 1264-1274, 1989.
- [3] D. Zhang and G. Lu, Study and evaluation of different Fourier methods for image retrieval, In Image and Visual Computing, 23: 33-49, 2005.
- [4] Q. Gao and A. Wong, Curve detection based on perceptual organization, Pattern Recognition, 26(1): 1039-1046, 1993.
- [5] X. Zheng, S. A. Sherrill-Mix, Q. Gao, Perceptual shape-based natural image representation and retrieval, In Proc: The 1st IEEE Int. Conf. on Semantic Computing (ICSC2007), 622-629, 2007.
- [6] G. Hu and Q. Gao, Interactive image feature visualization for supporting CBIR study, In Proc. Int. Conf. on Image Analysis and Recognition, LNCS 5627: 239-247, 2009.
- [7] H. Chen and Q. Gao, Integrating color and gradient into real-time curve tracking and feature extraction for video surveillance, Book Title: Video Surveillance, Chapter 12, Publisher InTechOpen, ISBN: 978-953-307-436-8: 217-230, 2011.
- [8] http://en.wikipedia.org/wiki/Non-parametric_statistics [retrieved 8 May 2012]
- [9] D.G. Lowe, Distinctive image features from scale-invariant keypoints, Journal of Computer Vision, 60(2):91-110, 2004.
- [10] K. Mikolajczyk, C. Schmid, A performance evaluation of local descriptors, IEEE Trans. on Pattern Analysis and Machine Intelligence, 27(10):1615-1630, 2005.
- [11] J.-P. Cocquerez et S. Philipp. « Analyse d'Images : filtrage et Segmentation ». Masson, 1995.
- [12] J. M. S. Prewitt, "Object enhancement and extraction," Picture Processing & Psychopictorics, Academic Press, New York, pp. 75-149, 1970.
- [13] I. Sobel. Neighbourhood coding of binary images for fast contour following and general array binary processing. Computer Graphics and Image Processing, vol.8, pp. 127-135, 1978.
- [14] R. Kirsch. Computer determination of the constituent structures of biomedical images. Computer and Biomedical Research. USA, vol. 4, n° 3, pp 315-328, 1971.
- [15] J. Shen and S. Castan. An optimal linear operator for edge detection. In Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'86), Miami Beach, Florida, pp. 109--114, USA, 1986.
- [16] S.Castan, J. Zhao and J. Shen. Une famille de détecteurs de contours basée sur filtre exponentiel.optimal. In AFCET-RFIA, Paris, 1989, p. 23-. 36. [6] J.
- [17] J. Shen and S. Castan_ An optimal linear operator for step edge detection. CVGIP. Graphics Models and Image Processing. 54(2). 112-133, Mars 1992.
- [18] J. Canny, A computational approach to edge detection, IEEE Trans. On Pattern Analysis and Machine Intelligence, vol. 8, n°6, pp. 679-698, 1986.
- [19] Y. Li. Generic edge feature extraction based on perceptual curve partitioning, Master thesis, Dalhousie University, 2004.

- [20] G. HU and Q. gao, a non-parametric statistics based method for generic curve partition and classification, 978-1-4244-7994-8/10, 2010 IEEE. pp. 3041–3044.
- [22] V. AHO, J. D. ULLMAN, the theory of parsing, translation and compiling, prentice-Hall, 1972.
- [23] H.BUNKE (ed.), advance in structural and syntactic pattern recognition, World scientific, 1992.
- [24] John R. Platt Science, New Series, Vol. 146, No. 3642. (Oct. 16, 1964), pp. 347-353.
- [25] Glynn Winskel. The Formal Semantics of Programming Languages: An Introduction, MIT Press, 1993, pp.10-26.
- [26] Jean-Yves Girard Proof Theory and logical complexity, ch. 6.
- [27] A.R. Anderson et N.D, Jr. Belnap, Entailment, vol. 1, Princeton, NJ, Princeton, 1975.
- [28] Jon Barwise et John Etchemendy, Language, Proof and Logic, Stanford, CSLI Publications, 2008.
- [29] Jacob Feldman. The role of objects in percetual grouping. Acta Psychologica, 102:137,163,1999.
- [30] S. Negahdaripour and A.K. Jain. Challenges in computer vision: Future research directions. IEEE Proc. Comput. Soc. Conf. Computer Vision and Pattern Recognition, 92:189,198, 1992.
- [31] QiGang Gao and A.K.C. Wong. Curve detection based on perceptual organization. Pattern Recognition, 26(1):1039,1046, 1993.
- [32] Qigang Gao, Yun Zhang and Alan Parslow Motion Stream Analysis based on Perceptual Feature Partitioning and Grouping IEEE Intelligent Transportation Systems Conference Washington, D.C., USA, October 36,2004.
- [33] D.G. Lowe. Three-dimensional object recognition from single two-dimensional images. Arti ficial Intelligence, 31(3):1121,1139, 1987.
- [34] W.E.L. Grimson. Object Recogniton By Computer: The Role of Geometric Constraints. MIT press, first edition, 1990.

Résumé

Nous présentons dans ce mémoire un système de reconnaissance de formes basé sur la description structurelle.

Les éléments formant la chaîne de description structurelle sont des arcs, lignes droites et les points partageants la courbe CPP.

Ces éléments sont issus d'un processus de segmentation avec l'utilisation des deux techniques, le théorème des traversant zéro pour les CPP forts et le schéma des OPABS pour les CPP faibles, après ils sont classés à l'aide d'un système d'inférence.

Pour nos applications, nous avons utilisés une base de données regroupant différentes formes.

Summary

We present in this paper a pattern recognition system based on the structural description.

The elements forming the structural description string are arcs, straight lines and curve partitioning points CPP.

These elements are derived from a process of segmentation using two techniques, the theorem of crossing zero for strong CPP and scheme of the OPABS for low CPP, after they are classified using a system of inference.

For our applications, we have used a data base of various forms.