

République Algérienne Démocratique et Populaire
Ministère de l'enseignement supérieur et de la recherche scientifique
Université des Sciences et de la Technologie Houari Boumedienne

Faculté d'Electronique et d'Informatique

MEMOIRE

Présenté pour l'obtention du diplôme de MAGISTER

En : INFORMATIQUE

Spécialité : Systèmes Intelligents et Ingénierie du Logiciel

Par : AIT-AOUDIA Amina

Sujet :

***Indexation et reconnaissance des
silhouettes d'objets en présence d'occlusion***

Soutenu publiquement, le 26 /12/2013 devant le jury composé de :

M. LARABI Slimane Professeur, à l'USTHB Président

Mme. AOuat Saliha Professeur/Maître de conférence/A, à l'USTHB Directrice de Mémoire

Mme. Baha Touzene Nadia Professeur/Maître de conférence/A, à l'USTHB Examineur

M. FEREDJ Mohamed Professeur/Maître de conférence/A, à l'USTHB Examineur

*A la mémoire de mon Père et
a ma très chère Mère.*

AVANT-PROPOS

Ce mémoire a été préparé au sein du laboratoire LRIA de la faculté d'électronique et d'informatique de l'USTHB.

Je tiens à remercier Madame Aouat Saliha, Directrice de ce mémoire, qui m'a encadré tout au long de la préparation de celui-ci.

Je tiens aussi à remercier Professeur Slimane LARABI, Dr. Nadia BAHA et Dr. Mohamed FEREDJ, d'avoir accepté de faire partie de mon jury.

Je remercie tous ceux qui m'ont aidé à réaliser ce mémoire.

Je remercie mon défunt père et ma mère pour m'avoir encouragé à faire ce mémoire. Sans leurs encouragements, je crois que je ne l'aurais jamais entamé. Je remercie spécialement ma mère, qui m'a toujours tout donné, pour moi et pour mes enfants.

Je remercie mes frères, mes neveux et nièces, mes beaux-parents, mes oncles, mes tantes, mes cousins et mes cousines.

Je remercie mon mari et mes enfants Abdelmalek et Chaïma, mes trésors, ma joie de vivre.

RÉSUMÉ.

La méthode de reconnaissance élaborée combine deux méthodes de reconnaissance d'objets par leurs silhouettes. La première méthode utilisée est la méthode du Quadtree mais traitant initialement des objets en absence d'occlusion et la deuxième méthode est la reconnaissance d'objets par leurs courbures. L'intérêt de cette combinaison est de palier à certains inconvénients de la méthode du Quadtree pour la reconnaissance d'objets comme par exemple sa non robustesse à la rotation. En plus de la combinaison des deux méthodes, le défi principal de notre approche est la possibilité de reconnaître les objets même en présence d'occlusion.

On définit dans ce mémoire ce que l'on nomme seuil de tolérance à l'occlusion, ce seuil a contribué dans la reconnaissance de l'objet partiellement occulté. Le système de reconnaissance établi est robuste à la rotation.

Table des matières :

RÉSUMÉ.....	4
Introduction générale.....	8
Chapitre 1 : Etat de l'art sur les méthodes d'indexation et la reconnaissance d'objets.	11
1. Introduction :	11
2. Reconnaissance d'objets (l'aspect général):	12
2.1. Segmentation :	13
2.2. Classification :	15
2.3. Reconnaissance d'objets :	16
2.3.1. Méthodes géométriques:	17
2.3.2. Vision par apparence :	18
2.3.2.1. Méthodes Globales :	19
2.3.2.2. Méthodes Locales :	20
2.3.2.3. Modélisation par histogrammes :	22
2.3.2.4. Modélisations statistiques et probabilistes :	22
2.3.2.5. Caractérisation par graphes :	24
3. Méthodes se basant sur les silhouettes pour la reconnaissance et l'indexation d'objets : ...	24
3.1. L'approche de Huttenlocher :	25
3.1.1. Distance de Hausdorff :	25
3.1.2. Méthodes de Hetttenlocher :	27
3.2. L'approche de Radack et Badler :	30
3.3. L'approche de Rodriguez et Aggarwal :	31
3.4. L'approche de Kamgar-Parsi :	33
3.5. L'approche de Nelson et Selinger :	34
3.6. L'approche de Belongie et Jitendra:	34
4. Conclusion :	35
Chapitre 2 : Les méthodes de reconnaissance d'objets en présence d'occlusion.	37
1. Introduction:	37
2. La vision Humaine:	38
3. Le schéma général des méthodes de reconnaissance d'objets :	39
4. Méthodes se basant sur la silhouette pour la reconnaissance d'objets en présence d'occlusion :	40
4.1. Techniques Corrélatives:	40
4.2. Techniques syntaxiques:	42

4.3. <i>Techniques d'analyse d'histogramme :</i>	43
4.3.1. Transformée de Hough pour la reconnaissance d'objets partiellement occultés :	43
4.4. <i>Techniques de recherche arborescente :</i>	44
5. Autres techniques :	45
6. Conclusion :	50
Chapitre 3 : Indexation et Reconnaissance.	52
1. Introduction :	52
2. Architecture globale d'indexation et de reconnaissance d'objets :	53
3. Etape indexation :	54
3.1. Technique d'indexation utilisant la structure Quadtree :	55
3.1.1. Définition :	55
3.1.2. Recherche :	56
3.1.3. Insertion :	57
3.1.4. Avantages des Quad-Tree :	57
3.1.5. Les images en Quad-Tree	57
3.1.6. L'arbre Quartenaire :	58
3.1.7. Codage de l'image	59
3.1.8. Algorithme de découpage :	62
3.2. Méthodologie utilisant les informations de courbures pour l'indexation et la reconnaissance d'objets :	63
3.2.1. Présentation de la méthodologie :	63
3.2.2. Algorithme d'indexation :	65
4. Reconnaissance d'objets :	66
4.1. Reconnaissance par la méthode du Quadtree:	66
4.1.1. Stratégie de recherche :	67
4.3. Etape de reconnaissance d'objet utilisant les informations de courbures :	71
4.4. Combinaison de la technique utilisant l'angle de courbure et la technique du Quadtree :	72
4.4.1. Indexation des images de la base :	73
4.4.2. La reconnaissance d'objet :	73
5. Généralisation de la méthode du Quadtree:	73
5.1. Exemple pour la généralisation de la méthode du Quadtree:	74
5.2. Démonstration par récurrence de la Généralisation :	77
6. Conclusion :	79

Chapitre 4 : Expérimentation.	81
1. Introduction :	81
2. Indexation des images par la méthode du Quadtree :	83
3. Indexation des Images par la méthode utilisant les informations de courbures pour la reconnaissance d'objets :	85
4. Quelques résultats d'Expérimentation :	88
4.1 Donnée de Synthèse :	88
4.1.1. Reconnaissance des objets :	89
• Seuil de tolérance :	89
4.2 Images réelles :	93
5. Conclusion :	98
Conclusion générale	99
Référence:.....	100

Introduction générale

La vision artificielle ou vision par ordinateur est une branche du domaine d'intelligence artificielle. Le but de la vision par ordinateur est de faire en sorte que la machine comprenne ce qu'elle « voit » par le biais de cameras auquel elle est connecté.

La vision par ordinateur recouvre L'analyse d'images consiste à reconnaître les éléments contenus dans une image.

Les applications du domaine de la vision par ordinateur vont de la vision industrielle (par exemple dans l'industrie de fabrication de bouteilles), à la recherche dans le domaine de l'intelligence artificielle et des ordinateurs ou robots capables de « comprendre » le monde qui les entoure.

La vision par ordinateur recouvre plusieurs applications dans différent domaines. Quelques domaines d'application de la vision par ordinateur : Procédés de contrôle, Navigation, Détection d'événements, Organisation d'informations, Modélisation d'objets ou d'environnements, Interaction, Inspection automatique.

Un problème très Important qui apparait dans de nombreuses applications (industrielles, médicales, etc.) est celui de la reconnaissance d'objets partiellement occultés. Pour mener à bien une telle reconnaissance, les techniques fondées sur des caractéristiques globales des objets risquent d'être inefficaces. Plusieurs tentatives ont été menées pour se focaliser sur des caractéristiques locales, telles que des coins, des trous, des textures locales, des segments, des contours, moins perturbées par une occlusion éventuelle. De nombreuses techniques ont été développées pour la reconnaissance d'objets en présence d'occlusion telle que (Papadourakis et al., 2011) (Liu et al., 2012) (Shotton et al., 2013) (Torralba et al., 2012) (Wu et al., 2012) (Zhu et al., 2010).

Nous présentons dans ce mémoire, une méthode de reconnaissance d'objets par leurs silhouettes en présence d'occlusion. La méthode élaborée combine deux méthodes de reconnaissance d'objets par leurs silhouettes. La première méthode utilisée est la méthode du Quadtree mais traitant initialement des objets en absence d'occlusion et la deuxième méthode est la reconnaissance d'objets par leurs courbures. L'intérêt de cette combinaison est de palier à certains inconvénients de la méthode du Quadtree pour la reconnaissance d'objets comme par exemple sa non robustesse à la rotation (Buffa, 2001). En plus de la combinaison des

deux méthodes, le défi principal de notre approche est la possibilité de reconnaître les objets même en présence d'occlusion.

Plusieurs techniques ont été utilisées pour pallier aux problèmes de rotation comme les boîtes englobantes, les boîtes minimales et la représentation par les moments invariants (Maître, 2004); ces techniques amènent à des calculs assez importants sur l'image avant même de commencer la reconnaissance. Nous avons décidé d'utiliser une technique qui nous permet de faire une rotation de l'objet de l'image requête de telle sorte qu'il y ait une correspondance avec un objet d'une image de la base. Ensuite nous utilisons la méthode du Quadtree pour une meilleure correspondance entre l'objet de l'image requête et l'objet de l'image de la base. Nous définissons dans ce mémoire ce que l'on appellera seuil d'occlusion.

Nous présentons dans le chapitre 1, la plupart des techniques utilisées dans le domaine de la vision artificielle pour l'indexation et la reconnaissance des objets dans une image.

Le chapitre 2 sera consacré à l'étude des techniques pour l'indexation et la reconnaissance d'objets et cela en présence d'occlusion.

Le chapitre 3 comprend la conception de notre projet de magister. Nous présentons dans ce chapitre la méthode proposée pour l'indexation et la reconnaissance de silhouette d'objet en présence d'occlusion et cela à l'aide de deux méthodes, la première méthode est la méthode d'indexation et de reconnaissance qui se base sur le Quadtree et la deuxième méthode est la méthode qui se base sur les informations de courbures pour la reconnaissance et l'indexation des objets dans une image.(Oliviera, 2008)

Dans le chapitre 4, nous présentons l'Expérimentation de notre méthode, nous avons utilisé pour les tests de notre logiciel des images de synthèse ainsi que des images réels de la base de Leibe B, Schiele B.

Chapitre 1

***Etat de l'art sur les méthodes
d'indexation et la reconnaissance
d'objets***

Chapitre 1 : Etat de l'art sur les méthodes d'indexation et la reconnaissance d'objets.

1. Introduction :

L'Intelligence Artificielle (IA) est la science dont le but est de faire par une machine des tâches que l'homme accomplit en utilisant son intelligence. L'interaction automatisée d'un ordinateur avec son environnement a toujours été l'un des buts fondamentaux de l'intelligence artificielle. La vision artificielle est une branche de l'intelligence artificielle dont le but est de permettre à une machine de comprendre ce qu'elle « voit » lorsqu'elle est connectée à une ou plusieurs caméras. Les applications qui découlent de la vision artificielle sont très vastes tel que : la vidéo surveillance, les commentaires automatiques de sport, la recherche d'images... etc. Toutes ces applications convergent vers un axe de recherche qui est la reconnaissance d'objets dans une image. La reconnaissance d'objets est l'un des domaines d'étude les plus actifs de la vision par ordinateur.

La reconnaissance d'objets est un problème épineux se plaçant au niveau supérieur dans la hiérarchie des tâches de vision et constitue la partie computationnelle la plus difficile. Pour surmonter cette difficulté, un système de vision doit être capable de combiner ses capacités de représentation interne dans le but de réussir des décisions.

Les schémas computationnels actuels de reconnaissance se limitent généralement à l'utilisation d'un petit nombre de modèles spécifiques à un domaine d'intérêt donné. Ils n'abordent qu'un nombre restreint de types d'objets (souvent un seul type d'objets est pris en compte, par exemple: la reconnaissance de visages humains). Cette approche est utile pour réaliser des actions requérant une certaine précision. Elle devient cependant réductrice lorsque des scènes complexes nécessitent l'utilisation d'un grand nombre de modèles qui est le cas des images aériennes. Beaucoup de difficultés apparaissent, notamment celles liées à la variabilité d'apparence liée : à la lumière, à l'orientation, etc. Il devient alors nécessaire de mettre en place une représentation d'objets qui émane d'une étude laborieuse des problèmes d'organisation intra objet (au sein de l'objet lui-même) et inter objets (relations dans la communauté d'objets). (Sadgal et al., 2003)

Nous nous intéressons dans ce qui suit à l'étude d'une approche de reconnaissance d'objets dans une image et qui est la reconnaissance d'objets se basant sur la silhouette.

Avant d'entamer l'étude de l'approche de reconnaissance d'objets en se basant sur la silhouette d'objet, situant celle-ci dans le cadre des approches de reconnaissance d'objets en vision artificielle.

But de la reconnaissance visuelle

La reconnaissance visuelle par ordinateur consiste à prédire à l'aide d'un algorithme quel objet se trouve dans une image. En raisonnant en termes d'entrées et sorties d'un algorithme, l'entrée de l'algorithme est une image, la sortie est la classe de l'objet (tâche de reconnaissance de catégorie d'objets) ou la référence d'un objet précis (tâche de reconnaissance d'instance d'objet). Si l'image contient plusieurs objets, la sortie de l'algorithme peut être une liste d'objets. Cette tâche diffère de la localisation d'objet (aussi appelée détection d'objet) ou de l'appariement d'objet. La première consiste à déterminer précisément la position d'un objet dans une image, la seconde consiste à trouver les correspondances point à point entre deux objets présents dans deux images.

2. Reconnaissance d'objets (l'aspect général):

L'approche classique de David Marr (Marr, 1982) a permis de proposer le premier modèle unificateur de la vision comme science et comme discipline du génie. Il énonce des principes généraux qui s'appliquent à tous les systèmes visuels et une méthodologie de travail pour concevoir et produire de systèmes artificiels. Le paradigme de David Marr identifie trois niveaux de représentation :

- **Niveau 1**: information extraite de l'image. Par exemple : arêtes, variation d'intensité, groupes de points, frontières, lignes virtuelles, organisation curviligne, *etc.*
- **Niveau 2** : orientation locale des surfaces observées,
- **Niveau 3** : description (hiérarchique) de la forme des objets.

Le point principal de ce paradigme est dans la structure hiérarchique de traitement et de représentation. Marr a posé un cadre de traitement rigoureux qui a inspiré un nombre important de travaux, autant pour la vision bidimensionnelle que tridimensionnelle.

Dans la pratique des travaux qui ont succédé, ces niveaux se sont souvent traduits par trois étapes de traitement : **segmentation, reconstruction et reconnaissance**. En abordant le

problème de la reconnaissance, Marr devait résoudre le problème de la représentation d'objets qui, malgré les modèles géométriques avancés (cylindres, cônes généralisés, ...), s'est restreint à des objets du Laboratoire. En réalité la reconnaissance doit tenir compte des images d'objets qui varient selon la quasi-totalité des paramètres de visualisation (Illumination, Orientation, Point de Vue, Occlusion, Taille, ...). Les seuls indices visuels : point, segment,... pratiqués dans l'approche classique s'avèrent insuffisants pour interpréter des images représentant des scènes naturelles, ce qui a menés les chercheurs à s'orienter vers des méthodes de vision par apparence. (Sadgal et al., 2003)

2.1.Segmentation :

En analyse d'images, on fait souvent la distinction entre traitements *bas niveau* et *haut niveau*. Les premiers travaillent sur les valeurs attachés aux pixels de l'image sans faire le lien entre la réalité qu'elles représentent. Tandis que les seconds opèrent sur des entités symboliques constituant une interprétation de la réalité extraite de l'image. La segmentation est un traitement bas niveau qui consiste à diviser l'ensemble de pixels en régions connexes, homogènes et différentes de ses voisins. Ici, on ne cherche pas à déterminer ce que les régions représentent. La qualité de l'interprétation d'une image dépend fortement de celle de la segmentation. Malgré la grande diversité de méthodes, les résultats de segmentation restent moyens et varient beaucoup en fonction de la technique choisie. Une méthode de segmentation générale et automatique est difficile à concevoir étant donnés les différents types de régions pouvant être présentes dans une image.

Généralement, la segmentation d'une image est effectuée par l'utilisation de l'une des deux grandes approches basée sur l'extraction de contours (frontières) ou la croissance des régions.

• L'approche contour :

L'approche contour consiste à identifier les transitions entre les régions. Remarquons qu'une discontinuité dans l'image n'est pas forcément liée à une variation géométrique ou physique de la surface observée : elle peut également être due à une différence d'éclairage, par exemple un effet d'ombre. L'approche contour n'aboutit pas directement à une segmentation, car les contours détectés ne sont pas toujours connexes. Il existe cependant des techniques permettant d'obtenir des contours fermés. Dans ce cas, on observe une parfaite dualité entre les contours et les régions. Dans cette approche la détection de contours consiste à balayer

l'image avec une fenêtre définissant la zone d'intérêt. A chaque position, un opérateur est appliqué sur les pixels de la fenêtre afin d'estimer s'il y a une transition significative au niveau de l'attribut choisi. A partir des pixels susceptibles d'appartenir à un contour, il faut ensuite extraire des contours fermés. Un pixel contour est souvent défini comme un minimum local du module du gradient dans la direction du gradient, ou encore comme un passage par zéro de la dérivée seconde dans cette même direction. Comme les opérateurs de dérivation sont très sensibles au bruit, des images bruitées doivent être préalablement lissées. Un grand nombre d'opérateurs gradient ont été proposés. Ils se distinguent entre eux principalement par le choix du filtre de lissage. Le lissage et la dérivation sont en pratique réunis dans un seul filtre.

• *L'approche région :*

Les méthodes de l'approche région cherchent à regrouper directement des pixels ayant une propriété commune; l'ensemble des regroupements de pixels définit à la fin une segmentation de l'image. Les plus importantes sont celles qui procèdent par croissance de régions et par division/fusion de régions : La segmentation par croissance de régions vise à regrouper les pixels adjacents de l'image dont les attributs varient de façon négligeable. Il faut choisir un prédicat ou critère d'uniformité P et un jeu de pixels initiaux, ou germes, à partir desquels la croissance de régions s'effectuera. Le critère P peut par exemple consister à exiger que la variance dans une région ne dépasse pas un certain seuil. On utilise souvent à la fois des critères radiométriques et des critères géométriques. Une initialisation très simple consiste à choisir comme germes un grand nombre de pixels aléatoirement répartis dans l'image. On peut éventuellement commencer par identifier de toutes petites zones vérifiant le prédicat P, afin de les utiliser comme points de départ.

La croissance de régions s'effectue à partir des germes par agrégation de pixels : on ajoute successivement aux régions des pixels voisins qui ont des propriétés similaires, en ce sens que les régions continuent à vérifier P après l'agrégation d'un pixel. Des régions adjacentes qui ensemble vérifient P peuvent être fusionnées. De nouveaux germes peuvent être introduits dans les zones ne pouvant pas être incluses dans les régions existantes.

Les méthodes du type division et fusion comportent, comme leur nom l'indique, deux étapes. Dans la première, l'image est divisée récursivement jusqu'à ce que toutes les régions vérifient le prédicat P. Dans la deuxième étape, des régions adjacentes sont regroupées, tant que les régions résultantes vérifient P.

- **Coopération :**

Concernant la segmentation, il est acquis depuis une dizaine d'années déjà qu'il faut faire coopérer les approches régions et les approches contours pour définir un bon critère de segmentation. Indépendamment des problèmes inverses à régulariser, de nombreux travaux ont été développés ces dernières années sur des modèles de contours. Ces modèles de contours sont développés pour la segmentation d'objet dans une image et n'étaient pas, jusqu'à récemment encore, envisagés comme des modèles de contours pour la segmentation par approche contour/région. Les travaux actuels tentent de combiner et d'unifier ces différents modèles.

2.2.Classification :

Contrairement à la segmentation d'images, ici on cherche à identifier ce que représente chacune des régions segmentées. En imagerie aérienne par exemple, on cherche à déterminer les zones urbaines, les forêts, zones d'eau, les montagnes, Il s'agit de regrouper les différents éléments (pixels) en thèmes correspondants à la vérité du terrain. Le résultat est bien entendu une image segmentée.

Dans ce domaine on procède, par l'attribution des pixels de l'image à des classes connues à priori (c'est la classification supervisée) ou à des classes inconnues (classification non supervisée). Plusieurs approches sont utilisées à ce sujet :

- **Minimisation de distance :**

Le principe des méthodes reposant sur la minimisation de distance consiste à rechercher la classe la plus proche pour chaque pixel, ou groupe de pixels si l'on travaille dans une fenêtre d'analyse centrée sur le pixel courant. La notion de proximité est liée à la distance considérée. Ces méthodes sont très simples et souvent utilisées, mais ne sont pas robustes au bruit car elles ne comportent pas de terme de régularisation.

On peut distinguer deux méthodes :

- a) Méthodes non itératives qui consistent à parcourir l'image et à déterminer la classe la plus proche parmi les K possibles.
- b) Méthodes itératives qui utilisent généralement un l'algorithme connu sous le nom de K means (ou K moyennes).

Cette méthode est supervisée dans le sens où le nombre de classes doit être donné, mais pas nécessairement leurs paramètres.

- Les approches stochastiques :(Champs de Markov, inférence bayésienne, règle de Bayes) :

Dans ce type d'approches, les modèles sont souvent discrets et de nombreux travaux reposent sur la théorie des champs de Markov et l'estimation bayésienne. Les champs de Markov permettent de ramener des propriétés globales à des contraintes locales, et le théorème de Hammersley Clifford en faisant l'équivalence entre champs de Gibbs et champs de Markov permet d'exprimer les contraintes locales sous forme de potentiel énergétique.

Les modèles markoviens en traitement d'images ont trouvé leur application dans divers problèmes de vision par ordinateur comme : segmentation, classification, restauration d'image, stéréovision, imagerie radar, modélisation de textures, Les premières approches proposées ont fait appel à des techniques de relaxation stochastique fondées sur le recuit simulé.

- Les Réseaux de neurones :

Les réseaux de neurones s'avèrent capables de traiter des problèmes complexes de reconnaissance de forme, ou de simulation de processus non linéaires et/ou dynamiques. De nombreuses applications en ingénierie et en recherche utilisent ces nouveaux outils statistiques (par exemple: en physique appliquée et en environnement). En effet, le nombre et la dimension des données ainsi que la non-linéarité des problèmes traités posent de nombreuses difficultés quant à l'utilisation des outils statistiques classiques. Les réseaux de neurones peuvent stocker des connaissances expérimentales acquises à partir d'un environnement formel épargnant ainsi l'étape difficile de formalisation. Les méthodes d'apprentissage automatiques font que plusieurs neurones participent à la représentation d'une connaissance et qu'il soit possible qu'un neurone participe au codage de plusieurs d'entre elles. Ils sont souvent utilisés pour leur capacité de classifier et ils sont classiquement employés dans des problèmes d'approximation pour simuler des fonctions de transfert non-linéaires et multidimensionnelles et pour résoudre des problèmes d'inversion.

2.3.Reconnaissance d'objets :

La reconnaissance d'objets est un domaine très actif en vision par ordinateur. L'approche générale consiste à considérer que l'on dispose d'une banque de données où sont stockés des

modèles d'objets et une vue du monde réel. On doit répondre à une question essentielle : L'objet extrait (par les moyens précédents) existe-t-il dans la banque de données. Le problème dépend de la modélisation d'objets adoptée. Dans la majorité des cas cette modélisation est purement géométrique : un ensemble de caractéristiques appelées primitives. L'opération de reconnaissance consiste en l'appariement de ces caractéristiques et celles décelées dans l'image.

Pour effectuer l'opération de reconnaissance, plusieurs méthodes ont vu le jour. Le but de ces méthodes est de déterminer si une image contient des objets de référence, et éventuellement la position de ces objets dans l'image. Pour ce faire, les objets de référence sont représentés par un modèle 3D (méthodes géométriques) ou par des bases de données d'images de référence ce qui est appelé en littérature par reconnaissance ou vision par apparence (méthodes globales et méthodes locales). (Nowak ,2008)

2.3.1. Méthodes géométriques:

Les premières méthodes de reconnaissance d'instances d'objet sont des méthodes géométriques (Huttenlocher et al., 1990) (Grimson et al., 1987) (Goad, 1983) (Fischler et al., 1981) (Lamdan et al., 1988) (Lowe, 1987) (Stockman, 1987) (Thompson et al., 1987) (Wolfson et al., 1997). Elles consistent à représenter les objets de référence par leurs structures. On distingue les méthodes à base d'alignement modèle-objet et les méthodes à base de tables de hachage. Les méthodes d'alignement géométrique disposent d'une modèle 3D de l'objet recherché. Les primitives (par exemple des droites) composant ce modèle 3D sont alignées sur les primitives détectées dans l'image, et la qualité de l'alignement détermine si l'objet est présent ou non (Huttenlocher et al., 1990) (Lowe, 1987) (Nowak ,2008).

On peut citer les arbres d'interprétation (Grimson et al., 1987) (Goad, 1983), qui sont des arbres de tous les alignements modèle-image possibles, élagués par l'application d'heuristiques. Par ailleurs, Fischler (Fischler et al., 1981) appliquent la méthode RANSAC pour obtenir un consensus d'alignement à partir d'un sous ensemble des primitives de l'image, et décident de la validité de l'alignement avec les primitives de l'ensemble complémentaire. Lamdan (Lamdan et al., 1988) et Wolfson (Wolfson et al., 1997) utilisent des tables de hachage pour la reconnaissance. Les primitives géométriques des différents modèles d'objets sont stockées dans une table de hachage, et lors de la reconnaissance, les primitives observées dans l'image sont recherchées dans la table de hachage, et des votes sont

accumulés pour les différents modèles. Un objet est reconnu si le modèle correspondant obtient une quantité minimale de votes (Nowak ,2008).

Les méthodes géométriques possèdent une grande robustesse aux changements d'illumination et de texture des objets. De plus, l'utilisation d'un modèle 3D permet de traiter les occultations et le fond chargé de manière très élégante.

2.3.2. Vision par apparence :

La vision par apparence ; comme son nom l'indique s'appuie sur toutes les manifestations visuelles possible des entités observées, c'est-à-dire en théorie, sur toutes les images de ces entités prises de tous les points de vue et éclairages possibles. En vision par apparence, les modèles sont construits dans le repère image (ce sont donc des modèles 2D).

Une méthode de reconnaissance par apparence est toute méthode de reconnaissance d'image qui cherche à modéliser des objets (2D ou 3D) directement par leur image perçue et non pas par un modèle construit par une conception abstraite particulière (Lamiroy, 1998).

Le facteur commun à toutes les méthodes qui entrent dans la catégorie « par l'apparence » est qu'elles modélisent les objets 3D par un ensemble d'images, prises dans des conditions particulières. Ces conditions sont liées aux contraintes de reconnaissances qui sont imposées au problème que le système est censé résoudre, et couvrent grossièrement toutes les apparences que l'objet peut prendre lorsqu'il est observé. Les conditions qui sont généralement les plus considérées sont liées aux contraintes d'éclairage et au déplacement de l'objet dans l'image.

Dans la majorité des applications qui implémentent une reconnaissance d'objets par une vision par apparence, cette dernière est couplée à une indexation. Dans ces applications, une structure globale des algorithmes se présente comme suit :

Phase de modélisation :

Paramètres d'entrée : IMAGE_MODELE¹,..., IMAGE_MODELEⁿ

Paramètres de sortie : STRUCTURE_INDEXATION /*Contenant les n modèles */

Début

Pouri allant de 1 à nfaire

Début

 Caractériser l'IMAGE_MODELEⁱ ;

Utiliser cette caractérisation pour indexer le modèle
dans STRUCTURE_INDEXATION ;
fin
retourner STRUCTURE_INDEXATION ;
fin

Phase de reconnaissance :

Paramètres d'entrée : IMAGE_INCONNUE, STRUCTURE_INDEXATION

Paramètres de sortie : IMAGE_MODELE

Début

Caractériser IMAGE_INCONNUE ;
Utiliser cette caractérisation pour accéder au(x) modèle(s)
Similairement indexé(s) dans STRUCTURE_INDEXATION ;

Eventuellement choisir le modèle le plus probant parmi ceux trouvés ;

Retourner ce modèle ;

fin

Dans la littérature, on distingue deux grandes classes dans les méthodes par apparence. L'une regroupe les approches qui considèrent une image comme une entité indivisible, donnant généralement lieu à un index unique par image, l'autre contient les méthodes qui modélisent les objets par un ensemble de caractéristiques hétérogènes désignant des parties plus ou moins grandes dans l'image. On parle de *méthodes globales* pour les unes et de *méthodes locales* pour les autres.

2.3.2.1.Méthodes Globales :

Les méthodes de reconnaissance globale (Turk et al., 1991) (Schiele et al., 2000) (Pontil et al., 1998) (Niblack et al., 1993) (Murase et al., 1995) calculent une signature de l'image prise dans sa globalité. Cette signature peut être, par exemple, la distribution des couleurs dans l'image. Une telle signature est calculée pour un ensemble d'images représentatives d'une instance d'objet. Lors de la reconnaissance, la signature de l'image inconnue est calculée,

puis est comparée aux images de références. L'objet prédit est l'objet de référence le plus similaire à la signature de l'image inconnue. La signature de l'image peut être constituée de l'ensemble des pixels mis les uns à la suite des autres dans un vecteur colonne. La comparaison avec les images de référence peut se faire avec une analyse en composante principale, ce qui est le cas des *eigenfaces* de Turk et Pentland (Turk et al., 1991), ou bien à l'aide d'une machine à vecteur support comme le font Pontil et al (Pontil et al., 1998). Les signatures utilisées par Niblack et al (Niblack et al., 1993) et Schiele et al (Schiele et al., 2000) sont respectivement des histogrammes de couleurs et de textures.

Les méthodes globales sont très simples mais souffrent d'un besoin d'une grande quantité de données. (Nowak ,2008)

Les inconvénients de la méthode globale sont de taille :

-Le fait que l'approche soit globale la rend particulièrement sensible à des *occultations* partielles des objets observés, de changement du fond sur lequel l'objet se trouve, etc.

-Les *changements d'éclairage*, ou des transformations géométriques lors de la prise de vue influent directement sur la représentation finale de l'image. (Lamiroy, 1998)

2.3.2.2.Méthodes Locales :

Les méthodes locales (Schmid et al., 1997) (Lowe, 1999) pallient les principaux défauts des méthodes globales. Elles ne considèrent plus les images comme un tout monolithique, mais comme une collection de régions locales, qui sont le plus souvent des parties d'images carrées ou rectangulaires, pouvant contenir une centaine de pixels ou bien plus. Ces régions locales s'affranchissent des difficultés rencontrées sur des images entières, car elles peuvent être ramenées à une apparence standard. Les modifications d'illumination sont localement uniformes, et la luminosité et le contraste peuvent donc être normalisés. Ces régions locales correspondent généralement à des surfaces planes, donc leur orientation et leur échelle peuvent être ramenées à des valeurs standard (Schmid et al., 1997) . Quand ces régions sont découpées à l'intérieur des objets, elles ne subissent pas d'influence du fond chargé, et en sont donc indépendantes. (Nowak ,2008)

Enfin, quand ces régions sont de petites tailles, elles sont le plus souvent présentes intégralement ou bien totalement occultées, mais elles sont rarement partiellement occultées, et donc leur apparence ne varie pas à cause des occultations. Le principe de ces méthodes est

de représenter les images de référence par des régions locales et de stocker les descripteurs de ces régions locales (par exemple leurs niveaux de gris) dans une base de données. Lors de la reconnaissance, les descripteurs des régions locales sélectionnées dans l'image sont recherchés dans la base de données, ce qui permet de voter pour les images de référence qui contiennent des descripteurs similaires. (Nowak ,2008)

Schmid et al (Schmid et al., 1997) utilisent des détecteurs de points d'intérêt pour obtenir des régions locales d'une orientation et échelle normalisées. Des contraintes géométriques sur les positions relatives des régions locales permettent de réduire le taux de fausse alarme. Lowe (Lowe, 1999) propose un système temps réel et robuste pour rechercher les objets similaires dans une base de données, et introduit le fameux descripteur SIFT, qui encode la distribution de l'orientation du gradient dans une région locale. Dans ce cas aussi, la présence des objets est vérifiée à l'aide de contraintes géométriques sur les positions relatives des régions locales. (Nowak ,2008)

En raison de leur rapidité, de leur bonne gestion des occultations, du fond chargé et des modifications de point de vue, les méthodes locales ont aujourd'hui la préférence de la communauté pour la reconnaissance visuelle d'instances d'objets.

Il est intéressant de noter l'approche de HUTTENLOCHER et al. Basée sur la distance de HAUSDORFF (Huttenlocher et al., 1992) (Huttenlocher et al., 1993)(Rucklidge, 1995). Elle est locale dans le sens qu'elle modélise les images par des contours représentatifs qu'elle réussit à mettre en correspondance. L'appariement des contours ne se fait pas pixel par pixel, mais on identifie globalement un contour avec un autre. Cette méthode présente de très bon résultats de reconnaissance, mais n'est malheureusement pas adaptable à une mise en œuvre pour indexation et est très consommatrice en temps de calcul. (Lamiroy, 1998)

Autres modélisations :

Il est difficile de classer les méthodes de modélisation décrites dans cette section parmi les deux méthodes précédentes. Autant les deux classes de méthodes précédentes étaient antagonistes dans leurs approches, autant celles-ci se font hybrides, mais néanmoins distinctes. Elles s'appuient généralement sur des mesures locales, tels que des points ou des structures de contours, mais les contraintes de reconnaissance n'ont pas de justification géométrique et se basent généralement sur des théories probabilistes ou statistiques. Il n'y a donc pas lieu de raisonner en *global* ou *local*.

2.3.2.3. Modélisation par histogrammes :

Utilisée dans les premiers travaux cherchant à caractériser des images par leur distribution de couleurs par SWAIN et BALLARD (Swain et al., 1991), la modélisation par histogrammes a été implémentée avec succès pour l'accès à de grandes bases d'images par leur contenu couleur. Elle a été étendue par SCHIELE (Schiele, 1997) pour d'autres descripteurs locaux. L'idée générale, et que l'on retrouve de façon simplifiée dans les travaux originaux, est de calculer, en tout point de l'image, un vecteur de descripteurs invariants, de manière similaire aux approches locales, à cette différence près qu'aucune segmentation préalable n'est effectuée. Une image modèle est donc représentée par son histogramme multidimensionnel de ces descripteurs. (Lamiroy, 1998)

Pour identifier une image à l'un des modèles connus, on calcule les mêmes types de descripteurs dans un ensemble quelconque de points. On obtient ainsi un « sous-histogramme » que l'on peut comparer aux histogrammes stockés afin de déterminer le modèle le plus semblable.

Cette méthode présente un grand nombre des avantages des approches locales. Du fait qu'elle ne nécessite pas de segmentation, et est seulement une prise en compte d'un nombre limité de mesures dans l'image requête, elle est très robuste. Elle ne procure pas, cependant, les mises en correspondance entre les différentes configurations comme dans le cas d'une modélisation locale. Elle se généralise aussi très bien à la catégorisation d'images, but difficile à atteindre avec des méthodes globales.

2.3.2.4. Modélisations statistiques et probabilistes :

Densités de probabilité Dans leurs travaux, HORNEGGER et NIEMANN (Hornegger et al., 1995) se basent sur une représentation par densité de probabilité de caractéristiques d'un objet pour le modéliser. La reconnaissance passe alors par une estimation de paramètres par maximum de vraisemblance, à partir d'un modèle probabiliste *à priori*. Les caractéristiques utilisées pour la reconnaissance sont considérées comme des variables aléatoires.

L'algorithme de modélisation et de reconnaissance se découpe alors en trois étapes principales. On suppose que k objets différents sont représentés par des fonctions de densité de probabilité paramétrées. Les caractéristiques et les paramètres de ces fonctions peuvent varier selon les méthodes et les objets. Dans la phase « d'apprentissage » les paramètres $B_R, 1 \leq R$

$\leq k$ de la fonction de densité de probabilité *a priori* doivent être estimés à partir d'un ensemble de vues $\{^pO\}$ de départ pour chaque modèle R.

Lorsqu'un objet doit alors être reconnu par rapport aux k modèles, on passe par deux phases. On estime d'abord les paramètres P qui représentent les degrés de liberté (par exemple ceux régissant les transformations géométriques entre modèle 3D et sa projection 2D) entre chaque modèle et l'objet à reconnaître. L'objet étant un ensemble d'observations de variables aléatoires C , on estime la probabilité $p(C|B_k, P)$, que l'on maximise ensuite.

La reconnaissance finale, c'est-à-dire la détermination du modèle k correspondant aux observations C , correspond à une application de la règle de BAYES.

Cette modélisation est certes élégante et bien développée mathématiquement, mais reste très dépendante des choix des modèles de densités de probabilité utilisés (ici, un mélange de distributions Gaussiennes multi-variables paramétriques) et gourmande en ressources, puisqu'elle est basée sur un processus d'optimisation, lié à l'estimation du maximum de vraisemblance des paramètres.

Chaînes de MARKOV et décision statistique

Les travaux de HERBIN (Herbin, 1997) pourraient être classés sous différentes dénominations. Lui-même définit son approche comme la reconnaissance en actes. Il rejette notamment la réduction de la reconnaissance à un simple appariement empirique, et ne la conçoit que dans une activité cognitive guidée par le contexte. Il développe alors une modélisation qui s'appuie sur les graphes d'aspects, enrichies d'une structure probabiliste markovienne. La théorie asymptotique des tests d'hypothèses lui permet ensuite d'analyser le problème de discrimination des modèles markoviens.

Se basant sur la théorie des graphes d'aspects, l'auteur considère que la véritable modélisation d'un objet réside dans les transitions entre nœuds du graphe de ses aspects. Un modèle est donc un ensemble d'états observables, formant les nœuds d'un graphe de transitions auxquelles sont attachées des probabilités (structure markovienne). L'acte de reconnaissance consiste ensuite, à partir de l'observation d'un état, à formuler la probabilité de son appartenance à chacun des modèles plausibles. Ensuite, le système décide de modifier ses paramètres de prise de vue afin de provoquer l'observation d'un autre aspect (et donc nœud dans le graphe d'aspects). On converge donc d'observation en observation vers une

certitude cumulée d'appartenance ou la non-appartenance de l'objet à l'un des modèles connus.

Ce type de modélisation a l'avantage d'être très rigoureux et d'intégrer dès sa conception le bruit et l'incertitude par son approche stochastique. Il aborde également des questions soulevées par des questions liées à la reconnaissance « biologique ». Dans le contexte d'un processus de reconnaissance automatique autonome, il est probablement suffisamment bien développé et générique pour donner lieu à des applications très intéressantes.

2.3.2.5. Caractérisation par graphes :

Dans sa thèse, SOSSA (Sossa, 1992) propose de modéliser des objets 3D par un nombre limité de vues qui les définissent. Sa méthode est basée sur une approche de caractérisation provenant de la théorie des graphes. Elle utilise comme entrée des images segmentées en contours, approchés par des lignes polygonales. Ces lignes polygonales forment un graphe qui est caractérisé par son polynôme de deuxième invariants. La mise en équations de la topologie et de l'apparence du graphe, combinée à une indexation des caractéristiques de sous graphes, permet à l'auteur de contourner le problème de la NP-complétude de la recherche de sous-graphes isomorphes.

L'approche n'est clairement pas globale, puisque seule la partie segmentée du modèle est utilisée. Elle devient robuste aux occultations grâce à l'introduction des sous-graphes, qui lui confèrent une certaine « localité ». (Lamiroy, 1998)

3. Méthodes se basant sur les silhouettes pour la reconnaissance et l'indexation d'objets :

Les méthodes dans cette catégorie utilisent la connaissance de la transformation entre deux cartes de profondeur (par exemple, mouvement sur un plan). Elles décomposent une surface en un ensemble de courbes (ou contours). Chaque contour doit être invariant à la transformation entre les deux cartes. Le problème d'appariement 3D devient un problème d'appariement 2D (la mise en correspondance de contours). (Zhang, 1993)

-La notion de silhouette

La silhouette (forme) est un trait visuel important et très utilisé pour la description basée sur le contenu de l'image. Cependant la description du contenu d'une forme est une tâche difficile en raison de la difficulté de définir ses traits perceptuels et de mesurer la

ressemblance entre formes. Le problème devient plus complexe lorsque la forme est altérée par un trait, une distorsion arbitraire ou une occlusion.

Les silhouettes sont importantes dans la reconnaissance des objets, ce sont elles qui détachent les objets du fond de l'image.

La silhouette est un dispositif visuel important et elle est l'un des dispositifs primitifs pour la description de l'image.

-La notion de contour (frontière)

Le contour est un ensemble de points qui donne la forme générale de la silhouette, on peut dire aussi que ce sont les points qui séparent la silhouette de son environnement (frontière).

Il s'agit de l'ensemble des points se trouvant aux extrémités d'une surface. Au sens mathématique, la frontière est l'ensemble des points de la surface qui correspondent à des discontinuités d'ordre 0.

3.1.L'approche de Huttenlocher :

L'approche de Huttenlocher (Huttenlocher et al., 1993) est l'une des méthodes se basant sur les contours de formes pour la reconnaissance d'objets. Cette approche est basée sur la distance de Hausdorff. Déterminer une mesure pour laquelle une forme est différente d'une autre, est un problème central dans la branche de reconnaissance d'objets en vision artificielle. Les travaux qui ont été menés par Huttenlocher visent à rechercher des fonctions pour déterminer le degré de différence entre deux formes. Le but de cette recherche est de développer des méthodes de comparaisons entre formes. L'algorithme présenté dans permet d'évaluer efficacement la distance de Hausdorff entre toutes les positions relative d'un modèle et une image.

Avant d'entamer l'approche de Huttenlocher, définissant l'outil de base qui a été utilisé pour l'élaboration de la méthode est qui est la distance de Hausdorff.

3.1.1. Distance de Hausdorff :

En topologie, la distance de Hausdorff mesure l'éloignement de deux sous-ensembles d'un espace métrique. Elle porte le nom du mathématicien allemand Felix Hausdorff.

Définition :

Soit un espace métrique (E, d) . Soient A et B deux sous-ensembles compacts non vides de E . On définit tout d'abord, pour tout sous-ensemble X de E , le r -voisinage ouvert de X comme étant l'ensemble : $V(r, X) = \{x \in E \mid d(x, X) < r\}$

C'est-à-dire l'union des boules ouvertes centrées sur un élément de X et de rayon r .

La distance de Hausdorff $D_H(A, B)$ entre A et B est définie comme étant le plus petit nombre réel r tel que le r -voisinage de A contienne B et le r -voisinage de B contienne A . En d'autres termes :

$$D_H(A, B) = \inf\{r > 0 \mid A \subset V(r, B), B \subset V(r, A)\}$$

Propriétés

La distance de Hausdorff sur E définit une distance sur l'ensemble $K(E)$ des compacts non-vides de E . $K(E)$ est alors un espace métrique et sa topologie dépend de celle de E .

Si E est un espace complet, alors $K(E)$ est complet. Si E est un espace compact, alors $K(E)$ est compact.

Par conséquent, toute suite $(A_n)_{n \in \mathbb{N}}$ d'ensembles de $K(E)$ décroissante au sens de l'inclusion admet une limite au sens de la distance de Hausdorff, à savoir $\bigcap_{n \in \mathbb{N}} A_n$.

Généralisation

La distance de Hausdorff peut être définie de la même manière pour les sous-espaces fermés non-compacts de E , mais la « distance » ainsi créée peut prendre des valeurs infinies et la topologie induite dépend de la métrique employée sur E .

Il est également possible de définir la distance de Hausdorff entre deux sous-ensembles non fermés de E comme la distance de Hausdorff entre leur adhérence. On munit ainsi l'ensemble des sous-ensembles de E d'une pseudo-métrique (puisque deux sous-ensembles distincts mais partageant la même adhérence auront une distance de Hausdorff nulle).

Mesure de similarité La **distance de Hausdorff** (D_H) est régulièrement utilisée en analyse d'image. D'après Rucklidge (Rucklidge, 1995), elle est considérée comme une mesure de similarité naturelle entre les formes.

Définition

La définition ci-dessous est habituellement utilisée en reconnaissance de forme.

Soient S et T deux ensembles de points. La distance de Hausdorff est définie par

$$D_H(S,T) = \max \{f_d(S,T), f_d(T,S)\},$$

où f_d est appelée la **distance de Hausdorff relative** (ou semi-distance de Hausdorff). Elle est définie par

$$f_d(S,T) = \max_{p \in S} d(p,T).$$

Habituellement, la distance d utilisée est la distance euclidienne.

Propriété

La distance de Hausdorff $D_H(S,T)$ est nulle si et seulement si $S = T$ et elle augmente lorsque des différences de plus en plus importantes apparaissent entre S et T .

Le calcul de la distance de Hausdorff peut se faire en utilisant une carte de distances.

Défaut

La distance de Hausdorff ne considère les objets géométriques seulement comme des ensembles de points et non selon leur propre nature. En conséquence, les mesures obtenues par cette distance peuvent être incohérentes par rapport à ce que nous pouvons observer. Par exemple, si nous prenons le cas de deux courbes se croisant en un grand nombre de points, la distance de Hausdorff entre ces deux courbes sera faible alors que visuellement elles paraissent très différentes.

3.1.2. Méthodes de Huttenlocher :

Un problème central dans les modèles de reconnaissance et en vision artificielle est de déterminer l'extension à laquelle une forme diffère d'une autre forme. Les modèles de reconnaissance tel que la corrélation, les modèles de correspondance et méthodes de vision basées modèles peuvent tous être vus comme des techniques pour déterminer la différence entre des formes. Les travaux de Huttenlocher ont permis de regrouper les fonctions qui permettent de déterminer le degré auquel une forme diffère d'une autre forme. Le but de ces travaux est de développer des méthodes de comparaisons de formes qui sont efficace et qui

donne des résultats corrects. Dans les travaux de Huttenlocher (Huttenlocher et al., 1993), des algorithmes ont été conçus pour utiliser efficacement la distance de Hausdorff pour la comparaison de positions relative d'un modèle et une image. Les travaux de Huttenlocher visent à traiter dans un premier temps le cas où le modèle peut être translaté à l'image, et dans un deuxième temps d'étendre le modèle à des cas plus général. Il existe plusieurs algorithmes théoriques qui utilisent efficacement la distance de Hausdorff comme une fonction de translation (Huttenlocher et al., 1990) (Huttenlocher et al., 1991). Huttenlocher a construit des algorithmes d'approximations qui sont efficace autant que dans la théorie que dans la pratique. Ces méthodes opèrent sur des trames binaires, les rendant particulièrement bien adaptées au traitement d'image et aux applications de vision, où les données sont généralement en forme de trame.

Les trois avantages de l'approche de Huttenlocher sont :

- Une insensibilité relative à de petites perturbations dans l'image.
- Une simplicité et une rapidité de traitement.
- Et l'avantage de permettre la comparaison de portions de formes à d'autres formes.

Trois différentes méthodes de l'utilisation de la distance de Hausdorff sont discutées dans (Huttenlocher et al., 1993) comme fonction pour la comparaison d'un modèle à une image. La première méthode est similaires de différentes manières à une convolution et une corrélation binaire excepté que la distance de Hausdorff est un opérateur non-linéaire. La deuxième méthode vise à étendre la définition de la fonction de distance pour permettre la comparaison de portion de modèle à des portions de formes dans une image. La troisième méthode est amélioré par rapport au deux premières, en utilisant certaines propriétés de la distance de Hausdorff pour exclure des positions possible des modèle et de l'image sans avoir à les considérés explicitement. Dans (Huttenlocher et al., 1993), plusieurs exemples utilisant des images réels ont été présentés. Ces exemples illustres l'application de la méthode de Huttenlocher à des images où les portions d'objets à identifier sont occultés de la vue.

3.1.3. Comparaison de portions de forme :

Dans plusieurs applications de vision artificielle et de reconnaissance, il est important d'être capable d'identifier des instances d'un modèle qui n'est que partiellement visible (soit en raison d'une occlusion ou à l'échec du dispositif de détection pour détecter la totalité de l'objet). C'est pour ces raisons que Huttenlocher a voulu étendre la définition de la distance

de Hausdorff pour permettre la comparaison de portions de deux formes. Cette extension sera utile d'une part pour les scènes contenant de multiples objets, et pour des objets qui sont partiellement occultés de la vue.

-Distance partiels basée sur un classement :

La distance de Hausdorff peut naturellement être étendue au problème de recherche de la meilleure distance partielle entre un modèle B et d'une image A. Huttenlocher a considéré dans un premier temps que la distance de Hausdorff de B a A, $h(B, A)$. Le calcul de $h(A, B)$ détermine la distance entre un point du modèle B qui est le plus loin de tous les points de l'image A. C'est ainsi que chaque point du modèle B est classé par une distance au plus proche point correspondant dans l'image A, et le point de B qui a été classé le plus loin de tous les points de l'image A.

Ceci amène à une définition d'une 'distance' pour un point K appartenant aux points modèles $q(1 \leq K \leq q)$ donnée en prenant le $K^{\text{ème}}$ point classé de B :

$$h_K(B, A) = K_{b \in B}^{\text{th}} \min_{a \in A} \|a - b\| \quad (**)$$

Où $K_{b \in B}^{\text{th}}$ correspond à la $K^{\text{ème}}$ valeur classée dans un ensemble de distances (qui correspondent à chaque élément de B). Pour chaque élément de B la distance à chaque point le plus proche de A est calculée, et ensuite les points de l'ensemble B sont classés par leurs valeurs respectives de cette distance. La $K^{\text{ème}}$ valeur classée, d, nous informe que K des points modèles B ont chacun une distance d d'un point de l'image (et quand $K=q$ tous les points ont été considérés, et la valeur est la distance de Hausdorff $h(B, A)$). Cette définition de la distance a une propriété avantageuse et qui est qu'elle sélectionne automatiquement les K points de B qui correspondent le mieux, parce qu'elle identifie un sous ensemble du modèle de taille K qui minimise la distance de Hausdorff.

En général, dans le but de mettre en place une distance partielle $h_K(B, A)$, Huttenlocher a spécifié quelques fraction $0 \leq f_1 \leq 1$ des points de B qui vont être considérés. Chaque point q de B est classé par une distance par rapport au point le plus proche de A. La $K^{\text{ème}}$ valeur classée, donnée par l'équation (**), donne une distance partielle, où $K = \lceil f_1 q \rceil$. Cette $K^{\text{ème}}$ valeur classée peut être évaluée en une complexité de $O(q)$.

La distance partielle mesure la différence entre une *portion* d'un modèle et une image : les K points de l'ensemble du modèle qui sont les plus proches des points de l'image. Un

point avantageux de cette méthode est qu'il n'est pas nécessaire de pré-spécifier quelle est la partie du modèle à comparer avec l'image. Ceci est dû à l'estimation de la distance de Hausdorff qui détermine la distance de chaque point du modèle du plus proche point de l'image, et sélectionne automatiquement les K points du modèle qui sont les plus proches aux points de l'image.

3.2. L'approche de Radack et Badler :

Radack et Badler (Radack et al., 1989) utilisent des contours de distance. Soit S une surface, et p un point dans l'espace. Le contour de distance sur S de rayon r autour de p est défini comme l'ensemble de points de S avec une distance r de p. Il est noté par $S(r)p$. $S(r)p$ est en fait l'intersection de S avec la sphère centrée sur p avec rayon r. En variant r, nous obtenons un ensemble de contours de distance sur S autour de p, qui s'appelle le profil de distance sur S autour de p.

Etant donné S1, la première image 3D, et S2, la deuxième image 3D. Supposons w1 et w2 sont appariés l'un à l'autre, où w1 et w2 sont deux points exprimés dans la première et la deuxième cartes, respectivement. Pour déterminer si S1 apparie S2 ou non, nous examinons si chaque contour $S1(r)w1$ apparie $S2(r)w2$ ou non. Nous devons donc être capable de choisir deux points w1 et w2 qui correspondent. D'après la géométrie différentielle, les courbures Gaussienne et moyenne sont des propriétés locales de la surface, et sont invariantes par une transformation rigide. Des points critiques où la courbure Gaussienne ou moyenne est au maximum ou minimum sont utilisés pour construire des profils de distance sur S1 et S2. Un contour de distance n'est pas nécessairement fermé, et souvent ne l'est pas. Toutefois, dans leurs implémentations ils supposent que les contours sont des courbes fermées simples. Un contour C est ensuite paramétrées par s, où s est l'abscisse curviligne normalisée pour varier entre 0 et 1. Une représentation de C qui est invariante vis-à-vis à toute rotation autour du point de centre s'appelle la carte de distance de C, qui est définie comme

$$\hat{C}(s) = d(C(s), \bar{C}), \quad 0 \leq s \leq 1 ;$$

où \bar{C} est le centroïde de C, i.e., $\bar{C} = \int C(s)ds$, et d() est la distance Euclidienne. Les cartes de distance peuvent être créées en tableaux de distances d'une taille fixe. Deux profils de distance P et Q sont considérés appariés si :

-
1. Les courbures sur les points de centre sont égales avec une tolérance préfixe.
 2. Chaque contour sur P apparie un contour sur Q, jusqu'à un niveau donné.

Deux courbes C1 et C2 sont considérées appariées s'il existe un décalage w qui minimise la fonction suivante :

$$e(w) = \int_0^1 [\hat{C}_1(s) - \hat{C}_2(s - w)]^2 ds,$$

$$= \int_0^1 \hat{C}_1^2(s) ds - 2 \int_0^1 [\hat{C}_1(s) - \hat{C}_2(s - w)] ds + \int_0^1 \hat{C}_2^2(s - w) ds, \text{ (Eq.*)}$$

et le minimum de $e(w)$ est inférieur à un seuil préfixe. Puisque le premier et le deuxième terme dans Eq. * sont constants, la minimisation de $e(w)$ devient un problème de corrélation standard. Les points critiques appariés (au moins 3 non-colinéaires) sont utilisés pour estimer la transformation rigide.

L'algorithme a été testé seulement avec des données synthétiques simples et bien formées, malgré l'idée de base est applicable aux cas plus généraux. C'est probablement parce que les auteurs n'ont pas de données réelles disponibles, ou probablement parce que l'implémentation actuelle a des limitations suivantes:

1. La représentation d'une surface par des contours dépend fortement de la précision de la location des points critiques. Mais la location est sensible au bruit.
2. Des contours de distance en général ne sont pas des courbes simples fermées dans des données réelles.
3. Le calcul de profils de distances n'est pas évident avec des scènes compliquées.

Il faut noter que c'est possible en général de calculer une transformation rigide à partir d'une paire de points critiques appariés ainsi que leurs profils de distance, qui aurait dû être plus stable qu'à partir de quelques points critiques.

3.3. L'approche de Rodriguez et Aggarwal :

Rodríguez et Aggarwal (Rodriguez et al., 1989) proposent un algorithme pour mettre en correspondance une séquence d'images aériennes avec un MNT à bord. Un système stéréoscopique basé sur des contours obtenus par le passage par zéro (zerocrossings) d'un opérateur LoG (Laplacian-of-Gaussian) est utilisé pour obtenir une carte de points 3D

clairsemés. Ces points sont ensuite utilisés pour interpoler une surface continue, qui est à la fin échantillonnée avec la même résolution que le MNT à bord. L'algorithme qui suit permet d'apparier la carte d'élévation ainsi générée avec le MNT. Notons que les deux cartes à apparier ont la même représentation, et nous les appelons chacune un MNT.

Un MNT est d'abord transformé en une carte de falaise (cliff en anglais). Une falaise est un endroit où l'élévation change brusquement. Pour détecter les falaises, le MNT est convolué avec un filtre LoG (Laplacian-of-Gaussian). Un filtre LoG continu a la forme:

$$\nabla^2 G(x, y) = \frac{1}{2\pi\sigma^6} (r^2 - 2\sigma^2) \exp(-r^2/2\sigma^2),$$

Où $r^2 = x^2 + y^2$. Dans leur implémentation, $\sigma = 4$ est choisi pour fournir suffisamment de lissage afin de réduire l'effet du bruit et la densité de contours de falaise. Finalement, les passages par zéro sont détectés, qui génère en général des contours fermés.

Maintenant le problème devient un problème d'appariement 2D: la mise en correspondance de contours de falaise. Un contour est représenté par un codage de chaîne de Freeman (la direction le long d'un contour est représentée par un entier de 0 (0 degrés) à 7 (315 degrés)). Pour trouver sa correspondance, un contour dans un MNT est comparé avec chaque contour dans l'autre MNT basé sur leurs codages de chaîne. Des points critiques le long de chaque contour sont d'abord détectés.

Un point critique ici est un point dont la courbure est localement maximale et supérieure à un seuil donné. Chaque point critique dans un MNT est comparé avec chaque point critique dans l'autre. Si l'erreur entre leurs voisins est petite, alors une hypothèse d'appariement est émise, et une transformation (2D) T peut être calculée. Cette hypothèse doit être testée en utilisant tous les autres points critiques, comme expliqué dans la suite. Tous les points critiques du premier MNT sont transformés par T. Le critère de qualité de T est le nombre de points critiques transformés qui se trouvent près d'un point critique de l'autre MNT avec une orientation similaire. Une transformation ayant 30% de points critiques alignés est déclarée valide.

Cet algorithme suppose implicitement le plan z d'un MNT est parallèle au plan z de l'autre MNT, qui n'est pas toujours le cas en pratique. Il a été testé avec un vrai MNT et une séquence d'images synthétiques obtenues en appliquant un modèle de réflectance Lambertien

au MNT. Les auteurs prétendent que leur algorithme arrive à apparier un MNT généré avec la stéréo, ayant même 40% erreurs dans les valeurs de disparité, avec le MNT original.

3.4. L'approche de Kamgar-Parsi :

La méthode de Kamgar-Parsi et al. (Kamgar-Parsi et al., 1991) est similaire à celle présentée ci-dessus, dans le sens que toutes les deux utilisent la représentation de contours pour apparier deux cartes de profondeur et que la mise en correspondance de deux contours est basée sur la comparaison de leurs codages en fonction du changement de l'orientation le long du contour. Mais elles se diffèrent sensiblement dans les détails. L'application en question est d'apparier deux cartes de profondeur du fond de la mer, acquises par un système de sondeur à multi-rayons à bord d'un bateau. Comme le tangage et le roulis du bateau peuvent être mesurés exactement, mais la direction du bateau (yaw en anglais) n'est pas connue précisément, les points 3D dans les deux cartes sont exprimés dans les systèmes de coordonnées $(x_1; y_1; z)$ et $(x_2; y_2; z)$ respectivement. Le problème consiste à trouver la transformation $T(\theta; t_x; t_y)$ entre les deux repères, où θ est l'angle de rotation, et t_x et t_y sont les composantes x et y de la translation.

Les contours utilisés sont simplement des contours de la même profondeur. Ils sont composés de chaînes de segments de droites qui peuvent changer leurs orientations quand intersectant une grille régulière.

Le critère pour trouver le recouvrement d'une paire de contours comporte deux parties.

- La première est la corrélation de deux codes de chaîne. Etant donné deux codes de chaîne, disons $\{a_i\}$ et $\{a_j\}$, en glissant un code de chaîne sur l'autre nous pouvons calculer *l'appariement erroné*: $(1/A) \sum_{\mathfrak{A}} |a_i - b_j|$ où \mathfrak{A} est la partie recouvrante, et A est le facteur de normalisation $N \ln N$, où N est le nombre d'éléments de \mathfrak{A} , et ln est le logarithme naturel. Nous utilisons *mm* pour indiquer le minimum de la valeur de *l'appariement erroné*.

- La deuxième est la proximité géométrique de deux contours correspondants. Si on trouve le recouvrement correct de deux contours, les points dans la partie recouvrante doivent se superposer dans le cas sans bruit. Le placement erroné, noté comme *mp*, est défini comme la somme de distances entre les points correspondants dans la partie recouvrante. Plus la valeur *mp* est petite, plus vraisemblablement le recouvrement est correct.

En combinant les deux, le meilleur recouvrement entre deux contours est la position où $g = mm * mp$ est minimale. Comme remarqué par les auteurs, le meilleur recouvrement ne garantit pas que le recouvrement soit correct. Il faut donc examiner la consistance entre les meilleurs recouvrements de différentes paires de contours. Ceci peut être effectué en regardant la similarité entre les transformations calculées pour chaque recouvrement. Parce que la translation calculée à partir d'un seul recouvrement n'est pas fiable, seulement l'angle de rotation est utilisé. Après avoir identifié les corrects recouvrements pour plusieurs paires (cinq ou six en général) de contours, une transformation globale $T(\theta; t_x; t_y)$ peut être calculée avec la technique des moindres carrés. Cet algorithme a été testé avec des données réelles, et le résultat ne semble pas très mauvais. (Zhang, 1993)

3.5.L'approche de Nelson et Selinger :

Nelson et Selinger (Nelson et al., 1998) utilisent des caractéristiques locales fondées sur des contours pour obtenir un système de reconnaissance robuste. Leur technique consiste à appliquer un détecteur de contours sur les images d'apprentissage puis à sélectionner les plus longs contours comme caractéristiques principales. L'adjonction à ces caractéristiques principales des contours les intersectant à l'intérieur d'une fenêtre 21 X 21 permet d'obtenir des indices visuels locaux et robustes aux variations d'éclairage, d'échelle et d'orientation. Le regroupement des indices de bas niveaux (des contours) permet d'obtenir des caractéristiques très discriminantes. Une base de 24 objets 3D assez simples est utilisée pour valider ces descripteurs dans le cadre d'un système de reconnaissance. La reconnaissance apparaît largement fondée sur les contours extérieurs des objets et la majorité de ces contours apparaissent dans les images de test même en présence d'occultation partielle ou d'un fond non uniforme. Le choix de caractéristiques fondées sur des contours paraît assez limitatif pour la gamme d'objets modélisables : des objets fortement texturés risquent, par exemple, de noyer la technique dans un trop grand nombre de contours peu informatifs. (Colin, 1999)

3.6. L'approche de Belongie et Jitendra:

Dans les travaux de Belongie et Jitendra (Belongie et al., 2002), l'approche présentée consiste à mesurer la similarité entre les formes et l'exploite pour la reconnaissance d'objets. Dans cette approche la mesure de similarité est précédé par 1) la mise en correspondance entre les points des deux formes, 2) utiliser la correspondance pour estimer la transformation d'alignement. Dans le but de résoudre le problème de mise en correspondance, Belongie et Jitendra joignent un descripteur, le *contexte de forme*, à chaque point. Le contexte de forme a

un point référence capture la distribution des points restants relative à celui-ci, en offrant une caractérisation globale discriminative. Les points en correspondance dans deux formes similaires ont un contexte de forme similaire, laissant le problème de mise en correspondance devenir un problème d'affectation optimal. Avec les correspondances de points, Belongie et Jitendra estime la transformation qui permettra de retrouver le meilleur alignement des deux formes. La dissemblance entre les deux formes est calculée comme étant la somme des erreurs d'appariement entre les points en correspondance, avec la mesure de la magnitude de la transformation pour l'alignement. Belongie et Jitendra traitent le problème de reconnaissance dans une plateforme de classification du plus proche voisin comme étant la recherche d'un modèle de forme qui est le plus similaire à la forme se trouvant dans l'image. Dans (Belongie et al., 2002), des résultats ont été présentés pour des silhouettes, des logos et marques de commerce et des chiffres manuscrits.

4. Conclusion :

La reconnaissance d'objets est un problème épineux se plaçant au niveau supérieur dans la hiérarchie des tâches de vision et constitue la partie computationnelle la plus difficile.

Ce chapitre englobe quelques méthodes de recherche pour l'indexation et la reconnaissance d'objets en se basant sur leurs silhouettes.

Les méthodes de reconnaissance et d'indexation d'objets en se basant sur leurs silhouettes font parties des méthodes locales de la vision par apparence. Une méthode de reconnaissance par apparence est toute méthode de reconnaissance d'image qui cherche à modéliser des objets (2D ou 3D) directement par leur image perçue et non pas par un modèle construit par une conception abstraite particulière.

Dans le chapitre 2, nous présenterons les méthodes de reconnaissance d'objets en présence d'occlusion.

Chapitre 2

*Les méthodes de reconnaissance
d'objets en présence d'occlusion.*

Chapitre 2 : Les méthodes de reconnaissance d'objets en présence d'occlusion.

1. Introduction:

La reconnaissance automatique d'objets dans les images est un des problèmes les plus difficiles en vision par ordinateur. Elle est en même temps une étape primordiale pour la mise en œuvre de plusieurs applications actuelles qui nécessitent une interprétation de haut niveau d'images. Par conséquent, il existe un intérêt croissant sur ce domaine de recherche dans les dernières années et une vaste littérature.

Le problème de la reconnaissance d'objets est de décider si un objet spécifique ou un objet appartenant à une catégorie d'objets prédéfinie existe dans une image. On peut poser le problème de la reconnaissance comme un appariement entre le modèle de l'objet cible et un ensemble de descripteurs qu'on extrait à partir de l'image de test. Cette généralisation, aussi simple qu'elle semble être, est capable d'expliquer la multitude d'approches qui se différencient par le choix des descripteurs qui représentent l'objet, le type et la complexité du modèle, et les méthodes utilisées pour l'apprentissage du modèle d'objet et l'appariement.

La difficulté majeure du problème de la reconnaissance d'objets repose sur les variations d'un objet ou des objets d'une même catégorie sur des images différentes. On peut grouper les sources de variations dans sept catégories principales :

- Les variations de point de vue : les objets réels sont des entités physiques tridimensionnelles. Leur apparence change de façon significative en fonction du point de vue.
- Les variations d'illumination : en fonction de changements de l'environnement et des conditions d'acquisition d'une image, un objet peut subir des variations d'illumination importantes.

Les objets dans les images des scènes naturelles sont très sensibles aux variations d'illumination qui modifient les couleurs et les valeurs radiométriques des objets considérablement.

- Les occlusions : dans une image les objets peuvent être partiellement masqués par d'autres objets ou par leurs propres composants.
- Les variations d'échelle : les objets d'une même catégorie peuvent avoir des tailles physiques différentes ou les images d'un même objet peuvent avoir des tailles différentes en fonction de la résolution de l'image.

-
- Les déformations : la plupart des objets réels n'ont pas une forme rigide. Particulièrement les objets articulés subissent généralement des déformations linéaires et non-linéaires de leurs composants.
 - Complexité du fond : en général, dans une image les objets ne sont pas isolés de leur environnement. Sur un fond complexe il est difficile de distinguer les régions appartenant à l'objet et au fond.
 - Variations intra-classe : c'est le plus important des problèmes envisagés pour la reconnaissance générique d'objets. Les objets d'une même classe peuvent avoir des apparences très variables, même quand on se limite aux catégories à un niveau élémentaire. Dans le cas extrême, les catégories fonctionnelles la classe "chaise" sont définies par un haut niveau d'abstraction et les indices visuels seuls ne sont pas suffisants pour les reconnaître.

Plusieurs raisons peuvent être à l'origine du problème de la visibilité partielle: objets déformables, recouvrements entre objets, conditions d'éclairage dégradées, objets en partie hors du champ de la caméra. Il est clair que dans ces différentes conditions, le recours à une modélisation par les primitives globales est d'emblée écarté. De nombreuses techniques ont été développées pour cet aspect du problème.

2. La vision Humaine:

L'étape la plus sensible dans un processus de reconnaissance d'objets est la mise en correspondance de points ou de contours dans le but de pouvoir détecter et reconnaître des objets dans une scène à partir d'un catalogue (ou dictionnaire) d'objet. La procédure devra être robuste à la disposition aléatoire d'objet (taille, location, orientation, contraste, bruit...) et aux occlusions.

Les contours d'un objet donnent suffisamment d'information sur sa forme. D'après la définition du dictionnaire Le Petit Robert [1973] le mot "forme" signifie :

Forme : Ensemble des contours d'un objet, d'un être, résultant de la structure de ses parties.

L'œil humain est capable de reconnaître un objet à partir des fragments de ses contours. (Figure 2.1).



Figure 2.1 : Les fragments de contours de quelques objets courants (Shotton,2007).

Un problème très Important est celui de la reconnaissance d'objets partiellement occultés qui apparaissent dans de nombreuses applications (industrielles, médicales, etc.). Pour mener à bien une telle reconnaissance, les techniques fondées sur des caractéristiques globales des objets risquent d'être inefficace. Plusieurs tentatives ont été menées pour se focaliser sur des caractéristiques locales, telles que des coins, des trous, des textures locales, des segments, moins perturbées par une occlusion éventuelle par un autre objet. De nombreuses techniques ont été développées pour mener à bien une reconnaissance d'objet en présence d'occlusion. Parmi ces techniques il y a (Bhanu et al.,1984)(Chow et al.,1977)(Turney et al., 1985).

3. Le schéma général des méthodes de reconnaissance d'objets :

Le fossé sémantique est un terme utilisé pour décrire la difficulté des systèmes de vision par ordinateur pour obtenir une interprétation de haut niveau à partir des descriptions visuelles de bas niveau d'une image. Le problème de la reconnaissance d'objets automatique est un exemple parfait pour illustrer cette difficulté. On cherche dans l'image, qui n'est qu'un ensemble de pixels ayant des valeurs d'intensité numériques et des coordonnées dans un espace bi-dimensionnel, des objets qui ont des descriptions correspondantes aux concepts sémantiques. Les premiers travaux de recherche en vision par ordinateur se sont surtout concentrés sur les traitements de bas niveau, les traitements qui opèrent uniquement sur l'intensité des pixels sans considérer les entités qu'ils représentent, comme la segmentation et la détection des contours. Après être arrivé à une certaine maturité aux traitements de bas niveau, on est confronté aujourd'hui à un intérêt croissant de la communauté scientifique pour des traitements de haut niveau qui visent à interpréter une image par des entités symboliques et leurs relations spatiales. Les méthodes de reconnaissance suivent en général ces deux niveaux de traitements, les traitements de bas niveau et les traitements de haut niveau qui correspondent aux étapes suivantes :

1. Extraction : l'extraction d'un ensemble de descripteurs à partir de l'image, qui reflètent informations pertinentes pour la reconnaissance de l'objet ;
2. Représentation : la construction d'un modèle d'objet pour représenter une catégorie

d'objet;

3. Apprentissage : l'apprentissage des paramètres du modèle à partir des images d'apprentissage étiquetées ;

4. Reconnaissance : l'appariement avec le modèle ou la classification en utilisant le modèle pour reconnaître l'objet cible dans une image de test.

Les méthodes de reconnaissance d'objets se différencient principalement au niveau de:

- La définition du problème de reconnaissance : les catégories d'objets recherchées, le nombre de catégories ;
- Le nombre d'images utilisées pour l'apprentissage : une extrémité étant l'apprentissage à partir d'une seule image, on peut aussi utiliser un très grand nombre d'images pour l'apprentissage ;
- Le niveau de supervision : en cas d'un apprentissage non-supervisé on ne dispose pas d'images étiquetées, et on essaye de grouper les images dans des catégories d'après leurs similarités (clustering). En cas d'apprentissage supervisé, on a un ensemble d'images d'apétiquetées. Il s'agit d'un niveau de supervision plus élevé quand on connaît la segmentation manuelle des parties de l'objet sur l'image ;
- La modélisation : ici, nous utilisons le terme modèle dans le sens général de "la représentation d'un objet". Un très grand nombre de modèles d'objets différents ont été proposés pour représenter les objets dans les images ;
- Les méthodes et les algorithmes choisis pour l'extraction des descripteurs, la construction du modèle et la reconnaissance.

4. Méthodes se basant sur la silhouette pour la reconnaissance d'objets en présence d'occlusion :

4.1. Techniques Corrélatives:

Ces techniques utilisent usuellement une représentation des images par une description locale de leurs contours par la fonction de courbure (ou une de ses variantes) dans l'espace (θ, s) . Une mesure de similarité entre un modèle sélectionné et l'image observée est calculée ensuite par une corrélation croisée entre ces fonctions de description d'une partie des contours du modèle et celles d'une partie des contours de l'image. Cette corrélation est faite dans l'espace (θ, s) , où $\theta(s)$ est une fonction de l'abscisse curviligne représentant l'angle que fait la tangente au point -d'abscisse s - du contour considéré avec une direction choisie arbitrairement.

McKee et Aggarwal (McKee et al., 1977) ont proposé un des premiers systèmes basé sur cette technique. Une mesure de dissimilarité entre le modèle et l'image est déterminée par l'évaluation de la surface normalisée entre les courbes $\theta(s)$ du modèle et celle de l'image. Ce système tient compte de l'aspect de visibilité partielle, mais suppose la présence d'un seul objet dans la scène. Cette hypothèse permet notamment la détermination des points d'intersection de l'objet avec le champ de vision afin d'évaluer la mesure de dissimilarité.

Parkins (Perkins, 1978) propose une méthode utilisant une représentation de contours comparable à la précédente. Une segmentation des contours en « concourbes » ou polylignes (chaines formées de segments de droite et d'arcs de cercle) est faite à partir de la détermination de la courbure en chaque points par le calcul de $\Delta\theta/\Delta s$ à partir de ses deux points adjacents, où θ est la valeur retournée par l'opérateur de Hueckel et s la distance sur le contour considéré. Le processus de reconnaissance d'un objet dans une image est élaboré en trois étapes.

1. Premièrement, la structure de contrôle établit l'ordre dans lequel les combinaisons polylignes modèle et polylignes image seront considérées. Ceci est effectué à partir d'une comparaison des mesures scalaires (longueur, surface, nombre de segments de droite, nombre d'arcs de cercle d'une polylignes) extraites du modèle et des polylignes de l'image. La comparaison est une procédure d'appariement exhaustif appliquée à toutes les paires possible polylignes image-polyligne modèle. Les résultats donnés en termes de mesures de ressemblance sont rangés dans une liste ordonnée. Un ordre de sélection des modèles est ainsi défini.

2. Ensuite, ces correspondances sont prises dans l'ordre décroissant de vraisemblance jusqu'à la détermination d'une transformation (translation (x, y) et une rotation θ) qui permet de superposer une partie du modèle à une partie de l'image. Pour cela, les polylignes image et modèle sont alignées par des corrélations croisées dans l'espace (θ, s) .

3. Enfin, une vérification de la transformation est faite par l'application de celle-ci au modèle complet. En cas de succès, la position de l'objet dans la scène est déterminée.

Dessimoz (Dessimoz, 1979) utilise une approche similaire avec corrélation croisée entre description des contours du modèle et celle de l'image. La principale différence est qu'il utilise la fonction intrinsèque du contour comme description de celui-là ($u(s) = d\theta/ds$), laquelle a l'avantage d'être invariante aux déplacements de courbe (translation et rotation).

Riad (Riad, 1983) emploie également une technique de corrélation à partir de primitives sélectionnées lors d'une phase d'apprentissage assez bien élaborée. Le processus d'apprentissage est interactif. Il permet de sélectionner à partir des portions de contour délimitées par un opérateur, les primitives les plus discriminantes pour chaque objet comparées à celles qui caractérisent l'ensemble du reste des autres objets.

Shirai (Shirai, 1978) utilise des caractéristiques similaires à celle utilisées par Perkins, mais où les modèles sont organisés hiérarchiquement en primitives principales (les plus visibles) et secondaires.

Ces méthodes donnent de bons résultats dans l'analyse de scène industrielles, elles sont insensibles au bruit à cause du lissage des contours avant la segmentation.

Mais elles requièrent néanmoins des prétraitements coûteux et ne sont pas adaptées aux variations du facteur d'échelle.

4.2. Techniques syntaxiques:

Les techniques syntaxiques font partie des premières techniques utilisées pour la reconnaissance de formes. Elles ont été développées initialement pour l'analyse et la reconnaissance de caractères. La représentation des formes est faite à partir d'une description par un ensemble de symboles. Pour chaque objet est associée une syntaxe propre décrite par une liste de règles constructives ou de production syntaxique appliquée à des sous-ensembles de symboles pour une description symbolique. Ces symboles représentent des entités graphiques de type segments de droite, arcs de cercle, etc.

Le processus de reconnaissance d'objets dans une image consiste à rechercher dans celle-ci les chaînes les plus longues de symboles compatibles avec les syntaxes des modèles correspondants.

Pavlidis (Pavlidis, 1979) ont proposé une méthode hiérarchique d'analyse syntaxique de formes. La description est faite après une approximation polygonale des contours par quatre types d'entités ayant chacune un nom formel: arcs approximés par des quadriques (orientés par leurs bissectrices), progressions ou intrusions, segments de droite longs, segments court irréguliers. Un analyseur grammatical de formes est utilisé; ce qui suggère son application à une large classe de problèmes de reconnaissance de formes sans utiliser des moyens plus puissants que des automates finis.

Tsai et Fu (Tsai et al., 1980) proposent une approche qui combine les techniques syntaxiques et les techniques statistiques pour la reconnaissance d'objets industriels. Dans cette approche, les contours sont segmentés par des segments de droite et des segments courbes selon la similarité de la courbure des points. Une représentation de chaque segment par un symbole distinct est effectuée. Elle permet une transformation des contours en suite de symboles pour une représentation structurelle. Des règles de production syntaxiques sont inférées à partir des représentations de chaînes pour former une partie de la grammaire attribuée. Enfin la grammaire attribuée inférée est utilisée pour analyser les représentations de formes. La combinaison des deux techniques dans cette approche a pour but selon les auteurs d'exploiter l'avantage de l'une, là où l'autre est faible.

Tsai et Yu (Tsai et al., 1985) ont introduit une nouvelle opération d'édition de primitives appelée (merging) qui peut être utilisée pour combiner et comparer un nombre quelconque de primitives de contour consécutives. Cette nouvelle méthode d'appariement de chaînes avec fusion s'est avérée utile pour la reconnaissance de formes distordues et présente ainsi certains avantages par rapport à l'appariement de chaînes conventionnel. Les résultats des expérimentations rapportées semblent satisfaisant, mais celles-ci ont été réalisées sur des images très simple : un objet à la fois dans la scène et trois modèles de pinces différentes.

On peut dire d'une manière générale que l'application des méthodes syntaxiques réussit d'avantage dans le cas des images ayant des contours simples et nets. Ceci explique son utilisation d'avantage pour l'analyse de caractères, cellules, chromosomes, etc; que pour la description des images de pièces industrielles complexes. Ceci est encore appuyé probablement par les difficultés rencontrées pour automatiser la phase de modélisation par nature complexe.

4.3. Techniques d'analyse d'histogramme :

4.3.1. Transformée de Hough pour la reconnaissance d'objets partiellement occultés :

La transformée de Hough a été développée par Paul Hough en 1962. Dans les dernières décennies, la transformée de Hough est devenu un outil standard dans le domaine de la vision artificielle. Elle permet la détection de droites, de cercles ou d'ellipses de façon traditionnelle. Elle peut aussi être étendue à des cas de description d'objets plus complexes. La transformée de Hough permet d'avoir une bonne robustesse contre les discontinuités et les

éléments manquants d'une structure.

Le principe général de la transformée de Hough est d'établir une projection entre l'espace de l'image et un espace de paramètres représentatif de la forme recherchée.

La transformée de Hough (Duda et al., 1972) est une méthode pour détecter les formes simples à partir d'un ensemble de points. L'application la plus classique est la détection des segments de droites. Le principe est de représenter les segments entre chaque paire de points dans un espace de coordonnées défini par l'angle et la distance du segment à l'origine et d'accumuler les nouvelles coordonnées dans une matrice. Les maxima locaux de cette matrice indiquent les segments. On peut appliquer la transformée de Hough également pour détecter les cercles. Cette fois-ci on fait la transformation vers un espace à trois coordonnées, position et le rayon. La recherche des cercles dans cet espace est difficile surtout quand on ne connaît pas les rayons des cercles recherchés.

La transformée de Hough traite les contours comme un ensemble de points indépendants. A la fin de l'étape de l'extraction des contours on dispose souvent des informations sur les connections entre les points de contours et la direction du gradient sur ces points. On peut exploiter ces informations pour une meilleure extraction des primitives. Venkateswar (Venkateswar, 1992) ont détecté les segments de droites dans les images aériennes en parcourant l'image pour affecter une étiquette de segment à chaque point de contour d'après la direction du gradient à ce point. On regroupe ensuite les points qui ont les mêmes étiquettes et on détermine les extrémités. Dans une méthode similaire, Desolneux (Desolneux et al., 2000) utilisent le principe d'événement significatif : dans une image, les directions des gradients de deux points voisins sont des variables aléatoires.

Les déviations de ce caractère aléatoire sont considérées comme des événements significatifs qui correspondent aux structures. Ils ont calculé la direction du gradient très localement, dans un voisinage de 2×2 et recherché les événements significatifs. Ils ont utilisé leur méthode pour détecter les structures linéaires dans les images.

(Du-Ming, 1997)

4.4. Techniques de recherche arborescente :

Grimson et Lozano-Perez (Grimson et al., 1985) décrivent un système de reconnaissance d'objets 2D et 3D dit de recherche d'arbre d'interprétation (« searching interpretation tree ») où ils limitent l'exploration totale par des tests de coupure (« search cutoff ») basés sur une mesure de qualité. Ils utilisent en outre la transformée de Hough à l'initialisation pour réduire

la dimension de l'espace de recherche. Cette approche pour les objets partiellement observés utilisant des mesures locales de positions et des normales aux surfaces est une extension à celle proposée par les mêmes auteurs dans (Grimson et al., 1984) pour la reconnaissance d'objets sans recouvrement.

Ayache dans (Ayache, 1983) et Faugeras et Ayache dans (Ayache et al., 1986) ont proposé un système HYPER-2D pour la reconnaissance d'objets partiellement cachés. La segmentation des contours est faite par une approximation polygonale. Le processus de reconnaissance fait premièrement une génération d'hypothèses sur la position de l'objet ensuite, réalise des appariements séquentiels des segments modèles et image en tirant parti de l'estimation de la position du modèle à chaque appariement. Cette position est affinée récursivement à chaque identification d'un segment par l'utilisation du filtre de Kalman pour estimer les paramètres de la transformation. Un score de qualité réactualisé à chaque appariement permet de sélectionner la meilleure hypothèse.

5. Autres techniques :

Dorkó et Schmid (Dorko et al., 2003) ont utilisé des descripteurs Harris-Laplace et invariants à l'échelle. Ils ont groupé les descripteurs similaires et appliqué deux types de classifieurs, les et le modèle de mixture gaussienne pour apprendre les descripteurs les plus significatifs.

Les modèles par l'apparence seule sont indifférents aux positions des composants et à leurs relations spatiales. Cela apporte certes une classification rapide et robuste aux occlusions.

Mais, d'un autre côté, le modèle n'est pas capable de distinguer les images d'objets qui possèdent des composants similaires mais avec des configurations spatiales différentes.

Shotton (Shotton, 2007) propose d'utiliser des fragments de contours comme descripteurs. A partir d'un petit nombre d'images d'objets segmentées on construit un dictionnaire des fragments.

Le dictionnaire est constitué des fragments des contours extérieurs de l'objet et des positions des fragments par rapport au centre de l'objet. Pour construire le dictionnaire, on génère aléatoirement un grand nombre de régions qui contiennent une partie de l'objet et du fond. On perturbe légèrement ces régions par des transformations aléatoires pour obtenir un dictionnaire plus représentatif, et on détecte les contours sur chaque région.

L'évaluation de Shotton (Shotton, 2007) démontre que les contours peuvent être utilisés avec succès dans la reconnaissance d'objets d'une large variété de classes d'objets en présence d'occlusion. La figure 2.2 illustre la détection d'objets en présence d'occlusion.

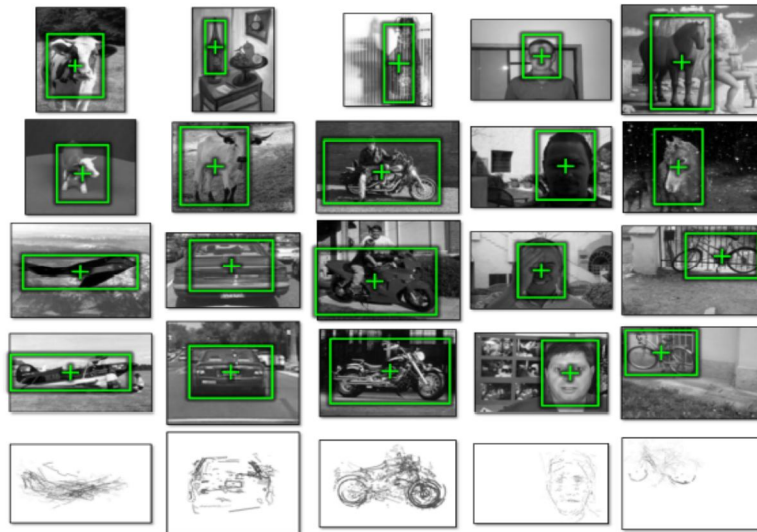


Figure 2.2 : Localisation d'objets avec de multiple objets et en présence d'occlusion. (Shotton, 2007)

Lowe (Lowe, 2004) présente un outil complet permettant: de détecter et sélectionner ce que l'on appellera les points-clés de l'image, de décrire un point-clé à l'aide de son environnement, de définir une mesure entre deux descripteurs de points-clés, et enfin de chercher les correspondances entre les points-clés de deux images. Cet outil -qui est amené à être utilisé dans nombre d'applications liées à la reconnaissance d'objets- est appelé **SIFT** (*Scale Invariant Feature Transform* en anglais).

La démarche proposée par Lowe se déroule en 3 phases:

- Une recherche de mise en correspondance de tous les points-clés extraits de l'image à analyser avec ceux de la base de données.
- Les points-clés dont les correspondants pointent vers un même objet sont rassemblés en un groupe (*cluster*).
- Chaque groupe est alors testé géométriquement afin d'être valide comme portion d'un objet.

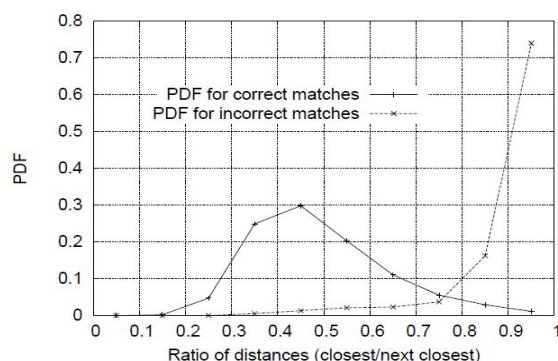
a) Le choix d'un critère pour la mise en correspondance des points-clés:

La solution la plus élémentaire visant à établir le plus proche voisin d'un point-clé dans une base de données est de choisir le représentant **minimisant la distance euclidienne entre les deux vecteurs de description**, ce qui est justifié par le choix du type de représentation

(histogramme d'orientation de gradient). Bien entendu, avec ce type de critère, il y a irrémédiablement beaucoup de fausses alarmes car, pour une base de données suffisamment volumineuse (quelques images suffisent étant donné le nombre moyen de points-clés par image, de l'ordre de 103), le risque qu'il y ait par hasard un point-clé différent mais avec un descripteur très proche est non négligeable. De plus, il peut arriver qu'en calculant la distance euclidienne, on trouve par exemple trois distances très faibles en comparaison des autres: il n'est pas satisfaisant soit de choisir le minimum minimorum, soit de prendre tous ces représentants car notre préoccupation majeure est **de minimiser au possible le taux de fausses alarmes**. En effet, il importe plus d'avoir des points-clés qui pointent vers le véritable objet, plutôt qu'un grand nombre de points qui nous amèneront à détecter des objets fantômes! Une possibilité serait alors de définir un seuil minimal mais pour les raisons que l'on vient d'évoquer, on comprend bien que cela n'apporte pas de réelles améliorations en pratique.

La solution retenue par Lowe (Lowe, 2004) est de ne conserver la solution minimisant la distance que si elle est significative vis à vis des autres solutions, ce qui est obtenu en définissant **un seuil maximal sur le rapport entre la distance du plus proche voisin sur celle du second plus proche voisin** (maximal car la distance du plus proche voisin est plus faible que le second plus proche voisin).

Pour sélectionner efficacement ce seuil, on trace la courbe expérimentale ci-dessous qui montre l'histogramme normalisé (donc l'estimation de la densité de probabilité) des fausses et bonnes correspondances en fonction de ce rapport des distances. Le fait le plus remarquable est de voir que la solution du minimum minimorum est catastrophique dans le cas où le rapport tend vers 1. Ceci a conduit Lowe à dans ces travaux (Lowe, 2004) choisir un rapport de **0.8** ce qui permet d'**éliminer 90% des fausses alarmes** et ne perdre que 5% des bonnes correspondances.



b) Recherche du plus proche voisin :

Sachant qu'il serait fastidieux de calculer toutes les distances entre les points-clés de l'image et les représentants du dictionnaire (par exemple, pour une base de 100 images soit 105

points-clés, cela équivaudrait au calcul de 107 distances entre deux vecteurs de dimension 128, soit environ $4 \cdot 10^9$ opérations...).

Lowe utilise en ce sens un algorithme BBF5 (*Best Bin First*) permettant de trouver plus rapidement (gain de calcul d'environ **2 ordres de grandeur** grâce notamment au critère du rapport maximal de 0.8) pour seulement **une perte de 5%** des bonnes correspondances.

c) Détection et positionnement d'un objet :

Dans le but de détecter et positionner un objet dans la scène (et ce même en cas d'inclusion), il faut réaliser deux étapes d'identification:

- La première consiste à choisir si un objet de la base de donnée est présent ou non. Ceci est rendu possible grâce au regroupement (clustering) des représentants sélectionnés pointant vers un même objet. Expérimentalement, il s'est avéré qu'il faut un cluster contenant au moins 3 points-clés pour avoir un très faible taux de fausses alarmes de détection.
- La seconde vise à établir quelles sont la localisation, l'échelle et l'orientation de l'objet que l'on a détecté grâce aux informations stockées avec les points-clés de l'image analysée (nous verrons le cas des transformations affines au paragraphe suivant). Ceci peut être obtenu en utilisant la Transformation de Hough qui identifie les points-clés ayant du sens vis à vis de l'objet détecté (inliers par opposition aux outliers) par un système de vote dans l'espace des 4 paramètres que sont la localisation 2D, l'orientation et l'échelle. L'auteur préconise l'utilisation de cellules de taille 30° pour l'orientation, variant par octave pour l'échelle (soit d'un facteur 2) et de prendre pour la localisation la taille correspondant à 0.25 fois la dimension maximale de l'image de l'objet donnée par l'échelle prédite. Afin d'éviter les effets de bords, à l'image de la construction de l'histogramme de l'orientation du point-clé, chaque point-clé vote pour les deux plus proches cellules en dimension 4 de ses paramètres.

Au final, cela permet d'obtenir l'estimation de l'orientation, de l'échelle et d'un point de repère pour localiser l'objet dans l'image et ainsi de tracer sur l'image le contour (rectangle par défaut) contenant l'objet détecté.

d) Extension aux transformations affines :

La méthode proposée au paragraphe précédent permet d'obtenir à la fois une bonne détection et un très bon positionnement pour les objets ayant seulement subi une rotation 2D. Pourtant,

si comme on l'a vu, la détection pour les objets plans subissant une rotation 3D reste performante, le positionnement ne convient plus; il faut alors réaliser une étape supplémentaire visant à établir quels sont les paramètres de la transformation affine qu'a subit l'objet dans la scène.

Une méthode simple est d'exprimer la solution au sens des moindres carrés:

La méthode d'interprétation des points-clés en fonction de l'objet détecté est alors modifiée:

- Utiliser les points-clés -d'un même cluster- extraits de l'image analysée et les représentants sélectionnés afin de déterminer les paramètres de la transformation affine entre ces deux images.
- Les points qui « collent » le moins au modèle sont alors éliminés en comparant les positions des points-clés théoriques $[u',v']^T$ obtenus par la solution des moindres carrés avec les cellules auxquelles ont été attribué les positions des points retenus par la transformée de Hough.
- La solution des moindres carrés est alors réévaluée avec les points restants et en ajoutant éventuellement les points rejetés par la transformée de Hough mais qui collent au modèle de la transformation affine.
- Réaliser ces opérations en boucle jusqu'à une solution stable.

S'il reste moins de 3 points-clés retenus, l'objet est rejeté. Dans le cas contraire, on réalise un dernier test, sur la probabilité P -a posteriori, basée sur une analyse bayésienne- que les points-clés retenus correspondent bien à l'objet choisit: l'objet n'est retenu que si $P > 0.98$, ce qui en pratique nécessite **3** points-clés pour un objet de petite dimension dans l'image analysée et **10** pour un objet plus grand.

e) Exemple de résultats :



Figure 2.3 : Les images de deux objets sont montrées à gauche. Les deux objets peuvent être reconnus dans l'image du milieu en présence d'occlusion. Le résultat de la reconnaissance est

montré à droite. (Lowe, 2004)

L'ensemble des opérations décrites dans les paragraphes précédents ont été menées sur l'image centrale vis-à-vis de la base de données constituée des 2 objets à gauche. On observe d'une part, que malgré **l'occlusion, le changement d'échelle, le changement de point de vue 3D, et l'orientation des objets**, ces derniers sont bien détectés et en outre, que la détermination de la transformation affine donne de bons résultats.

6. Conclusion :

Dans ce chapitre, nous avons abordé plusieurs techniques de littérature concernant l'indexation et la reconnaissance des silhouettes en présence d'occlusion.

Ces techniques consistaient à reconnaître un ou plusieurs objets dans une image qui ont étaient partiellement occultés.

Dans le chapitre 3, nous détaillerons notre méthode proposée concernant la reconnaissance d'objets en occlusion.

Chapitre 3

Indexation et reconnaissance

Chapitre 3 : Indexation et Reconnaissance.

1. Introduction :

Plusieurs approches de reconnaissance et d'indexation d'objets ont été investies ces dernières années. Un problème très important est apparu ; celui de la reconnaissance d'objets partiellement occultés. Ce problème apparaît dans de nombreuses applications (industrielles, médicales, etc.). Effectivement, dans le monde réel, les objets peuvent être partiellement occultés dans une image. Plusieurs raisons peuvent être à l'origine de la visibilité partielle : objets déformables, recouvrement entre objets, conditions d'éclairage dégradées, objets en partie hors du champ de la caméra. Et malgré cette occlusion d'après des études psychologiques (De Winter et al., 2004) (Biederman et al., 1988) , l'œil humain peut détecter et reconnaître des objets partiellement occultés. La question qui se pose est comment arriver à construire un système capable de palier à cet aspect du problème ?

Pour mener à bien une telle reconnaissance, il est clair que les techniques fondées sur les caractéristiques globales risquent d'être inefficaces. De nombreuses techniques ont été développées pour la reconnaissance d'objets en présence d'occlusion telle que (Papadourakis et al., 2011) (Liu et al., 2012) (Shotton et al., 2013) (Torralba et al., 2012) (Wu et al., 2012) (Zhu et al., 2010).

Nous présentons dans ce chapitre, notre méthode d'indexation et de reconnaissance d'objets par leurs silhouettes en présence d'occlusion.

La méthode élaborée combine deux méthodes de reconnaissance d'objets par leurs silhouettes. La première méthode utilisée est la méthode du Quadtree traitant initialement des objets en absence d'occlusion et la deuxième méthode est la reconnaissance d'objets par leurs courbures. En plus de la combinaison des deux méthodes, le défi principal de notre approche est la possibilité de reconnaître les objets même en présence d'occlusion. L'intérêt de cette combinaison est de palier à certains inconvénients de la méthode du Quadtree (sans occlusion) pour la reconnaissance d'objets comme par exemple sa non robustesse à la rotation (Buffa, 2001).

Plusieurs techniques ont été utilisées pour pallier aux problèmes de rotation comme les boîtes englobantes, les boîtes minimales et la représentation par les moments invariants (Maître, 2004); ces techniques amènent à des calculs assez importants sur l'image avant même de commencer la reconnaissance. Nous avons décidé d'utiliser une technique qui nous permet

de faire une rotation de l'objet de l'image requête de telle sorte qu'il y ait une correspondance avec un objet d'une image de la base. Ensuite nous utilisons la méthode du Quadtree pour une meilleure correspondance entre l'objet de l'image requête et l'objet de l'image de la base.

Nous définissons dans ce chapitre ce qu'on appellera seuil d'occlusion.

Pour concevoir un système de reconnaissance d'objets, la première étape à définir est l'indexation des objets dans une image.

Plusieurs méthodes d'indexation ont été développées aux besoins de différentes applications dans différents domaines d'applications. Parmi ces méthodes, il existe des méthodes calquées sur les techniques du séquentiel indexé. Leur principe est de décomposer l'espace en sous-ensemble bornés, appelé feuilles, et d'indexer les objets par rapport à ces feuilles et d'effectuer les recherches séquentiellement dans les feuilles sélectionnées pour le domaine d'étude.

On peut classer les différentes méthodes utilisées selon la définition et la structure des feuilles :

- les structures en arbres quaternaires ou *region-Quad-Tree*
- les structures en arbres k-d (*k-dtree*)
- les structures en arbres R (*R-tree*).

Dans ce qui suit, notre travail sera basé sur une structure en arbres quaternaires ou *région Quad-Tree* (découpage en *Quad-Tree*) ainsi que sur une technique de mise en correspondance utilisant les informations de courbure de la silhouette de l'objet.

2. Architecture globale d'indexation et de reconnaissance d'objets :

Le système que nous avons mis en place pour l'indexation et la reconnaissance d'objets et cela même en présence d'occlusion, se compose de deux étapes : l'étape d'indexation et l'étape de recherche.

La figure 3.1 ci-dessous présente le schéma global du système conçu pour l'indexation et la recherche et reconnaissance d'un objet dans une image.

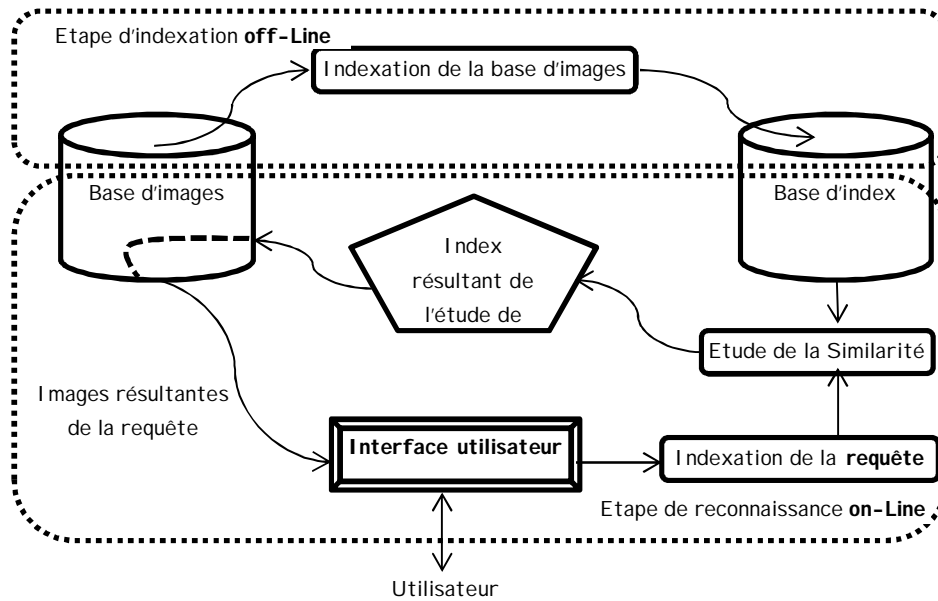


Figure 3.1 : Architecture globale du système d'indexation et de reconnaissance d'objets.

L'étape d'indexation est une étape qu'on a choisi de mettre en off-line car cette étape est construite lors de la conception du système. Elle permet de construire une base d'index correspondante à une base d'images que nous possédons. La seconde partie correspond à la phase de recherche proprement dite, elle nécessite la mise en place de métriques de comparaison entre un ensemble de caractéristiques issues de la requête et les caractéristiques des images sélectionnées dans la base d'images.

3. Etape indexation :

Nous nous intéressons dans ce chapitre à la technique du Quadtree, pour établir une indexation de notre base d'images. Ainsi qu'à une deuxième technique qui utilise les informations de courbures d'une silhouette pour la reconnaissance d'un objet dans une image.

Dans le système que nous avons conçu, L'étape d'indexation consiste à extraire automatiquement des caractéristiques de l'image et à stocker ces caractéristiques dans un vecteur (dans notre cas fichier texte) numérique qu'on appelle *descripteur visuel*.

En réalité, dans notre système pour chaque image nous avons deux index stockés. Nous verrons dans ce qui suit l'intérêt qu'apporte le deuxième index à notre système.

Les étapes pour arriver d'une image A de la base d'images à un index de l'image sont montrés dans la figure 3.2.

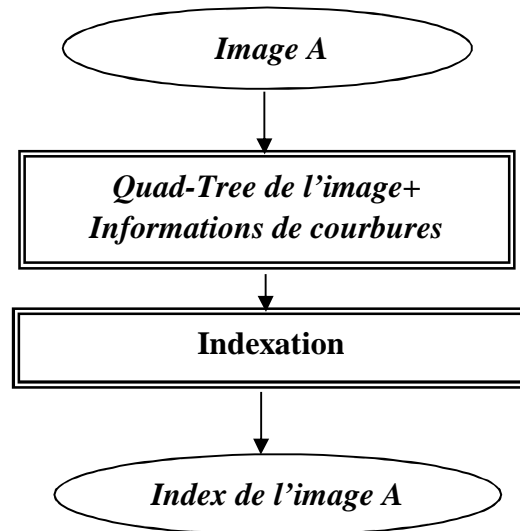


Figure 3.2 : Etapes d'indexation.

Nous décrivons dans ce qui suit, les étapes permettant l'extraction des caractéristiques de l'image et la création de l'index d'une image A, en premier lieu avec la technique du Quadtree et en deuxième lieu la technique utilisant les informations de courbure pour la reconnaissance de l'objet.

3.1. Technique d'indexation utilisant la structure Quadtree :

3.1.1. Définition :

Les QuadTree, (ou arbres quaternaires en français) sont une structure de données qui représente des arbres, dont chaque nœud possède quatre fils (Samet, 1984).

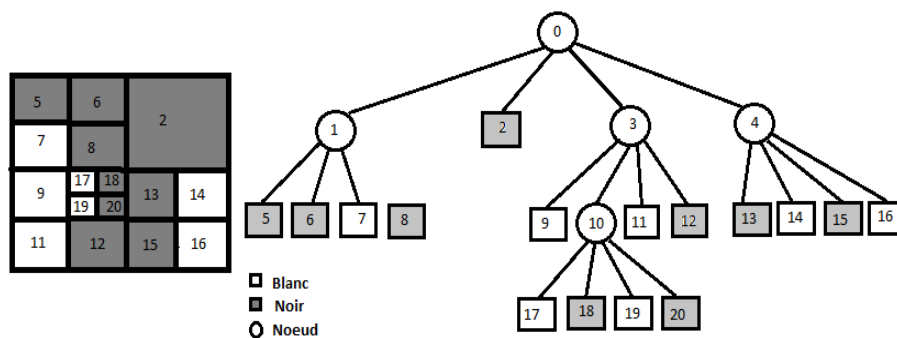


Figure 3.3 : Représentation d'un Quad-tree. (Aouat et al., 2010).

Il est simple de le représenter par analogie avec les arbres binaires que tout le monde connaît, la seule différence vient du fait que chaque nœud possède non pas deux mais quatre feuilles / fils. (Figure 3.3)

Un arbre quaternaire est une structure de données permettant de représenter des images à deux dimensions (Manouvrier et al., 2002). Les images sont découpées en quadrants ou carrés réguliers selon un critère de découpage donné (l'homogénéité, la texture, etc.). Un arbre quaternaire peut être implanté de manière hiérarchique, à l'aide de pointeurs, ou de manière linéaire, sous forme d'une liste de valeurs. L'arbre quaternaire permet de tenir compte, lors de la recherche d'images similaires, de la localisation spatiale des caractéristiques d'images, telles que la couleur (Albuz et al., 2000) et le contour (Chakkrabarti et al., 2000).

3.1.2. Recherche :

◆ Recherche d'un point précis :

La recherche d'un point dans un Quad-Tree commence en comparant ses coordonnées avec celles de la racine.

Après avoir déterminé les fils correspondant au rectangle qui contient le point requête, on compare de nouveau les coordonnées requêtes avec celles de ce nœud fils, et ainsi de suite jusqu'à la localisation de la requête, ou la rencontre d'un point vide (le point requête n'appartient pas à la base).

La recherche d'un point précis dans un Quad-Tree est très simple et efficace, puisque si l'on dispose d'un ensemble de données contenues dans $4n$ feuilles réparties uniformément, on n'aura besoin que de $(n+2)$ accès en tous pour la requête. (Garici et al., 2002)

◆ Recherche de points similaires

Un autre type de recherche dans un arbre est la recherche de points similaires à un point donné (les points se trouvant à une certaine distance de ce point, ou les points se trouvant dans un rectangle dont le centre est ce point).

Cette recherche s'effectue au début en déterminant le nœud correspondant au plus petit rectangle englobant la zone de recherche.

Si ce nœud pointe sur une feuille alors, on détermine par calcul des points appartenant à la zone requête, sinon pour chacun de ses fils dont le rectangle correspondant se chevauche avec la zone requête, on lance la même procédure récursive de recherche. (Gariciet al., 2002)

3.1.3. Insertion :

L'insertion dans un Quad-Tree est une opération relativement aisée, car sont peu de cas spéciaux se présentent.

Pour insérer un nouveau point, on commence par rechercher le nœud dont le rectangle correspondant est le plus petit rectangle contenant ce point, deux cas se présentent :

- Si ce nœud ne pointe pas sur une feuille correspondant à ce rectangle, alors une nouvelle feuille est allouée et le point est y inséré.
- Ce nœud pointe sur une feuille déjà existante, alors le point y est inséré. Si elle est pleine, on procède à son éclatement, c'est-à-dire que le rectangle auquel elle correspond sera divisé en quatre nouveaux rectangles, donc cette feuille va être remplacée par un nœud pointant sur quatre nouvelles feuilles, et ces vecteurs seront insérés dans ces feuilles. (Garici et al., 2002).

3.1.4. Avantages des Quad-Tree :

1. Simple d'utilisation : il suffit de trouver la relation récursive existant entre les quadrants de l'image, et bien choisir les structures de données à utiliser.
2. Pas de débordement en mémoire centrale : En sauvegardant les informations relatives aux quadrants dans un fichier, la mémoire est libérée au fur et à mesure de découpage en Quad-Tree, et par conséquent la mémoire ne risque pas d'être débordée.

3.1.5. Les images en Quad-Tree

Puisqu'une image peut comporter un ensemble de régions avec des propriétés distinctes au niveau des caractéristiques, il serait intéressant de procéder à un découpage automatique de l'image.

La méthode couramment utilisée consiste à faire une dichotomie par blocs de l'image. Pour cela, on commence par définir un bloc de la taille de l'image, puis on examine le contenu de ce bloc. (Avec des conditions bien définies qu'on va voir en détail dans les sections qui suivent). Si le bloc est homogène (contient exclusivement des pixels similaires) alors on

arrête la décomposition. Sinon, on découpe le bloc en 4 sous blocs et on examine le contenu de chaque sous blocs.

Le découpage de l'image (ou bien l'algorithme de découpage) se fait selon la courbe de *Peano* (parcours en *Z*), qui est décrit ci-après.

(Voir la figure 3.4)

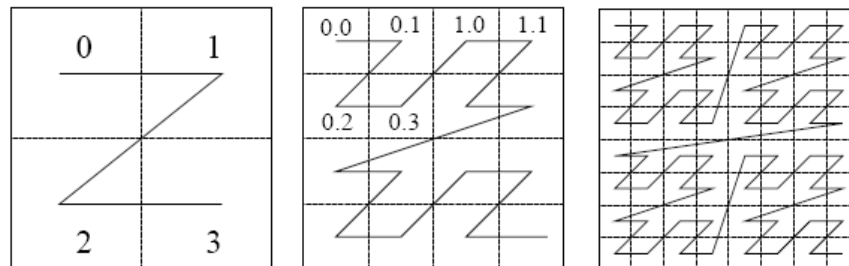


Figure 3.4 :schéma montrent le parcours du découpage.

Et ainsi de suite jusqu'à ce qu'il n'y ait plus besoin de décomposer les blocs. Le résultat obtenu est donc un ensemble jointif de blocs de différentes tailles qui recouvre entièrement l'image. (Voir figure 3.5)

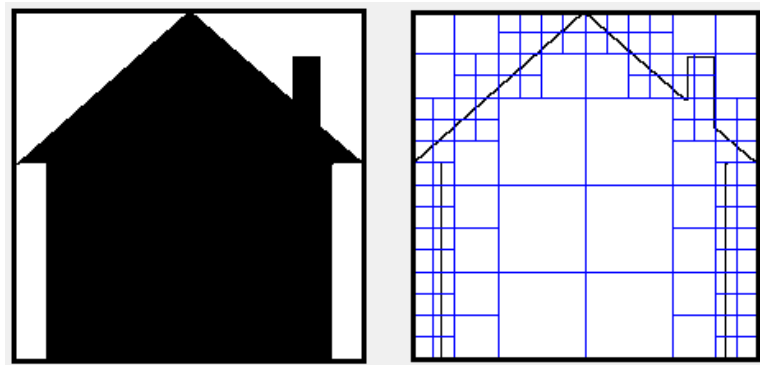


Figure 3.5 : schéma d'un découpage en Quadtree.

3.1.6. L'arbre Quartenaire :

L'un des types d'arbres quaternaires les plus connus est celui basé sur les régions. Selon le critère de décomposition qui peut être l'homogénéité de la couleur ou de la texture d'une région d'une image, cette dernière est récursivement décomposée jusqu'à l'obtention de zones homogènes.

Par exemple, chaque région va être décomposée en quatre quadrants (carrés) identiques. Chaque quadrant est représenté par un nœud qui aura quatre nœuds fils correspondant aux

quatre sous quadrants (NO : Nord-Ouest, NE : Nord-Est, SO : Sud-Ouest et SE : Sud-est) du quadrant initial si ce dernier est décomposé.

La racine de l'arbre quaternaire représente l'image initiale. Pour savoir à quelle partie de l'image correspond un nœud de l'arbre, on a besoin d'une fonction d'identification des nœuds. Une façon de faire est de numéroter les nœuds par des séquences de numéros compris entre 0 et 3 tel que: 0 correspond au quadrant NO, 1 au NE, 2 au SO et 3 au SE.

Le nœud racine est identifié par 0. Ses 4 fils sont identifiés respectivement par 00, 01, 02 et 03. Chaque fils du nœud 00 à pour identificateur 00x, $x \in [0;3]$, chaque fils du nœud 01 à pour identificateur 01x, $x \in [0;3]$, etc. Cette numérotation est employée dans la figure 3.6. Il s'agit d'une numérotation en Z. (Voir figure 3.6)

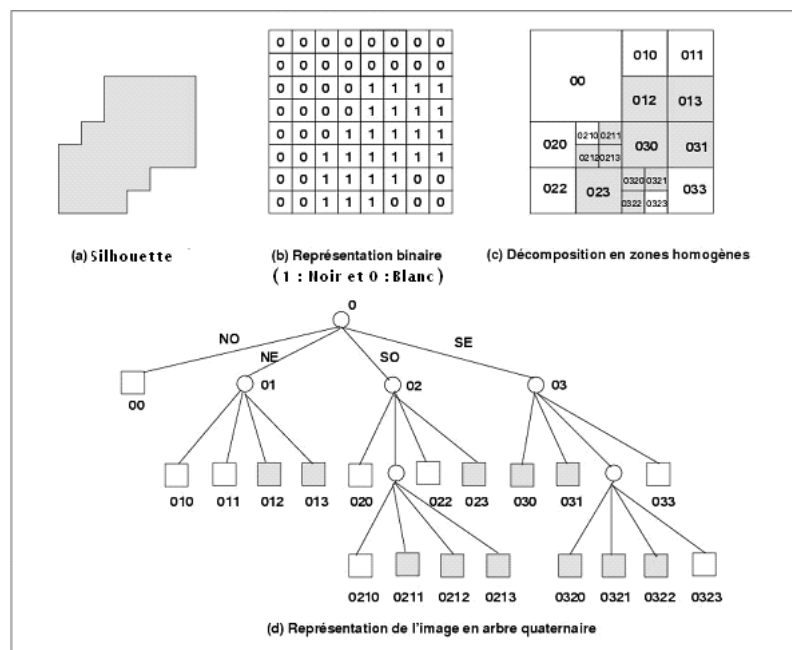


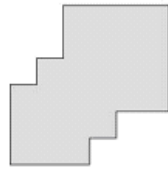
Figure 3.6 : Représentation d'une image en arbres quaternaires.

3.1.7. Codage de l'image

La méthode de stockage linéaire, que nous proposons code chaque nœud feuille **noir** par un 'N' (où il existe au moins un pixel noir dans ce quadrant), et chaque nœud feuille **blanc**, par un 'B' (où il n'existe aucun pixel noir dans le quadrant).

La hiérarchie de la structure est conservée par l'utilisation de parenthèses, une **parenthèse** '('code un nœud interne. Pour récapituler ce codage nous donnons l'exemple suivant :

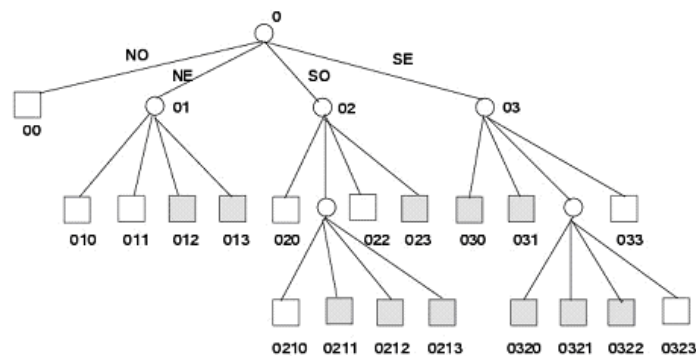
▪ *Exemple*



00		010	011
		012	013
020	0210 0211	030	031
	0212 0213		
022	023	0320 0321	033
		0322 0323	

(a) :silhouette

(b) :découpage enQuad-Tree



(c): arbre quaternaire

Figure 3.7 : image en Quad-Tree

Il existe deux manières de coder un arbre Quaternaire soit d'écrire le code en suivant la profondeur de l'arbre d'abord ou alors ou suivant la largeur de l'arbre d'abord.

• *Profondeur d'abord*

L'arbre quaternaire de l'image de la figure 3.7 est codé, via cette méthode, par la séquence:

(**B** (**BBNN** (**B** (**BNNNBN** (**NN** (**NNNBB**. La première parenthèse indique que la racine de l'arbre quaternaire est un nœud interne, l'image a été découpée en quatre quadrants principaux identifiés de 00 à 03. Le premier **B** code la couleur du nœud 00 de l'arbre quaternaire, qui est blanc. La deuxième parenthèse signifie que le nœud 01 a été découpé en quatre. La séquence **BBNN** code les couleurs des nœuds fils de 01. La troisième parenthèse indique que le nœud 02 est interne, et le **B** qui suit signifie que le nœud 020 est une feuille blanche. La séquence (**BNNNBN**, pour être comprise, doit être décomposée en trois parties:

(1) la parenthèse signifie que le noeud **021** est interne, (2) la sous séquence **BNNN** code les couleurs des nœuds fils de **021**, (3) les deux lettres restantes, **BN**, correspondent aux couleurs respectives des noeuds **022** et **023**.

- **Largeur d'abord**

L'arbre quaternaire de l'image de la figure 3.7 est codé, via cette méthode, par la séquence :

B322 BBNN B3BN NN3B BNNN NNNB. Les quatre premiers chiffres représentent le codage des nœuds du premier niveau de l'arbre quaternaire (identifiés par **00**, **01**, **02** et **03**). Le nœud identifié par **00** est blanc, il est donc codé par un **B**. Le nœud identifié par **01** est interne et tous ses fils sont feuilles, il est donc codé par un **3**. Les nœuds identifiés par **02** et par **03** sont internes et possèdent des fils qui sont eux-mêmes nœuds internes, ils sont donc codés par un **2**. La séquence **BBNN** correspond au codage des noeuds **010** à **013**. La séquence **B3BN** code les noeuds **020** à **023**, etc (De la même façon nous continuons pour le sous code correspondant au nœud **03** et ses fils).

Un travail précédent montre que le descripteur trié (ordonné) selon le niveau augmente la qualité de l'indexation (Aouat et Al. 2010).

Pour obtenir un descripteur linéaire trié par niveau, nous ajoutons au descripteur (profondeur d'abord), un numéro de niveau sur chaque nœud et chaque feuille de l'arbre. Le résultat sera ensuite trié par niveaux. Le code précédent devient alors :

(0 B1 (1 (1 (1 B2 B2 N2 N2 B2 (2 B2 N2 N2N2 (2 B2 B3 N3 N3N3N3N3N3 B3.

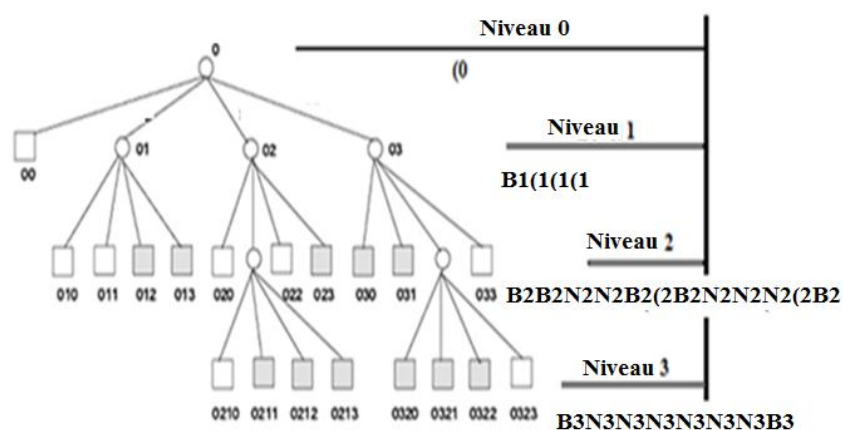


Figure 3.8 :Codification du Quadtree de la Figure 3.7 par niveaux

3.1.8. Algorithme de découpage :

Le découpage d'une image A se fait comme suit :

- La racine de l'arbre (le nœud 0) est l'image entière.
- On découpe l'image du nœud courant en quatre images égales qui représentent les quatre fils du nœud.
- On récupère les caractéristiques de chaque fils (niveau de détails, niveau d'homogénéité).
- On compare les caractéristiques des fils. Si une différence entre deux fils apparaît, alors on découpe la racine en quatre fils. Sinon, le nœud n'est pas découpé.
- On agit récursivement de la même manière sur chacun des fils.

▪ **Remarque :**

L'algorithme s'arrête quand toutes les feuilles sont homogènes (chaque quadrant possède les mêmes caractéristiques de couleurs, (noir ou bien blanc) en caractéristiques ou quand la taille de l'image du nœud devient trop petite (dernière taille égale à un pixel).

Chaque nœud représente un quadrant de l'image. On enregistre dans chaque nœud interne des informations concernant le quadrant (le taux de remplissage, le niveau du découpage) dans une structure de données bien définie car ces informations sont primordiales pour l'étape de reconnaissance.

▪ ***Seuils de découpage***

Nous définissons les conditions d'arrêt de notre algorithme comme étant l'homogénéité des feuilles (ou bien quadrant) en caractéristiques et en niveau de la taille du nœud (quadrant).

Dans notre projet nous traitons les images binaires correspondant aux silhouettes d'objets (c'est-à-dire pixel noir et pixel blanc).

Nos conditions pour le découpage est qu'on arrive à un quadrant blanc c'est-à-dire qu'il n'existe aucun pixel noir dans ce quadrant ou bien on arrive un niveau de découpage prédéfini.

Nous constituons de ces deux cas des conditions d'arrêt pour la récursivité du découpage en Quadtree.

3.2. Méthodologie utilisant les informations de courbures pour l'indexation et la reconnaissance d'objets :

3.2.1. Présentation de la méthodologie :

Le travail décrit ici consiste à rechercher l'adéquation optimum globale entre les contours de deux objets représentés dans les images, qui sont échantillonnés par nombre égal de points. Ainsi, pour déterminer l'adéquation optimum globale entre les deux points de contours, on utilise les informations de courbure, qui sont totalement invariantes aux transformations rigides.

Nous avons utilisé et implémenté la méthodologie proposée par (Oliviera, 2008), pour déterminer la correspondance entre les points de contours représentés dans une image.

Comme il est bien connu, les transformations rigides, peuvent produire des changements dans un objet, comme dans sa position, dans l'échelle ou dans son orientation, mais ils n'impliquent pas de changements dans sa forme. Une caractéristique invariante qui persiste entre deux objets, où l'on obtient l'une à partir de l'autre par une transformation rigide, est la courbure.

Etant donnée deux objets à comparer. Ces derniers sont représentés par un ensemble de points. La courbure d'un contour est définie par un angle, qui peut être estimé pour chaque collection de trois points consécutifs. Il est également important de considérer la direction du contour dont la courbure est analysée, sinon quelques cas de confusion peuvent apparaître figure3.9. Ainsi l'hypothèse que les points soient correctement ordonnés est importante.

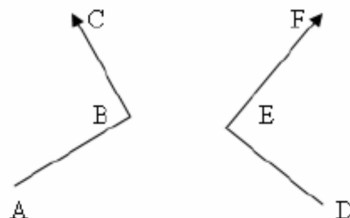


Figure3.9 : Dans la méthodologie de (Oliviera, 2008) une direction est considérée pour l'analyse d'une courbure. Par conséquent, nous avons $AB\hat{C} = 270^\circ$ et $D\hat{E}F = 90^\circ$.

Considérons qu'au point i du contour 1 est associée une amplitude d'angle α_i , et au point j du contour 2 est associée une amplitude d'angle θ_j . Il convient de noter que les séquences d'angles obtenus pour chaque contour ne dépendent que de sa forme, et non sur ses dimensions ou la position dans l'image d'origine.

L'étape suivante de la méthodologie de mise en correspondance consiste à rechercher les correspondances entre les angles des deux contours avec un coût global minimum de mise en correspondance. Autrement dit, l'objectif de cette étape est de minimiser la somme des valeurs absolues des différences angulaires entre les points appariés. Considérons deux contours à mettre en correspondance. En utilisant l'approche de (Oliviera, 2008). On obtient deux séquences d'amplitudes d'angles associées à chaque courbure des deux contours à appariées $\alpha_1, \alpha_2, \dots, \alpha_n$ et $\theta_1, \theta_2, \dots, \theta_n$.

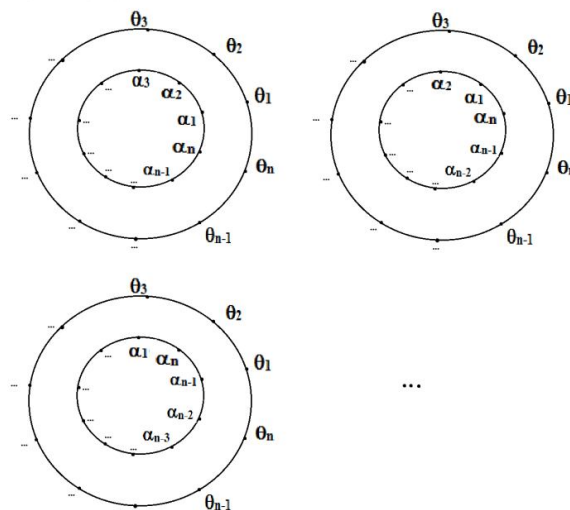


Figure 3.10 : Exemple de recherche de la meilleure mise en correspondance entre deux contours échantillonnés par un nombre égal de points.

La prochaine étape est de calculer successivement les n hypothèses de mise en correspondance, figure 3.10. Les valeurs de mise en correspondance sont données par la matrice des coûts suivante :

$$\begin{bmatrix} |\alpha_1 - \theta_1| & |\alpha_1 - \theta_2| & \dots & |\alpha_1 - \theta_n| \\ |\alpha_2 - \theta_2| & |\alpha_2 - \theta_3| & \dots & |\alpha_2 - \theta_1| \\ \dots & \dots & \dots & \dots \\ |\alpha_n - \theta_n| & |\alpha_n - \theta_1| & \dots & |\alpha_n - \theta_{n-1}| \end{bmatrix}$$

Par conséquent, la somme des éléments de la première colonne de la matrice de coûts représente le premier coût de la mise en correspondance angulaire; la somme des éléments de la deuxième colonne représente le deuxième coût de la mise en correspondance angulaire, et ainsi de suite. Avec cette approche, le meilleur appariement sera donnée par:

$$\min \left\{ \sum_{i=1}^n |\theta_i - \alpha_i| ; \sum_{i=1}^n |\theta_{i+1} - \alpha_i| ; \dots ; \sum_{i=1}^n |\theta_{i+n-1} - \alpha_i| ; \right\}$$

with $\theta_{i+n} = \theta_i$.

Ainsi, afin de pouvoir utiliser cette méthode, les points de chaque contour à appairer doivent être préalablement ordonnés, et les deux contours doivent être analysés dans une direction commune. Notez que le point i du contour 1 ne doit pas nécessairement correspondre au point i du contour 2. La seule condition qui doit être respectée est que, par exemple, si le point i du contour 1 correspond au point j du contour 2, alors le voisin successeur de i dans le contour 1 (point $i+1$) doit correspondre au voisin successeur de j dans le contour 2 (point $j+1$).

Cette restriction d'ordre réduit l'hypothèse du problème de mise en correspondance de $n!$ à n . Dans la figure 3.10, une représentation graphique de la méthodologie de mise en correspondance est effectuée. L'idée associée est la suivante: les deux contours sont chevauchés et l'un d'autre eux est mis en rotation successivement dans le but de retrouver la meilleure adéquation entre les angles de courbures.

Dans la figure 3.11, une mise en correspondance adéquate a été trouvée par la rotation d'un contour par un angle de 175 degrés.

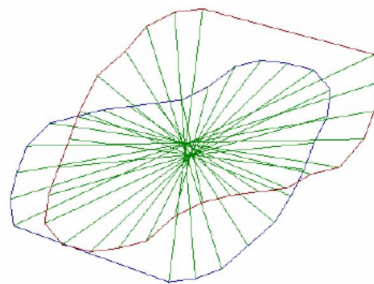


Figure 3.11 : une mise en correspondance adéquate par la rotation d'un contour par un angle de 175 degrés.

3.2.2. Algorithme d'indexation :

L'étape d'indexation est cruciale pour le bon fonctionnement d'un système de recherche d'objets.

Préparation de la base d'image :

Dans le cas de la reconnaissance d'objet en fonction de sa courbure, nous remarquons que dans la méthode que nous avons choisie, les variables dont on a besoin pour traiter la mise en correspondance de deux contours sont un ensemble d'amplitude d'angle associé en chaque point défini du contour.

Nous décidons d'indexer une image par son ensemble d'amplitude d'angle qui correspond aux courbures des contours apparaissant dans cette image.

Nous avons choisi huit points à analyser pour notre mise en correspondance.

Exemple :

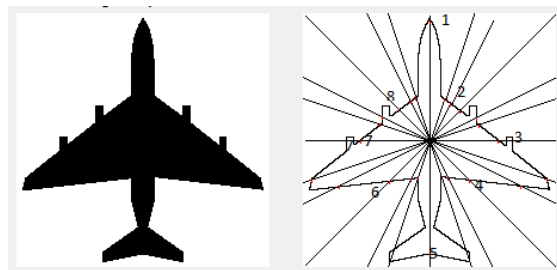


Figure 3.12 : Indexation de l'image Avion par angle de courbure en degrés, à l'image Avion correspond l'ensemble d'angles suivant {(angle 1)21, (angle2) 166, (angle 3)178, (angle 4) 179, (angle 5) 17, (angle 6)179, (angle 7)177, (angle 8) 168}.

Le code de l'image est {21, 166,178, 179, 17, 179, 177,168}.

Nous indexons toutes les images de notre base d'images dans un fichier texte indexation1.

4. Reconnaissance d'objets :

4.1. Reconnaissance par la méthode du Quadtree:

Dans l'étape *on-line*, il s'agira de mettre en place un système complet de recherche. La première étape consiste à créer une structure d'indexation de la base de données (déjà réalisé), permettant d'organiser les données de façon à ne pas les parcourir toutes lors d'une recherche. Ensuite, en se basant sur cette structure, il faudra développer une **stratégie de recherche** permettant de retrouver les images les plus proches d'une image requête donnée

Dans cette étape, le système prend une **requête** et donne le résultat correspondant à une liste d'images. La stratégie de recherche que nous devons développer doit prendre en compte les occlusions d'un objet donné dans une image requête.

4.1.1. Stratégie de recherche :

Rappel :

Les Quad-Tree, ou arbre quaternaire en français est une structure de données qui représente des arbres, dont chaque nœud a quatre fils. Il est simple de le représenter par analogie avec les arbres binaires que tout le monde connaît, la seule différence vient du fait que chaque nœud possède non pas deux mais quatre feuilles / fils.

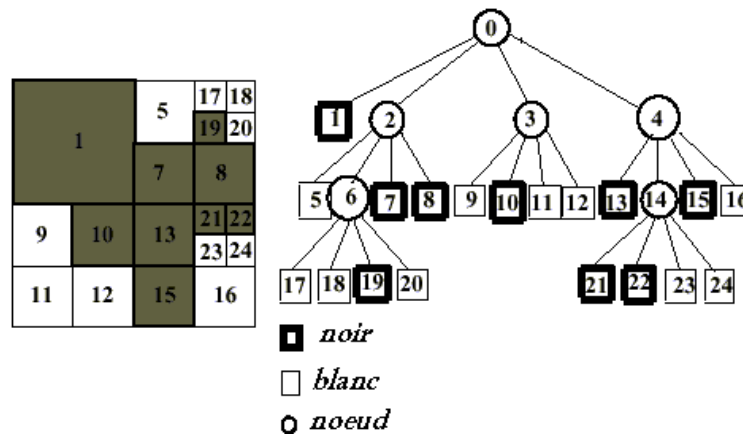


Figure 3.13 : représentation d'un Quad-Tree.

Un arbre quaternaire est une structure de données permettant de représenter des images à deux dimensions. Les images sont découpées en quadrants ou carrés réguliers selon un critère de découpage donné (l'homogénéité, la texture, etc.). Un arbre quaternaire peut être implanté de manière hiérarchique, à l'aide de pointeurs, ou de manière linéaire, sous forme d'une liste de valeurs.

Peut-on réaliser une comparaison d'une image ou un Objet est partiellement occulté et une image ou l'objet est entier avec la technique du Quadtree ?

De la définition de quadtree nous observons que la représentation d'un objet dans une image est un arbre et la représentation de ce même objet partiellement occulté est le même arbre tronqué.

Par exemple nous avons l'image de la Figure 3.13 qui est représentée par un Quadtree, nous établissons le Quadtree de la même image mais avec une occlusion du quatrième quadrant de l'image. Nous obtenons le Quadtree de la Figure 3.14, Nous réalisons que la partie de l'arbre qui a été tronquée est la quatrième feuille du niveau1 du Quadtree.

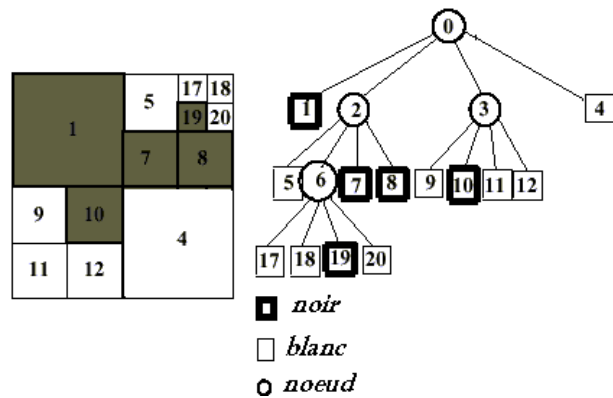


Figure 3.14 : Représentation d'une image en Quadtree.

Nous avons réalisé un système de reconnaissance d'objet partiellement occulté avec la technique du Quadtree.

Nous procédons au codage de l'image requête avec n_0 découpage et nous obtenons le descripteur linéaire trié, nous comparons le descripteur de l'objet requête avec tous les descripteurs de la base d'images en faisant appel au fichier indexation.

La comparaison est faite en vérifiant si le descripteur de l'image requête est sous descripteur de l'image de la base mais avec une **tolérance**. C'est-à-dire pour un nœud donné ; l'image de la base sur laquelle se fait la comparaison est acceptée comme image correspondante même si ce nœud est interne dans le Quadtree de l'image de la base et feuille dans le Quadtree de l'image requête.

Exemple :

Pour l'image Avion partiellement occulté dans la figure 3.15, le résultat donné par notre système avec un seuil de tolérance égale à 3 est représenté sur la figure 3.16:

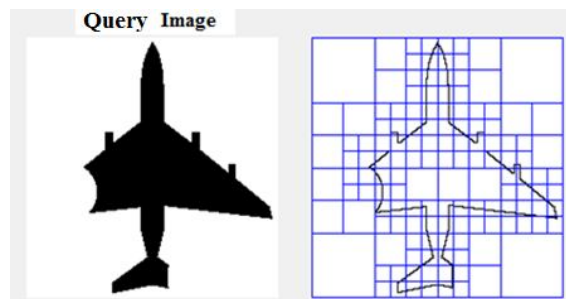


Figure 3.15: Quad-tree d'une image où l'objet est partiellement occulté.

Le résultat de notre système après la recherche dans la base d'image est:

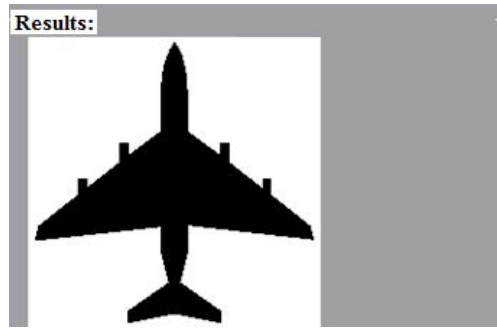


Figure 3.16: Le Modèle retrouvé pour la requête de l'image 3.15.

On a pour la figure 3.15 le descripteur linéaire suivant: (0 (1(1(1(1 B2(2(2(2 (2B2(2(2 (2(2B2(2 (2(2(2B2 B3(3B3(3 B3B3B3(3 (3(3(3(3 **B3**B3(3B3 (3(3(3(3 B3B3(3B3 (3(3(3(3 B3B3(3(3 B3(3(3(3 B3B3(3(3 **B3**(3(3(3 (3B3**B3**(3.

On a pour la figure 3.16 le descripteur linéaire égal a : (0 (1(1(1(1 B2(2(2(2 (2B2(2(2 (2(2B2(2 (2(2(2B2 B3(3B3(3 B3B3B3(3 (3(3(3(3 (**3**B3(3B3 (3(3(3(3 B3B3(3B3 (3(3(3(3 B3B3(3(3 B3(3(3(3 B3B3(3(3 (**3**(3(3(3 (3B3(**3**(3.

Nous pouvons voir que les deux descripteurs sont différents dans trois quadrants (texte en gras dans les deux descripteurs) qui correspondent au seuil de tolérance à l'occlusion qui est égal à 3, soit au nombre de différence qu'il y a entre les deux descripteurs. La différence des trois quadrants s'explique que dans la figure 3.15, le quadrant décrit par un B3 c'est-à-dire blanc est vide, c'est l'un des quadrants qui représente l'occlusion partielle de l'objet Avion. Alors que dans le descripteur de la figure 3.16 le quadrant est décrit par une parenthèse soit qu'il représente un nœud racine soit que ce quadrant n'est pas vide.

4.2. Seuil de tolérance :

Combien de nœuds peut-on accepter qu'il soit interne dans le Quadtree de l'image de la base et qu'il soit feuille dans le Quadtree de l'image requête ?(Ait-Aoudia et al., 2013a)

Après plusieurs tests effectués sur différentes images nous remarquons que le seuil de tolérance est propre à chaque image requête et au niveau auquel se trouve le nœud dans le Quadtree.

Nous avons effectué nos tests sur une base de données comportant 40 Image de base. Chaque image étant occultée par différents pourcentage d'occlusion de 10 à 70 pourcent, nous avons effectué donc des tests sur 280 images (voir chapitre 4).

La figure suivante illustre un histogramme montrant la correspondance entre le seuil de tolérance et le pourcentage d'occlusion.

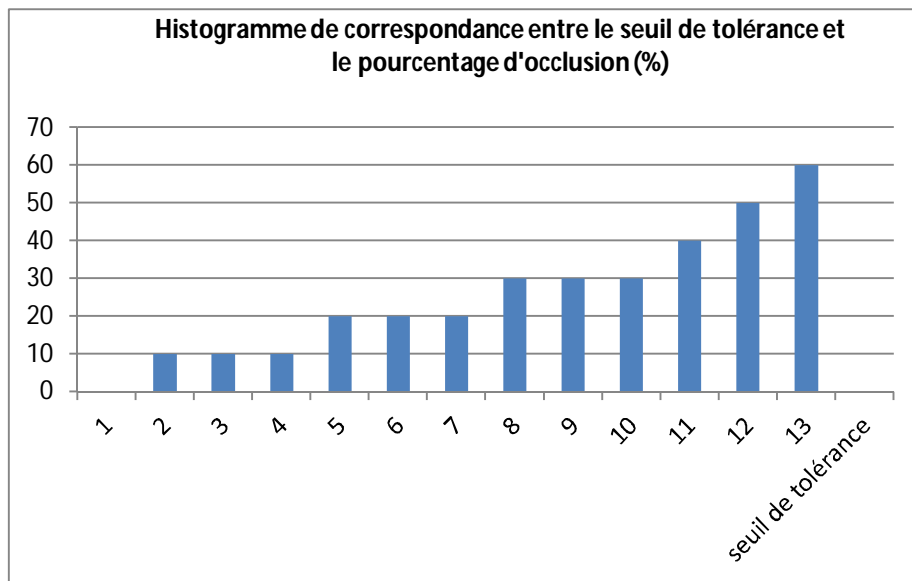


Figure 3.17 : Histogramme montrant la correspondance entre le seuil de tolérance et le pourcentage d'occlusion.

Nous remarquons qu'à chaque fois que le pourcentage d'occlusion augmente le seuil de tolérance doit augmenter pour que la reconnaissance de l'image occultée soit faite.

Nous remarquons que le système reconnaît l'objet Avion même en présence d'occlusion. Mais en approfondissant notre étude nous remarquons que la technique du Quadtree permet de reconnaître que les objets requêtes qui ont la même orientation qu'une image indexée dans la base de données, et ceci même en cas d'occlusion !

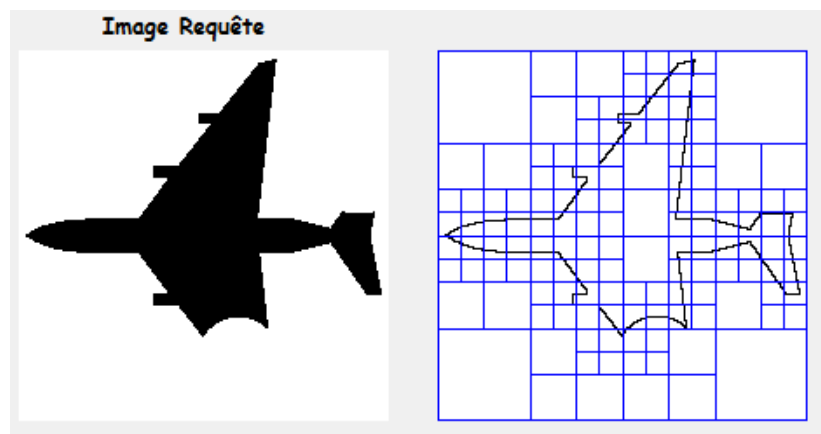


Figure 3.18 : Représentation en Quadtree.

La requête de la figure 3.15 est associée à l'image Avion. Par contre la requête de la figure 3.18 n'est associée à aucune image. Alors que c'est le même objet qui a subi une rotation.

Pour pallier à ce problème nous avons décidé d'implémenter une technique qui permet de reconnaître des objets par leurs courbures et de déduire l'angle de rotation d'un objet requête d'un objet de la base.

4.3. Etape de reconnaissance d'objet utilisant les informations de courbures :

Pour la reconnaissance d'un objet dans une image nous devons en premier lieu extraire la silhouette de l'objet de l'image A, ensuite mettre en correspondance les courbures de l'image requête avec chacune des images de la base. L'Algorithme suivant décrit la technique de mise en correspondance. (Ait-Aoudia et al., 2013b)

Début

Soit une image requête A ;

Extraction de la silhouette de l'objet de l'image A ;

Calculer l'ensemble d'amplitude d'angle au points i de l'objet; $i \in [1, n]$;

Pour chaque image B de la base **faire**

Début

Extraire le code de l'image du fichier texte Indexation1 ;

Etablir la matrice des coûts ;

Pour chaque colonne de la matrice ;

Début

 Calculer la somme des éléments ;

 Sauvegarder la somme dans un Tableau Tab ;

Finpour

Calculer le minimum du Tableau Tab ;

Si le minimum tend vers zéro **alors**

 Afficher l'image B comme image correspondante à l'image requête A ;

Finsi

Finpour

Fin

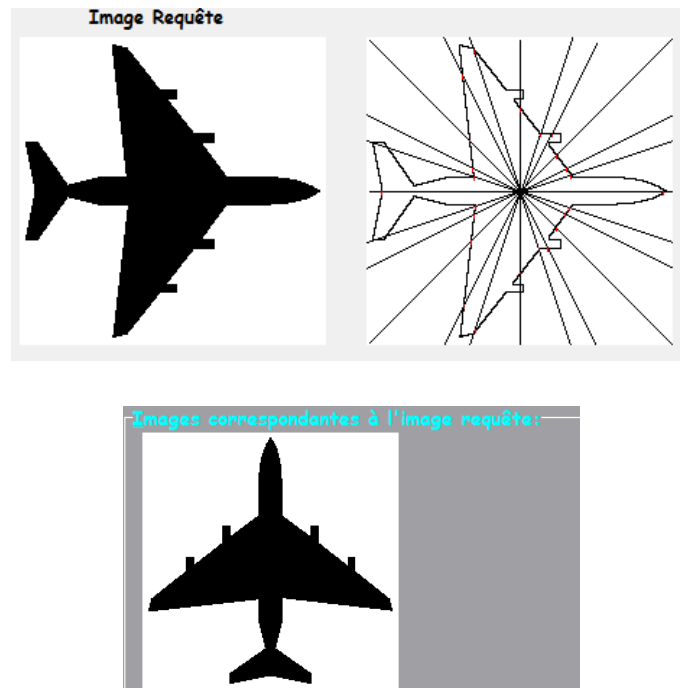


Figure 3.19 : Exemple de résultats pour l'image Avion avec une rotation de 90° degré de notre Système.

4.4. Combinaison de la technique utilisant l'angle de courbure et la technique du Quadtree :

La présence d'occlusion d'objet dans de nombreuses images requête de la vie réelle, incitent à trouver un système adéquat pouvant reconnaître les objets partiellement occultés dans ces dernières.

Lors l'utilisation de la technique du Quadtree pour reconnaître des objets partiellement occultés, nous avons remarqué que la technique est rapide efficace pour cette reconnaissance. Mais un problème est apparu lorsque l'orientation d'un objet changé. C'est-à-dire qu'un objet partiellement occulté est reconnu lorsqu'il a la même orientation que l'image indexé dans la base ; mais ce même objet n'est pas reconnu quand il avait une orientation différente. Pour pallier à ce problème nous avons décidé d'avoir recours à une technique de reconnaissance d'objet par leurs courbure, qui nous permet d'avoir l'angle de rotation d'un objet requête avec l'objet qu'il lui correspond dans la base (si celui-ci existe bien sûr).

Nous mettons en œuvre en premier lieu la technique de courbure, pour reconnaître un objet occulté et avoir un angle de rotation R qui correspond à l'adéquation de l'objet requête avec l'objet présent dans la base. Ensuite nous appliquons la technique du Quadtree sur l'objet requête transformé par la rotation R .

4.4.1. Indexation des images de la base :

On effectue deux traitements sur les images. Le premier est pour retrouver l'ensemble d'amplitude de l'angle au point $i, i \in [1, n]$; i étant un point de la silhouette de l'objet dans l'image, et le deuxième pour retrouver le code linéaire trié en utilisant le Quadtree.

Les images de notre base sont indexées dans un fichier texte. On a pour chaque image deux codes : le premier est le code linéaire trié et le deuxième code est l'ensemble d'amplitude de l'angle au point $i, i \in [1, n]$; i étant un point de la silhouette de l'objet dans l'image.

4.4.2. La reconnaissance d'objet :

Pour une image requête, nous calculons l'ensemble d'amplitude de l'angle au point $i, i \in [1, n]$; i étant un point de la silhouette de l'objet dans l'image. Nous effectuons la recherche sur notre base par la technique utilisant l'information de courbures; dès que nous avons un résultat, nous effectuons l'extraction du code linéaire trié par la méthode du Quadtree et enfin nous effectuons une recherche avec la combinaison des deux méthodes.

La figure 3.20 montre le résultat obtenu par notre système après l'application des deux techniques.

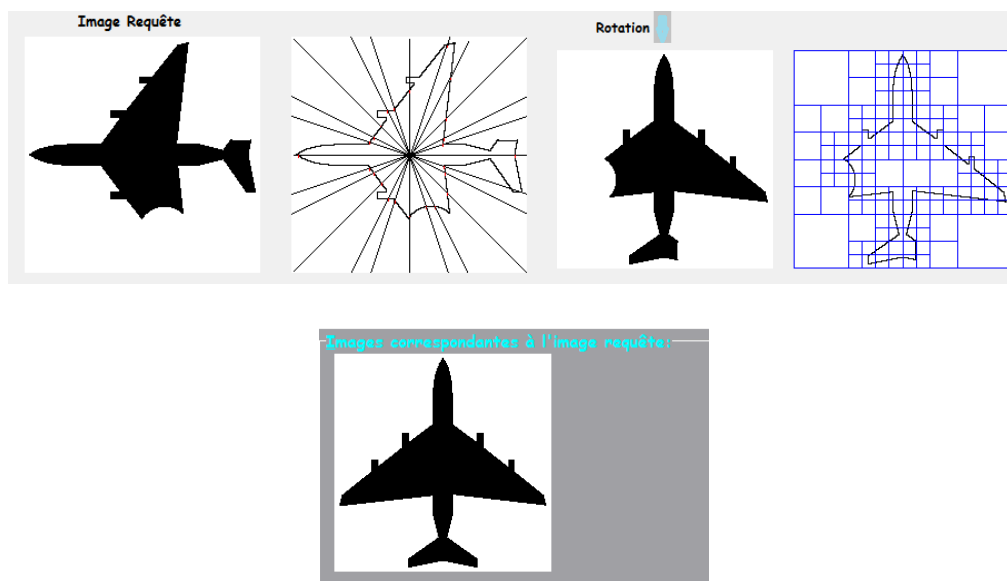


Figure 3.20 : Résultat obtenu par notre système après l'application des deux techniques utilisant l'information sur la courbure et le Quadtree.

5. Généralisation de la méthode du Quadtree:

Dans ce qui suit, nous avons généralisé la méthode d'Indexation et de reconnaissance d'objets en présence d'occlusion que nous avons construite dans ce projet. La méthode

établie se base sur la technique du Quadtree pour l'indexation des objets. Il se trouve que la technique du Quadtree est une méthode que nous avons pu adapter à l'indexation et la reconnaissance des objets partiellement occultés.

Nous avons décidé d'illustrer notre généralisation par un exemple et de commencer la description de la généralisation étape par étape par un exemple.

5.1. Exemple pour la généralisation de la méthode du Quadtree:

Voici un schéma de découpage d'une image A (figures 3.21 et 3.22).

Dans la figure 3.23 l'image A, a été partiellement occultée, les parties a4, a12, a13 ont été occultés l'Image A devient alors l'image B de la figure 3.24.

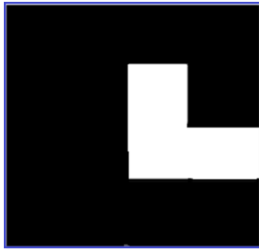


Figure 3.21 : Image A.

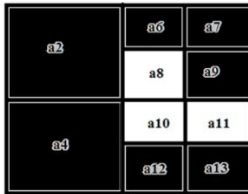


Figure 3.22: Image(A) non occultée

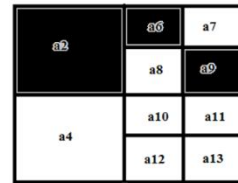


Figure 3.23 : Image (A) occulté



Figure 3.24 : Image B

La représentation Quadtree de l'image A est la suivante :

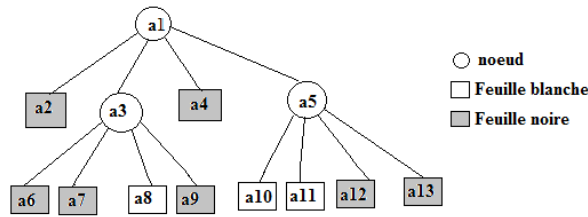


Figure 3.25 : Représentation en Quadtree de l'image A.

La représentation Quadtree de l'image B est la suivante :

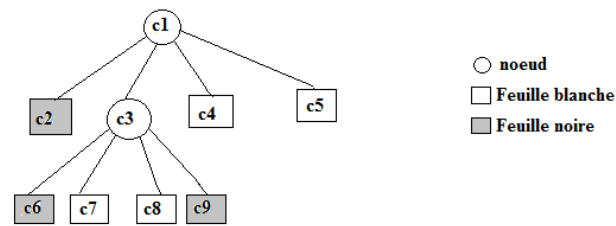


Figure 3.26 : Représentation en Quadtree de l'image A.

Le code linéaire trié de l'image A est :

A: {a1 . a2 . a3 . a4 . a5 . a6 . a7 . a8 . a9 . a10 . a11 . a12 . a13}.

A : (0 N1 (1 N1 (1 N2 N2 B2 N2 B2 B2 N2 N2.

Où: a1 ='(' a2='N' a3='(' a4='N' a5='(' a6='N' a7='N' a8='B' a9='N' a10='B' a11='B'
a12='N' a13='B'

Le code linéaire trié de l'image B est:

B : {c1 . c2 . c3 . c4 . c5 . c6 . c7 . c8 . c9}

B : (0 N1 (1 B1 B1 N2 B2 B2 N2.

Où : c1='(' c2='N' c3='(' c4='B' c5='B' c6='N' c7='B' c8='B' c9='N'.

Nous remarquons de ces deux codes linéaire trié qu'il y a une correspondance entre le code linéaire trié de l'image A et le code linéaire trié de l'image B qui est l'image A partiellement occultée.

$$B : \{a1 . a2 . a3\} \cup \{'B'+'B'\} + \{a6\} \cup \{'B'\} + \{a8 . a9\} \quad \dots (1)$$

Nota: on fait abstraction dans la correspondance du niveau parce que le code linéaire des deux images est déjà trié et on compare chaque nœud ou feuille de chaque niveau de l'image avec le même nœud ou feuille du même niveau dans l'image B.

Nous remarquons que le code linéaire trié de l'image partiellement occulté s'écrit en fonction de l'image non occultée en groupe de a_i suivi d'un blanc ou d'un groupe de blanc qui représente l'occlusion. Nous pouvons imaginer généraliser cette écriture sous la forme suivante :

$$\{\sum a_i \cup \{ 'B' + \dots + 'B' \} \} + \{\sum a_j \cup \{ 'B' + \dots + 'B' \} \} + \{\sum a_k \cup \{ 'B' + \dots + 'B' \} \} + \dots \quad (2)$$

$$1^{\text{er}} \text{ groupe} \quad + \quad 2^{\text{ème}} \text{ groupe} \quad + \dots \dots \dots$$

Nous remarquons de l'exemple précédent (1) que l'indice j du deuxième groupe de la formule est en relation avec l'indice i du premier groupe, l'indice k du troisième groupe est en relation avec l'indice j du deuxième groupe et ainsi de suite.

De (1), le premier indice j du deuxième groupe est égale au dernier indice i du premier groupe plus le nombre de blanc qu'il y a dans le premier groupe plus 1.

De ceci, nous pouvons réécrire la formule (2) par :

$$\left\{ \sum_{i=1}^f (a_i) \cup \sum_{p=0}^{m_1} \{ 'B' \} \right\} + \left\{ \sum_{i=f+m_1+1}^d (a_i) \cup \sum_{p=0}^{m_2} \{ 'B' \} \right\} +$$

$$\left\{ \sum_{i=d+m_2+1}^g (a_i) \cup \sum_{p=0}^{m_3} \{ 'B' \} \right\} + \dots \dots \dots (3)$$

Nous remarquons qu'il y a une relation entre les indices f, d et g, nous pouvons généraliser la formule (3) en changeant la suite f, d, g par f_1, f_2, f_3 , jusqu'à arriver a_n . Nous obtenons la formule suivante :

$$\left\{ \sum_{i=1}^{f_0} (a_i) \cup \sum_{p=0}^{m_0} \{ 'B' \} \right\} + \left\{ \sum_{i=f_0+m_0+1}^{f_1} (a_i) \cup \sum_{p=0}^{m_1} \{ 'B' \} \right\} +$$

$$\left\{ \sum_{i=f_1+m_1+1}^{f_2} (a_i) \cup \sum_{p=0}^{m_2} \{ 'B' \} \right\} + \dots + \left\{ \sum_{i=f_{n-1}+m_{n-1}+1}^{f_n} (a_i) \cup \sum_{p=0}^{m_n} \{ 'B' \} \right\} \quad (4)$$

De la formule (4), on peut conclure que :

$$\sum_{i=1}^k (c_i) = \sum_{n=0}^s \left\{ \sum_{i=f_{n-1}+m_{n-1}+1}^{f_n} (a_i) \cup \sum_{p=0}^{m_n} \{ 'B' \} \right\}$$

$$\text{Avec } \begin{cases} n = 0, i = 1; \\ 2 \leq f_0 \leq k - 1; \\ f_{n-1} \leq f_n \leq k - 1; \\ \text{et} \\ c_1 = a_1 \end{cases} \quad (5)$$

Où $\{a_1, a_2, \dots, a_l\}$ le code linéaire trié d'une image indexé dans la base non occultée, et $\{c_1, c_2, \dots, c_k\}$ Le code linéaire trié par niveau d'une image requête partiellement occultée, la relation entre ces deux codes linéaires triés si l'image partiellement occultée correspond à l'image indexée dans la base est formulé par (5).

5.2. Démonstration par récurrence de la Généralisation :

$$\sum_{i=1}^k (c_i) = \sum_{n=0}^s \left\{ \sum_{i=f_{n-1}+m_{n-1}+1}^{f_n} (a_i) \cup \sum_{p=0}^{m_n} \{ 'B' \} \right\}$$

$$\text{Avec } \begin{cases} n = 0, i = 1; \\ 2 \leq f_0 \leq k - 1; \\ f_{n-1} \leq f_n \leq k - 1; \\ \text{et} \\ c_1 = a_1 \end{cases} \quad (6)$$

Pour $n=1$;

La formule (6) devient :

$$\sum_{i=1}^k (c_i) = \left\{ \sum_{i=1}^{f_0} (a_i) \cup \sum_{p=0}^{m_0} \{ 'B' \} \right\} + \left\{ \sum_{i=f_0+m_0+1}^{f_1} (a_i) \cup \sum_{p=0}^{m_1} \{ 'B' \} \right\}$$

Exemple : pour $f_0=2$; $m_0=1$; $f_1=6$; $m_1=2$; et on a : pour $n=0$; $i=1$;

$$\sum_{i=1}^k (c_i) = a_1 + a_2 \cup B + a_4 + a_5 + a_6 \cup B + B$$

Pour n :

$$\sum_{i=1}^k (c_i) = \sum_{n=0}^s \left\{ \sum_{i=f_{n-1}+m_{n-1}+1}^{f_n} (a_i) \cup \sum_{p=0}^{m_n} \{ 'B' \} \right\}$$

Pour n+1 :

$$\sum_{i=1}^k (c_i) = \sum_{n=0}^s \left\{ \sum_{i=f_n+m_n+1}^{f_{n+1}} (a_i) \cup \sum_{p=0}^{m_{n+1}} \{ 'B' \} \right\}$$

... (7)

La formule (7) se décompose ainsi :

$$\begin{aligned} & \sum_{i=1}^k (c_i) \\ &= \left\{ \sum_{i=1}^{f_0} (a_i) \cup \sum_{p=0}^{m_0} \{ 'B' \} \right\} + \left\{ \sum_{i=f_0+m_0+1}^{f_1} (a_i) \cup \sum_{p=0}^{m_1} \{ 'B' \} \right\} + \left\{ \sum_{i=f_1+m_1+1}^{f_2} (a_i) \cup \sum_{p=0}^{m_2} \{ 'B' \} \right\} \\ &+ \dots + \left\{ \sum_{i=f_{n-1}+m_{n-1}+1}^{f_n} (a_i) \cup \sum_{p=0}^{m_n} \{ 'B' \} \right\} + \left\{ \sum_{i=f_n+m_n+1}^{f_{n+1}} (a_i) \cup \sum_{p=0}^{m_{n+1}} \{ 'B' \} \right\} \end{aligned}$$

$$\begin{aligned} \sum_{i=1}^k (c_i) &= \sum_{n=0}^s \left\{ \sum_{i=f_{n-1}+m_{n-1}+1}^{f_n} (a_i) \cup \sum_{p=0}^{m_n} \{ 'B' \} \right\} + \\ & \left\{ \sum_{i=f_n+m_n+1}^{f_{n+1}} (a_i) \cup \sum_{p=0}^{m_{n+1}} \{ 'B' \} \right\} \end{aligned}$$

.... (8)

On remplace l'indice $n+1$ par m ; Et $n+1$ par l ; ainsi que n par $l-1$.

La formule 8 devient :

$$\sum_{i=1}^k (c_i) = \sum_{n=0}^s \left\{ \sum_{i=f_{l-2}+m_{l-2}+1}^{f_{l-1}} (a_i) \cup \sum_{p=0}^{m_{l-1}} \{ 'B' \} \right\} + \left\{ \sum_{i=f_{l-1}+m_{l-1}+1}^{f_l} (a_i) \cup \sum_{p=0}^{m_l} \{ 'B' \} \right\}$$

$$= \sum_{n=0}^s \left\{ \sum_{i=f_{l-1}+m_{l-1}+1}^{f_l} (a_i) \cup \sum_{p=0}^{m_l} \{ 'B' \} \right\} \dots (9)$$

La formule 9 est vraie.

6. Conclusion :

Dans l'étape d'indexation, l'implantation linéaire des arbres quaternaires a été proposée pour permettre un stockage efficace des images organisées en arbre quaternaire, sans occuper la mémoire par les pointeurs du stockage hiérarchique.

Ce type de stockage permet également de transmettre plus rapidement et à moindre coût des images sur un réseau (*téléchargement*).

La technique du Quadtree a montré un problème lors de son application sur des objets partiellement occultés, nous avons décidé de palier à ce problème en utilisant la technique de mise en correspondance utilisant les informations de courbure.

Les images de la base ont été indexées à l'aide de deux codes : le code linéaire trié et l'ensemble d'amplitude d'angle des courbures de l'objet dans l'image.

Le système que nous avons mis en place combine deux méthodes Quadtree et mise en correspondance en utilisant les informations de courbures pour la reconnaissance d'objets partiellement occultés. Nous avons décrit précédemment la généralisation de cette méthode à une multitude d'objets. Cette généralisation nous a permis de démontrer que la relation entre le code du quadtree occulté avec celui non occulté est correcte quel que soit le niveau ou l'endroit de l'occlusion dans l'image.

Dans le chapitre 4, nous allons présenter l'expérimentation que nous avons établie sur notre méthode.

Chapitre 4

Expérimentation

Chapitre 4 : Expérimentation.

1. Introduction :

Nous présentons dans ce chapitre, une description du fonctionnement de notre logiciel de reconnaissance d'objets par leurs silhouettes en présence d'occlusion.

Nous avons réalisé ce logiciel à l'aide du langage C++, avec l'outil de développement Borland C++ Builder version 6. C++ Builder 6 offre l'environnement de développement visuel que des milliers de développeurs C++ recherchent pour créer leurs applications.

Nous avons choisi cet outil car il permet de créer des interfaces interactives pour notre application et de façon rapide.

Nous avons travaillé sur une base d'une trentaines d'images de synthèse et nous avons utilisé quelques images construites par B. Leibe et B. Schiele [Leibe et Al. 2003] .

Nous allons présenter dans ce qui suit les interfaces que nous avons décidé de mettre en œuvre de manière à ce que notre application soit la plus interactive possible avec l'utilisateur.

Interface principale :

Dans la Fenêtre principale (Figure1), nous avons mis en place un menu où il y a les fonctionnalités d'un menu classique.

Dans l'onglet Fichier nous avons l'onglet Ouvrir et Quitter.

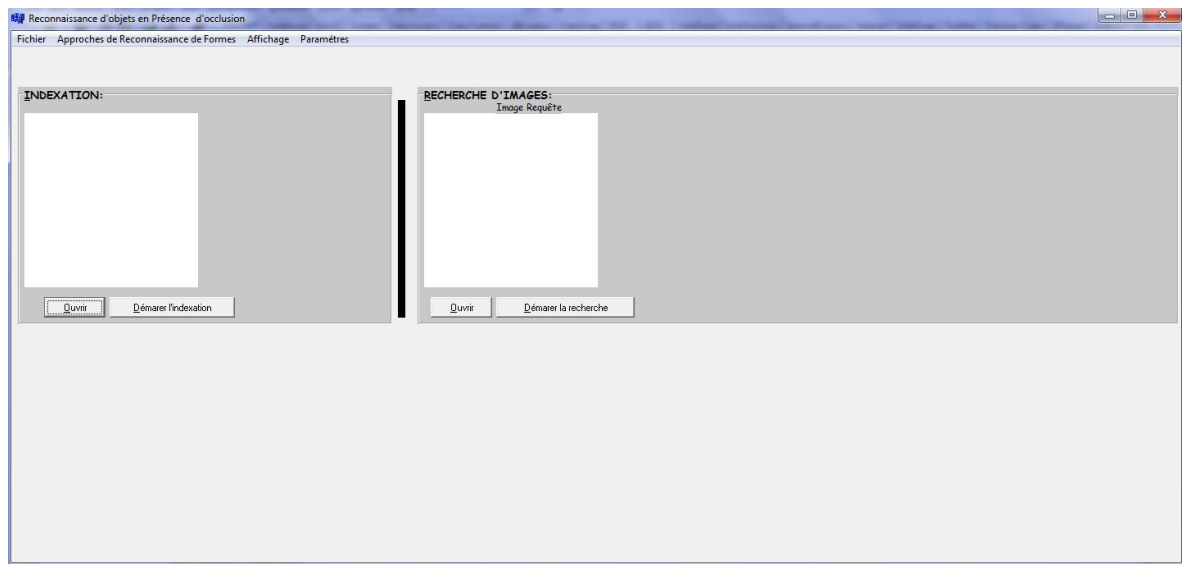


Figure 4.1 : Interface principale.

Dans l'onglet Approches de reconnaissance de Formes, nous donnons le choix à l'utilisateur de sélectionner soit l'approche de Quadtree ou l'approche de reconnaissance de formes avec les informations sur les courbures de silhouette, ou bien l'approche Hybride qui combinent les deux méthodes.

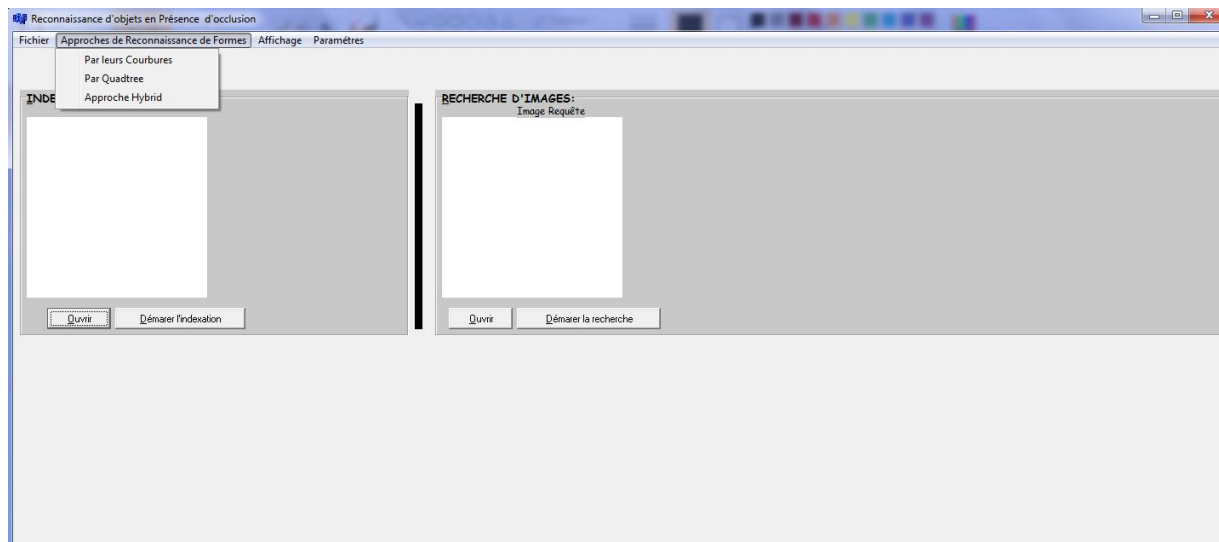


Figure 4.2 : Interface principale, Affichage du menu principale.

Dans l'onglet Affichage nous pouvons intervenir sur l'affichage de l'interface principale, effacer... etc.

Dans l'onglet paramètres, nous pouvons donner un niveau de découpage pour la méthode du Quadtree et de l'approche hybride de notre application.

Une caractéristique invariante qui persiste entre deux objets est l'information des courbures.

On montre sur la figure 4.9, les deux objets devant être appariés représentés par un ensemble de points, les informations de la courbure le long du contour de l'objet peuvent être estimées en utilisant l'angle défini, pour chaque ensemble de trois points consécutifs.

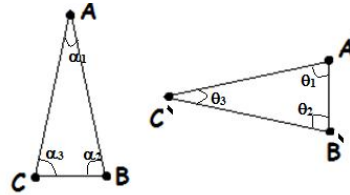


Figure 4.9 : Deux configurations de contours différents qui sont définis par le même ensemble de points.

Dans cette approche, nous avons considéré que pour le point i du contour 1 est associé l'amplitude de l'angle α_i et au point j du contour 2 correspond l'amplitude d'angle θ_j .

Nous considérons que les deux contours a appariés sont échantillonnés par n points. Grâce à notre approche, nous obtenons deux séquences d'amplitudes angulaires associés à la courbure le long de chaque contour à mettre en correspondance:

$\alpha_1, \alpha_2, \dots, \alpha_n$. et $\theta_1, \theta_2, \dots, \theta_n$.

L'étape suivante consiste à tester les n hypothèses de mise en correspondance, de sauvegarder les valeurs correspondantes dans le tableau suivant:

$$\begin{bmatrix} |\alpha_1 - \theta_1| & |\alpha_1 - \theta_2| & \dots & |\alpha_1 - \theta_n| \\ |\alpha_2 - \theta_2| & |\alpha_2 - \theta_3| & \dots & |\alpha_2 - \theta_1| \\ \dots & \dots & \dots & \dots \\ |\alpha_n - \theta_n| & |\alpha_n - \theta_1| & \dots & |\alpha_n - \theta_{n-1}| \end{bmatrix}$$

Par conséquent, la somme des éléments de la première colonne de la matrice représente le coût angulaire de la première mise en correspondance globale; la somme des éléments de la deuxième colonne représente le coût angulaire de la deuxième mise en correspondance globale, et ainsi de suite, et le meilleur correspondant sera donné par:

$$\min \left\{ \sum_{i=1}^n |\theta_i - \alpha_i| ; \sum_{i=1}^n |\theta_{i+1} - \alpha_i| ; \dots ; \sum_{i=1}^n |\theta_{i+n-1} - \alpha_i| ; \right\}$$

avec $\theta_{i+n} = \theta_i$.

Exemple: Nous avons choisi d'analyser huit points pour notre indexation:

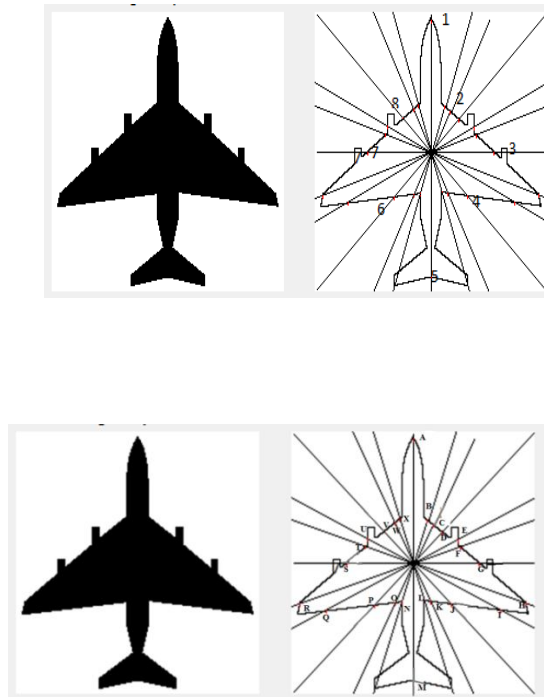


Figure 4.10: L'indexation de l'image Avion par les angles de courbures de la silhouette de l'objet Avion.

Dans la figure 4.10, l'angle au point 1 qui correspond à α_1 dans la matrice est l'angle \widehat{xab} égal à 21 degrés. L'angle au point 2 qui correspond à α_2 dans la matrice est l'angle \widehat{cde} égal à 166 degrés. L'angle au point 3 qui correspond à α_3 dans la matrice est l'angle \widehat{fgh} égal à 178 degrés. L'angle au point 4 qui correspond à α_4 dans la matrice est l'angle \widehat{ijk} égal à 179 degrés. L'angle au point 5 qui correspond à α_5 dans la matrice est l'angle \widehat{lmn} égal à 17 degrés. L'angle au point 6 qui correspond à α_6 dans la matrice est l'angle \widehat{opq} égal à 179 degrés. L'angle au point 7 qui correspond à α_7 dans la matrice est l'angle \widehat{rst} égal à 177 degrés. L'angle au point 8 qui correspond à α_8 dans la matrice est l'angle \widehat{uvw} égal à 168 degrés.

L'ensemble des angles qui correspond à l'image Avion est: {(angle 1)21, (angle 2) 166, (angle 3)178, (angle 4) 179, (angle 5) 17, (angle 6)179, (angle 7)177, (angle 8) 168}.

Le code de l'image Avion est: {21, 166,178, 179, 17, 179, 177, et 168}.

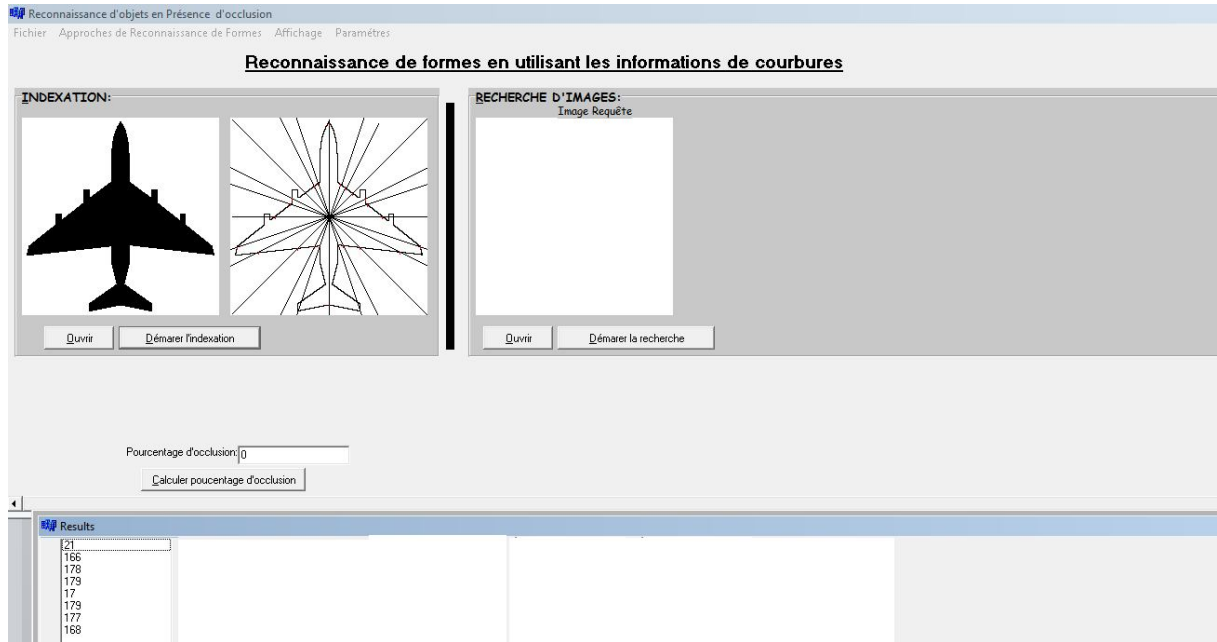


Figure 4.11 : Indexation de l'objet avion par la méthode utilisant la technique utilisant les informations de courbures pour l'indexation et la reconnaissance d'objets.

Dans la fenêtre « Results » qui apparaît nous obtenons l'ensemble des angles qui définissent l'image Avion. Nous avons indexés toutes les images de notre base dans un fichier d'indexation texte.

4. Quelques résultats d'Expérimentation :

4.1 Donnée de Synthèse :

Pour l'expérimentation de notre méthode de reconnaissance et d'indexation, nous avons utilisé les images de synthèse (Ikeskinen, 2010) montrés sur la figure

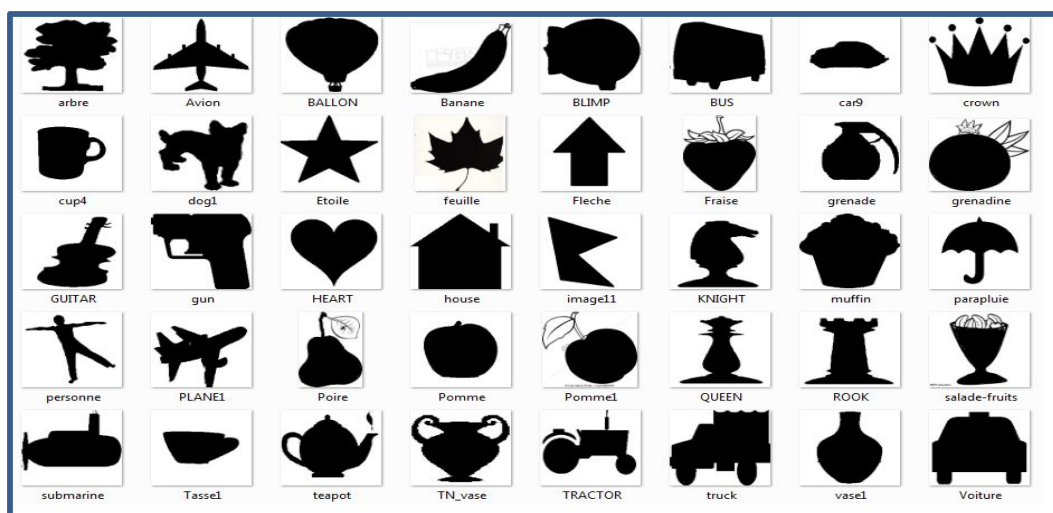


Figure 4.12 : Images de synthèse utilisée pour l'expérimentation de notre logiciel.

4.1.1. Reconnaissance des objets :

Notre but principal dans ce projet est d'arriver à établir une reconnaissance d'objets partiellement occulté pour cela nous avons fait appel à la méthode du Quadtree plus une deuxième méthode se basant sur les informations de courbures de l'objet. Nous avons défini dans ce projet ce que l'on nomme seuil de tolérance, nous donnons dans ce qui suit des résultats de notre application par rapport à ce seuil d'occlusion.

Rappel :

- **Seuil de tolérance :**

Combien de nœuds peut-on accepter qu'il soit interne dans le Quadtree de l'image de la base et qu'il soit feuille dans le Quadtree de l'image requête ?

Après plusieurs tests effectués sur différentes images nous remarquons que le seuil de tolérance est propre à chaque image requête et est lié au pourcentage d'occlusion de l'image requête.

Nous avons effectué nos tests sur une base de données comportant 40 Image de base. Chaque image étant occultée par différents pourcentage d'occlusion de 10 à 70 pourcent nous avons effectué donc des tests sur 280 images.

Histogramme montrant la correspondance entre le seuil de tolérance et le pourcentage d'occlusion.

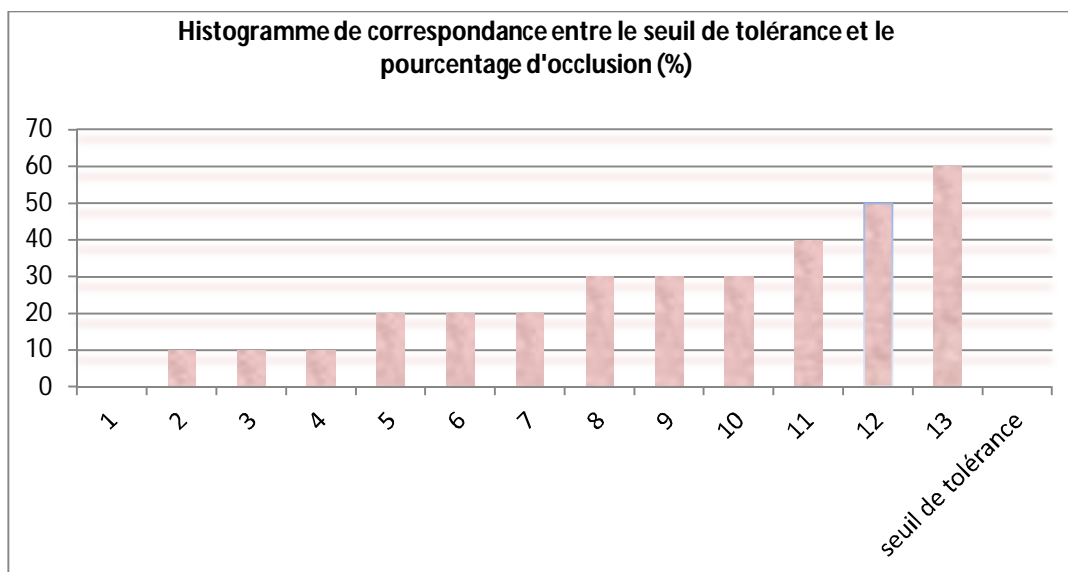


Figure 4.13 : Histogramme montrant la correspondance entre le seuil de tolérance et le pourcentage d'occlusion.

Nous remarquons qu'à chaque fois que le pourcentage d'occlusion augmente le seuil de tolérance doit augmenter pour que la reconnaissance de l'image occultée soit faite.

Nous montrons dans le graphe suivant le pourcentage de reconnaissance de notre méthode par rapport au pourcentage d'occlusion des images requêtes. Les tests ont été effectués sur 280 Image.

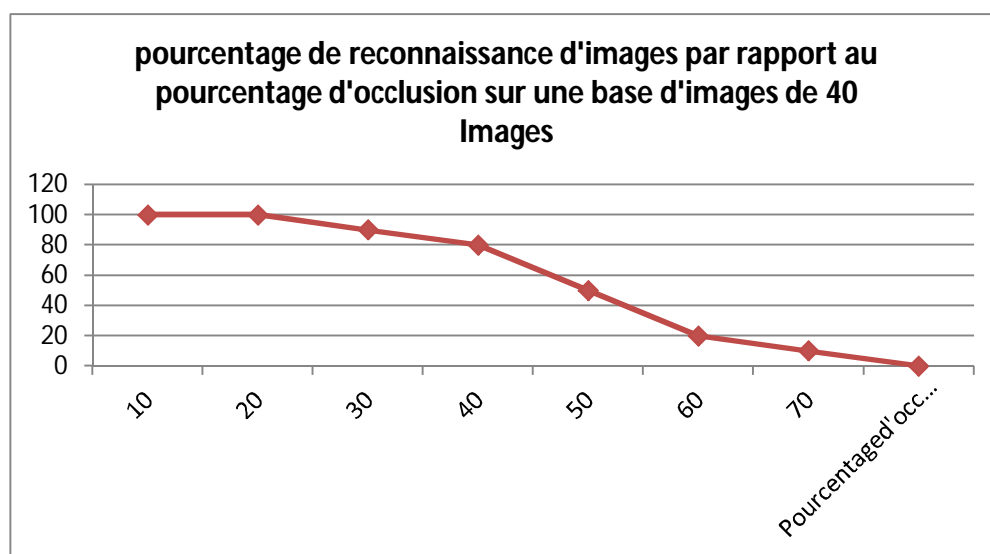


Figure 4.14 : Pourcentage de reconnaissance d'image par rapport au pourcentage d'occlusion sur une base de 40 images.

La figure 4.15 montre le résultat obtenu par notre système après l'application des deux techniques avec un seuil de tolérance égal à 4 (ce qui correspond d'après le diagramme de la figure 4.31 à 10% d'occlusion) et la figure 4.16 avec un seuil de tolérance à l'occlusion égal à 3 (ce qui correspond 20% d'occlusion).

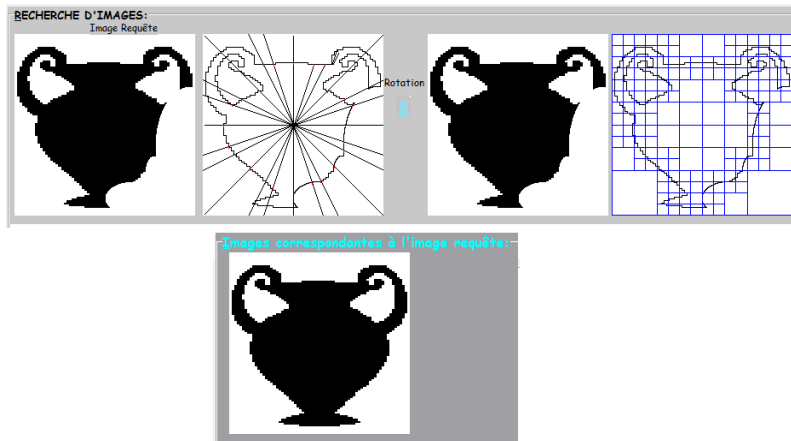


Figure 4.15. Image Vase

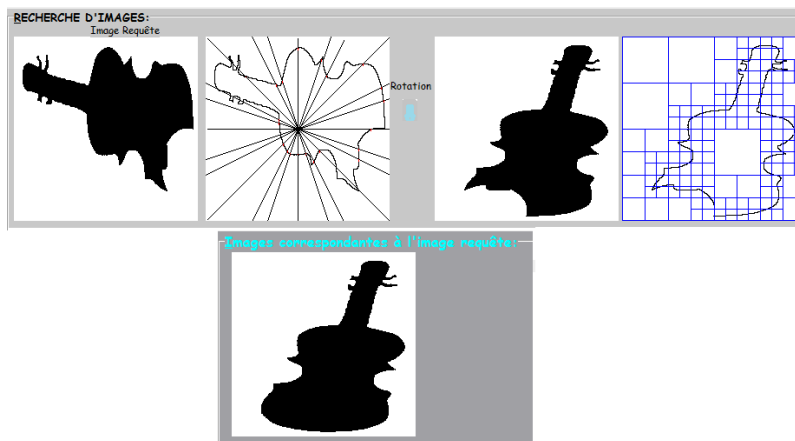

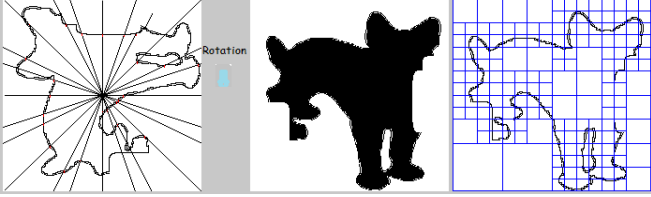


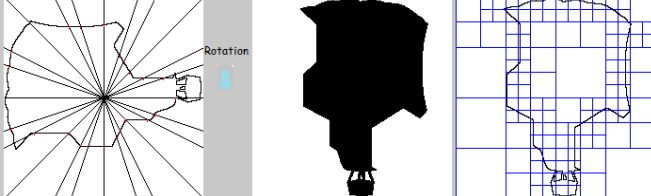


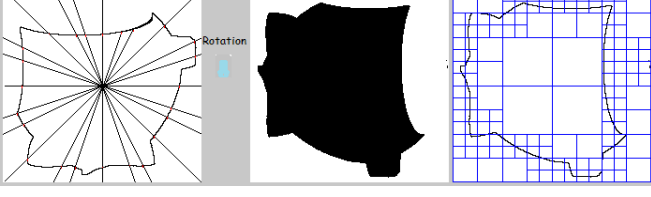











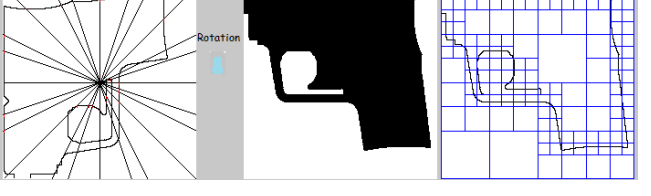






Figure 4.16 : Image Guitare.

Objet requête	Recherche d'une image correspondante dans la base de donnée	Objet retrouvé
 <p>(1)</p>		
 <p>(2)</p>		
 <p>(3)</p>		
 <p>(4)</p>		
 <p>(5)</p>		
 <p>(6)</p>		
 <p>(7)</p>		
 <p>(8)</p>		

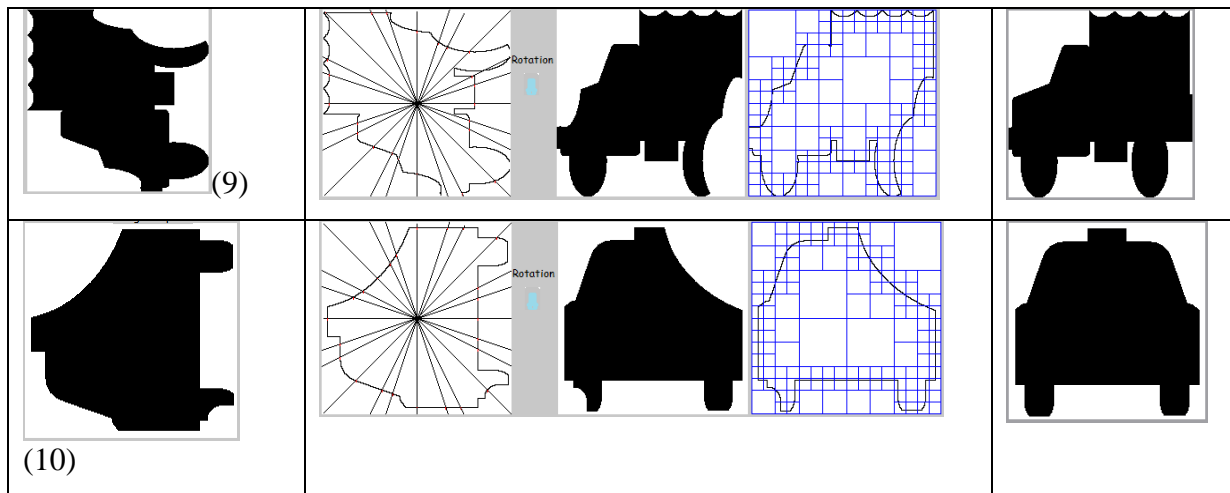


Figure 4.17 : Quelques résultats d'expérimentation.

Résultat obtenu par notre système après l'application des deux techniques utilisant l'information sur la courbure et le Quadtree.

Nous montrons dans la figure 4.17, d'autres expérimentations avec différents objets de requête et avec différentes valeurs de seuil de tolérance et le pourcentage d'occlusion.

De la ligne 1 à la ligne 10 de la figure 4.16 nous avons 10 images requêtes: un chien, un ballon, un dirigeable, un arbre, un bus, une tasse, un pistolet, une tour, un camion et un arbre. Le seuil de tolérance et le pourcentage d'occlusion de chaque requête sont donnés respectivement par les paires de valeurs suivantes: (3, 10%), (8, 30%), (3, 20%), (5, 10%), (5, 10%), (6, 10%), (2, 13%), (5, 15%), (5, 10%), (6, 10%).

A partir de ces résultats, nous pouvons déduire que jusqu'à 30% de l'occlusion et également en présence de rotation, notre système peut reconnaître correctement l'objet de requête.

4.2 Images réelles :

Nous utilisons quelques images construites par B. Leibe et B. Schiele [Leibe et Al. 2003] où chaque objet est représenté par 41 vues différentes. Nous utiliserons quelques images de la base de B. Leibe et B. Schiele.

La base de Leibe [Leibe et Al. 2003] est constituée de 80 objets (10 pommes, 10 vaches, etc..) dont les types sont montrés dans la figure 4.18. Pour chacun de ces 80 objets, 41 vues différentes sont obtenues.

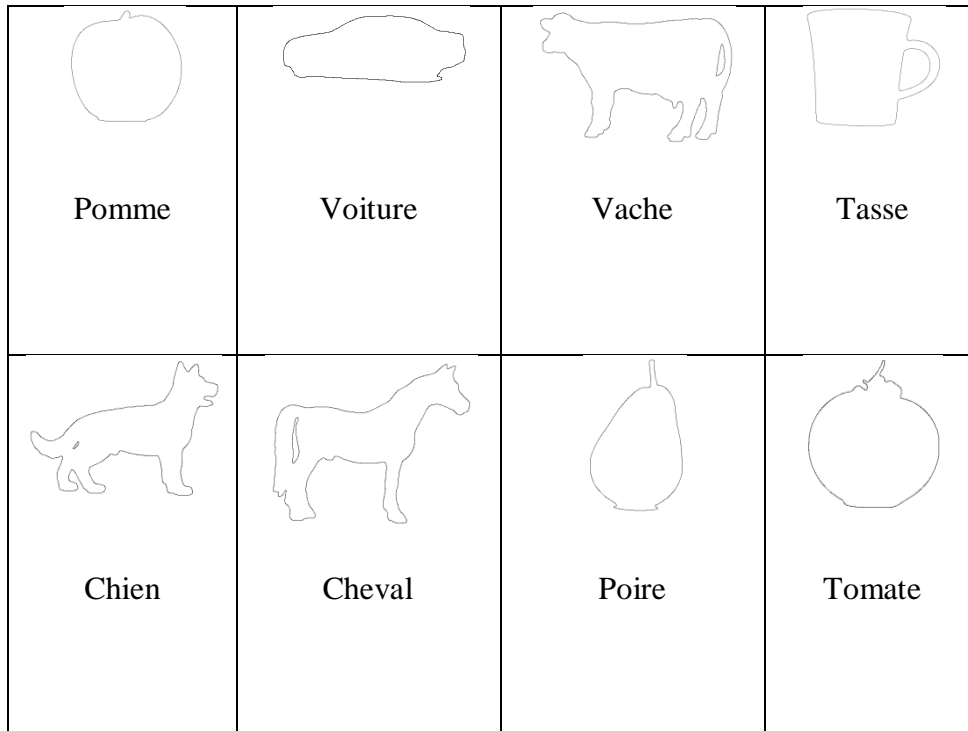


Figure 4.18 : Les types d'objets dans la base [Leibe et Al. 2003]

Pour l'expérimentation de notre méthode de reconnaissance et d'indexation, nous avons utilisé les images de la base de [Leibe et Al. 2003] montrés sur la figure suivante :



Figure 4.19 : Images réel [Leibe et Al. 2003] utilisée pour l'expérimentation de notre logiciel.

Nous avons effectué nos tests sur une base de données comportant 40 Image de base. Chaque image étant occultée par différents pourcentage d'occlusion de 10 à 70 pourcent.

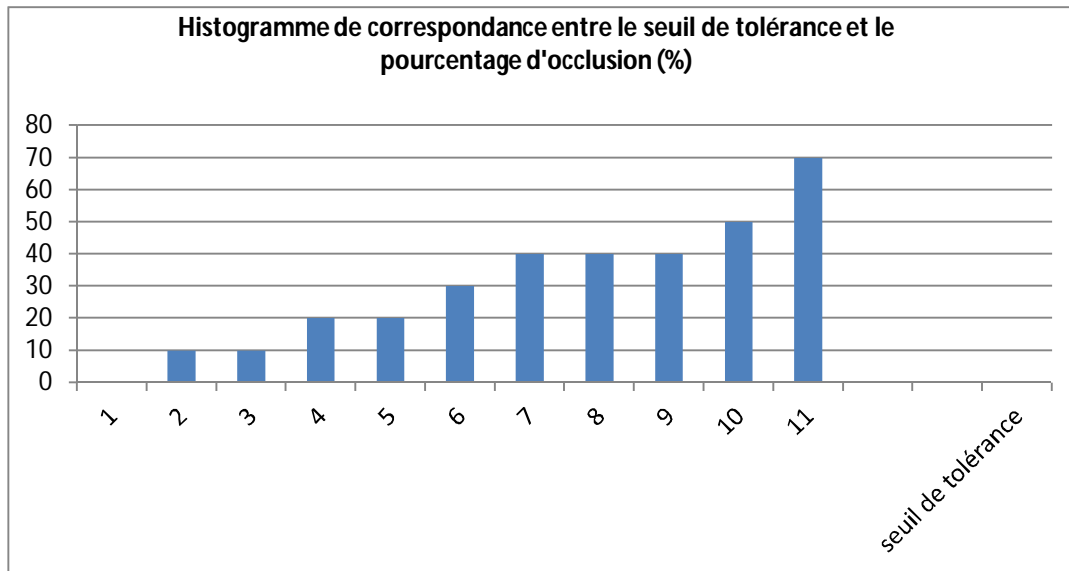


Figure 4.20 : Histogramme montrant la correspondance entre le seuil de tolérance et le pourcentage d'occlusion.

Nous remarquons qu'à chaque fois que le pourcentage d'occlusion augmente le seuil de tolérance doit augmenter pour que la reconnaissance de l'image occultée soit faite.

Nous montrons dans le graphe suivant le pourcentage de reconnaissance de notre méthode par rapport au pourcentage d'occlusion des images requêtes. Les tests ont été effectués sur 40 Image, on a choisi les huit image de la figure 4.19 et cinq vue sur chaque image qui ont été occulté de 10 a 70 pourcent.

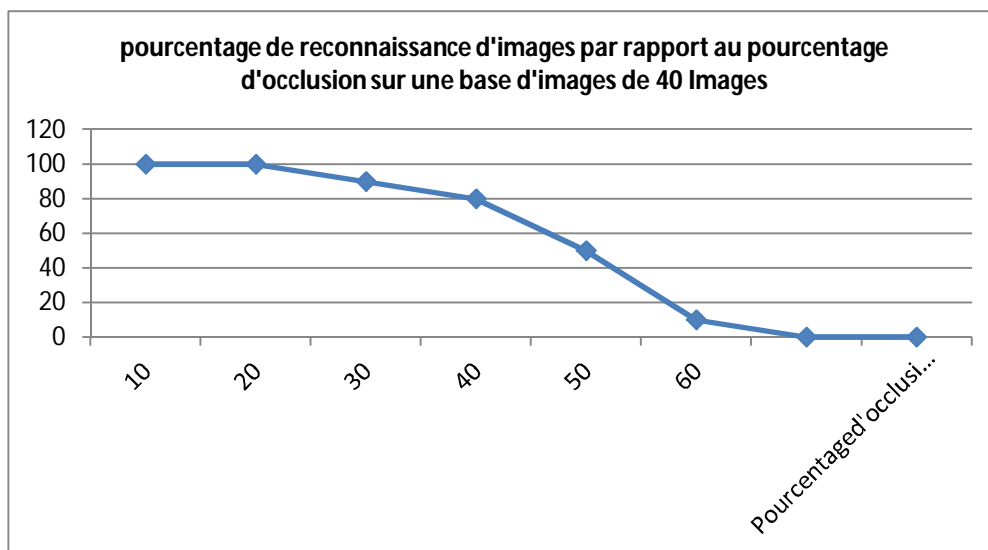

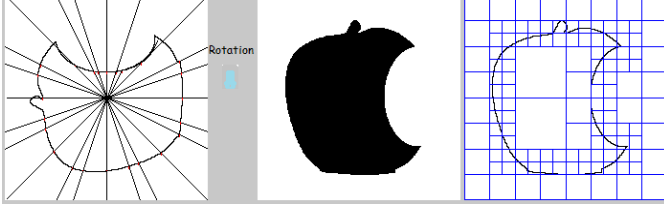
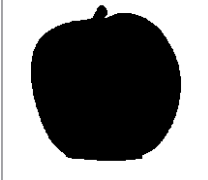

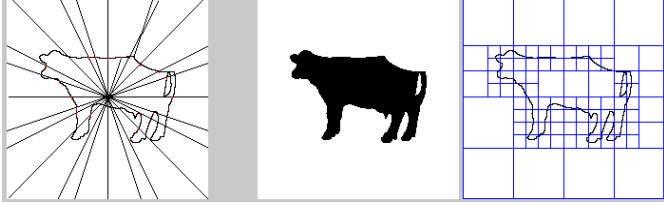

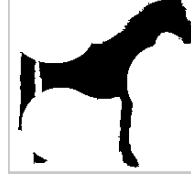
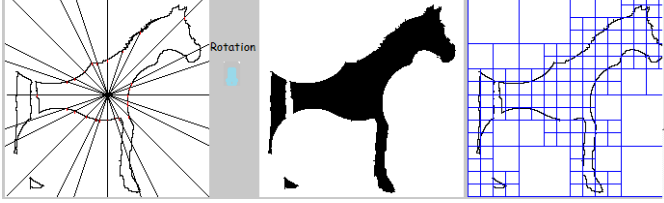

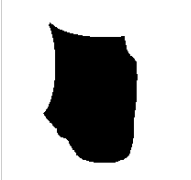
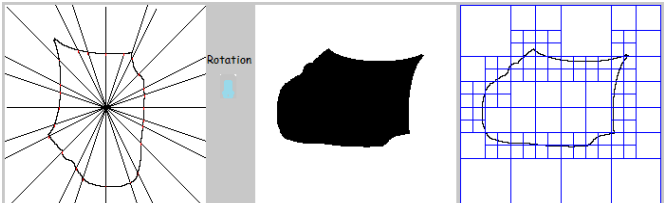
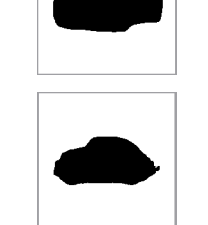
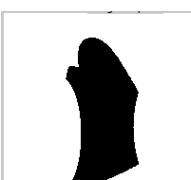

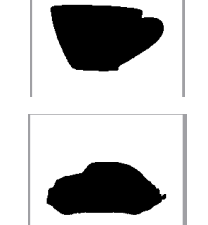
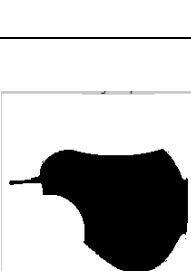
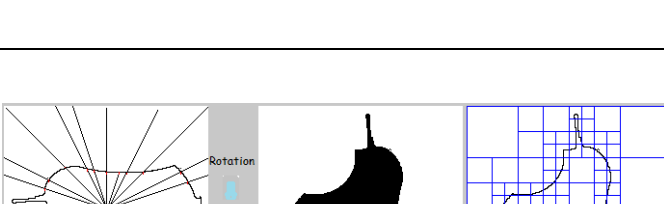
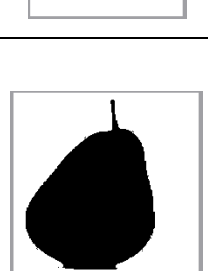


Figure 4.21 : Pourcentage de reconnaissance d'image par rapport au pourcentage d'occlusion sur une base de 40 images.

Objet requête	Recherche d'une image correspondante dans la base de donnée	Objet retrouvé
 <p>(1)</p>		
 <p>(2)</p>		
 <p>(3)</p>		
 <p>(4)</p>		
 <p>(5)</p>		
 <p>(6)</p>		

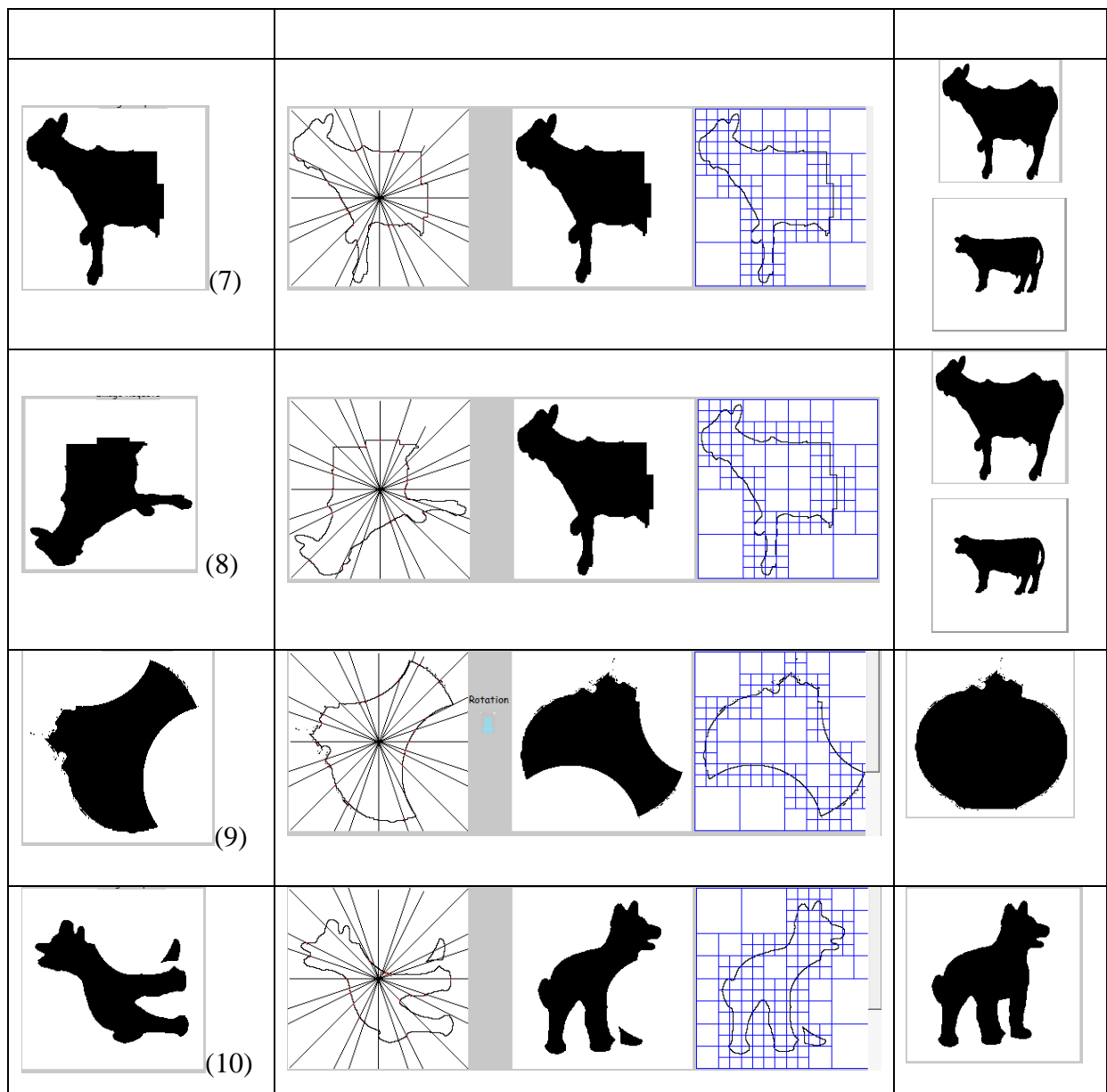


Figure 4.22 : Quelques experimentation de notre logiciel d'images de la base de [Leibe et Al. 2003].

De la ligne 1 à la ligne 10 de la figure 4.22 nous avons 10 images requêtes: une pomme, une vache, un cheval, une voiture, une tasse, une poire, une vache avec une autre vue, une vache en rotation de 90 degré, une tomate et un chien.

Le seuil de tolérance et le pourcentage d'occlusion de chaque requête sont donnés respectivement par les paires de valeurs suivantes: (3, 10%), (5, 30%), (4, 30%), (4, 20%), (8, 47%), (5, 20%), (3, 23%), (4, 29%), (6, 20%), (4, 10%).

A partir de ces résultats, nous pouvons déduire que jusqu'à 40% de l'occlusion et également en présence de rotation, notre système peut reconnaître correctement l'objet de requête. Cependant il y a quelques exceptions, par exemple au niveau de l'exemple 5, notre système reconnaît l'objet mais affiche aussi une autre image qui correspond à sa recherche dans la base et qui ne correspond pas à l'objet. Cela est dû à une certaine ressemblance de la forme de l'objet dans l'image trouvée et la forme de l'objet partiellement occulté. Cette confusion est due à l'utilisation du seuil de tolérance car si on tolère certaines non ressemblances à cause de l'occlusion, on peut tomber dans certains cas comme dans le cas de l'exemple 5 dans une reconnaissance erronée.

On peut déduire de cela que plus le seuil de tolérance est élevé, plus les cas de confusion de reconnaissance peuvent apparaître.

5. Conclusion :

Nous avons réussi à construire un système capable d'indexer et de reconnaître des objets partiellement occultés avec une combinaison de deux techniques : la technique du Quadtree et la technique utilisant les informations de courbures pour la reconnaissance des objets. Nous avons présenté dans ce chapitre notre logiciel et quelques interfaces principales pour l'exécution d'une indexation et d'une reconnaissance d'objets partiellement occultés.

Conclusion générale

Nous avons proposé dans ce mémoire, une nouvelle méthode d'indexation et de reconnaissance d'objets en présence d'occlusion. Cette technique consiste à combiner deux méthodes, la méthode de Quadtree et une méthode de reconnaissance d'objets se basant sur l'information de courbure de l'objet pour la reconnaissance de celui-ci. Notre technique permet d'indexer et de reconnaître un objet partiellement occulté et est robuste à la rotation de l'objet, grâce à la technique de reconnaissance d'objet par les informations de courbure de l'objet, nous avons établis une reconnaissance d'un objet en rotation.

La technique du Quadtree a montré un problème lors de son application sur des objets partiellement occulté avec une rotation, nous avons décidé de palier à ce problème en utilisant la technique de mise en correspondance utilisant les informations de courbure et ceci après avoir résolu le problème d'occlusion basé sur les arbres quaternaires.

Les images de la base ont été indexées à l'aide de deux codes, le code linéaire trié et l'ensemble d'amplitude d'angle des courbures de l'objet dans l'image.

Dans l'étape d'indexation, l'implantation linéaire des arbres quaternaires a été proposée pour permettre un stockage efficace des images organisées en arbre quaternaire, sans occuper la mémoire par les pointeurs du stockage hiérarchique. Ainsi qu'une indexation par la technique utilisant les informations de courbures pour pouvoir prendre en charge la rotation.

Le système que nous avons mis en place combine deux méthodes, La technique du Quadtree et la mise en correspondance en utilisant les informations de courbures pour la reconnaissance d'objets partiellement occultés.

Nous avons décrit précédemment la généralisation de cette méthode. Cette généralisation nous a permis de démontrer que la relation entre le code du quadtree occulté avec celui non occulté est correcte quel que soit le niveau ou l'endroit de l'occlusion dans l'image. Pour nos perspectives, nous souhaitons élargir cette méthode pour reconnaître un objet dans image caché par plusieurs objets.

Référence:

- (Ayache et al., 1986) N.Ayache and O.D. Faugeras, HYPER: a new approach for the recognition and positioning of 2D objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(1):44-54, 1986.
- (Ayache, 1983) Ayache, N. Un système de vision Bidimensionnelle En Robotique Industrielle. Thèse de troisième cycle 578, Univ. Paris-Sud, Centre d'Orsay, Paris, France, June 1983.
- (Ayache et al., 1986) Ayache, N et Faugeras, O.D. Hyper : a new approach for recognition and positioning of 2-D objects. *IEE Trans. On PAMI*, 8(1):44-54, 1986.
- (Albuz et al., 2000) Albuz E., Kocalar E., Khokhar A., Quantized CIELab Space and Encoded Spatial Structure for Scalable Indexing of Large Color Image Archives, *Proc. Of the IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, June 2000.
- (Aouat et al., 2010) Aouat S., Larabi S., Indexing binary images using quad-tree Decomposition, *International Conference on System, Man, and Cybernetics, Istanbul, Turkey*, 10-13 October, 2010.
- (Ait-Aoudia et al., 2013) Ait-Aoudia A., Aouat S., recognition of partially occluded object, the 10th acs/ieee international conference on computer systems and applications, Morocco, 27-30 May, 2013.
- (Ait-Aoudia et al., 2013b) Ait-Aoudia A., Aouat S., Quad-tree technique to retrieve occluded objects, *The ASM SIVA, the International Conference on Signal, Image, Vision and their Application*, Guelma, Algérie, Novembre 2013.
- (Bolles et al., 1986) R.C. Bolles and R.Horaud, 3DPO: A three-dimensional Part Orientation system, *The International Journal of Robotics Research*, 5(3):3-26, 1986.
- (Bhanu et al., 1984) Bhanu,B. et Faugeras, O.D. Shape Matching of Two-dimensional Objects. *IEEE Trans. On Pattern Analysis and Machine Intelligence*, vol 6, pp. 137-155, 1984.
- (Buffa, 2001) Buffa M., The Virtual Diver, an Architectural "Swim-Around" System Incorporating Real Imagery, In *workshop Proc Computer Graphics technology for the exploration of the sea*, 1999.
- (Biederman et al., 1988) Biederman I., Ju G., Surface vs. edge-based determinants of visual recognition, *Cognitive Psychology*, 20(1):38-64, January 1988.
- (Belongie et al., 2002) S. Belongie, M. Jitendra, Shape Matching and Object Recognition Using Shape Contexts, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol 24, Avril 2002.
- (Colin, 1999) Vincent Colin, représentation et reconnaissance d'objets par champs réceptifs, Thèse de Doctorat, Décembre 1999.

-
- (Chen et al., 1991) C.H. Chen and P.G. Mulgaonkar, CAD-based feature-utility measures for automatic vision programming, In *Direction in Automated CAD-Based Vision*, pages 106 -114, IEEE Computer Society Press, 1991.
- (Chow et al., 1977) Chow, W.K. and Aggarwal, J.K. Computer Analysis of Planar Curvilinear Moving Images. *IEEE Transactions on Computers*, 26(2):179-185, 1977.
- (Chakkrabarti et al., 2000) Chakrabarti K., Ortega-Binderbergem., Porkaew K., Zuo P., Mehrotra S., Similar Shape Retrieval in MARS, *IEEE Int. Conf. on Multimedia and Expo (ICME II)*, New York, NY, USA, 2000, p. 709–712.
- (Desolneux et al., 2000) Desolneux, A., L. et al., J. Meaningful alignments. *J. Comput. Vision*, 40(1):7–23, 2000.
- (Dessimoz, 1979) Dessimoz, J.D. Recognition and Handling of overlapping industrial parts. In *9th Symposium on Industrial Robots*, Washington, 1979.
- (Dorko et al., 2003) Dorkó, G. et Schmid, C. Selection of scale-invariant parts for object class recognition. *ICCV '03 : Proceedings of the Ninth IEEE International Conference on Computer Vision*, page 634, Washington, DC, USA. IEEE Computer Society, 2003
- (Du-Ming, 1997) Du-Ming, T. An improved generalized Hough transform for the recognition of overlapping objects. In *Image and vision computing*. 15(1997) 877-888, 1997.
- (Duda, 1972) Duda, R. O., Use of the Hough transformation to detect lines and curves in pictures. *ACM*, 15(1):11–15, 1972.
- (De Winter et al., 2004) De Winter J., Wagemans J., Contour-based object identification and segmentation: stimuli, norms and data, and software tools, *Behavior Research Methods, Instruments, and Computers*, 36(4):604–624, November 2004.
- (Fischler et al., 1981) M.A. Fischler and R.C. Bolles, Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography, *ACM*, 24(6):381–395, 1981.
- (Grimson et al., 1984) Grimson, W.E.L et Lozano-Perez. Model-based recognition and localization from sparse or tactile data. *Int. J. Robotics Res.*, 3(3):3-35, 1984.
- (Grimson et al., 1985) Grimson, W.E.L et Lozano-Perez. Recognition and localization of overlapping parts from sparse data in two and three dimensions. In *IEEE, International Conference on Robotics and Automation*, St Louis (USA), page 61-66, St.Louis, 1985.
- (Grimson et al., 1987) W.E.L. Grimson and T. Lozano-Pérez, Localizing overlapping parts by searching the interpretation tree, *PAMI*, 9(4):469–482, 1987.
- (Goad, 1983) C.Goad, Special purpose automatic programming for 3d model-based vision, In *Proceedings of the DARPA Image Understanding Workshop*, pages 371–381, 1983.
-

-
- (Garici et al., 2002): M.A.GARICI, S.AKKOUCHE, Utilisation des Quasi Invariants Géométriques pour L'indexation d'une Base D'image, Mémoire d'ingénieur USTHB 2002.
- (Huttenlocher et al., 1990) D.P. Huttenlocher and S Ullman, Recognizing solid objects by alignment with an image, *IJCV*, 5(2):195–212, 1990.
- (Hornegger et al., 1995) J.Hornegger and H. Niemann. Statistical learning, localization, and identification of objects. In *Proceedings of the 5th International Conference on Computer Vision*, Cambridge, Massachusetts, USA, pages 914-919, Juin 1995.
- (Huttenlocher et al., 1992) D.P Huttenlocher and W.J. Rucklidge. A multi-resolution technique for comparing images using the Hausdorff distance. In *Proceedings of the Conference on Computer Vision Pattern Recognition*, New York, USA, pages 705-706, 1993.
- (Huttenlocher et al., 1993) D.P Huttenlocher , G.A. Klanderman, and W.J. Rucklidge. Comparing images using the Hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9): 850-863, Septembre 1993.
- (Herbin, 1997) S.Herbin. Eléments pour la formalisation d'une reconnaissance active. Application à la vision tridimensionnelle. PhD thesis, Ecole Normale Supérieure de Cachan, juillet 1997.
- (Huttenlocher et al., 1990) D.P Huttenlocher and K.Kedem. Efficiently computing the Hausdorff distance for point sets under translation. In *Proceeding of Sixth ACM Symposium on Computational Geometry*, pages 340-349, 1990.
- (Huttenlocher et al., 1991) D.P Huttenlocher, K.Kedem, and M.Sharir. The upper of Voronoi surfaces and its applications. In *Proceeding of Seventh ACM Symposium on Computational Geometry*, pages 194-293, 1991.
- (Ikeuchi et al., 1996) K.Ikeuchi and T.Kanade, Automatic generation of object recognition programs, *Proceedings of the IEEE*, 76(8):016-1035, 1996.
- (Ikeuchi, 1987) K.Ikeuchi, Generating an interpretation tree from a CAD model for 3D object recognition in binpicking tasks, *International Journal of Computer Vision*, pages 145-165, 1987.
- (Ikeskinen, 2010) Ikeskinen, <http://www.fr.123rf.com>.
- (Kamgar-Parsi et al., 1991) B. Kamgar-Parsi, J. L. Jones, and A. Rosenfeld. Registration of multiple overlapping range images: Scenes without distinctive features. *IEEE Trans. PAMI*, 13(9):857_871, September 1991.
- (Lamdan et al., 1988) Y. Lamdan, J.T. Schwartz, and H.J. Wolfson, Object recognition by affine invariant matching, In *CVPR*, pages 335–344, 1988.
- (Leibe et al, 2003) Leibe B, Schiele B. Analyzing Appearance and Contour Based Methods for Object Categorization. *Int. Conf. (CVPR'03)*, Madison, Wisconsin, June 2003.
- (Lowe, 1987) D.G. Lowe, The viewpoint consistency constraint, *IJCV*, 1(1):57–72, 1987.
-

(Lamiroy, 1998) B.Lamiroy, Reconnaissance et modélisation d'objet 3D à l'aide d'invariants projectifs et affine, juillet 1998.

(Lowe, 1999) D. Lowe. Object recognition from local scale-invariant features. In CVPR, pages 1150–1157, 1999.

(Lowe, 2004) Lowe, D. Distinctive Image Features from Scale-Invariant Keypoints, International Journal of Computer Vision, 2004.

(Liu et al., 2012) Liu K., Kaleas D., Ruuspa R., Prototyping interaction with everyday artifacts: training and recognizing 3D objects via Kinects, TEI '12: Proceedings of the Sixth International Conference on Tangible, Embedded and Embodied Interaction, 2012.

(Marr, 1982) D. Marr, Vision: A Computational Investigation into the Human Representation and Processing of Visual Information, W.H. Freeman, 1982.

(McKee et al., 1977) McKee, J.M. and Aggarwal, J.K. Computer recognition of partial views of curved objects. IEEE Trans. On Computer, 26(8):790-900, 1977.

(Murase et al., 1995) H.Murase and S Nayar. Visual learning and recognition of 3d objects from appearance. IJCV, 14(1):5–24, 1995.

(Maitre et al., 2000) Maître H., Joinville D., and M. Roux. How to design dem assessment maps. In International Workshop on Pattern recognition in Remote Sensing, Andorre-La-Vieille, Andorre, septembre 2000.

(Manouvrier et al., 2002) Manouvrier M., Rukoz M., Jomier G., Quadtree representations for storage and manipulation of clusters of images, Image and Vision Computing, vol. 20, n 7, 2002, p. 513–527.

(Munkelt et al., 1995) O.Munkelt, Aspect-trees: Generation and interpretation, Computer Vision and Image Understanding, 61(3): 365-386, Mai 1995.

(Nelson et al., 1998) R. C. Nelson et A. Selinger. A Cubist approach to Object Recognition. Dans International Conference on Computer Vision (ICCV'98), janvier 1998.

(Nowak ,2008) E.Nowak, Reconnaissance de catégories d'objets et d'instances d'objets à l'aide de représentations locales, Mars 2008.

(Niblack et al., 1993) W. Niblack, R. Barber, W. Equitz, M.D. Flickner, D. Glasman, D. Petkovic, and P. Yanker. The qbic project: Querying image by content using color, texture, and shape. SPIE, 1998:173–187, Février 1993.

(Oliveira, 2008) Oliveira F., Matching contours in images using curvature information, Computational Vision and Medical Image Processing, 2008.

(Pontil et al., 1998) M. Pontil and A. Verri. Support vector machines for 3d object recognition. PAMI, 20(6):637–646, 1998.

-
- (Pavlidis, 1979) Pavlidis, T. The use of a syntactic analyzer for contour matching. *IEEE Trans. On PAMI*, 1(3):307-310,1979.
- (Perkins, 1978) Perkins, W.A. A model based system for industrial parts. *IEEE Trans. Comput.*, 27(2):126-143,1978.
- (Papadourakis et al., 2011) Papadourakis V., Argyros A., Multiple objects tracking in the presence of long-term occlusions, *Computer Vision and Image Understanding*, 2011.
- (Rucklidge, 1995) W.J. Rucklidge. Locating objects using the Hausdorff distance. In *Proceeding of the 5th International Conference on Computer Vision*, Cambridge, Massachusetts, USA, pages 57-464, 1995.
- (Radack et al., 1989) G. M. Radack and N. I. Badler. Local matching of surfaces using a boundary-centered radial decomposition. *Comput. Vision, Graphics Image Process.*,45:380_396, 1989.
- (Rodriguez et al., 1989)J. J. Rodríguez and J. K. Aggarwal. Navigation using image sequence analysis and 3-D terrain matching. In *Proc. Workshop on Interpretation of 3D Scenes*, pages 200_207, Austin, TX, November 1989.
- (Riad, 1983) Riad, A. Identification d'Objets Partiellement observés en vue de leur manipulation en robotique. Thèse de troisième cycle 2890, Université Paul Sabatier, Toulouse, 1983.
- (Sadgal et al., 2003)M.Sadgal, A.El fazziki, Une Architecture pour la reconnaissance d'objets en imagerie aérienne, *Journal électronique d'intelligence artificielle*, Juillet 2003.
- (Stockman, 1987)G Stockman, Object recognition and localization via pose clustering, In *Computer Vision, Graphics and Image Processing*, pages 361–387, 1987.
- (Schmid et al., 1997) C. Schmid and R. Mohr. Local greyvalue invariants for image retrieval. *PAMI*, 19(5):530–534, 1997.
- (Schiele, 1997) B.Schiele. Reconnaissance d'objets utilisant des histogrammes multidimensionnels de champs réceptifs. Thèse de doctorat, GRAVIR-IMAG-INRIA Rhône-Alpes, Juillet 1997.
- (Sossa, 1992) H.Sossa. Reconnaissance d'objets polyédriques dans une base de modèles. Thèse de doctorat, Institut National Polytechnique de Grenoble, France, Décembre 1992.
- (Swain et al., 1991) M.J. Swain and D.H. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1): 11-32, 1991.
- (Schiele et al., 2000) B. Schiele and J. Crowley. Recognition without correspondence using multidimensional receptive field histograms. *IJCV*, 36(1):31–50, 2000.
- (Shirai, 1978) Shirai, Y. Recognition of real-word objects using edge cues, pages 353-362. Academic Press, New York, 1978.

(Shotton, 2007) Shotton, J. Multi-scale categorical object recognition using contour fragments. Transactions on Pattern Analysis and Machine Intelligence. 2007.

(Samet, 1984) Samet H., The Quadtree and Related Hierarchical Structures, Computing Surveys, vol. 16, n2, 1984, p. 187–260.

(Shotton et al., 2007) Shotton J., Blake A., Cipolla R., Multi-Scale Categorical Object Recognition Using Contour Fragments, IEEE Transaction of pattern analysis and machine intelligence, 2007.

(Shotton et al., 2013) Jamie Shotton, Toby Sharp, Alex Kipman, Andrew Fitzgibbon, Mark Finocchio, Andrew Blake, Mat Cook, Richard Moore, Real-time human pose recognition in parts from single depth images, Communications of the ACM, Volume 56 Issue 1, 2013.

(Thompson et al., 1987) D.W. Thompson and J.L. Mundy, Three-dimensional model matching from an unconstrained viewpoint, In International Conference on Robotics and Automation, pages 208–220, 1987.

(Turk et al., 1991) M. Turk and A. Pentland. Face recognition using eigenfaces. In CVPR, pages 586–591, 1991.

(Tsai et al., 1980) Tsai, W.H et Fu, K.S. A Syntactic-statistical approach to recognition of industrial objects. Proc. Of IEEE 5th Int. Conf on Patt. Recogn., 1980.

(Tsai et al., 1985) Tsai, W.H et Yu, S.S. Attributed string matching with merging for shape recognition. IEEE Trans. On PAMI, 7(4), 1985.

(Turney et al., 1985) Turney, J.L. et Mudge, T.N. Recognizing partially occluded parts. In IEEE Transactions on Pattern Analysis and Machine Intelligence, 1985.

(Torralba et al., 2012) Torralba A., Murphy K.P., Freeman W.T., Using the Forest to See the Trees: Exploiting Context for Visual Object Detection and Localization, Communications of the ACM volume 53, Issue 3, 2012.

(Venkateswar, 1992) Venkateswar, V, Extraction of straight lines in aerial images. Transactions on Pattern Analysis and Machine Intelligence, 14(11):1111–1114, 1992.

(Wolfson et al., 1997) H.J. Wolfson and I Rigoutsos, Geometric hashing: An overview, Computational Science and Engineering, 4(4):10–21, 1997.

(Wu et al., 2012) Wu C., Kuo Y., Hsu W., Large-Scale Simultaneous Multi-Object Recognition and Localization via Bottom Up Search-Based Approach, MM '12: Proceedings of the 20th ACM international conference on Multimedia, 2012.

(Zhu et al., 2010) Zhu B., Yan J., Li Q., Yang C., Liu J., Xu N., Yi M., Cai K., Attacks and Design of Image Recognition CAPTCHAs, CCS '10: Proceedings of the 17th ACM conference on Computer and communications security, 2010.

(Zhang, 1993) Z.Zhang, Le problème de la mise en correspondance: L'état de l'art, *Research Report*, No.2146, INRIA Sophia-Antipolis, Dec. 1993.