

N° d'ordre :

Thèse

Présentée à

L'Université d'Evry-Val d'Essonne

Par

Djamel MERAD

Pour obtenir le diplôme de

Docteur de L'Université d'Evry-Val d'Essonne

Spécialité : **Robotique**

**Reconnaissance 2D/2D et 2D/3D d'objets à
partir de leurs squelettes**

Soutenue le 13 Décembre 2004

Devant le jury Composé de :

Mr S. MIGUET	Professeur, Université Lumière Lyon II	Président / Rapporteur
Mr P. GORRIA	Professeur, Université de Bourgogne	Rapporteur
Mme S. LELANDAIS	Maître de conférences, Université d'Evry	Examineur
Mr J. TRIBOULET	Maître de conférences, CUFN Nîmes	Examineur
Mr M. MALLEM	Professeur, Université d'Evry	Directeur de thèse

Remerciements

Je voudrais exprimer ma gratitude et mes sincères remerciements à :

Messieurs Serge MIGUET et Patrick GORRIA, respectivement Professeur à l'université Lumière Lyon 2 et Professeur à l'université de Bourgogne qui m'ont fait l'honneur d'être les rapporteurs de ce mémoire.

Monsieur Malik MALLEM, Professeur à l'université d'Evry-Val d'Essonne, pour la confiance qu'il m'a accordée en m'accueillant dans son équipe ainsi que pour les conseils qu'il m'a apportés.

Madame Sylvie LELANDAIS, Maître de conférences à l'université d'Evry et Monsieur Jean TRIBOULET, Maître de conférence au CUFR Nîmes, d'avoir encadré ce travail et d'avoir corrigé ce manuscrit dans les meilleurs délais. Je suis également reconnaissant pour les conseils et les encouragements qu'ils m'ont dispensés.

Un remerciement spécial à Mr Joël Leroux Professeur à l'université de Nice qui m'a encadré en DEA et qui m'a transmis le goût de la recherche.

Je voudrais aussi remercier tout simplement Pierre Drap Chercheur-CNRS à Marseille pour qui je dois tout.

Je remercie également toutes les personnes que j'ai côtoyées: Bachir, Fakhrou, Krimou, Bad, Salim, Bruce, Mr Didier, Djaber, Zorba, ... et bien sûr sans oublier le fantastique Khiair ainsi que tous les chercheurs, enseignants, techniciens et administratifs, pour leur sympathie, leur grande disponibilité et leurs indispensables coups de main.

Un merci particulier à Patrick BABIN et à Omar AIT AIDER qui m'a aidé dans la rédaction de mon mémoire de soutenance en génie civil en 1994.

*L'homme sage est celui qui cherche à s'instruire auprès des hommes.
L'homme fort est celui qui sait briser ses désirs.
L'homme riche est celui qui se contente de son lot.
L'homme honoré est celui qui honore les autres.*

Sentence populaire juive.

Cette thèse est dédié à **Ahmed Iflis** et à **Ahsene Azembo**.
*Ils se sont consacrés et consumés pour définir et construire
un langage dans le langage.*

Table des matières

Liste des figures	xi
INTRODUCTION ET OBJECTIFS	1
1 PROBLEMATIQUE ET ETAT DE L'ART SUR L'INSPECTION AUTOMATIQUE	7
1.1 Introduction	7
1.2 Présentation de les machines à usinage rapide URANE 25	8
1.3 Problématique	10
1.4 Etat de l'art	11
1.5 Méthode d'inspection proposée	13
1.6 Conclusion	14
2 PRETRAITEMENT DE L'IMAGE	17
2.1 Introduction	17
2.2 Restauration d'images	17
2.2.1 Les méthodes Markoviennes	18
2.2.2 Les méthodes variationnelles	18
2.2.3 Les méthodes par Equations aux Dérivées Partielles	19
2.2.4 Méthode implémentée	20
2.3 Extraction des segments de contours	23
2.3.1 Méthodes dérivatives	23
2.3.2 Segmentation des contours	29
2.4 Détection des points de forte courbure	31
2.4.1 Analyse de quelques approches classiques	32
2.4.2 Méthode implémentée	34
2.4.3 Résultats	39
2.5 Mesures de similarité	41
2.5.1 Coefficient de corrélation des segments	41
2.5.2 Coefficient de corrélation des PFC	42
2.5.3 Coefficient de corrélation des niveaux de gris	42
2.6 Conclusion	43

3	METHODES DE CLASSIFICATION	45
3.1	Introduction	45
3.2	Saisie des données	45
3.3	Méthode empirique	46
3.4	Analyse des paramètres	49
3.4.1	Analyse en composantes principales	50
3.5	Classification automatique	55
3.5.1	Discrimination Bayésienne	55
3.5.2	Approche neuronale	57
3.5.3	Combinaison des classifieurs	63
3.6	Conclusion	65
4	SYSTEMES DE RECONNAISSANCE D'OBJETS 3D DE FORME LIBRE	67
4.1	Introduction	67
4.2	Etat de l'art sur la reconnaissance d'objets 3D de forme libre	68
4.2.1	Méthodes basées sur l'apparence	68
4.2.2	Méthodes basées sur l'extraction de silhouettes	72
4.2.3	Méthodes basées sur le "shape from shading"	74
4.3	Présentation de l'approche développée	75
4.3.1	Méthode de reconnaissance par squelettes	76
4.4	Conclusion	78
5	SQUELETTISATION	81
5.1	Introduction	81
5.2	Définitions et propriétés	82
5.3	Squelettisation 2D	83
5.3.1	Algorithme d'amincissement	83
5.3.2	Simulation de la propagation de feu de prairie	84
5.3.3	Squelette à partir de la carte des distances	84
5.3.4	Squelette à partir du diagramme de Voronoï	84
5.3.5	Bilan	85
5.4	Méthode hybride développée	86
5.4.1	Algorithme	87
5.4.2	Evaluation de la méthode proposée	90
5.5	Squelette homotopique 3D	95
5.5.1	Algorithme d'amincissement	95
5.6	Conclusion	100

6	MISE EN CORRESPONDANCE DE GRAPHERS	103
6.1	Introduction	103
6.2	Terminologie et notations de base	103
6.3	Etat de l'art dans la littérature	104
6.3.1	Appariement par mesures de distance	104
6.3.2	Appariement de graphes par les algorithmes génétiques . .	105
6.3.3	Appariement de graphes utilisant les techniques basées sur la théorie probabiliste	105
6.3.4	Appariement de graphes utilisant les réseaux de neurones .	106
6.4	Classification des problèmes d'appariement de graphes	106
6.4.1	Appariement exact et inexact de graphes	107
6.4.2	Appariement de graphes en utilisant les nœuds factices (nuls)	108
6.4.3	Appariement de graphes nécessitant plus d'une correspon- dance par nœud	109
6.5	Complexité d'appariement de graphes	110
6.6	Appariement de deux squelettes	110
6.6.1	Construction du graphe à partir du squelette	111
6.6.2	Caractérisation des squelettes par les valeurs propres . . .	113
6.6.3	Indexation du modèle dans la base de données	115
6.6.4	Mise en correspondance un-à-un des nœuds	116
6.6.5	Problèmes et améliorations	120
6.7	Conclusion	125
7	RÉSULTATS ET APPLICATIONS	127
7.1	Introduction	127
7.2	Vérification des hypothèses	127
7.3	Evaluation des performances de la reconnaissance	129
7.3.1	Conditions expérimentales	129
7.3.2	Différentes situations étudiées	129
7.4	Application au suivi d'objets de forme libre	140
7.4.1	Principe du suivi	140
7.4.2	Suivi d'objets 3D par squelettes	142
7.4.3	Résultats du suivi par les squelettes	143
7.5	Conclusion	144
	Conclusion générale	149

Annexes	153
A Analyse Multi-échelles	155
A.1 Généralités	155
A.1.1 Axiome de la structure pyramidale	155
A.1.2 Axiome de comparaison locale	156
A.1.3 Axiome de régularité	156
A.1.4 Invariance morphologique	156
A.1.5 Invariance euclidienne ou affine	156
A.1.6 Linéarité	157
A.2 Théorèmes généraux	157
A.3 Rappel théorique et implémentation	158
B Triangulation de Delaunay	161
B.1 Diagramme de Voronoi	161
B.2 Delaunay, dual de Voronoi	162
B.2.1 Algorithmes de construction	163
B.3 Squelette et triangulation de Delaunay	165
B.3.1 Construction du squelette	165
C Squelette par Carte de Distance	167
C.1 Calcul de la carte de distance	167
C.1.1 Principe	167
C.1.2 Méthode de Danielsson	168
C.1.3 Distance du chanfrein	168
C.2 Détection du squelette	170
C.2.1 Centre des boules maximales	170
C.2.2 Approche gradient	170
C.2.3 Détection du squelette par géométrie différentielle	171
D Recalage 3D	173
D.1 Définition du problème	173
D.2 Algorithme d'itération orthogonale (IO)	176
D.2.1 Problème de l'orientation absolue	177
D.2.2 Algorithme	178
Références bibliographiques	181

Liste des figures

1.1	Machine URANE 25	9
1.2	Pièce sans et avec défaut	10
1.3	Pièces présentant une forte similitude	10
1.4	Méthode d'inspection automatique proposée	15
2.2	Restauration par diffusion linéaire - filtre Gaussien -	22
2.1	Restauration par diffusion non linéaire - Alvarez -	22
2.3	Méthodes derivatives par différences finies	25
2.4	Méthodes dérivatives par filtrage optimal	27
2.5	Extraction de segments	30
2.6	Image originale	35
2.7	Sortie du filtre de Harris avant seuillage	35
2.8	Sortie du filtre de Harris après seuillage	36
2.9	Image réelle	37
2.10	Opérateur de Harris avant traitement	37
2.11	Filtre de Harris après seuillage	38
2.12	Points de forte courbure	38
2.13	Segments avant restauration	39
2.14	Segments après restauration	40
2.15	PFC avant restauration	40
2.16	PFC après restauration $n = 5$	41
3.1	Images de la scène réelle	46
3.2	Cercles de corrélation issus de l'ACP sur les trois variables	51
3.3	Cercles de corrélation issus de l'ACP avec 15 paramètres	54
3.4	Schéma général d'un réseau de neurones	59
4.1	Algorithme de reconnaissance de forme par squelettes	78
4.2	Reconnaissance d'objets 3D de forme libre par squelettes	79
5.1	Algorithme de la methode de squelettisation hybride	87
5.2	Image origine	88
5.3	Squelette par diagramme de Delaunay	88
5.5	Elagage des segments	89
5.4	Squelette par carte de distance	89
5.6	Squelette par Delaunay + carte de distance	90

5.7	Squelette résultant	90
5.8	Test de robustesse	91
5.9	Test d'homotopie	92
5.10	Squelette théorique	93
5.11	Squelette avec un pas de $1/2$	93
5.12	Squelette avec un pas de $1/5$	94
5.13	Squelette avec un pas de $1/15$	94
5.14	Voisins d'un voxel	95
5.15	Masque de Chanfrein $d_{3,4,5}$	96
5.16	Algorithme de calcul de la carte de distance 3D	97
5.17	Cube et squelette curviligne	98
5.18	Cylindre et squelette curviligne	99
5.19	Cactus et squelette curviligne	99
5.20	Squelettes 2D et 3D -1-	101
5.21	Squelettes 2D et 3D -2-	101
5.22	Squelettes 2D et 3D -3-	102
5.23	Squelettes 2D et 3D -4-	102
5.24	Squelettes 2D et 3D -5-	102
6.1	Construction du graphe à partir du squelette	112
6.2	Calcul de la signature topologique	115
6.3	Algorithme d'indexation	116
6.4	Algorithme d'appariement un-à-un des nœuds	118
6.5	Algorithme de recherche en profondeur	118
6.6	Graphe bipartite	119
6.7	Nœuds de cardinalité maximale et de poids minimal	120
6.8	Exemple de mise en correspondance	121
7.1	Algorithme de vérification des hypothèses	128
7.2	Objet Cactus	131
7.3	Squelette 2D de l'objet Cactus	131
7.4	Graphe du squelette 2D	132
7.5	Projection du modèle à partir du premier ensemble d'appariements	133
7.6	Projection du modèle à partir du second ensemble d'appariements	133
7.7	Résultat de la reconnaissance sur l'aspect -1-	134
7.9	Résultat de la reconnaissance sur l'aspect -3-	135
7.8	Résultat de la reconnaissance sur l'aspect -2-	135
7.10	Résultat de la reconnaissance sur l'aspect -4- : cas dégénéré	136
7.11	Reconnaissance dans le cas d'occlusions -1-	138
7.12	Reconnaissance dans le cas d'occlusions -2-	138
7.13	Reconnaissance dans le cas d'occlusions -3-	139

7.14	Reconnaissance dans une scène à plusieurs objets	140
7.15	Principe général du suivi	141
7.16	Algorithme de suivi par les squelettes	142
7.17	Séquence de suivi d'un objet	146
7.18	Base de données de modèles	147
B.1	Diagramme de Voronoï de points dans le plan.	162
B.2	Dualité Delaunay/Voronoï.	163
B.3	Simplexes de Delaunay en conflit avec S_i	164
B.4	Nouvelle triangulation	165
C.1	Masques séquentiels d_{34} et d_{5711} : avant (a, c) et arrière (b, d) . .	169
C.2	Carte de distance	169
C.3	Squelette par le gradient	171
C.4	Squelette par Courbure moyenne	172
D.1	Les différents repères mis en jeu dans le problème de recalage 3D .	174
D.2	Les erreurs de colinéarité dans l'espace image et l'espace objet . .	176

INTRODUCTION ET OBJECTIFS

Le monde réel est composé essentiellement d'objets tridimensionnels. Une des tâches principales d'un système de vision artificielle est de pouvoir identifier et localiser ces objets à partir d'informations reçues de l'environnement.

La reconnaissance d'objets a pour but d'extraire automatiquement et efficacement le contenu intéressant, pertinent et utile de la scène c'est-à-dire de déterminer l'identité des objets formant la scène et leur disposition spatiale dans le contexte d'une tâche ou d'un processus à réaliser.

Il s'agit d'un point très important en vision par ordinateur puisqu'il est indispensable pour toutes sortes d'applications. On peut citer par exemple :

1. la robotique et l'automatisation des procédés industriels (inspection, assemblage, tri),
2. L'analyse d'images aériennes ou satellites (télédétection),
3. L'analyse d'images médicales (aide au diagnostic).
4. ...

Il est toutefois complexe et les approches pour le résoudre sont très différentes suivant la connaissance a priori de l'objet ou non, le type et le nombre de capteurs utilisés (une ou plusieurs caméras 2D, caméras 3D, Télémètre, etc.), et le type d'objet à détecter (2D, 3D, Polyédrique, forme libre, ..).

Cependant il existe deux grandes familles de méthodes pour construire un système de reconnaissance de formes :

- les méthodes structurelles,
- les méthodes statistiques.

Les méthodes structurelles sont ainsi dénommées parce que l'on suppose que l'on est capable de définir des composants élémentaires de la forme de l'objet à reconnaître, qui constituent des primitives structurelles, et de caractériser, pour chaque classe, la structure de l'objet selon ces primitives. On parle aussi dans ce cas de primitives de haut niveau (par opposition au bas niveau, plus près du signal brut délivré par les capteurs). Les primitives ainsi définies sont donc, le

plus souvent, de nature discrète, définies par exemple à partir de prédicats. Elles seront pertinentes, c'est-à-dire qu'elles seront définies en fonction de la forme à reconnaître plutôt que des caractéristiques des informations fournies par le capteur. En conséquence, leur extraction ne sera pas immédiate et nécessitera la mise en œuvre d'algorithmes d'extraction souvent sophistiqués.

Les algorithmes à mettre en œuvre pour effectuer une classification à partir des primitives structurelles relèvent le plus souvent de l'algorithmique géométrique. On trouve ainsi typiquement :

- Exploration d'arbres de décision : c'est la forme la plus générale (la moins structurée) de vérification de prédicats.
- Mise en correspondance de graphes ou sous-graphes, ou encore recherche de cliques maximales dans des graphes : l'accent est ici mis non seulement sur la présence de primitives, mais sur les relations qui les lient. Une forme est décrite par un graphe encodant ces relations, et la reconnaissance s'effectue en cherchant un appariement entre le graphe décrivant une classe et celui construit à partir des primitives décrivant une forme à reconnaître.

Les méthodes statistiques de reconnaissance de formes reposent sur une description extensive, plutôt que compréhensive, des classes. Bien entendu, il est assez rare que l'on puisse prévoir a priori toutes les occurrences possibles de formes observées pour les différentes classes. On considère plus précisément qu'une classe est décrite par un ensemble représentatif d'exemplaires de formes appartenant à cette classe.

Les primitives statistiques, qu'on appelle souvent aussi des paramètres, sont de natures variées, mais par essence simple : valeurs réelles, ou représentation quantifiées de valeurs réelles (par exemple des niveaux de gris de pixels dans une image). L'approche statistique comprend les étapes suivantes :

- choix des paramètres,
- construction d'une description statistique des classes à partir d'un ensemble d'apprentissage,
- calcul de la distance d'une forme à chacune des classes à partir de la description statistique précédemment élaborée,
- choix de la classe la plus proche.

Dans le travail de recherche que nous présentons, il s'agit de reconnaître des objets 3D rigides relativement complexes, à l'aide d'une caméra 2D fixe. Peu de systèmes peuvent prétendre à la résolution du problème de reconnaissance d'objets 3D à partir de séquences d'images monoculaires. Nous avons apporté des contributions à ce problème en proposant, en premier lieu, une méthode efficace

et robuste de reconnaissance d'objets 2D/2D par des méthodes statistiques. En second lieu, nous avons proposé une méthode originale de reconnaissance d'objets 3D de formes libres à base de squelettes. Deux applications différentes ont servi de support pour valider nos apports:

La première application liée à notre sujet est l'aide au diagnostic dans la surveillance de machines à usinage rapide de type Urane 20 ou Urane 25. En effet la société Renault Automation-Comau a demandé au LSC d'étudier la faisabilité d'un système orienté vision permettant de répondre aux questions suivantes :

Est-ce le bon objet ?

Est-il correctement placé ?

Présente-t-il des défauts ?

En raison des informations dont on dispose a priori, comme le type et la position de la pièce à usiner, nous avons opté pour une méthode de reconnaissance statistique 2D/2D.

Dans un environnement tel que l'enceinte de travail d'une machine-outil, les conditions de vision sont dégradées. Il est donc nécessaire de faire subir à l'image des traitements préliminaires de restauration et de rehaussement avant d'extraire les informations intéressantes.

Après restauration de l'image, nous commençons par l'extraction de primitives structurales (points singuliers, segments) sur une image qui nous servira de référence. Ensuite pour chaque nouvelle image nous appliquons la même procédure d'extraction de primitives, et par une simple comparaison de leurs attributs géométriques nous déterminons deux coefficients de corrélation qui seront utilisés comme paramètres de classification. En plus de ces deux paramètres nous avons défini d'autres paramètres à partir des niveaux de gris des deux images. Ce travail a été complété par une analyse en composante principale permettant d'évaluer la pertinence et l'indépendance de ces paramètres.

Afin d'obtenir une classification automatique et robuste, nous avons opté pour trois approches. La première est une approche bayésienne, la seconde est fondée sur la définition d'un réseau de neurones et la combinaison de ces deux techniques, nous a permis de créer une troisième voie améliorant les résultats de la classification. Cette classification a été validée par une série de tests menés dans des conditions réelles.

La deuxième application pratique est le guidage par vision indirecte au moyen d'une caméra d'un bras manipulateur à six degrés de libertés via internet. L'opérateur, devant son poste de travail, commande le bras manipulateur à travers une liaison internet. L'opérateur éprouve des difficultés à guider le bras

manipulateur vers un objet 3D de forme libre. À l'aide d'une caméra 2D observant la scène, l'algorithme proposé permet de reconnaître les objets présents ainsi que leurs positions. Dans ce contexte, nous supposons que les modèles 3D des objets sont connus.

De ce fait, nous avons à résoudre un problème de reconnaissance 2D/3D à l'aide d'une méthode structurelle.

Depuis longtemps, les chercheurs en vision par ordinateur traitent le problème de la reconnaissance de formes à base de modèles. Il n'est donc pas étonnant de trouver une littérature abondante à ce sujet. Les méthodes employées visent à connaître quel objet est présent dans l'image et à le localiser. Le système suit une phase d'apprentissage dans laquelle les formes de références sont décrites en termes de paramètres discriminants. Vient ensuite la phase proprement dite de classification, où les données du capteur sont traitées pour en tirer une série de paramètres de même nature, afin de les comparer aux attributs des formes de références. Une décision finale permet de désigner la forme reconnue et de la localiser. Pour les objets de formes libres le problème réside dans l'extraction de primitives structurelles qui devront être mises en correspondance.

Ces deux dernières décennies, nous trouvons dans la littérature plusieurs travaux qui portent sur la reconnaissance d'objet 3D de forme libre. Les tests effectués sont réalisés soit avec des images de synthèse, soit dans des conditions contrôlées où des problèmes omniprésents dans les scènes 3D réelles (comme les occlusions, les ombrages ..) sont rarement pris en compte. Afin de prendre en compte ces problèmes, une approche originale qui utilise le squelette des objets comme descripteur de formes a été développée.

Il existe plusieurs applications pour les squelettes d'objets 2D et 3D dans le traitement d'images (codage, compression,..) et dans la vision en général. En effet, nous retrouvons dans le squelette de l'objet sa structure topologique et surtout, la plupart des informations contenues dans la silhouette de la forme se retrouvent dans le squelette. Un autre avantage qu'il faut signaler, c'est le fait que de nature les squelettes ont une structure de graphe.

A partir du processus de squelettisation nous obtenons deux squelettes. Le premier représente l'objet dans la scène "Squelette 2D on-line" et le second représente le modèle de l'objet dans la base de données "Squelette 3D curviligne off-line". Cette méthode code les informations géométriques et topologiques sous la forme d'un graphe de squelette, ensuite elle utilise les techniques d'isomorphisme de graphes (de sous-graphes) pour effectuer un appariement entre les différents nœuds de chaque graphe. Cette correspondance un-à-un des nœuds nous permet de reconnaître l'objet et de le localiser.

Ce document est composé de deux parties

- La Partie 1 constituée des 3 premiers chapitres, est dédiée à la présentation de la première approche de reconnaissance 2D/2D. Dans le premier chapitre, nous présentons un tour d’horizon des méthodes d’inspection automatiques existantes. Nous introduisons par la suite notre approche de reconnaissance 2D/2D. Dans le chapitre 2, nous commençons par une analyse bibliographique et une discussion générale des différents outils développés (outils de restauration d’images et d’extraction d’amers géométriques) pour mettre en œuvre notre méthode. Dans le chapitre 3, nous détaillons le choix des primitives de classifications utilisées et l’analyse en composante principale qui a suivi. Ensuite nous présentons le second aspect concernant les approches de classification qui nous permettent d’effectuer un diagnostic. Nous développons plus particulièrement la théorie bayésienne et neuronale et une troisième qui est une combinaison des deux. En dernier, nous présentons des résultats expérimentaux qui permettent alors d’évaluer cette méthode.
- La Partie 2 constituée des chapitres 4 à 7, est consacrée à la présentation de notre méthode de reconnaissance d’objet 2D/3D de forme libre qui utilise la transformation en squelettes des objets. Cette technique nous permet de contourner le problème lié aux formes libres d’une manière élégante. Dans le chapitre 4, un état de l’art sur la reconnaissance d’objets de forme libre est présenté ainsi que le principe général de notre système de reconnaissance. Ensuite, dans le chapitre 5, nous exposons les méthodes de squelettisation 2D et 3D existantes et les améliorations apportées à l’une d’elles. Le sixième chapitre concerne la mise en correspondance de la scène avec le modèle (squelette 2D avec le squelette 3D). Après avoir étudié les méthodes d’isomorphisme de graphes existantes, nous avons mis en œuvre une méthode basée sur la signature topologique. Cette technique nous permet d’apparier l’objet de la scène avec un modèle de la base de données. Finalement dans le chapitre 7, nous développons la méthode de vérification utilisée pour valider l’étape de reconnaissance et nous présentons les résultats expérimentaux obtenus sur des images réelles et son application pour le suivi d’objets 3D de forme libre.
Dans la conclusion générale, le bilan de ces travaux ainsi que les perspectives envisagées seront présentés.

Chapitre 1

PROBLEMATIQUE ET ETAT DE L'ART SUR L'INSPECTION AUTOMATIQUE

1.1 Introduction

Ces dernières décennies sont caractérisées par la nécessité de mettre sur le marché des produits de qualité et ce, rapidement. La fabrication d'objets manufacturés n'échappe pas à cette tendance et de nombreux outils informatiques facilitent la conception des pièces (CAO : conception assistée par ordinateur) de même que, de plus en plus, leur fabrication (FAO : fabrication assistée par ordinateur). Ce foisonnement d'outils informatiques a permis de diminuer considérablement les temps de conception et de fabrication de pièces en série.

Bien que plusieurs de ces activités soient aujourd'hui assistées par ordinateur, le domaine de l'inspection automatisée, indispensable pour assurer la qualité de la production, demeure une des tâches de l'activité manufacturière à être la moins automatisée.

Ceci est dû au fait que celle-ci demande un minimum de capacité de raisonnement, puisqu'il s'agit de rejeter ou d'accepter une pièce (inspection qualitative) ou de classer un type de défaut constaté (inspection quantitative).

L'inspection visuelle automatisée comble en partie les défauts d'un évaluateur humain, dont la constance (aptitude à résister à la lassitude due à la répétitivité) et l'efficacité ne sont pas optimales. Par ailleurs, certains défauts ne sont pas détectables par l'œil humain et certains environnements sont hostiles à l'intervention humaine. L'inspection visuelle automatique coûte globalement moins cher, elle est plus rapide, et procure une inspection de bonne qualité sans contact, donc sans endommagement de la production.

Pour ces raisons la société Renault Automation a voulu étudier la possibilité d'intégrer un tel module dans les machines à usinage rapide URANE 20 et URANE 25.

Dans ce chapitre nous commencerons par définir le contexte de l'approche que nous proposons. Nous allons d'abord décrire la machine à laquelle elle est dédiée ainsi que les problèmes qui lui sont liés. Nous présenterons ensuite un état de l'art sur l'inspection automatique et finalement nous exposerons l'architecture générale de notre système, en soulignant les problèmes relatifs à l'approche utilisée.

1.2 Présentation de les machines à usinage rapide URANE 25

Les machines à usinage rapide URANE 25 figure 1.1, qui allient productivité et flexibilité, sont destinées prioritairement à l'usinage "en moyenne et grande cadence" des composants et organes mécaniques de l'industrie automobile. Avec ses moteurs linéaires et ses deux magasins d'outils, elles autorisent des changements d'outils en temps masqués, supprimant ainsi les temps morts. De cette façon, ces machines permettent d'usiner des pièces rapidement et avec une grande précision.

Les machines URANE 25 ont deux types de configurations de fonctionnement, soit "à poste fixe" pour un type d'opérations sur une pièce particulière (culasse par exemple) réalisées par une seule machine, soit "en ligne" où la même pièce sera usinée par deux ou plusieurs machines. Les travaux réalisés sont, de manière non exhaustive, alésage, tournage, perçage, taraudage. . .

Le volume de travail de la colonne est de $500 \times 500 \times 200$ (mm). Les vitesses de rotation de la broche sont de plusieurs milliers de tours par minute et les accélérations lors des déplacements ou des changements d'outils peuvent aller jusqu'à 3.5 g.

La pièce à usiner est montée sur un support permettant sa mise en place. Le tout est ensuite bridé à un plateau rotatif permettant de présenter la pièce sous l'angle adéquat pour l'usinage.

Durant les phases d'usinage, trois cas peuvent être observés :

1. Dans le cas d'une rectification de surface, le travail est réalisé "à sec" sans ajout de lubrifiant.
2. Dans le cas d'une opération de perçage ou de taraudage, l'outil est vecteur d'un lubrifiant qui est projeté avant et pendant l'usinage.
3. Lorsque l'outil n'est pas équipé du système précédent, une "douche" permet d'assurer cette opération de lubrification.

L'ensemble est confiné dans une ou deux enceintes successives permettant d'assurer la protection du voisinage (casse d'un outil, projection de lubrifiant, projection de copeaux ...).

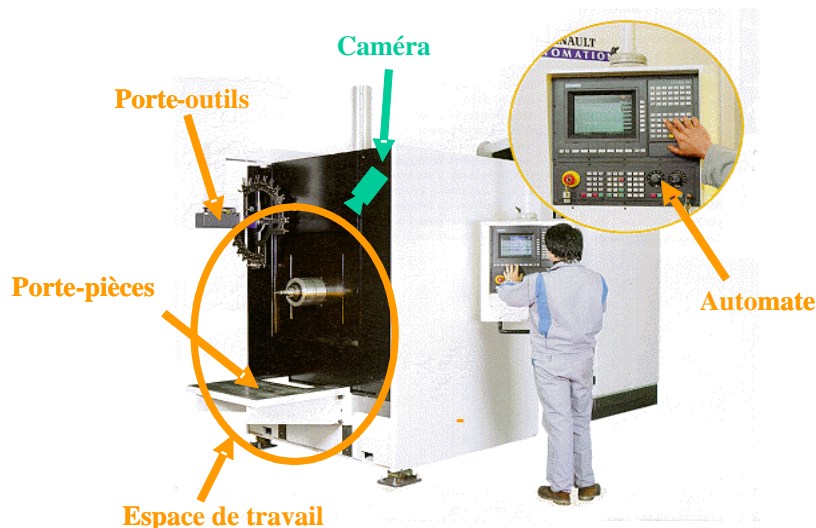


Figure 1.1: Machine URANE 25

Lors des opérations de changement d'outil, une roue située dans le plan de déplacement de la colonne permet de recevoir jusqu'à vingt outils. Dans le cas d'opérations plus complexes, une seconde roue peut être ajoutée de l'autre côté de la broche et un mouvement de rotation autour d'un axe vertical permet de présenter les outils à la broche.

La mise au point des phases automatisées d'usinage est faite via un système de CFAO. A l'issue de cette phase de mise au point "virtuelle", les informations sont téléchargées dans l'automate de la machine. La phase d'usinage peut alors commencer moyennant une éventuelle procédure de mise en correspondance.

Ces machines sont équipées d'une caméra intérieure. Les images sont transmises à l'opérateur sur un moniteur au-dessus du pupitre de commande permettant une observation globale du volume de travail. Durant la phase d'usinage, l'opérateur pourrait observer le déroulement des opérations grâce à des vitres aménagées autour de l'enceinte de travail, mais hormis le cas d'usinage sans lubrifiant, l'espace de travail est difficilement observable.

1.3 Problématique

Etant donné les vitesses de déplacement importantes et l'ambiance de travail lors des phases d'usinage, une surveillance directe (par l'opérateur) ou indirecte (par caméra), bien que rassurante pour l'opérateur, ne lui accorde pas un temps de réaction suffisant en cas d'un mauvais positionnement de la pièce. De plus, des défauts peuvent passer inaperçus au moment de la mise en route du centre d'usinage engendrant des conséquences fâcheuses pour la machine et pour la pièce.

Les figures ci-dessous illustrent les différents défauts qui peuvent se retrouver durant le processus de fabrication.



Figure 1.2: Pièce sans et avec défaut



Figure 1.3: Pièces présentant une forte similitude

Le premier cas (figure 1.2), se présente lorsque la pièce est initialement endommagée et les opérations d'usinage ne permettront pas de la rectifier. Le deuxième cas (figure 1.3), se produit lorsque l'opérateur se trompe complètement de pièce. En effet, dans un atelier de fabrication, nous pouvons trouver facilement des pièces qui présentent une grande similitude avec d'autres pièces, comme par exemple entre une culasse d'une Renault 25 et une culasse d'une Renault Safrane.

Le troisième cas correspond à un mauvais positionnement de la pièce sur le porte-pièce. Bien que des brides permettent de fixer convenablement la pièce à usiner, pour des raisons diverses, il arrive que celle-ci soit mal calée et une erreur de quelques millimètres peut provoquer des dommages importants pour la pièce et parfois même pour la machine.

La société Renault Automation-Comau a demandé au Laboratoire des Systèmes Complexes d'étudier la faisabilité d'un système permettant d'offrir une aide au diagnostic à l'opérateur lors de la mise en fonction de la machine. Ce système lui permettra d'assurer que la réalité est bien en accord avec les données du modèle transmis à l'automate, c'est-à-dire, vérifier que la pièce à usiner est bien celle attendue, qu'elle est placée convenablement et qu'elle ne présente pas de défauts.

Pour solutionner ce problème, nous avons commencé à étudier les systèmes d'inspection automatique existants. Un ensemble non exhaustif de ces techniques est présenté dans le paragraphe suivant.

1.4 Etat de l'art

Historiquement, la plupart des tâches d'inspection automatique sont accomplies à l'aide de systèmes d'inspection à contact, qui exigent que le dispositif soit minutieusement placé. Cependant, la constante amélioration des technologies de production à grande vitesse a créé le besoin de nouvelles techniques d'inspection rapide. Dans ce contexte qui correspond à notre problématique, l'inspection visuelle convient parfaitement, car elle permet de réaliser des contrôles précis en suivant les cadences de fabrication.

Les techniques d'inspection automatique visuelles peuvent se classer en deux catégories :

- Extraction des caractéristiques de l'objet scanné et comparaison à une description ou à une liste de règles qui décrivent le modèle idéal. Si toutes les règles sont satisfaites, alors l'objet est considéré sans défaut
- Mise en correspondance de parties d'une image modèle (image de synthèses ou image stockée) avec l'image fournie par le capteur.

Nous présentons rapidement quelques-uns des ces travaux.

Edinbarough et al [EDI 95] présentent des modèles mathématiques et des algorithmes pour une synthèse visuelle d'inspection de tolérances géométriques d'objets circulaires. La détermination de l'erreur de circularité est basée sur l'approche de minimisation de zone. Le système utilise une caméra CCD noir

et blanc. Après la numérisation de l'objet à contrôler, l'erreur de circularité est calculée à partir du contour de l'objet en la comparant avec le modèle de circularité.

Chen et Su [CHE 96] présentent un système d'inspection qui mesure les dimensions et la concentricité d'assemblage. Des images 2D sont prises avec une caméra CCD. Après quelques traitements, des algorithmes pour la reconnaissance de la forme des trous intérieurs sont utilisés. Dans le processus d'inspection les dimensions et la concentricité trouvées sont comparées aux valeurs de référence dans une base de données.

Moron [MOR 96] a présenté un travail qui porte sur l'inspection automatique d'objets comportant des surfaces complexes à partir de leur modèle CAO et de données 3D provenant d'un capteur télémétrique. La particularité de ce travail est l'utilisation d'un modèle triangulé simple pour représenter des objets complexes. Afin de comparer le modèle géométrique et les données 3D, il utilise une méthode classique de mise en correspondance qui minimise la distance globale entre les points 3D et les points du modèle.

Modayur et al [MOD 92] ont développé un système d'inspection basé sur une théorie de tolérancement cohérente pour vérifier des pièces manufacturées 3D à partir des images 2D. Les images 2D sont comparées aux modèles CAO des pièces. Les points importants de leur travail sont : 1. L'utilisation de définitions précises des tolérances géométriques appropriées pour leur usage dans le traitement d'image. 2. Le développement des algorithmes de mesure correspondant directement à ces définitions. 3. L'utilisation d'incertitudes dans la tâche de mesure. 4. L'utilisation de cette information d'incertitude dans le processus décisionnel. Les résultats expérimentaux vérifient les valeurs d'incertitude obtenues statistiquement et montrent que les probabilités d'erreur obtenues avec la propagation d'incertitude sont inférieures à celles obtenues sans la propagation d'incertitude.

Trucco et al [TRU 97] présentent un système complet d'inspection impliquant : le positionnement de l'objet 3D, la planification optimale de la position d'inspection et la mesure de certaines caractéristiques géométriques pour lesquelles le modèle géométrique (CAO) est connu. Le recalage 3D est résolu par une technique qui combine l'utilisation des invariants algébriques avec les modèles approximatifs de visibilité. Le bruit est modélisé par une distribution gaussienne, et les points aberrants sont séparés en utilisant une adaptation de l'algorithme LMS (Least Median of Squares). L'inspection se concentre sur des mesures de forme telles que la planéité d'une surface, l'angle entre surfaces et les diamètres des trous, en utilisant le vecteur normal des surfaces.

Prieto [PRI 99] a travaillé sur l'inspection automatique de pièces industrielles à partir du modèle CAO de la pièce et de son image 3D obtenue à l'aide d'un

capteur laser de haute précision. Tout d'abord il détermine et modélise les différents paramètres qui influencent la précision du capteur laser lors du processus de numérisation d'une pièce. Ce modèle est utilisé pour générer automatiquement une stratégie d'acquisition pour la numérisation complète et précise d'une pièce. Le nuage de points issu de la numérisation est recalé avec son modèle CAO postérieurement segmenté. En utilisant un modèle de bruit, il a introduit une valeur de dispersion sur chacun des points 3D acquis par le capteur. Cette valeur de dispersion est présentée comme un facteur de pondération dans les résultats de l'inspection.

Nous pouvons aussi citer les travaux de Moganti et al qui présentent dans [MOG 96] une bonne revue des méthodes d'inspection de PCB (Printed Card Board).

Pour cette problématique des travaux ont été déjà menés dans notre laboratoire. Triboulet et Shaheen [TRI 01] ont proposé un algorithme de vision par ordinateur pour cette application d'inspection. Cette méthode permet de localiser un objet en 3D à l'aide d'une caméra, connaissant le modèle géométrique de cette caméra ainsi que le modèle 3D de l'objet. Ensuite, une méthode de superposition du modèle filaire de l'objet est réalisée en utilisant les plans d'interprétation fournis par la caméra. Une procédure ultime permet de recalibrer le modèle et d'affiner la localisation 3D de l'objet par rapport à un repère fixe. Les méthodes de localisation 3D ainsi que la procédure de recalage proposée sont issues des travaux de thèse de Shaheen [SHA 99]. Cet algorithme permet de bien localiser la pièce, en revanche il ne détecte pas les éventuels défauts que la pièce est susceptible de contenir. A ceci s'ajoute le problème de temps de calcul qui n'est pas compatible à la tâche pour laquelle il est dédié.

Pour ces raisons, nous proposons une méthode de reconnaissance 2D/2D efficace et rapide, qui nous permet de détecter les défauts, d'une part, et d'optimiser le temps de calcul, d'autre part.

1.5 Méthode d'inspection proposée

L'objectif de cette étude est la détection d'éventuelles anomalies des pièces à usiner. Cela revient à vérifier que la pièce souhaitée est au bon endroit par rapport à un ensemble de spécifications données et qu'elle ne possède pas de dégradations.

Ayant connaissance du type de pièces qui seront usinées et de leurs emplacements optimaux pour l'usinage, la méthode d'inspection mise en œuvre consiste à comparer les images de chaque pièce placée pour l'usinage avec une image de

référence. Cette image de référence est préalablement créée pour chaque modèle de pièce. Cela revient à saisir une image d'une bonne pièce placée au bon endroit. Elle est par la suite sauvegardée puis transmise à l'automate avec les données du système. Le schéma global de notre système d'inspection est illustré sur la figure 1.4.

Dans notre contexte, l'utilisation des images 2D est suffisante car les objets industriels traités sont assez complexes, et par conséquent, les informations contenues dans leurs silhouettes sont riches. Cela nous permet de nous affranchir des méthodes 3D comme celles présentées précédemment, qui sont gourmandes en temps de calcul. Pour ces mêmes raisons, nous avons aussi voulu insister sur l'apport de l'information géométrique de bas niveau, et nous différencier des méthodes plus classiques basées sur une analyse de plus haut niveau où l'agrégation de l'information géométrique joue un rôle principal. Rappelons que notre système d'inspection est dédié à une machine à usinage rapide, elle nécessite une grande vitesse d'exécution ainsi qu'une grande précision.

Dans l'environnement industriel tel que l'enceinte de travail, la présence de poussières et d'éclairage défavorable, exigent la restauration des images. Nous avons alors choisi de commencer par des traitements d'images dits de bas niveau pour améliorer la qualité des images traitées et ensuite nous avons procédé à l'extraction de segments après approximation polygonale des contours et de points de fortes courbures. L'intérêt d'utiliser ces primitives provient du fait qu'ils constituent une représentation intrinsèque de l'objet. En effet, ceux-ci contiennent une description mathématique de la forme qui sera utilisée pour quantifier le degré de ressemblance de deux formes. L'extraction de ces primitives, leur comparaison avec celles de l'image de référence, nous a permis de construire plusieurs coefficients de corrélations.

Par la suite, nous nous sommes consacrés à l'analyse de ces paramètres et nous avons mis au point une règle de décision empirique. Afin d'obtenir une classification plus robuste, nous avons opté pour deux autres approches. La première est une approche bayésienne et la seconde est fondée sur la définition d'un réseau de neurones. Cette démarche nous a offert la perspective d'intégrer les informations issues des deux classifieurs afin d'obtenir un diagnostic plus fiable. Les détails de cette partie feront l'objet du chapitre 3.

1.6 Conclusion

Le travail qui nous a été demandé par Renault-Automation est de mettre en oeuvre un système d'inspection automatique par vision monoculaire. Ce système

devra être capable de reconnaître et de localiser des pièces et aussi de détecter d'éventuels défauts présents sur ces pièces.

Comme nous l'avons vu dans l'état de l'art, il n'existe pas de méthode d'inspection standard. Nous avons alors développé une méthode qui prend en compte les informations a priori dont on dispose.

Notre système d'inspection est composé essentiellement de deux parties, une partie traitement d'images et une partie classification. Dans le chapitre suivant nous présenterons en détail la première partie, nous allons alors justifier le choix des méthodes de traitement d'images utilisées ainsi que les résultats de leurs implémentations.

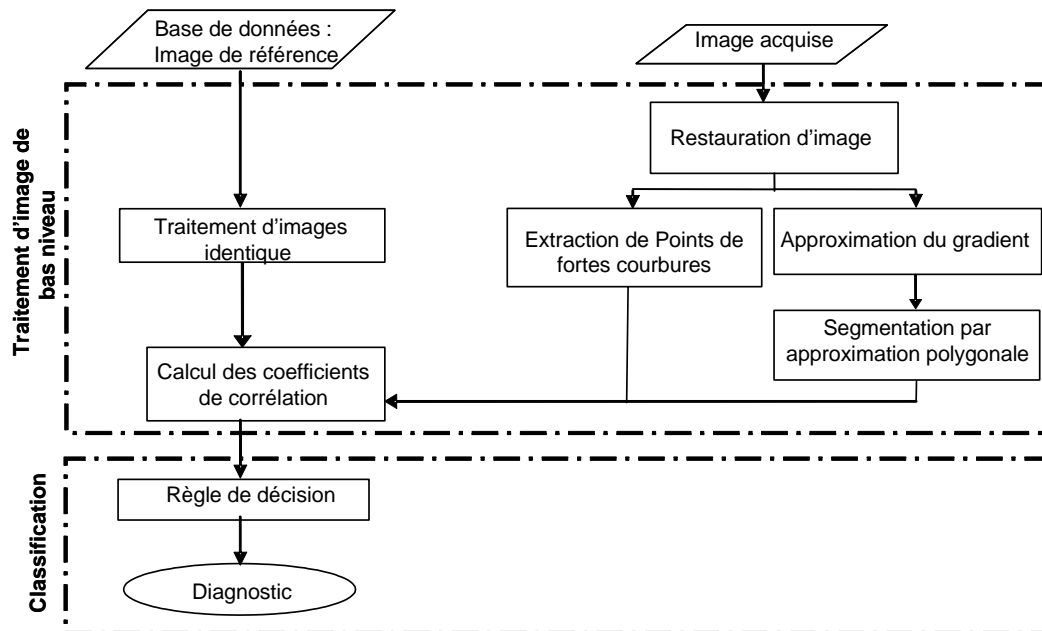


Figure 1.4: Méthode d'inspection automatique proposée

Chapitre 2

PRETRAITEMENT DE L'IMAGE

2.1 Introduction

La stratégie d'inspection que nous proposons est constituée principalement de deux étapes. La première étape consiste à extraire des informations géométriques de la forme par des méthodes de traitement d'image. Ces informations seront utilisées comme paramètres de classification dans la seconde étape. Celle-ci a pour but de déterminer si l'objet présent sur l'image est identique à l'objet de référence et ainsi établir un diagnostic.

Dans ce chapitre, nous nous intéressons principalement aux méthodes de traitement d'images utilisées. Nous regroupons sous ce terme le prétraitement de l'image qui consiste à améliorer et à restaurer les images bruitées et l'extraction d'amers géométriques. Dans les paragraphes suivants, nous présenterons une étude sommaire des méthodes existantes et nous exposerons en détail celles que nous avons implémentées.

2.2 Restauration d'images

Dans un environnement tel que l'enceinte de travail d'une machine-outil, les conditions de vision sont dégradées. Il semble donc nécessaire de faire subir à l'image des traitements préliminaires de restauration afin d'améliorer la qualité des images et par conséquent de parfaire la détection des primitives. Pour restaurer les images, il existe de nombreuses méthodes. Les méthodes classiques, qui utilisent des masques de convolution, fournissent des résultats présentant des oscillations près des discontinuités. Pour contourner cette difficulté, de nouvelles approches ont été proposées.

2.2.1 Les méthodes Markoviennes

L'utilisation des champs de Markov dans la restauration des images s'est développée depuis quelques années [BLA 87], [GEM 84], [ZER 90]. Elles présentent les problèmes classiques de vision et de traitement d'images d'une manière très élégante. Le principal avantage des champs de Markov est de fournir un modèle mathématique simple permettant de prendre en compte les interactions locales au niveau du pixel. Un inconvénient majeur est le coût de calcul relatif à certains algorithmes obtenus à partir de telles modélisations. Les temps de calcul sont aussi liés à la forme de l'énergie utilisée pour prendre en compte les contours.

Les recherches effectuées ces dernières années sont plus orientées vers les méthodes d'optimisation pour calculer les estimateurs de processus que sur l'approche stochastique elle-même.

2.2.2 Les méthodes variationnelles

L'approche variationnelle fait appel à des outils mathématiques liés au calcul des variations ainsi qu'à l'analyse fonctionnelle. Le principe consiste à trouver la solution d'un problème par la minimisation d'une fonctionnelle. Cette fonctionnelle joue le rôle d'énergie par analogie avec la physique, et contient différents termes qui doivent traduire les contraintes que l'on souhaite imposer sur la solution cherchée. Pour le problème de restauration d'images la fonctionnelle peut être, par exemple, sous la forme (2.1). Nous présentons le cas monodimensionnel pour alléger les notations :

$$J(h) = \int_{\Omega} (h(x) - I(x))^2 dx + \int |\nabla h(x)|^2 dx \quad (2.1)$$

La minimisation de la fonctionnelle J doit conduire à une solution h proche des données I et dont les discontinuités, associées à de grandes valeurs du gradient $|\nabla h(x)|^2$ sont minimisées. Cette contrainte régularisante est connue sous le nom de régularisation de Tikhonov [TIK 77].

L'existence et l'unicité de la solution de minimisation fait appel à des théorèmes d'optimisation des fonctionnelles.

Si h est solution de la minimisation de (2.1), et compte tenu des conditions choisies, alors h vérifie nécessairement l'équation d'Euler-Lagrange.

$$\nabla J = \frac{\partial L(h, \nabla h, x)}{\partial h} - \operatorname{div} \left(\frac{\partial L(h, \nabla h, x)}{\partial \nabla h} \right) = 0 \quad (2.2)$$

L'équation d'Euler-Lagrange est une condition nécessaire d'optimalité, et dans le cas (2.1) elle donne :

$$(h(x) - I(x)) - \Delta h = 0 \quad (2.3)$$

La minimisation de (2.1) se ramène donc à la résolution de l'équation aux dérivées partielles (*EDP*) (2.3) par le biais de l'équation d'Euler-Lagrange [DER 95]. Le lien entre minimisation de fonctionnelle et approche par *EDP* n'est pas toujours aussi évident. De nombreux travaux consistent à définir directement une *EDP* à résoudre, sans que celle-ci ne résulte forcément de la minimisation d'une fonctionnelle. C'est ce qui caractérise les approches par *EDP*.

2.2.3 Les méthodes par Equations aux Dérivées Partielles

Ces nouvelles approches posent le problème de la restauration d'images comme un problème de régularisation avec l'introduction des *EDP* où l'image restaurée peut s'écrire comme produit de convolution de l'image bruitée avec un opérateur de lissage [DER 95] :

$$I(x, y, t) = \int_{\Omega} G(x - \nu, y - \nu, t) I_0(\nu, \nu) d\Omega, \quad (\nu, \nu) \in \Omega \quad (2.4)$$

où nous notons que $I_0(x, y)$, est une fonction de R^2 qui représente l'image originale bruitée, $I(x, y, t)$ représente l'image restaurée et t un paramètre qui contrôle l'importance du lissage de l'opérateur $G(x, y, t)$.

2.2.3.1 Diffusion linéaire

Koenderink [KOE 84] est le premier à remarquer que si G de l'équation (2.4) est gaussien et d'une variance $\sigma^2 = 2t$, alors elle peut se mettre sous la forme d'une équation aux dérivées partielles de type parabolique linéaire (2.5), qui n'est autre que l'équation de la chaleur, connue dans la communauté scientifique (mécanique des fluides).

$$\begin{cases} \frac{\partial I}{\partial t}(x, y, t) = I_{xx}(x, y, t) + I_{yy}(x, y, t) \\ I(x, y, 0) = I_0(x, y) \end{cases} \quad (2.5)$$

Koenderink a mis en évidence le lien étroit entre un processus de restauration d'images et un processus de diffusion modélisé par des équations aux dérivées partielles. Cette *EDP* permet une diffusion isotrope, ce qui présente des inconvénients notamment au niveau de la qualité visuelle de l'image, en effet, il ne lisse pas uniquement le bruit mais il gomme aussi les contours, les rendant difficilement identifiables. L'analyse linéaire élimine le bruit mais introduit le flou.

2.2.3.2 Diffusion non linéaire : Modèle de Malik et Perona

Pour résoudre les problèmes issus de la diffusion isotrope, Perona et Malik dans [PER 90], [PER 94] ont proposé un modèle de diffusion anisotrope (2.6). La motivation essentielle de l'utilisation de ce genre de modèles est la construction d'un opérateur de diffusion dépendant des propriétés locales de l'image. Malik et Perona ont proposé un filtre adaptatif, permettant d'atténuer la diffusion dans les régions à fort gradient et de la maintenir dans les zones à faible gradient.

$$\begin{cases} \frac{\partial I}{\partial t}(x, y, t) = \operatorname{div}(g(|\nabla I(x, y, t)|) \nabla I(x, y, t)) \\ I(x, y, t = 0) = I_0(x, y) \end{cases} \quad (2.6)$$

où div et ∇ indiquent les opérateurs de divergence et de gradient par rapport aux variables spatiales respectivement et où $g(s)$ est une fonction décroissante par rapport à s .

Pour des tâches de restauration d'images bruitées, ce modèle présente clairement un avantage. Dans les régions d'intensité homogène, ce processus permettra de réduire effectivement le bruit. Et dans les régions présentant des discontinuités au niveau de l'intensité en niveau de gris, celles-ci ne seront pas lissées et le contraste visuel de ces parties sera meilleur. Il a été prouvé toutefois que cette méthode possède plusieurs inconvénients dont le premier est l'inefficacité dans les zones où le bruit présente de grosses discontinuités. Le second est d'ordre théorique, en effet, la résolution de cette *EDP* est un problème mal posé au sens de Hadamar. Pour une bonne analyse numérique des problèmes issus de la mise en œuvre de la méthode de Perona Malik, nous pouvons se référer à [NIE 94]. Une amélioration de cette idée de diffusion anisotrope, qui prend en compte ces problèmes, a été ainsi proposée par Alvarez et al dans [ALV 92].

2.2.4 Méthode implémentée

Les régions formant l'image sont caractérisées par leur intensité moyenne. Les fluctuations autour de cette intensité moyenne, qui correspondent au bruit, peuvent provenir du dispositif d'acquisition (camera, quantification ...) et aussi de l'environnement du travail (vibrations, rajout de copeaux, lubrifiant ...).

Pour restaurer nos images, nous avons utilisé un filtrage adaptatif. Le modèle choisi est un modèle très proche de celui utilisé par Malik et Perona [PER 94]. C'est un cas particulier du modèle général qu'Alvarez a mis au point dans sa théorie unificatrice des processus de diffusion dans le cadre de l'analyse multi-échelle [ALV 92], [ALV 94], [ALV 96]. Il a ainsi démontré la relation étroite qui existe entre une analyse multi-échelles et les *EDP*.

L'analyse multi-échelles d'une image $I_0(x, y)$ consiste à générer une séquence d'images $I(x, y, t)$, où chaque image de la séquence apparaît comme une version lissée, à l'échelle référencée par le paramètre t , de l'image initiale $I_0(x, y)$. En d'autres termes, un ensemble d'opérateurs T_t appliqués à une image initiale $I_0(x, y)$ conduit à une séquence d'images $I(x, y, t)$ et l'axiomatisation de l'analyse multi-échelles permet d'obtenir l'unicité de cet opérateur.

Nous avons choisi d'utiliser la diffusion euclidienne représentée par l'EDP (2.7) car elle n'introduit pas de flou, rehausse les contours et paraît la plus adaptée pour la segmentation [SCH 00]. Alvarez a démontré que (2.7) est la solution de l'analyse multi-échelle $I(x, y, t) = T_t(I)(x, y)$ si $T_t(I)$ satisfait aux axiomes d'invariances euclidienne et morphologique, de structure pyramidale et de comparaison locale de régularité.

$$\begin{cases} \frac{\partial I}{\partial t}(x, y, t) = |\nabla I(x, y, t)| G\left(\operatorname{div}\left(\frac{\nabla I(x, y, t)}{|\nabla I(x, y, t)|}\right), t\right) \\ I(x, y, t = 0) = I_0(x, y) \end{cases} \quad (2.7)$$

Où $\operatorname{div}\left(\frac{\nabla I}{|\nabla I|}\right)$ désigne la courbure en chaque point (x, y, t) de la ligne de niveau défini par $I(x, y, t) = C$, G est une fonction avec $G(s, t)$ non décroissante par rapport à s .

Le cas particulier où la fonction G est constante et égale à $+1$ ou -1 correspond aux opérations élémentaires de morphologie mathématique que sont la dilatation et l'érosion [SCH 00].

Dans notre implémentation nous avons utilisé la fonction $G(s, t) = s.t$ ce qui conduit à l'équation (2.8).

$$\begin{cases} \frac{\partial I(x, y, t)}{\partial t} = t |\nabla I(x, y, t)| \operatorname{div}\left(\frac{\nabla I(x, y, t)}{|\nabla I(x, y, t)|}\right) \\ I(x, y, t = 0) = I_0(x, y) \end{cases} \quad (2.8)$$

Elle correspond à une diffusion anisotrope dans la direction des lignes de niveaux de l'image. Nous avons opté pour cette fonction à cause de son schéma de résolution, qui est simple à mettre en œuvre et qui n'est pas gourmand en temps de calcul. Les détails de cette approche ainsi que le schéma de résolution sont présentés en annexe A.

Les figures 2.1 et 2.2 illustrent le résultat de l'application de la diffusion euclidienne et de la diffusion linéaire (lissage gaussien) sur des images représentant une portion d'une pièce industrielle où n représente le nombre d'itération et σ^2 la variance du filtre Gaussien.

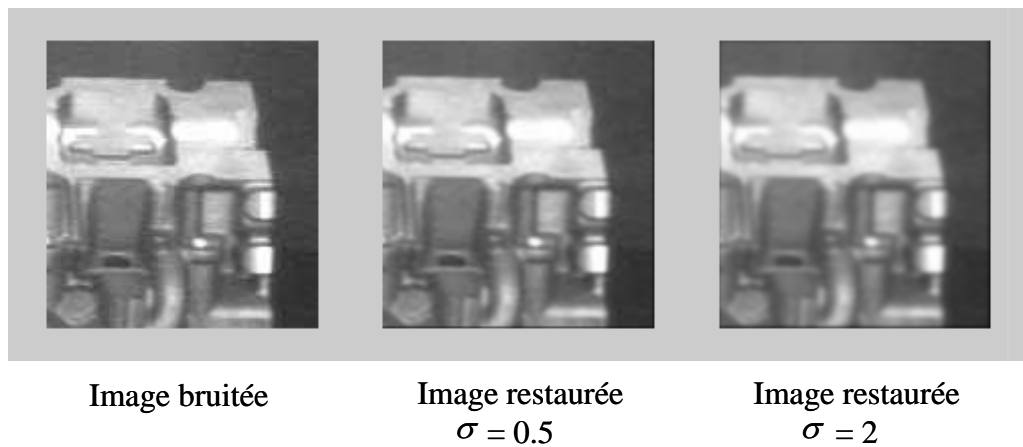


Figure 2.2: Restauration par diffusion linéaire - filtre Gaussien -

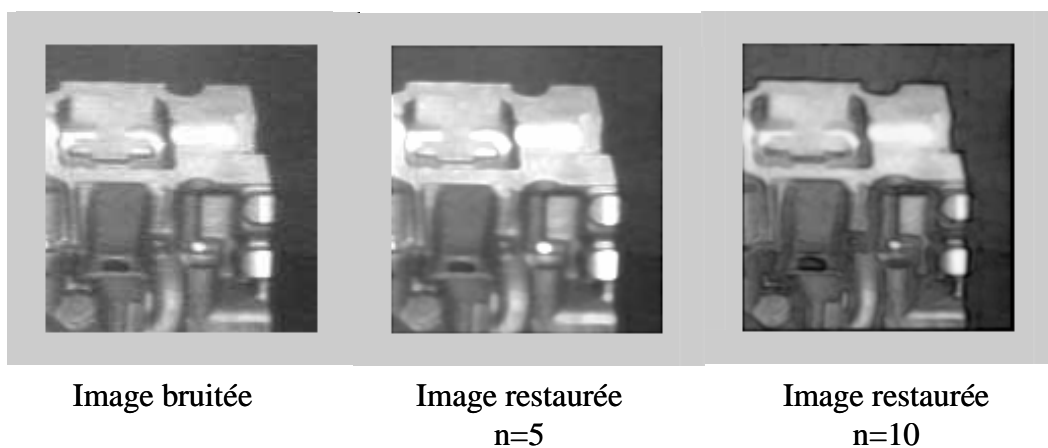


Figure 2.1: Restauration par diffusion non linéaire - Alvarez -

Nous remarquons clairement que, contrairement au filtrage gaussien, la diffusion euclidienne n'introduit pas de flou, rehausse les contours et, par conséquent, elle augmente la qualité visuelle de l'image. Cependant nous constatons qu'à partir de 10 itérations, nous perdons des détails qui peuvent être significatifs, ce qui nous présage l'utilisation de cet opérateur avec un petit nombre d'itération.

Nous allons maintenant présenter l'extraction des amers géométriques retenus comme caractéristiques de l'image. Nous détaillerons d'abord l'extraction des segments obtenu à partir des contours puis le calcul des points de forte courbure.

2.3 Extraction des segments de contours

Dans notre système d'inspection automatique, le diagnostic se fait en mettant en correspondance des primitives robustes issues de deux images différentes.

Les contours représentent la principale entité structurelle qui peut être extraite des deux images. En effet, Ils constituent des indices riches au même titre que les points d'intérêts, pour toute interprétation ultérieure de l'image, et sont caractérisés par la discontinuité de la fonction d'intensité dans les images. Pour détecter les contours, de nombreuses méthodes ont été proposées, parmi lesquelles :

- les méthodes dérivatives.
- les méthodes variationnelles, qui à l'inverse des méthodes dérivatives, prennent en compte l'information globale sur le contour.
- les méthodes stochastiques qui utilisent les champs de Markov.
- les méthodes issues de la morphologie mathématique.

Le choix d'une méthode de détection de contours dépend du type d'images à analyser et des objets que l'on désire extraire. Le choix doit tenir compte également des méthodes que l'on va utiliser par la suite, et de la stratégie d'interprétation. Il n'y a donc pas de méthodes meilleures que d'autres, mais des méthodes plus ou moins bien adaptées aux buts poursuivis. Néanmoins la nécessité de récupérer rapidement des contours fins et précis, élimine les méthodes de morphologie mathématique qui sont trop imprécises et les méthodes stochastiques et variationnelles qui sont coûteuses en temps de calcul.

Pour notre application dont la précision et la vitesse sont de fortes contraintes, les approches dérivatives semblent être les plus adéquates car elles présentent un bon compromis entre performance et temps de calcul.

2.3.1 Méthodes dérivatives

Les différentes approches existantes se classent suivant la manière d'estimer les dérivées de la fonction d'intensité, parmi elles citons les méthodes par différences finies et les méthodes par filtrage optimal [COC 95], [DER 96], [HOR 95].

2.3.1.1 Dérivation par différences finies

Une image est discrète par nature. Les premières approches ont donc consisté à approximer les dérivées par différence (2.9), (2.10).

$$\nabla_x I(x, y) = I(x, y) - I(x - n, y) \quad (2.9)$$

Ou

$$\nabla_y I(x, y) = I(x + n, y) - I(x - n, y) \quad (2.10)$$

Avec en général $n = 1$.

Ces dérivées sont calculées par convolution de l'image avec un masque de différences. Nous trouvons dans la littérature plusieurs types de masque plus ou moins célèbres parmi lesquels :

Opérateurs de Roberts Roberts approxime les dérivées directionnelles suivant les axes orientés à 45 degrés à l'aide d'une convolution de l'image $I(x, y)$ avec les masques 2×2 suivant :

$$h_0 = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \quad h_1 = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$$

Soient $I_x = I * h_0$ et $I_y = I * h_1$ les images obtenues par convolution de l'image originale I avec les deux masques h_0 et h_1 .

la norme du gradient est donnée par $\|\vec{\nabla I}\| = \sqrt{I_x^2 + I_y^2}$ et la direction du contour par $\theta = \frac{\pi}{4} + \arctan\left(\frac{I_y}{I_x}\right)$.

Opérateurs de Prewitt et Sobel Prewitt et Sobel approximent les dérivées directionnelles suivant les axes O_x et O_y à l'aide d'une convolution de l'image $I(x, y)$ avec les masques 3×3 suivant :

$$h_x = \begin{bmatrix} 1 & 0 & -1 \\ c & 0 & -c \\ 1 & 0 & -1 \end{bmatrix} \quad h_y = \begin{bmatrix} -1 & -c & -1 \\ 0 & 0 & 0 \\ 1 & c & 1 \end{bmatrix}$$

Pour $c = 1$, nous avons les masques de Prewitt et pour $c = 2$ les masques de Sobel.

La direction du contour est donnée par $\theta = \arctan\left(\frac{I_y}{I_x}\right)$.

Dans cette même catégorie nous trouvons les opérateurs de Kirsh et Robinson.

Ces approches sont très populaires mais elles ont l'inconvénient d'être très sensibles aux bruits et leurs performances dépendent de la taille des objets traités. Un exemple de ces méthodes est illustré dans la figure 2.3. Sur cette image représentant une partie d'une culasse, nous constatons que les contours détectés sont mal localisés et mal fermés (ils possèdent de nombreux trous).

Un autre type d'approches repose sur la définition de critères d'optimalité de la détection de contours ; ces critères conduisent à des filtres de lissage optimaux qui sont plus performants que les précédents.

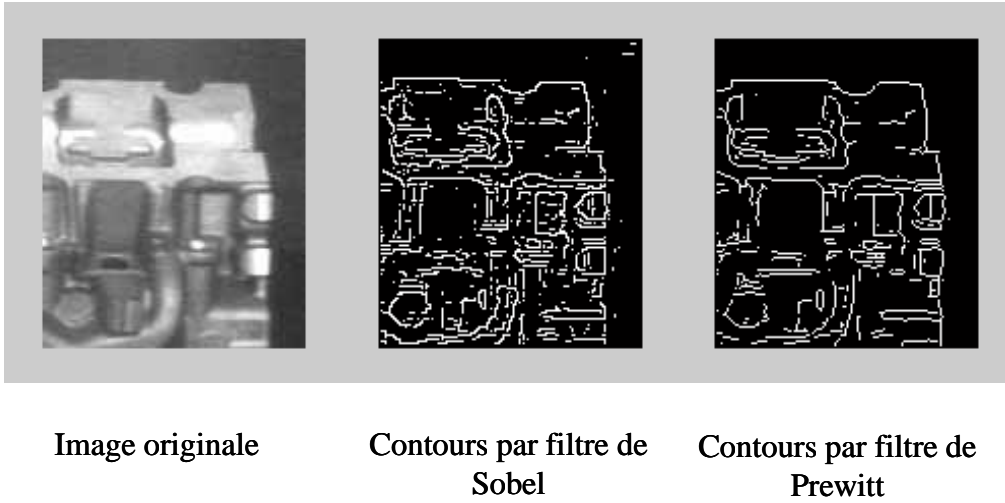


Figure 2.3: Méthodes derivatives par différences finies

2.3.1.2 Dérivation par filtrage optimal

Filtre Gaussien Ce filtre a été initialement introduit par Marr et Hildreth [MAR 80] dans le cadre du calcul du laplacien. Cette approche consiste en une opération de filtrage préalable de l'image par l'opérateur isotrope Gaussien $G(x, y)$ suivi par une localisation des passages par zéro du laplacien $\nabla^2 \hat{I}(x, y)$ où $\hat{I}(x, y)$ représente l'image filtrée par l'opérateur Gaussien $G(x, y)$:

$$G(x, y) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x^2+y^2)}{2\sigma^2}} \quad (2.11)$$

L'opérateur Gaussien est le filtre optimal au regard de la minimisation du produit d'incertitude, défini comme le produit de la variance spectrale par la variance spatiale et c'est la raison essentielle de son utilisation.

Filtre de Shen-Castan Le filtre de Shen et Castan [SHE 86] est obtenu par optimisation d'un critère incluant la détection et la localisation. Le formalisme mathématique développé s'inspire fortement de celui de Canny qui sera présenté juste après. La formule (2.12) représente la solution obtenue, qui est le filtre exponentiel symétrique.

$$h(x) = ce^{-\alpha|x|} \quad (2.12)$$

où

$$c = \frac{1 - e^{-\alpha}}{1 + e^{+\alpha}}$$

est choisi de façon à normaliser le filtre.

Le filtre de dérivation est de la forme (2.13).

$$h'(x) = \begin{cases} de^{-\alpha|x|} & \text{si } x \geq 0 \\ -de^{-\alpha|x|} & \text{sinon} \end{cases} \quad (2.13)$$

avec $d = 1 - e^{-\alpha}$ choisi également de façon à normaliser le filtre.

Le paramètre α détermine la largeur, plus α est petit plus le lissage est important (perte en localisation). Il est aussi performant que le filtre de Canny avec la possibilité de l'implémenter d'une manière récursive

Filtre de Canny Parmi les approches par optimisation, celle développée par Canny [CAN 86] semble être la plus intéressante et mérite une présentation plus en détail.

Nous nous plaçons dans le cas monodimensionnel. Soit $I(x)$ un signal représentant un saut d'amplitude A , noyé dans un bruit Gaussien de variance n_0 . Nous supposons que la détection est effectuée en convoluant le signal $I(x)$ par un filtre impulsionnel $h(x)$, les contours étant caractérisés par les extrema de la sortie du filtre. Les contours envisagés sont des contours type marche et le bruit est supposé blanc (de moyenne nulle). Les critères d'optimalités sont :

Détection : Ce critère revient à chercher $h(x)$ antisymétrique et tel que le rapport signal sur Σ soit maximum.

$$\Sigma = \frac{A \int_{-\infty}^0 h(x) dx}{n_0 \sqrt{\int_{-\infty}^{+\infty} h^2(x) dx}} \quad (2.14)$$

Localisation : Ce critère correspond à la minimisation de la variance σ^2 de la position des passages par zéro et revient à maximiser la localisation Λ définie comme l'inverse de σ .

$$\Lambda = \frac{A |h(0)|}{n_0 \sqrt{\int_{-\infty}^{+\infty} h^2(x) dx}} \quad (2.15)$$

Réponse unique : Ce critère correspond à la limitation du nombre de maxima locaux détectés en réponse à un seul contour. La distance moyenne entre les maxima locaux, notée x_{\max} est alors contrainte à l'égalité suivante :

$$x_{\max} = 2\pi \sqrt{\frac{\int_{-\infty}^{+\infty} h'^2(x) dx}{\int_{-\infty}^{+\infty} h''^2(x) dx}} \quad (2.16)$$

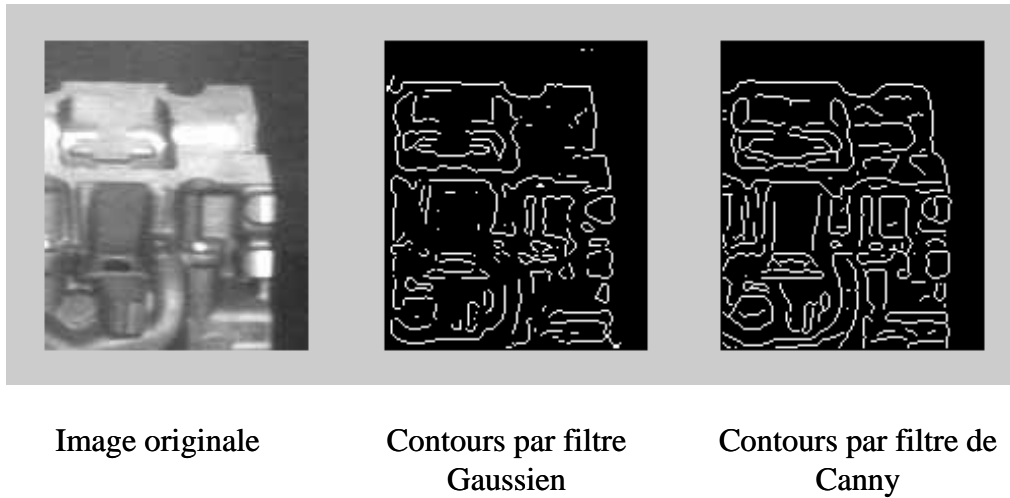


Figure 2.4: Méthodes dérivatives par filtrage optimal

La minimisation de ces critères dans le contexte du modèle proposé débouche sur une équation différentielle dont la solution est de la forme (2.17) :

$$h(x) = a_1 e^{\alpha x} \sin(\omega x) + a_2 e^{\alpha x} \cos(\omega x) + a_3 e^{-\alpha x} \sin(\omega x) + a_4 e^{-\alpha x} \cos(\omega x) \quad (2.17)$$

Cherchant l'opérateur $h(x)$ sous forme de filtre à réponse impulsionnelle finie définie sur $[-W, +W]$ et présentant une pente S à l'origine, Canny a imposé les conditions aux limites suivantes :

$$h(0) = 0 \quad h(W) = 0 \quad h'(0) = S \quad h'(W) = 0 \quad (2.18)$$

Ces 4 conditions aux limites permettent alors de déterminer les coefficients a_1 à a_4 . L'opérateur trouvé n'étant pas simple à mettre en oeuvre, Canny l'approxima au vu de sa forme par la dérivée 1^{ère} d'une gaussienne. Dans la figure 2.4, nous pouvons voir la différence entre les contours obtenus par le filtre Gaussien classique et les contours obtenus par le filtre de Canny.

Nous voyons clairement que pour ce type d'images, les contours détectés par le filtre de Canny sont bien meilleurs que ceux obtenus par le filtre Gaussien. Par le filtre de Canny, on obtient des contours plus longs et connexes par rapport à ceux obtenus par le filtre Gaussien.

Filtre de Deriche Contrairement à l'approche de Canny celle développée par Deriche [DER 87], a consisté en la recherche de l'opérateur optimal sous forme

de filtre à réponse impulsionnelle infinie (RII). Il en résulte la même équation différentielle à résoudre avec cependant de nouvelles conditions aux limites.

$$h(0) = 0 \quad h(+\infty) = 0 \quad h'(0) = S \quad h'(+\infty) = 0 \quad (2.19)$$

Appliquées à la solution générale, ces conditions donnent avec α , ω et S réels positifs comme solution l'opérateur (2.20).

$$h(x) = \frac{S}{\omega} e^{-\alpha|x|} \sin(\omega x) \quad (2.20)$$

Le meilleur indice de performance correspond à la limite de l'opérateur pour ω tendant vers 0.

Cette limite correspond à l'opérateur $g(x)$ donné par (2.21).

$$g(x) = Sx e^{-\alpha|x|} \quad (2.21)$$

La constante S peut être fixée par la contrainte de normalisation suivante :

$$\sum_{n=0}^{+\infty} h(n) = - \sum_{n=-\infty}^0 h(n) = 1$$

ce qui donne :

$$S = \frac{(1 - e^{-\alpha})^2}{e^{-\alpha}} \quad (2.22)$$

L'opérateur $g(x)$ présente des indices de performance bien meilleurs que celui donné par Canny. Il a l'avantage de ne présenter qu'un seul paramètre à régler correspondant à la résolution α adoptée pour détecter les contours. Ses indices de localisation et de détection sont très simplement exprimés en fonction de cette résolution. Diminuer α permet de favoriser la détection au détriment de la localisation et vice-versa.

Un autre avantage de cet opérateur est sa possibilité de mise en œuvre de manière récursive. Ce qui nous permet d'avoir un nombre d'opérations extrêmement faible et surtout indépendant de la valeur de la résolution à laquelle les contours sont détectés. En effet la forme du filtre déterminé par le paramètre α peut varier mais le nombre d'opérations reste le même. Il présente aussi l'avantage de ne pas introduire le bruit de troncature qu'on retrouve dans la plupart des méthodes dérivatives.

Pour ces raisons, nous avons choisi d'utiliser le filtre de Deriche dans notre application pour la détection de contours. Celle-ci sera suivie par une opération de segmentation qui consiste à subdiviser les contours en un ensemble de courbes

analytiques. Cette étape permet donc d'engendrer une description synthétique de l'image plus exploitable qu'un ensemble de pixels.

2.3.2 Segmentation des contours

L'estimation du gradient par le filtre de Deriche, bien que constituant la partie essentielle de la détection de contours, ne fournit pas directement les points de contour et des étapes supplémentaires sont nécessaires pour aboutir à une représentation des images par des segments de droite. Par conséquent, nous avons utilisé les méthodes décrites ci-après en post-traitement du calcul du gradient.

2.3.2.1 Extraction des maxima locaux de la norme du gradient

L'algorithme d'extraction des maxima locaux consiste à comparer la norme du gradient $m(i, j)$ à celle de ses deux voisins m_1 et m_2 situés dans la direction du gradient à une distance unité de part et d'autre de m . Pour plus d'efficacité, nous avons utilisé l'algorithme implémenté à l'INRIA.

2.3.2.2 Seuillage par hystérésis

Les fluctuations provenant du bruit et apparaissant en sortie de l'opérateur de détection de contours font que, quand un seuil unique est utilisé sur l'image des maxima locaux pour obtenir les contours, il y a risque de création de contours discontinus. Nous avons limité ce défaut en employant un seuillage par hystérésis avec deux valeurs de seuil Sh et Sb ($Sh > Sb$). Ce seuillage a été présenté pour la première fois par Canny [CAN 86]. Nous seuillons d'abord avec un seuil haut Sh et nous conservons ensuite toutes les chaînes ayant au moins un point dont la norme du gradient est supérieure au seuil bas Sb . Le grand avantage de cette procédure est qu'elle nous permet d'obtenir plus d'ensembles de points connexes, ce qui facilite les étapes de chaînage et d'approximation prévues en amont de cette chaîne.

2.3.2.3 Chaînage des contours

La description matricielle des contours, fournie après le seuillage par hystérésis, n'est pas suffisante. Il est alors nécessaire de passer à une description sous forme de liste de contours.

Le chaînage permet par codage local de stocker les pixels retrouvés afin de former des ensembles connexes qui caractérisent le passage d'un pixel à son successeur. Ceci est réalisé par les techniques de balayage d'image à l'aide de

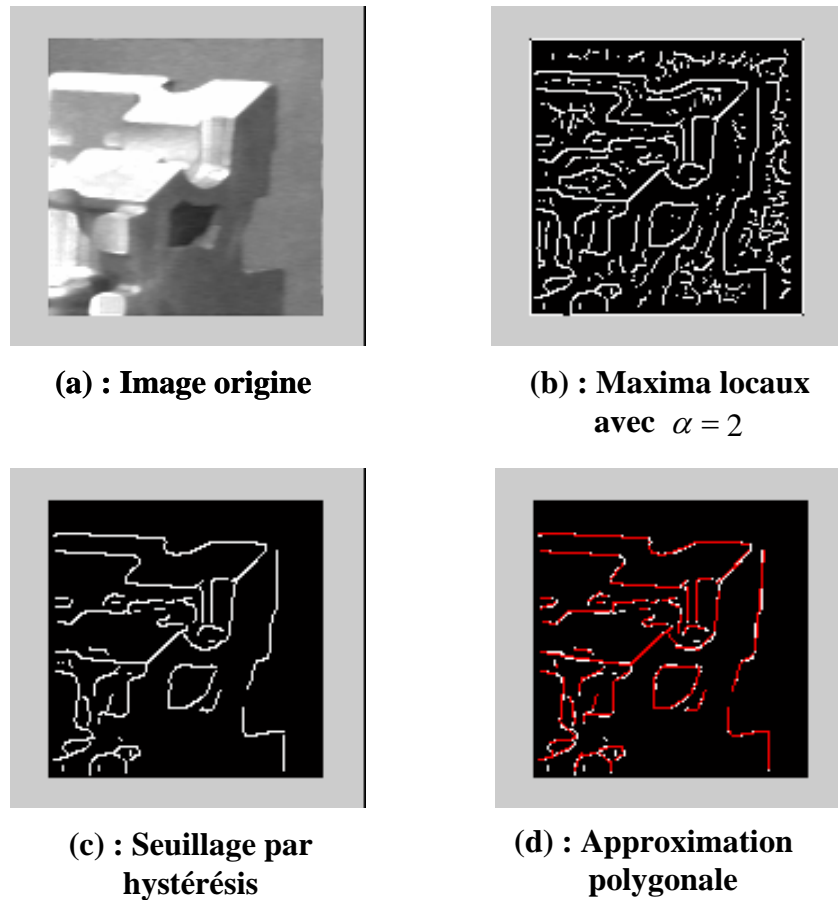


Figure 2.5: Extraction de segments

masques. Nous obtenons ainsi une suite de codes locaux donnant le codage du contour dans son intégralité. L'algorithme de chaînage utilisé [COC 95] permet aussi d'éliminer les points isolés et aussi d'amincir le contour.

2.3.2.4 Approximation Polygonale

La finalité de notre processus est de comparer deux images. La dernière étape consiste alors à quantifier l'information contenue dans les contours et ceci passe par une approximation polygonale. Il existe dans la littérature plusieurs approches possibles et nous avons choisi d'utiliser la méthode dite de découpage-fusion. Cette méthode procède à un découpage de la chaîne puis à une évaluation des paires de segments consécutifs pour éventuellement les fusionner. Les paramètres des segments sont estimés par la méthode des moindres carrés présentée par Pavlidis [PAV 77]. Cette méthode nous permet de résoudre certains problèmes,

tels que, la forte sensibilité au bruit et l'erreur de localisation. Les résultats de ces différentes étapes sont représentés sur la figure 2.5.

Nous avons appliqué le filtre de Deriche sur une vue partielle d'une culasse (figure 2.5.a). La valeur du paramètre qui nous semble donner un meilleur résultat est $\alpha = 2$. La détection des maxima locaux dans la direction du gradient nous donne le résultat de la figure 2.5.b, nous trouvons naturellement plusieurs maxima qui ne correspondent pas au contour de l'objet. Après le seuillage par hystérésis, nous récupérons des ensembles de pixels connexes représentant le contour de l'objet, néanmoins des contours parasites subsistent après ce traitement (figure 2.5.c).

La figure 2.5.d représente le résultat de l'approximation polygonale (en rouge) superposé sur l'image des contours (blanc). Nous constatons que les segments approximatés sont bien localisés et que des points isolés, correspondant à des parasites, ne sont pas pris en compte. Ce qui montre la faculté de filtrage de l'approximation polygonale.

2.4 Détection des points de forte courbure

Toujours dans l'optique de superviser une séquence d'usinage par l'application d'un processus de vision, nous avons mis en œuvre, outre l'extraction de segments, une détection de Points de Forte Courbure (PFC). Comme pour les contours, les PFC sont très importants et représentent des informations très pertinentes pour la vision. Ils peuvent être utilisés, par exemple, pour l'identification d'objets dans une scène, pour l'appariement en stéréovision, pour la mesure des vecteurs de déplacement ... etc.

Plusieurs approches pour résoudre le problème de la détection de points de forte courbure peuvent être trouvées dans la littérature. Elles peuvent être divisées en deux groupes :

- Les approches du premier groupe, basées sur le contour, effectuent d'abord une extraction des contours. Ensuite, plusieurs solutions sont possibles pour extraire les PFC. Asada et Brady [ASA 86] calculent les changements de courbure. Médioni et Yasumoto [MED 87] effectuent une approximation des contours par des B-splines, puis recherche les extrema de courbure ; Horaud et al. [HOR 90] effectuent une approximation polygonale et les points d'intérêt sont donnés par les points d'intersection et d'inflexion des segments calculés.
- Les approches du second groupe, consistent à travailler directement sur les images en niveaux de gris. Plusieurs méthodes ont été proposées qui sont

basées, soit sur des méthodes heuristiques, soit sur le calcul de courbures des surfaces. Dans la section suivante nous allons décrire brièvement quelques unes de ces méthodes.

2.4.1 Analyse de quelques approches classiques

Il existe un grand nombre de méthodes de détection de PFC. Dans ce paragraphe nous exposerons sommairement les approches les plus connues. Les analyses comparatives données dans chaque paragraphe sont issues des travaux de [DER 93], [SMI 95] et [JOL 98].

Dans ce paragraphe nous considérons les notations suivantes : pour une image $I(x, y)$, I_x représente la dérivée première directionnelle suivant x , I_y représente la dérivée première directionnelle suivant y , I_{xx} représente la dérivée seconde suivant x , ...

2.4.1.1 Beaudet

Beaudet dans [BEA 78] a proposé un opérateur invariant par rotation qu'il nomme Det (2.23). Il provient du développement en série de Taylor de l'image $I(x, y)$, qui est considérée comme une surface d'élevation des intensités de niveau de gris. Cette opérateur correspond au déterminant de la matrice Hessienne définie par (2.24) et qui est très proche de la courbure Gaussienne (2.25).

$$Det = I_{xx}I_{yy} - I_{xy}^2 \quad (2.23)$$

$$H = \begin{bmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{bmatrix} \quad (2.24)$$

La détection est basée sur le seuillage des valeurs absolues des extrema de l'opérateur Det .

L'angle n'est pas localisé précisément car il y a un déplacement engendré par la déviation standard du processus de filtrage. Par contre, cette approche est plus stable que celle de Rosenfeld que nous présenterons plus loin.

2.4.1.2 Dreschler et Nagel

Cette méthode, exposée dans [DRE 82], s'inspire fortement de celle de Beaudet et elle s'applique de la manière suivante :

- Calcul de la courbure Gaussienne (2.25).

$$K_{\min}K_{\max} = \frac{Det}{1 + I_x^2 + I_y^2} \quad (2.25)$$

- Sélection des extrema de la courbure Gaussienne : courbures positives et négatives (elliptique E et hyperbolique H).

- Appariement de chaque max positif E (point elliptique) avec le point H (maximum hyperbolique).

- Sélection du point T pour lequel la courbure principale passe par 0 entre E et H .

Le point T ainsi trouvé ne correspond pas à la position exacte de l'angle, car cette position évolue sur la bissectrice de l'angle dans l'échelle spatiale. La précision de localisation des angles obtenus par cette méthode subit la même délocalisation que celle obtenue par l'approche de Beudet.

2.4.1.3 Kitchen et Rosenfeld

Ils proposent dans [KIT 82] d'utiliser un opérateur appelé "cornerness" (2.26), il n'est autre que le produit de la courbure par la norme du gradient, qui est en fait une représentation explicite de la dérivée seconde directionnelle dans la direction orthogonale au gradient.

$$K = \frac{I_{xx}I_y^2 + I_{yy}I_x^2 - 2I_{xy}I_{yx}}{I_x^2 + I_y^2} \quad (2.26)$$

Les maxima locaux de cet opérateur représentent les angles dans une image. Les angles détectés par cet opérateur sont très mal localisés notamment pour les coins dont l'angle est inférieur à 45° .

2.4.1.4 Noble

Noble dans [NOB 88] a essayé de donner une formulation théorique pour le problème de détection de PFC par la géométrie différentielle. Elle a présenté un opérateur (2.27) basé sur les dérivées premières de l'image lissée.

$$C_p = \frac{Trace(\hat{C})}{Det(\hat{C})} \quad (2.27)$$

où \hat{C} est la matrice (2.28):

$$\hat{C} = \begin{bmatrix} \hat{I}_x^2 & \hat{I}_x \hat{I}_y \\ \hat{I}_x \hat{I}_y & \hat{I}_y^2 \end{bmatrix} \quad (2.28)$$

Avec \hat{I} , image de I après l'opération de lissage.

Comme pour les précédentes méthodes, les coins sont détectés par les maxima locaux de l'opérateur (2.27). Noble a précisé que ce détecteur d'angle est adapté pour les doubles jonctions (angles) mais que sa performance n'est pas garantie pour les jonctions d'ordre supérieure (sommets ...). Par contre la localisation des angles est meilleure par rapport aux méthodes citées auparavant.

2.4.1.5 Harris

Le détecteur d'angle développé par Harris et Stephen [HAR 88] est une version légèrement modifiée du fameux détecteur d'angle appelé 'Plessey Corner detector' et développé par Moravec [MOR 77].

Cet opérateur R est défini par :

$$R(x, y) = Det[\hat{C}] - k.Trace^2[\hat{C}]$$

avec

$$\hat{C} = \begin{bmatrix} \hat{I}_x^2 & \hat{I}_x \hat{I}_y \\ \hat{I}_x \hat{I}_y & \hat{I}_y^2 \end{bmatrix}$$

Comme pour le filtre de Noble \hat{I} représente l'image I filtrée par une opération de lissage.

Harris donne une valeur de k de 0.04 pour effectuer une distinction par rapport aux contours à forts contrastes. Par contre il ne justifie pas l'utilisation de cette valeur, elle est déterminée empiriquement et montre seulement que le nombre de PFC détectés dépend fortement de ce paramètre.

2.4.2 Méthode implémentée

Parmi les méthodes de détection de PFC existantes, la méthode de Harris procure, selon Jolion [JOL 98], de meilleurs résultats suivant deux importants critères :

- Robustesse par rapport aux rotations et aux changements d'échelles de la caméra.
- Robustesse par rapport aux variations d'éclairage.

En plus de ces deux avantages, le filtre de Harris est robuste au bruit codé (image compressée en 'jpeg' puis décompressée) [DER 93] et c'est l'opérateur qui convient le mieux aux images couleurs [GOU 00]. Par contre, il est moins robuste pour les bruits additifs impulsionsnels, à cause de l'utilisation trop importante des dérivées.

Pour notre application l'extraction des PFC se fera à l'aide de ce filtre. Contrairement à Harris qui a utilisé le filtre de Sobel pour estimer les dérivées directionnelles, nous avons utilisé le filtre de Deriche qui présente les avantages cités précédemment. A la sortie de ce filtre, les valeurs obtenues sont positives au voisinage d'un coin, négatives au voisinage d'un contour et faibles dans une région d'intensité constante. La détermination des points de forte courbure est alors obtenue par seuillage sur les extrema. Les résultats obtenus à la sortie de l'opérateur de Harris sont représentés sur la figure 2.8. Elle représente la sortie de l'opérateur de Harris appliquée au rectangle de l'image 2.6 après le seuillage du résultat brut illustré sur la figure 2.7. Les pics représentent les valeurs maximum de cet opérateur et ils sont localisés au niveau des coins du rectangle.



Figure 2.6: Image originale

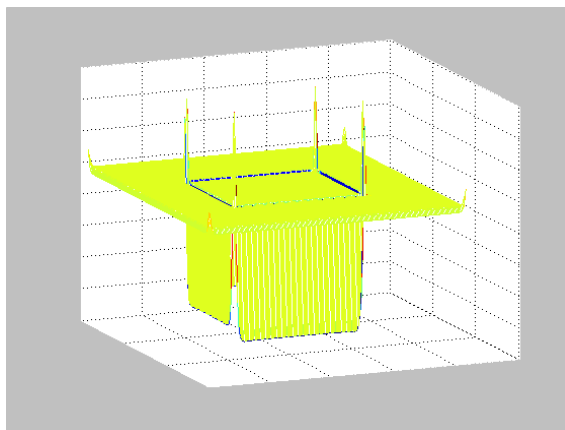


Figure 2.7: Sortie du filtre de Harris avant seuillage

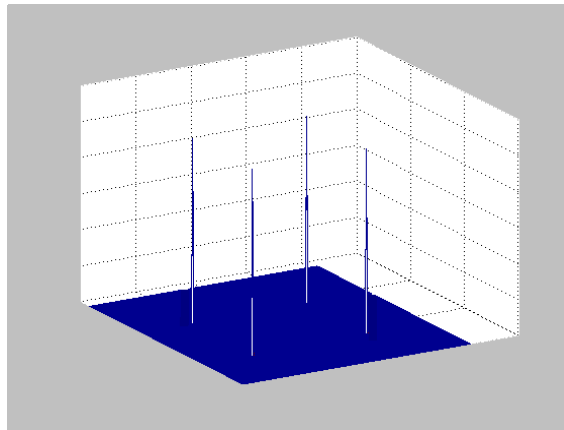


Figure 2.8: Sortie du filtre de Harris après seuillage

Nous rappelons que les PFC détectés serviront au calcul d'un coefficient de corrélation entre deux images et que le calcul de ce degré de ressemblance passe par la mise en correspondance efficace de ces PFC. Cette étape est rendue difficile dans le cas où il y a plusieurs candidats. La phase d'appariement sera entachée d'erreurs, ce qui entraîne un coefficient de corrélation irréaliste.

Ce problème est essentiellement dû aux types d'objets utilisés et au fait que les évolutions d'un contour dans le plan d'une image sont discrètes. Elles sont caractérisées par des changements brutaux de direction (à 45° ou à 90°). Sur une image réelle un simple seuillage ne suffit pas pour extraire les bons PFC. La tâche est rendue plus difficile dans le cas d'images de pièces mécaniques brutes car souvent les surfaces de telles pièces sont rugueuses et présentent de fortes irrégularités.

La solution consiste donc à faire un lissage afin d'adoucir les transitions, de détecter les maxima locaux dans un voisinage 5×5 et finalement d'effectuer un seuillage afin de ne retenir que les PFC les plus pertinents. Nous obtenons les résultats suivants sur une image réelle 2.9. La figure 2.10 montre le comportement du filtre de Harris sur cette image. Les pics de la figure 2.11 représentent les maxima de la figure 2.10 qui subsistent après les traitements sus-cités. Sur la figure 2.12, nous montrons les PFC ainsi détectés sur l'image.

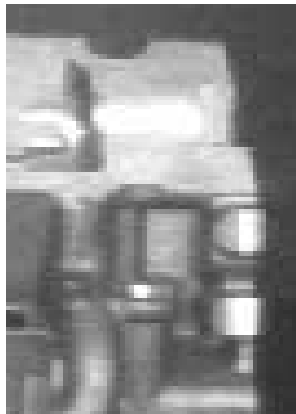


Figure 2.9: Image réelle

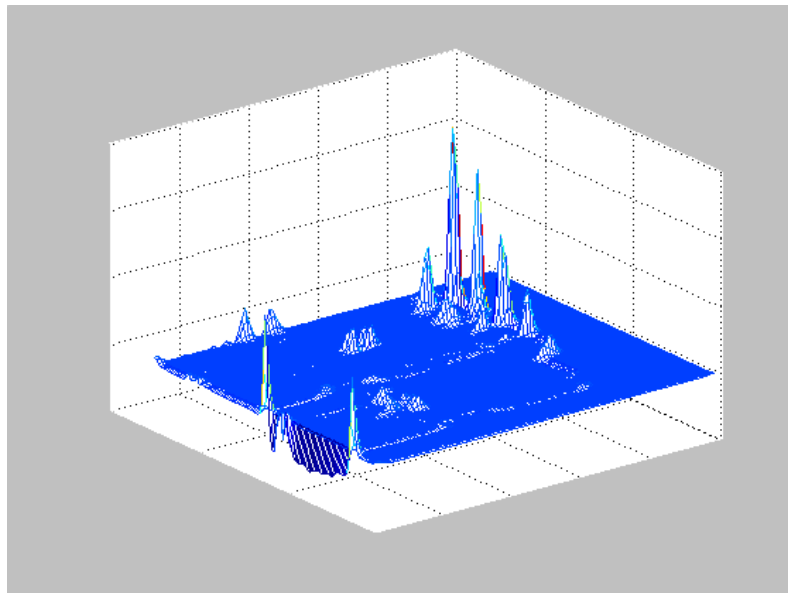


Figure 2.10: Opérateur de Harris avant traitement

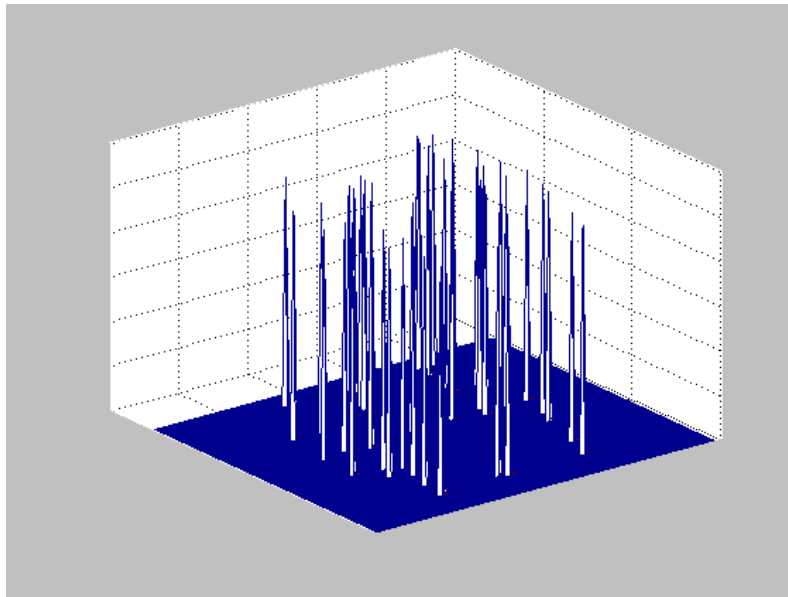


Figure 2.11: Filtre de Harris après seuillage

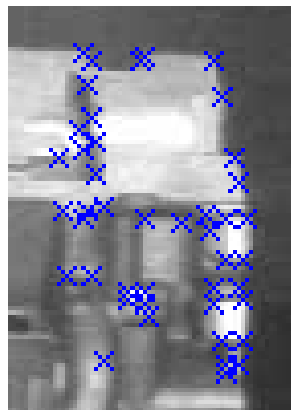


Figure 2.12: Points de forte courbure

Notre intérêt pour la détection de PFC provient du fait qu'il s'agit d'une solution simple pour comparer les positions de deux objets. En effet les PFC sont plus fiables que les contours, car ils sont plus stables par rapport aux différents bruits et ils ne nécessitent pas de post-traitements (chaînage, ...).

Nous avons saisi une image d'une pièce industrielle "culasse" avec une caméra, de focale de 8 mm, placée à 1 m de la pièce. Nous avons lancé la détection des PFC sur cette image, considérée comme la référence. Ensuite, à l'aide d'un banc de mesure, nous avons effectué plusieurs déplacements de la culasse, des

translations allant de 0.1 mm à 1mm, et saisi plusieurs images. Nous avons détecté les nouvelles coordonnées pixel des PFC de ces images et les avons comparé avec ceux de l'image de référence. Nous obtenons les résultats suivants :

<i>Déplacement</i>	<i>0,1 mm</i>	<i>0,2 mm</i>	<i>0,5 mm</i>	<i>0,8 mm</i>	<i>1 mm</i>
<i>% de PFC déplacés</i>	4,7 %	17,5 %	52 %	70,5%	86,7 %

Nous constatons clairement que les PFC sont sensibles à un déplacement faible. Sachant que la tolérance exigée par le constructeur et de l'ordre de 1 mm, nous pouvons considérer qu'à partir de 10% de PFC non appariés, la pièce est mal positionnée.

2.4.3 Résultats

Dans ce paragraphe nous montrerons le comportement de la restauration par diffusion euclidienne par rapport à l'extraction d'amers géométriques. Les figures 2.13 et 2.14 montrent le résultat de la segmentation des contours, un sur l'image initiale sans restauration et l'autre sur l'image restaurée par diffusion euclidienne. Les deux gradients sont calculés à l'aide du filtre de Deriche (avec $\alpha = 2$). Le nombre des itérations pour la restauration est de $n = 5$. Nous distinguons clairement qu'après l'approximation polygonale, il y a moins de contours parasites et surtout moins de trous. Ceci est encore plus visible pour le contour extérieur pour lequel nous obtenons des segments plus longs et moins nombreux.

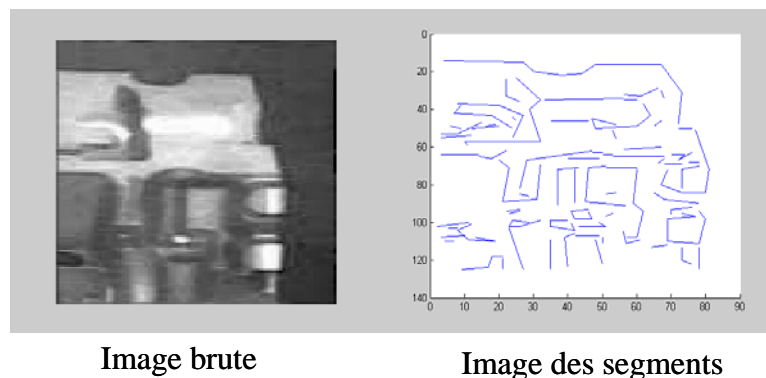


Figure 2.13: Segments avant restauration

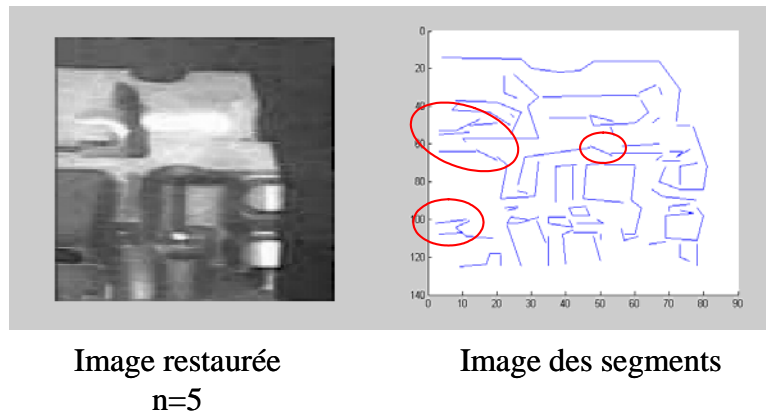


Figure 2.14: Segments après restauration

Dans les figures 2.15 et 2.16, nous avons appliqué l'opérateur de Harris sur deux images. La première est une image brute sans restauration et la seconde est une image restaurée par l'opérateur d'Alvarez. Nous rappelons que malgré les seuillages appliqués à la sortie du filtre de Harris, des PFC qui ne correspondent pas à des éléments structurels subsistent. Nous remarquons qu'à partir de 5 itérations quelques uns de ces faux PFC disparaissent, et à partir de 10 itérations ce sont les vrais PFC qui commencent à disparaître et ceux qui restent subissent une importante délocalisation.

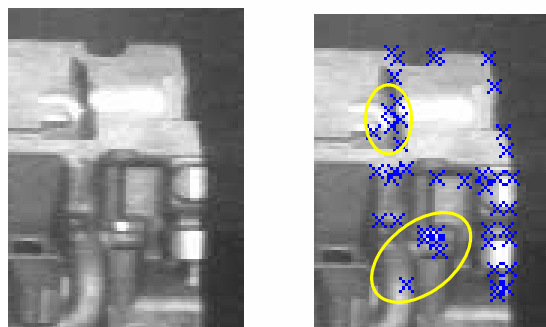
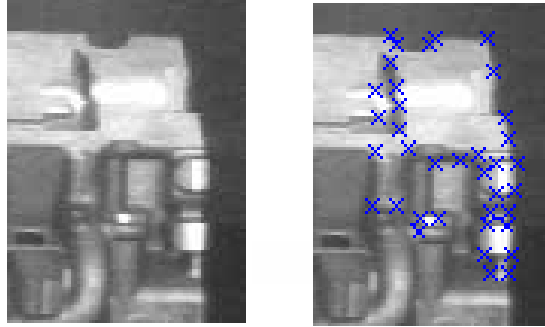


Figure 2.15: PFC avant restauration

Figure 2.16: PFC après restauration $n = 5$

2.5 Mesures de similarité

Il s'agit, comme nous l'avons précisé précédemment de comparer deux images et de détecter si les objets qui y figurent sont similaires du point de vue de leurs formes et de leurs positions spatiale.

Pour ce faire nous avons utilisé des algorithmes de traitement d'images afin d'extraire des attributs géométriques. Nous avons extrait des segments qui conservent les informations structurales des objets et des PFC dont la contribution réside surtout dans la détection de la position de l'objet. En plus de l'utilisation de ces deux informations, nous avons voulu introduire une mesure photométrique pour prendre en compte les informations de luminance globale des deux images.

Trois mesures de similarité entre une image modèle et une image à traiter, ont ainsi été élaborées qui vont définir des paramètres de classification pour la phase de reconnaissance de formes.

2.5.1 Coefficient de corrélation des segments

A partir d'une image de référence, nous extrayons les segments par le filtre de Deriche et approximation polygonale. Ensuite nous saisissons une image et nous relançons les mêmes filtres avec les mêmes paramètres.

Le coefficient de corrélation des segments $CSEG$ (2.29) est déterminé par le nombre de segments appariés $Nb_app(S_r, S_1)$ sur le nombre total de segments.

$$CSEG = \frac{Nb_app(S_r, S_1)}{Card S_r + Card S_1} \quad (2.29)$$

où S_r et S_1 représentent respectivement l'ensemble de segments de l'image de référence et de l'image acquise. La mise en correspondance des segments est

réalisée en comparant leurs orientations et les coordonnées de leurs extrémités. Soit deux segments $s_r(P_1, P_2, \theta_R) \in S_r$ et $s_1(P_3, P_4, \theta_1) \in S_1$ où $P_{i=1..4}(x_i, y_i)$ sont les extrémités et θ_R, θ_1 les orientations. Les segments s_r et s_1 seront considérés comme étant identiques si $|x_1 - x_3| \leq 3$ et $|x_2 - x_4| \leq 3$ et $|y_1 - y_3| \leq 3$ et $|y_2 - y_4| \leq 3$ et si $|\theta_R - \theta_1| < 10$. Les valeurs de ces tolérances ont été choisies empiriquement.

2.5.2 Coefficient de corrélation des PFC

Pareillement que pour les segments, nous extrayons les PFC par le filtre de Harris sur une image de référence. Ensuite nous relançons le même opérateur avec les mêmes paramètres sur l'image qui sera testée. Le coefficient de corrélation des points de forte courbure *CPFC* (2.30) est déterminé par le nombre de PFC appareillés $Nb_app(Pfc_r, Pfc_1)$ sur le nombre total de PFC.

$$CPFC = \frac{Nb_app(Pfc_r, Pfc_1)}{Card Pfc_r + Card Pfc_1} \quad (2.30)$$

où Pfc_r et Pfc_1 représentent respectivement l'ensemble des PFC de l'image de référence et de l'image acquise. Deux PFC $P_1(x_1, y_1)$ et $P_2(x_2, y_2)$ sont considérés comme étant similaires si $|x_1 - x_2| < 2$ et $|y_1 - y_2| < 2$.

2.5.3 Coefficient de corrélation des niveaux de gris

Pour comparer deux nombres, nous pouvons utiliser leur écart, c'est-à-dire, la valeur absolue de leur différence par contre pour deux images de mêmes tailles, il n'existe pas une solution unique pour cette comparaison. C'est pourquoi nous trouvons une grande variété de mesures de similarité. Nous présentons ici les principales en nous basant sur celles appliquées par Moravec [MOR 80] et Anadan [ANA 87] pour le suivi par corrélation, ainsi que sur l'étude comparative réalisée par Aschwanden [ASC 92]. Dans la suite du paragraphe, les mesures de similarité sont calculées pour deux images I_1 et I_2 de tailles $(m \times n)$ et de moyenne \bar{I}_1 et \bar{I}_2 .

La similarité de deux images peut être vue comme l'extension de celle de deux nombres : elle est choisie comme étant la somme des écarts des valeurs des pixels de ces deux images. Plus cette somme est proche de zéro et plus les images sont similaires. Nous la notons *SAD* (Sum of Absolute Differences) et elle se calcule selon la formule (2.27).

$$SAD = \frac{1}{m \times n} \sum_{i=1}^m \sum_{j=1}^n |I_1(i, j) - I_2(i, j)| \quad (2.31)$$

En pratique, il est plus intéressant de sommer les carrés des différences d'intensité des pixels car cela fait intervenir une multiplication moins coûteuse en temps que le calcul d'une valeur absolue. Nous parlons alors de *SSD* (Sum of Squared-Differences) (2.32).

$$SSD = \frac{1}{m \times n} \sum_{i=1}^m \sum_{j=1}^n (I_1(i, j) - I_2(i, j))^2 \quad (2.32)$$

Nous remarquons que si la caméra effectue une mesure photométrique exacte, et si les conditions lumineuses restent inchangées entre les différentes images, ces formules sont idéales.

Malheureusement, la réponse de la caméra varie en fonction de la position de la cible dans son champ. Ceci est dû aux distorsions de son optique et à certains phénomènes de la numérisation. De même, les conditions lumineuses englobants la cible varient en fonction du temps et de sa position.

D'autres mesures de similarité ont été conçues afin de limiter l'effet de ces distorsions. La corrélation normalisée (Zero-mean Normalized Cross-Correlation) (2.33) est une mesure de similarité qui présente une fiabilité supérieure aux autres mesures particulièrement sur les variations lumineuses car elle suppose une relation affine entre les intensités lumineuses des deux images. C'est pour cette raison que nous l'avons incluse comme paramètre dans notre méthode de reconnaissance.

$$ZNCC = \frac{\sum_{i=1}^m \sum_{j=1}^n (I_1(i, j) - \bar{I}_1) \times (I_2(i, j) - \bar{I}_2)}{\sqrt{\sum_{i=1}^m \sum_{j=1}^n (I_1(i, j) - \bar{I}_1)^2 \times \sum_{i=1}^m \sum_{j=1}^n (I_2(i, j) - \bar{I}_2)^2}} \quad (2.33)$$

2.6 Conclusion

Nous avons vu dans ce chapitre comment extraire des informations à partir d'une paire d'images, qui serviront à établir un diagnostic lors d'un processus industriel.

Nous avons donc opté pour une solution à base de traitement d'images de bas niveau, à savoir la détection de segments et de points de forte courbure et pour rendre ces entités géométriques robustes et fiables, nous avons effectué préalablement une restauration des images.

Pour réaliser ces étapes, nous avons fait un tour d'horizon des différentes approches existantes et nous avons choisi d'utiliser celles faisant intervenir le formalisme mathématique des équations aux dérivées partielles. Pour avoir un bon compromis entre la qualité des traitements et le temps de calcul, nous avons utilisé l'opérateur d'Alvarez pour la restauration et les filtres de Deriche et Harris

pour extraire les segments de contours et les points de forte courbure respectivement. Nous avons montré les bases théoriques de ces méthodes et aussi leur implémentation en illustrant leurs comportements sur des images réelles.

Enfin, nous avons construit trois coefficients de corrélations *CSEG*, *CPFC* et *ZNCC* qui nous serviront de paramètres dans la phase de reconnaissance par classification que nous présenterons au chapitre suivant.

Chapitre 3

METHODES DE CLASSIFICATION

3.1 Introduction

Le cadre général de l'étude est l'analyse de pièces industrielles de type mécanique. L'objectif étant de comparer une pièce à un modèle et de décider si nous pouvons considérer que la pièce vue est identique à la référence. Dans notre cas la notion de similitude inclut l'absence de défauts et la position équivalente à la référence. En effet nous souhaitons superviser le bon fonctionnement d'une manœuvre d'usinage et de tels processus ne peuvent pas être lancés si les conditions initiales ne sont pas respectées. Pour cet objectif, nous avons défini dans le chapitre précédent trois coefficients de corrélation, *CPFC*, *CSEG* et *ZNCC*, entre l'image de référence et l'image acquise.

Nous allons maintenant commencer à estimer leurs performances (leurs pouvoirs à établir un diagnostic). Ensuite, afin de pouvoir évaluer la pertinence et la corrélation de ces paramètres nous effectuerons une Analyse en Composantes Principales (ACP).

Face à la nécessité de mettre au point un système efficace, nous allons tester plusieurs classifieurs. Nous commencerons par une règle empirique ensuite nous testerons un classifieur Bayésien et un autre neuronal et finalement nous essayerons un classifieur hybride qui est une combinaison du Bayésien et du neuronal.

3.2 Saisie des données

Nous avons acquis, sur le site de production et dans des conditions variables de luminosité, de position et de qualité, 20 images d'une pièce issue de l'industrie automobile. Chaque image a été subdivisée en 36 imagerie de taille 128×96

pixels. Nous disposons ainsi d'une base de 720 imagerie. Pour chacune de ces imagerie nous sommes capable de prendre une décision en la comparant à une image de référence. Nous avons pu ainsi diviser cet ensemble en deux classes :

- La classe "0" correspond aux imagerie pour lesquelles il y a une différence entre l'image d'origine (ou de référence) et l'image en cours de test. 146 sous-images ont été affectées à cette classe. Les différences résident dans la présence de défauts, de copeaux ou encore dans des problèmes mineurs de positionnement.
- La classe "1" correspond aux imagerie qui sont jugées identiques à l'image de référence. 574 imagerie ont été affectées à cette classe.

Les figures 3.1.a montrent les images de références, les figures 3.1.b des images sans défauts, qui sont considérées comme appartenant à la classe "1" et les figures 3.1.c présentent des images avec défauts relevant de la classe "0". Nous pouvons constater, en observant les figures 3.1.c, que les images de la classe "0" présentent des différences relativement faibles avec l'image de référence. Nous comprenons donc qu'il nous faut des paramètres robustes et une règle de décision fiable pour détecter ces différences.

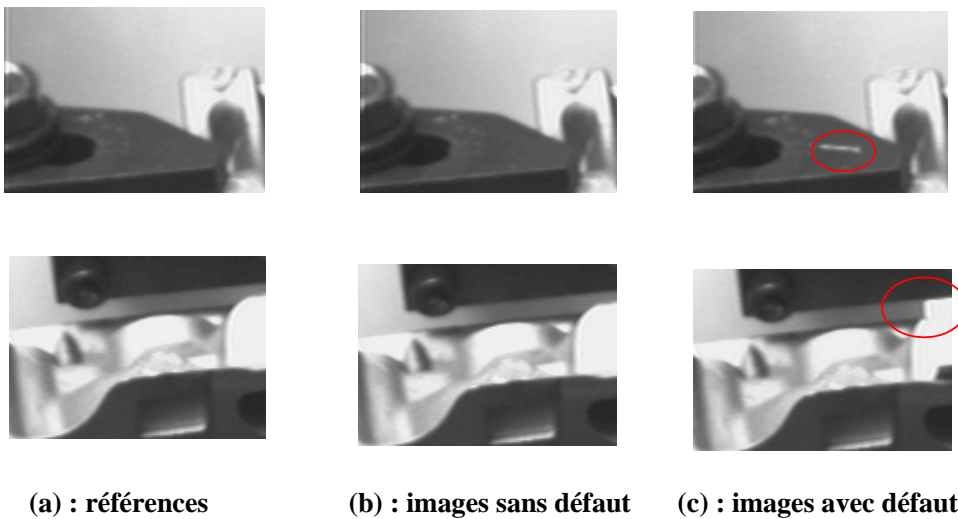


Figure 3.1: Images de la scène réelle

3.3 Méthode empirique

Les trois paramètres qui ont été présentés à la section (3.2) ont pour but d'évaluer la ressemblance entre deux images au sens des points de forte courbure (*CPFC*),

des segments (*CSEG*) et de la corrélation des niveaux de gris (*ZNCC*). Nous recherchons une règle qui utilise ces trois paramètres et délivre une décision du type : l'image testée relève de la classe "0" ou de la classe "1". Du fait de l'objectif industriel de ce travail, cette règle doit minimiser les erreurs de classification de la classe "0" vers la classe "1". En effet, si un appel à l'opérateur est effectué alors que la pièce en place dans le centre d'usinage est correcte, cela se soldera par une perte de temps. Par contre, si nous décidons d'usiner une pièce défectueuse ou mal positionnée, cela peut avoir de graves conséquences sur le centre d'usinage en provoquant d'éventuelles détériorations d'outils. Il faut toutefois chercher à maximiser le pourcentage global de bonne classification. Le problème que nous souhaitons résoudre est un problème de classification de données pour lequel les prédicteurs sont des variables quantitatives. Etant donné que l'on dispose de deux classes, définies a priori, que l'on cherche à différencier nous nous plaçons dans une hypothèse de classification supervisée [DID 82].

Une première approche a été présentée dans [MER 01]. Suite à de nombreuses observations et des discussions avec des experts, une règle empirique a été définie et chacune des 720 images a été classée selon les résultats obtenus. Cette règle est la suivante :

Si ($CPFC > 0,9$ et $ZNCC > 0,9$) alors image affectée à classe "1" ;

Sinon, Si ($CPFC > 0,8$ et $CSEG > 0,7$ et $ZNCC > 0,85$)

alors image affectée à classe "1" ;

Sinon image affectée à classe "0".

Après plusieurs tests et observations nous avons remarqué le pouvoir discriminant du coefficient *CPFC*. En effet les PFC sont des primitives ponctuelles robustes par rapport aux occlusions. Conceptuellement, ils représentent les extrémités de la pièce (notamment après avoir réalisé les différents filtrages). Nous avons alors déduit que si il y'a plus de 10% de PFC d'une image qui ne possèdent pas d'homologues dans l'image de référence alors la pièce est défectueuse ou bien mal positionnée. De même pour le coefficient *ZNCC*, si ce dernier est inférieur à 0,9, nous sommes nécessairement en présence d'un objet différent à l'objet de référence. Nous rappelons que *ZNCC* est un coefficient robuste aux changements d'éclairage et la pièce à usiner est cloîtrée dans une enceinte où la lumière extérieure n'influe pas beaucoup.

Nous avons constaté par la suite qu'il existe des images ne possédant pas beaucoup de PFC et ceci pose un problème pour la première condition. Par exemple dans le cas d'une image de référence dont le nombre de PFC est de 4, pour des raisons diverses (de filtrage excessif par exemple), il arrive que sur la nouvelle image nous ne trouvons que 3 PFC. Ceci conduit à un coefficient *CPFC* égale à 0.85 qui va aboutir à un faux diagnostic. C'est pour cette raison que nous avons rajouté la nouvelle condition qui prend en compte le coefficient *CSEG*.

Une telle règle conduit à un taux global de bonne classification de plus de 94% ce qui est satisfaisant pour une telle démarche. Mais, si la classe "1" est très bien identifiée (97,85%), cela se fait au détriment de la classe "0" pour laquelle presque 10% d'images sont mal classées.

Le second travail est l'étude exhaustive des résultats de classification obtenus avec *CPFC*, *CSEG* et *ZNCC*. Nous avons d'abord étudié les résultats fournis par les paramètres isolés en utilisant une règle très simple du type :

Si (Paramètre > seuil) alors image affectée à classe "1" ;

Sinon image affectée à classe "0".

Nous avons fait varier le seuil par pas de 0,1 de 0,49 à 0,99 sachant qu'au dessous d'un seuil de 0,49, la pièce est forcément défectueuse. Pour chaque paramètre nous avons relevé la valeur du seuil qui conduisait au meilleur résultat. Puis en utilisant les valeurs de seuils issues ou proches de cette première expérience, nous avons combiné deux paramètres en utilisant l'opérateur "**ET**" et enfin nous avons combiné les trois paramètres. Nous avons calculé le taux de bonne classification pour chaque classe ainsi que pour l'ensemble des 720 images. Ce dernier taux prend en compte la disparité numérique de la classe "0" et de la classe "1". Le tableau 3.1 présente les résultats obtenus.

Nous remarquons dans ce tableau que l'on atteint un taux total de bonne classification un peu meilleur qu'avec la règle décrite précédemment (94,72%) tout en ayant ramené le taux d'erreur de la classe "0" à environ 6%. Ce résultat est obtenu en utilisant conjointement *CPFC* et *ZNCC*. En combinant les trois paramètres il est même possible de faire chuter le taux d'erreur de la classe "0" à 5%.

Comme il est impératif d'augmenter ce résultat, une analyse sur le comportement des trois paramètres est nécessaire. Pour guider ce travail, nous nous sommes inspirés de travaux précédents réalisés au LSC par H. Loaiza [LOA 99].

Cas	Valeurs des seuils (résultat optimal)	%Total	%Classe "0"	%Classe "1"
<i>CPFC</i>	$CPFC > 0.79$	94,72	89.73	95.9
<i>CSEG</i>	$CSEG > 0.49$	86.53	38.36	98.%
<i>ZNCC</i>	$ZNCC > 0.99$	93.47	93.84	93.38
<i>CPFC</i> et <i>ZNCC</i>	$CPFC > 0.79$ and $ZNCC > 0.98$	94.72	93.84	94.9
<i>CPFC</i> et <i>CSEG</i>	$CPFC > 0.79$ and $0.4 < CSEG < 0.45$	94.58	89.73	95.82
<i>ZNCC</i> et <i>CSEG</i>	$ZNCC > 0.99$ and $0.4 < CSEG < 0.45$	93.33	93.84	93.2
<i>CPFC</i> et <i>CSEG</i> et <i>ZNCC</i>	$ZNCC = 0.99$ and $0.5 < CPFC < 0.59$ and $0.4 < CSEG < 0.45$	93.61	95.21	93.2

Tableau 3.1: Analyse exhaustive des paramètres *CPFC*, *CSEG* et *ZNCC*

3.4 Analyse des paramètres

Le but de cette section est d'analyser rigoureusement les trois paramètres (*CPFC*, *CSEG* et *ZNCC*) et de vérifier leur pertinence au sens de la discrimination entre les différentes classes. Cette analyse nous permettra d'examiner leurs taux de corrélation et d'en définir de nouvelles variables si c'est nécessaire. Pour cela nous allons utiliser une méthode statistique descriptive qui va nous faciliter la visualisation et l'analyse préliminaire des données. Cette opération se fera à l'aide d'une ACP.

L'analyse des correspondances, l'analyse canonique, l'analyse factorielle discriminante sont aussi des méthodes factorielles conduisant à des représentations graphiques et ont de ce fait des traits communs avec l'ACP. Ce qui fait la spécificité de l'analyse en composantes principales est qu'elle traite exclusivement de caractères numériques jouant tous le même rôle. Alors que l'analyse des correspondances traite des caractères qualitatifs, et qu'en analyse canonique comme

en analyse discriminante les caractères sont répartis en groupes bien distincts.

3.4.1 Analyse en composantes principales

L'ACP est une méthode factorielle d'analyse de données multidimensionnelles. Les méthodes factorielles présentent de multiples intérêts lors de l'élaboration de procédures de discrimination. Elles sont principalement utilisées pour la réduction de la dimension de l'espace originel, lorsque celle-ci est importante, et que les variables qui décrivent cet espace sont redondantes. Ainsi que dans un but descriptif, puisqu'elles fournissent un ensemble étendu de données, une représentation graphique et une information synthétique sur l'organisation des individus en classes [CEL 90].

Plus précisément, le but de l'ACP est d'obtenir une représentation d'un nuage d'individus dans un espace de dimension réduite de telle manière que l'inertie portée par cet espace soit la plus grande possible. La principale opération de l'ACP est donc de déterminer les axes principaux d'inertie du nuage de points autour de son centre de gravité. Ces axes principaux d'inertie appelés axes factoriels permettent de représenter les points du nuage sur des espaces de dimensions réduites [CEL 89].

En ce qui nous concerne, l'opération de l'ACP servira à estimer la qualité de représentation des variables (*CPFC*, *CSEG* et *ZNCC*). Nous commençons d'abord à centrer et à réduire ces paramètres car dans la suite nous aurons à traiter des variables hétérogènes. L'opération de l'ACP calcule la matrice des covariances, la diagonalise, en extrait les valeurs et vecteurs propres. Les axes factoriels sont définis comme les vecteurs propres de cette matrice. Les données réduites sont ensuite projetées sur ces axes et les coefficients de corrélation entre facteurs et variables sont alors mesurés par la quantité suivante qui représente le carré du coefficient de corrélation :

$$q_j = \sum_{l=1}^k \cos^2 \alpha_j^l \quad (3.1)$$

où α_j^l est l'angle formé par la variable l et la composante principale j .

3.4.1.1 ACP sur les paramètres *CPFC*, *CSEG* et *ZNCC*

Nous avons réalisé une Analyse en Composantes Principales sur l'ensemble des données en utilisant *CPFC* (*Var1*), *CSEG* (*Var2*) et *ZNCC* (*Var3*). Les résultats obtenus nous ont montré que deux valeurs propres fournissent 90% de l'énergie, ce qui est positif sur la représentativité de ces variables. Par ailleurs

l'axe F1 est constitué par les trois variables dans des proportions quasiment équivalentes. Nous constatons toutefois une corrélation importante des paramètres *CPFC* et *ZNCC* comme l'illustrent les trois cercles de corrélation présentés sur la figure 3.2.

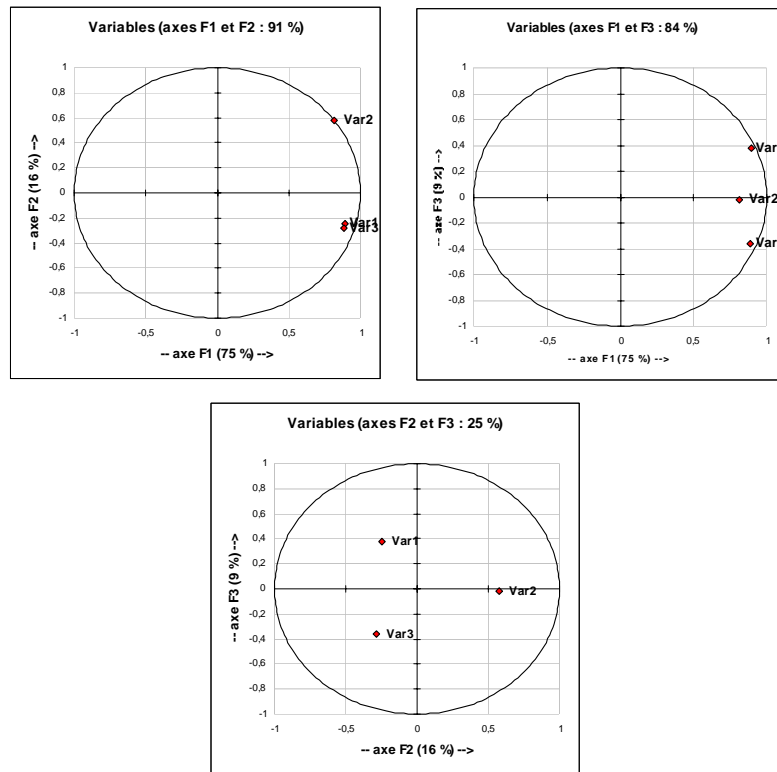


Figure 3.2: Cercles de corrélation issus de l'ACP sur les trois variables

Nous remarquons également sur ces cercles que la variable 2 (*CSEG*) a une contribution nulle pour l'axe 3, contrairement aux deux autres variables. Pour affiner l'analyse du pouvoir discriminant de ces variables, nous avons relancé l'ACP de manière séparée, sur les éléments de la classe "0" et sur ceux de la classe "1". Nous présentons dans les tableaux 3.2, 3.3 et 3.4, les matrices de corrélation issues de ces trois analyses.

Ces résultats illustrent que la corrélation est importante entre les trois variables, spécialement pour *CPFC* et *ZNCC*, pour les éléments de la classe "0". Par contre, pour les éléments de la classe "1", nous pouvons constater dans le tableau 3.4, que cette corrélation est inexistante. Nous validons ainsi le choix de ces trois paramètres qui présentent bien un pouvoir discriminant. Toutefois

nous pensons qu'il est intéressant, dans l'optique d'améliorer les résultats de la classification, d'introduire de nouveaux paramètres.

	Var1	Var2	Var3
Var1	1	0,581	0,728
Var2	0,581	1	0,567
Var3	0,728	0,567	1

Tableau 3.2: Matrice de corrélation

	Var1	Var2	Var3
Var1	1	0,460	0,575
Var2	0,460	1	0,566
Var3	0,575	0,566	1

Tableau 3.3: Matrice de corrélation pour la classe "0"

	Var1	Var2	Var3
Var1	1	0,023	0,076
Var2	0,023	1	0,057
Var3	0,076	0,057	1

Tableau 3.4: Matrice de corrélation la classe "1"

3.4.1.2 Définition et pertinence de nouveaux paramètres

Nous avons réfléchi aux conditions qui pouvaient influencer le résultat. Nous avons alors constaté qu'un certain nombre d'erreurs de classification étaient commises sur des images qui présentaient très peu de points de forte courbure (< 5) ou très peu de segments. En effet, si une seule de ces caractéristiques n'est pas détectée, l'incidence de cette non-détection est très forte sur la valeur des coefficients *CPFC* ou *CSEG*. Nous avons également pensé à prendre en compte la différence du nombre de points de forte courbure ou de segments entre l'image de référence et l'image à classer. Par ailleurs, il nous a semblé nécessaire de faire intervenir les conditions d'éclairage. Nous avons donc introduit la moyenne et l'écart-type des niveaux de gris des images de référence et à classer, ainsi que la différence entre ces deux paramètres. Les 15 paramètres auxquels nous avons aboutis sont présentés dans le tableau 3.5.

Nous avons à nouveau procédé à une analyse en composantes principales en utilisant cette fois ces 15 variables. Il faut cette fois 6 axes, issus des 6 valeurs propres principales, pour obtenir 90% de l'énergie. La figure 3.3 présente les cercles

Nom du paramètre	N° ACP	Nature du paramètre
<i>CPFC</i>	<i>Var1</i>	Coefficient sur les points de forte courbure
<i>CSEG</i>	<i>Var2</i>	Coefficient sur les segments
<i>ZNCC</i>	<i>Var3</i>	Corrélation sur les niveaux de gris
$\neq PFC$	<i>Var4</i>	Différence entre le nombre de points de forte courbure de la référence et de l'image
$\neq SEG$	<i>Var5</i>	Différence entre le nombre de segments de la référence et de l'image
$\neq MOY$	<i>Var6</i>	Différence entre la moyenne des niveaux de gris de la référence et de l'image
$\neq STD$	<i>Var7</i>	Différence entre l'écart-type des niveaux de gris de la référence et de l'image
<i>NPFC_ref</i>	<i>Var8</i>	Nombre de points de forte courbure dans la référence
<i>NPFC_test</i>	<i>Var9</i>	Nombre de points de forte courbure dans l'image
<i>NSEG_ref</i>	<i>Var10</i>	Nombre de segments dans la référence
<i>NSEG_test</i>	<i>Var11</i>	Nombre de segments dans l'image
<i>MOY_test</i>	<i>Var12</i>	Moyenne des niveaux de gris de l'image
<i>STD_test</i>	<i>Var13</i>	Ecart-type des niveaux de gris de l'image
<i>MOY_ref</i>	<i>Var14</i>	Moyenne des niveaux de gris de la référence
<i>STD_ref</i>	<i>Var15</i>	Ecart-type des niveaux de gris de la référence

Tableau 3.5: Présentation des 15 paramètres utilisés pour la classification des images

de corrélation concernant les 3 premiers axes. Nous remarquons sur ces cercles que certaines variables présentent une très forte corrélation. C'est le cas pour les couples (8, 9), (10, 11), (12, 14) et (13, 15). Une corrélation, mais plus faible, apparaît également entre les variables (1, 2, 3), d'une part, et (4, 5, 6, 7) d'autre part. L'analyse des matrices de corrélation obtenues à l'issue de l'ACP sur les 720 images, mais aussi sur les 146 éléments de la classe "0" et les 574 éléments de la classe "1" confirme les fortes corrélations des quatre couples cités précédemment. Nous remarquons sur ces matrices, comme dans le cas à trois variables, le comportement différent des 3 paramètres, *CPFC*, *CSEG* et *ZNCC*, pour les deux classes. Suite à ces différentes analyses, nous avons décidé de rechercher une méthode de classification automatique des images qui ne soit pas dépendante des seuils choisis empiriquement. Nous avons donc décidé de tester deux approches, la méthode bayésienne et la méthode neuronale. Les paramètres retenus pour ces deux approches ont été sélectionnés à partir de l'analyse précédente. Nous avons retenu 7 paramètres, les trois paramètres initiaux ainsi que ceux décrivant la différence d'une valeur entre l'image de référence et l'image à classer. En effet, comme nous pouvons le constater sur la figure 3.3, ces paramètres se situent dans des positions opposées sur les cercles de corrélation de l'ACP.

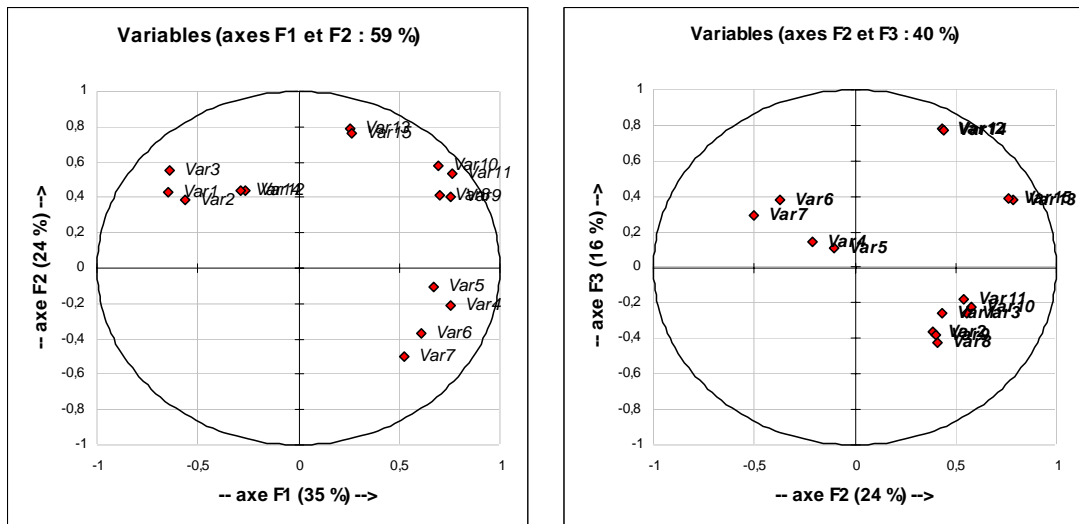


Figure 3.3: Cercles de corrélation issus de l'ACP avec 15 paramètres

3.5 Classification automatique

Le choix du classifieur est très important. Il constitue l'élément de décision dans un système de reconnaissance des formes. Le but des traitements antérieurs était d'extraire l'information essentielle caractérisant les objets et de la présenter sous la forme la plus condensée possible au classifieur.

Il est important pour cela de remarquer que le problème qui nous concerne est un problème de classification supervisée, c'est-à-dire pour lequel l'utilisateur connaît l'information (concernant l'appartenance aux classes des vecteurs d'apprentissage), que le système utilise pour l'apprentissage [DEV 82]. Nous disposons donc d'un ensemble de vecteurs étiquetés, qui vont être présentés au classifieur afin que celui-ci puisse en déduire une règle de décision en optimisant un certain critère. Ce processus est généralement connu sous le nom d'apprentissage inductif. Il dépend de la nature des données [RAD 99] et lorsque celles-ci sont numériques, comme dans notre cas, les méthodes utilisées s'apparentent à l'approche Bayésienne.

Nous rappelons que la principale contrainte dans notre système de vision est le temps de calcul. Pour cette raison nous avons choisi de tester les classifieurs Bayésien et neuronal car ils présentent le meilleur compromis entre temps d'exécution et efficacité. Les analyses de ces méthodes ont été présentées dans [MER 03b].

3.5.1 Discrimination Bayésienne

Le théorème de Bayes fournit un cadre théorique pour la problématique de la classification. Si l'on considère un problème de classes C_1, \dots, C_k , le théorème de Bayes permet de calculer les probabilités a posteriori, connaissant les distributions des observations a priori. (3.2) :

$$P(C_i/x) = \frac{p(x/C_i) P(C_i)}{p_i(x)} \quad (3.2)$$

$P(C_i|x)$ est la probabilité a posteriori d'appartenir à la classe C_i connaissant le vecteur des descripteurs x , $p(x|C_i)$ est la densité de probabilité du vecteur d'exemples $x(x = [x_1, x_2, \dots, x_n])$ dans la classe C_i , $P(C_i)$ est la probabilité a priori de la classe C_i et $p_i(x)$ est la densité de probabilité non conditionnelle définie par (3.3) :

$$p_i(x) = \sum_{i=1}^k p(x/C_i) P(C_i) \quad (3.3)$$

Dans le cas d'un problème de classification, la formule (3.3) définit une règle de décision. La probabilité de mauvaise classification est minimisée en sélectionnant la classe qui a la plus grande probabilité a posteriori.

Ce théorème est au cœur de la problématique de la classification : nous pouvons distinguer (i) les méthodes de classification qui estiment les densités de probabilité pour calculer les probabilités a priori et (ii) les méthodes qui estiment directement les probabilités a posteriori.

Dans notre cas, où il s'agit de décider de l'appartenance d'une image quelconque à l'une des deux classes G_1 et G_2 (classe "0" et classe "1" respectivement), nous allons utiliser la règle de décision optimale de Bayes δ^* qui s'écrit :

Si $h(x) > 1$ alors on affecte x à G_1 .

Si $h(x) < 1$ alors on affecte x à G_2 .

$$\text{Avec, } h(x) = \frac{C_{12}p_1f_1(x)}{C_{21}p_2f_2(x)}.$$

Où p_1 et p_2 sont les probabilités a priori des classes, $f_1(x)$ et $f_2(x)$ sont les fonctions de probabilité des classes G_1 et G_2 , et C_{ij} représente le coût de mauvais classement d'un individu de la classe G_i dans la classe G_j .

A partir de la règle de décision δ^* nous pouvons constater que la surface de discrimination est définie par l'équation $h(x) = 1$.

Cas gaussien : nous considérons les variables étudiées comme des variables aléatoires gaussiennes. Cette hypothèse sera rejetée si les pourcentages de discrimination entre classes ne sont pas satisfaisants. En cela nous suivons une démarche de raisonnement comparable à celle proposé par B. Burstschy et F. Nivelles dans [CEL 90]. En conséquence, nous analyserons la méthode de discrimination bayésienne sous l'hypothèse que les densités $f_l(x)$ sont des lois normales $N_p(\mu_l, \Gamma_l)$ où μ_l et Γ_l sont respectivement le vecteur moyen et la matrice variance-covariance des classes a priori G_l dans R^p .

$$f_l(x) = \frac{1}{(2\pi)^{p/2} |\Gamma_l|^{1/2}} \exp \left[-\frac{1}{2} (x - \mu_l)' \Gamma_l^{-1} (x - \mu_l) \right] \quad (3.4)$$

En appliquant le logarithme à la fonction $h(x)$, l'équation de la surface discriminante peut s'écrire : $\log h(x) = 0$

Soit

$$\log \frac{f_1(x)}{f_2(x)} + \log \frac{p_1 C_{12}}{p_2 C_{21}} = 0$$

En remplaçant $f_l(x)$ nous obtenons l'équation du second degré en x pour la surface discriminante :

$$\log \frac{|\Gamma_2|}{|\Gamma_1|} + x' [\Gamma_2^{-1} + \Gamma_1^{-1}] x + 2 [\mu_1' \Gamma_1^{-1} + \mu_2' \Gamma_2^{-1}] x + \mu_2' \Gamma_2^{-1} \mu_2 - \mu_1' \Gamma_1^{-1} \mu_1 = 2 \log \frac{p_2 C_{21}}{p_1 C_{12}}$$

3.5.1.1 Résultats

Les variables que nous allons utiliser dans cette partie sont donc les 7 premières énoncées dans le tableau 3.5. Puisque nous avons deux classes, nous avons deux coûts à définir, ces coûts représentent les mauvaises classifications d'une classe dans une autre. Pour éviter des choix a priori, nous avons fait varier ces coûts, notés C_{01} et C_{10} , sur un espace important de valeurs (de 0 jusqu'à 10 d'un pas de 0.1). Nous avons ensuite analysé les résultats et recherché quels étaient ceux qui nous conduisaient aux meilleurs taux de classification pour chaque classe. Ces résultats sont présentés dans le tableau 3.6. Si ces résultats sont du même ordre de grandeur que ceux présentés dans le tableau 3.1, nous pouvons malgré tout constater que la méthode bayésienne nous procure quelques améliorations. En effet, nous avons pu atteindre un taux global de bonne classification supérieur à 96%. De plus, dans l'objectif de maximiser le taux global, sous la contrainte de réduire les erreurs de classification de la classe "0" dans la classe "1", nous avons obtenu un taux global de bonne classification supérieur à 95% avec pour la classe "0" un taux de 93,8%, mais un taux pour la classe "1" légèrement supérieur, soit 95,8%. Ce résultat est donc tout à fait encourageant.

Objectif	C_{01}	C_{10}	% classe "0"	% classe "1"	% total
MaxClasse "0"	3 à 3,3	0,1	93,84	95,82	95,42
Max Classe "1"	0,1	> 9,6	86,99	98,43	96,11
Max % total	0,1	1,7 à 3,3	88,36	98,26	96,25

Tableau 3.6: Résultat de la classification bayésienne avec 7 paramètres

Toutefois notre contexte industriel nous oblige à améliorer le taux de classification de la classe "0", car rappelons le, si une pièce avec défauts est considérée comme bonne cela peut avoir des conséquences désastreuses pour la machine. Pour cette raison nous avons choisi d'utiliser un perceptron multicouche, entraîné avec un algorithme d'apprentissage par rétropropagation du gradient.

3.5.2 Approche neuronale

Les réseaux de neurones artificiels ou réseaux connexionnistes sont fondés sur des modèles qui tentent d'expliquer comment les cellules du cerveau et leurs

interconnexions parviennent, d'un point de vue global, à exécuter des calculs complexes. Ces systèmes qui stockent et retrouvent l'information de manière "similaire" au cerveau sont particulièrement adaptés aux traitements en parallèle de problèmes complexes comme la reconnaissance automatique de la parole, la reconnaissance de visages ou bien la simulation de fonctions de transfert.

Dans les réseaux de neurones artificiels, de nombreux processeurs appelés cellules ou unités, capables de réaliser des calculs élémentaires, sont structurés en couches successives capables d'échanger des informations au moyen des connexions qui les relient.

Perceptron multicouche Un réseau de neurones non bouclé est une composition de fonctions réalisées par des neurones formels interconnectés entre eux [MIL 93]. Certaines applications peuvent nécessiter plusieurs sorties (dans le cas d'une classification à plusieurs classes par exemple). Les possibilités d'arrangements entre les neurones sont multiples. La configuration la plus classique est appelée perceptron multicouche (PMC).

Dans cette architecture, les neurones sont organisés en couches comme le montre la figure 3.4 : une couche intermédiaire entre la couche d'entrée et la couche de sortie appelée couche cachée. Les connexions se font d'une couche à l'autre sans qu'il y ait de connexion entre couches non adjacentes. Cette architecture est également appelée réseau à deux couches puisqu'il y a deux couches de poids ajustables : celle qui relie les entrées aux neurones cachés et celle qui relie les neurones cachés aux neurones de sortie.

Le choix du nombre de neurones cachés est très important. Plus ce nombre est important, plus le nombre de degrés de liberté est élevé et plus la fonction modélisée par le réseau de neurones peut être complexe. Tous les poids sont fixés initialement de manière aléatoire ou de manière plus élaborée et ils atteignent des valeurs stables à la fin du processus d'apprentissage.

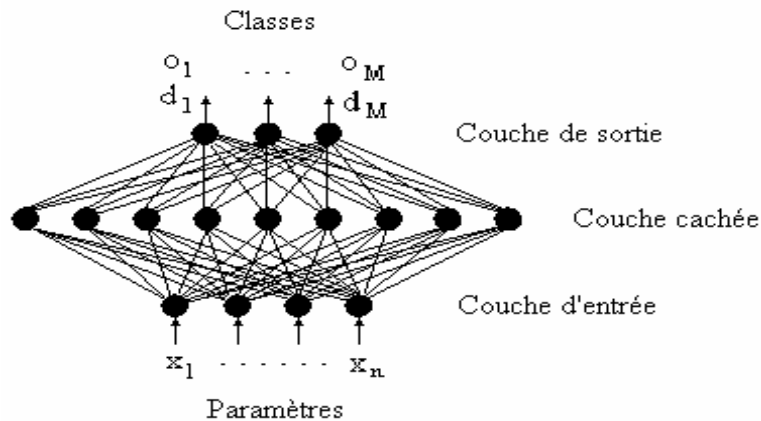


Figure 3.4: Schéma général d'un réseau de neurones

La phase d'apprentissage L'apprentissage du perceptron multicouche est supervisé. Il consiste à adapter les poids des neurones de manière à ce que le réseau soit capable de réaliser une transformation donnée, représentée par un ensemble d'exemples constitué d'une suite de n vecteurs d'entrées. L'apprentissage consiste à présenter au réseau, à plusieurs reprises (appelées itérations), toutes les paires :

- Vecteur d'apprentissage $x = [x_0 \ x_1 \ \dots \ x_n]^T$,
- Sortie souhaitée $d = [d_0 \ d_1 \ \dots \ d_n]^T$,

et à modifier les poids au sens d'une certaine règle (gradient descendant, compétitive, etc.) afin de minimiser l'erreur entre la sortie réelle et la sortie souhaitée (d). La validation de l'apprentissage est réalisée à l'aide d'une base d'exemples différents de ceux appris, appelée base de test.

Les réseaux de neurones apprennent ainsi sur le principe de la "boîte noire". Car il suffit de leur présenter à l'entrée un ensemble représentatif de vecteurs d'apprentissage, durant un nombre d'itérations suffisamment important, pour qu'ils en induisent les surfaces de séparation entre les classes. Ces fonctions ne sont pas déterminées explicitement, mais elles sont encryptées dans les poids du réseau. Grâce à leur structure, le calcul est réalisé en parallèle. La conséquence immédiate est leur possibilité de classifier en temps réel les vecteurs inconnus, car le temps de classification est égal au temps de propagation dans les couches du réseau.

3.5.2.1 Caractéristiques du réseau retenu

Pour construire notre réseau de neurones, nous devons tenir compte des paramètres suivants :

Nombre de couches : plusieurs travaux réalisés au laboratoire nous ont conduit à utiliser un réseau à trois couches [BAR 96],[VER 98]. Ce choix est confirmé en prenant en compte les remarques suivantes :

- La modélisation de n'importe quelle fonction continue est possible à l'aide d'un réseau composé, d'une couche cachée non linéaire, d'une couche d'entrée et d'une couche de sortie linéaires [BIS 95][HOR 89].

- Le temps de calcul devient prohibitif avec l'augmentation du nombre de neurones.

- La taille des bases de données d'apprentissage doit augmenter avec la taille de réseau pour obtenir une bonne généralisation (plus de neurones, plus de paramètres à calculer)[VAP 71].

Couche d'entrée : le nombre de neurones dans la couche d'entrée est égal au nombre de variables. La fonction d'activation présente en chaque neurone est une fonction linéaire.

Couche cachée : le nombre de neurones est choisi selon l'heuristique des résultats de classification sur les données de généralisation. Les méthodes heuristiques consistent principalement à lancer en parallèle plusieurs réseaux d'architectures différentes et à comparer leurs performances à la fin de l'apprentissage par un critère qui est souvent fonction de l'erreur quadratique [ALL 98]. La fonction d'activation utilisée par les neurones de cette couche est la fonction sigmoïde [RIP 96].

Couche de sortie : étant donné que nous connaissons le nombre de classes, nous avons fixé à deux le nombre de neurones dans la couche de sortie, ces neurones présentent une fonction d'activation linéaire.

L'apprentissage : le but de la rétropropagation du gradient est de minimiser la distance entre les réponses du réseau et les réponses désirées selon un certain critère. Dans notre cas nous avons utilisé une fonction d'erreur quadratique moyenne. Cette étape d'apprentissage démarre avec une étape d'initialisation des poids. Pour que l'algorithme converge, les valeurs initiales des poids ne doivent pas être nulles. Dans notre application, nous avons initialisé les poids d'une manière aléatoire. Il existe des études visant à éliminer toutes les initialisations aléatoires, par exemple les travaux de P. Hirschauer et al [HIR 94].

3.5.2.2 Résultats

Nous avons séparé l'ensemble des données en trois ensembles : un ensemble d'apprentissage contenant 360 images, un ensemble de validation contenant 180 images et enfin un ensemble de généralisation (ou test) contenant également 180 images. Tout d'abord nous avons utilisé les 7 mêmes variables que pour l'approche bayésienne et nous avons cherché quel était le nombre de neurones de la couche cachée qui nous permettait d'obtenir le meilleur résultat. Nous avons testé différentes configurations du réseau avec un nombre de neurones de la couche cachée allant de 1 jusqu'à 10. Les résultats sont représentés sur le tableau 3.7. Le meilleur score est obtenu avec 2 neurones dans la couche cachée.

Toujours dans le but d'améliorer ce résultat, nous avons intégré dans le réseau les paramètres écartés après l'ACP. Malgré la forte corrélation constatée lors de cette analyse, ces paramètres apporteraient tout de même une petite information supplémentaire pour le classifieur.

Nous avons d'abord ajouté les variables 8 et 10 qui expriment le nombre de points de forte courbure et le nombre de segments dans l'image de référence, ce qui nous fait un réseau avec 9 neurones en entrée. Nous avons ensuite rajouté les variables 9 et 11 qui donnent le nombre de points de forte courbure et de segments dans l'image à classer (11 neurones pour la couche d'entrée). Et enfin nous avons testé l'intérêt d'utiliser les quinze variables simultanément.

Avec la même procédure que pour les 7 variables, nous avons expérimenté plusieurs configurations de la couche cachée. Les meilleurs scores pour ces différents schémas sont présentés sur le tableau 3.8.

A partir de ces résultats, nous constatons que l'approche neuronale permet d'améliorer les résultats de la classification totale. Par contre le pourcentage d'images de la classe "0", qui sont bien classées, est un peu inférieur au meilleur résultat du classifieur bayésien.

Ceci peut s'expliquer par le fait que l'étape d'apprentissage du réseau n'a pas été réalisé convenablement car le nombre d'éléments de la classe "0" n'est pas assez élevé (146 images). Nous avons également remarqué que l'augmentation du nombre de variables ne permet pas d'accroître notablement les résultats, et même, nous avons noté une petite diminution des performances lorsque nous passons de 11 à 15 paramètres. Ceci confirme les conclusions de l'analyse en composantes principales effectuée. Par ailleurs, nous constatons que, le plus souvent, le meilleur résultat, ou quasiment le meilleur, est fourni en utilisant un seul neurone sur la couche cachée, ce qui s'explique peut-être par le fait que la séparation des deux classes serait linéaire.

Nombre de neurones	1	2	4	5	7	10
% classe "0"	90,28	92,47	91,64	89,81	87,57	88,12
% classe "1"	98,07	97,04	97,79	98,2	96,73	96,65
% Total	96,49	96,98	96,54	96,49	94,87	94,92

Tableau 3.7: Résultat de la classification neuronale avec 7 variables

Nombre de paramètres	7	9	11	15
Nombre de neurones de la couche cachée	2	1	1	1
% classe "0"	92,47	91,45	92,41	89,79
% classe "1"	97,04	98,45	98,37	98,27
% Total	96,98	97,03	97,16	96,55

Tableau 3.8: Résultat de la classification neuronale

3.5.2.3 Comparaison des approches bayésienne et neuronale

Au vu des résultats précédemment obtenus et dans l'optique d'obtenir une règle de classification automatique la plus robuste possible, nous avons décidé d'analyser la complémentarité des résultats délivrés par chacune des deux méthodes. Pour cet essai nous avons comparé, image par image, les résultats issus de la classification bayésienne qui procure le meilleur score pour la classe "0" (la première ligne du tableau 3.6) et les résultats fournis par une classification neuronale opérée sur les 720 images. Pour les deux méthodes nous avons restreint notre étude à l'utilisation des 7 descripteurs issus de l'ACP par image. Les pourcentages de bonne classification obtenus, à l'issue d'une itération, sont donnés dans le tableau 3.9.

Méthodes	Neuronale(2 neurones)	Bayésienne
% classe "0"	92,47	93,83
% classe "1"	97,04	95,82
% Total	96,11	95,42

Tableau 3.9: Comparaison des résultats neuronale et bayésienne

Nous avons analysé finement les cas conduisant à des erreurs de décision. Ils sont au nombre de 33 pour la méthode bayésienne (9 pour la classe "0" et 24 pour la classe "1") et de 28 pour la méthode neuronale (11 pour la classe "0" et 17 pour la classe "1"). Dans seulement 12 cas les désaccords sont communs aux deux méthodes. Cela signifie que dans près de 3% des cas la méthode neuronale délivre une réponse juste alors que la méthode bayésienne propose un résultat

faux. La situation est inverse dans un peu plus de 2% des cas. Si l'on arrive à tirer parti de cette disparité de réponse, nous pourrions espérer améliorer encore les taux de bonne classification. Par exemple, dans le cas présenté ici, si les seules erreurs étaient dues au vote simultané des deux méthodes, les taux de bonne classification atteints seraient ceux présentés dans le tableau 3.10, ce qui est tout à fait satisfaisant. Le problème est de trouver la méthode permettant de fusionner les réponses des deux classifieurs. Un certain nombre de travaux existent dans cette voie qu'il nous appartiendra d'exploiter [KIT 98].

	Taux de bonne classification
% classe "0"	95,2
% classe "1"	99,13
% total	98,33

Tableau 3.10: Résultats optimaux possible

3.5.3 Combinaison des classifieurs

Les observations précédentes ont motivé notre intérêt pour la combinaison des classifieurs. Cette idée consiste à utiliser tous les résultats individuels des classifieurs et à les combiner pour obtenir un consensus sur le résultat final.

Plusieurs schémas de combinaison ont été conçus et il a été démontré que certains d'entre eux donnent de meilleures performances que celui qui a donné individuellement le meilleur taux de classification [KIT 98] [LOA 99]. Cependant il subsiste une compréhension insuffisante des circonstances suivant lesquelles certains schémas de combinaison sont meilleurs que d'autres.

3.5.3.1 Principe

Dans la classification, le problème consiste à assigner un individu z décrit par le vecteur de mesure x à une des m classes possibles $(\omega_1, \omega_2, \dots, \omega_m)$.

Soit R le nombre de classifieurs utilisés et soit x le vecteur de mesures utilisés par les classifieurs. Chaque classe ω_k est modélisée par la fonction de densité de probabilité $p_i(X/\omega_k)$ pour le classifieur. La probabilité a priori d'occurrence des classes est donnée par $p(\omega_k)$.

Or la théorie bayésienne établit qu'un individu z peut être assigné à la classe ω_j si la probabilité a posteriori de l'interprétation est maximale, c'est-à-dire

$$\text{Assigner } Z \rightarrow \omega_j \text{ si} \quad (3.5)$$

$$P_{conjoint}(\omega_j/X) = \max_{k=1}^m P_{conjoint}(\omega_k/X)$$

La règle de décision bayésienne établit que pour utiliser toute l'information correctement pour atteindre une décision, il est essentiel de calculer la probabilité confondue, $P_{conjoint}$, de toutes les hypothèses. Cette formulation est correcte du point de vue de la classification, mais difficilement applicable, car la probabilité a posteriori peut dépendre d'équations mettant en jeu des mesures statistiques d'ordre important, difficiles à résoudre.

En utilisant des simplifications, nous pouvons exprimer la règle (3.5) en termes des probabilités a posteriori données par les classifieurs individuellement [KIT 98]. Cette démarche conduit à une règle plus facile à calculer, ainsi qu'à des règles de combinaison qui sont couramment utilisées dans la pratique, comme celle du produit ou celle de la somme :

- Règle du produit

$$\begin{aligned} & \text{Assigner } Z \rightarrow \omega_j \text{ si} \\ P_j^{-(R-1)}(\omega_j) \prod_{i=1}^R P_i(\omega_j/X) &= \max_{k=1}^m \left[P^{-(R-1)}(\omega_k) \prod_{i=1}^R P_i(\omega_k/X) \right] \end{aligned} \quad (3.6)$$

- Règle de la somme

$$\begin{aligned} & \text{Assigner } Z \rightarrow \omega_j \text{ si} \\ (1-R)P(\omega_j) + \sum_{i=1}^R P_i(\omega_j/X) &= \max_{k=1}^m \left[(1-R)P(\omega_k) + \sum_{i=1}^R P_i(\omega_k/X) \right] \end{aligned} \quad (3.7)$$

3.5.3.2 Règle de décision choisie et résultats

Un aspect important pour la combinaison des classifieurs est que ces derniers soient différents. Cette différence peut être obtenue selon plusieurs manières (vecteurs de variables, ensembles d'entraînement, ...). Pour nous [MER 03a], cette différence réside dans le type des classifieurs utilisés, le bayésien et le neuronal.

La règle de décision résultant de la combinaison que nous avons adoptée est décrite ci-dessous. Cette règle est fondamentalement une règle de la somme, comme présentée dans l'équation (3.7). Elle a été retenue en suivant les conclusions de l'étude réalisée par J.Kittler dans [KIT 98] sur la combinaison des classifieurs où il a démontré que la règle somme est la plus robuste et la plus efficace.

$$\begin{aligned} & \text{Assigner } Z \rightarrow \omega_j \text{ si} \\ P_B(\omega_j/X_k) + P_N(\omega_j/X_k) &= \max_{k=1}^m [P_B(\omega_j/X_i) + P_N(\omega_j/X_i)] \end{aligned} \quad (3.8)$$

Où $P_B(\omega_j/X_k)$ et $P_N(\omega_j/X_k)$ sont les probabilités a posteriori que l'image appartienne à la classe. Ces probabilités sont issues respectivement des classifieurs bayésien et neuronal.

Du point de vue de la représentation des individus, il existe deux techniques de combinaison. Dans la première technique le même vecteur de variables est utilisé pour représenter les individus. Dans la deuxième, chaque classifieur utilise son propre vecteur pour représenter les individus. Dans notre application nous avons utilisé un même vecteur de variables comme entrée aux classifieurs neuronal et bayésien.

Dans le tableau 3.11, la performance de la combinaison des deux classifieurs est montrée. Avec cette approche, nous avons amélioré les résultats du diagnostic, de 95,82% à 96,1% pour la classe "0" et de 97,04% à 98,89% pour la classe "1".

	Taux de bonne classification
% classe "0"	96,1%
% classe "1"	98,89%
% total	98,12%

Tableau 3.11: Classifieur combiné

3.6 Conclusion

Dans ce chapitre, Nous avons présenté le processus de classification qui est une tâche centrale dans notre système d'inspection automatique. Le problème concerné est un problème de classification supervisée. Nous avons alors commencé par construire nos deux classes à partir des images saisies dans des conditions réelles. Nous avons par la suite utilisé une méthode empirique pour classer les images dans l'une des deux classes à l'aide des trois paramètres *CPFC*, *CSEG* et *ZNCC*.

Afin d'améliorer les résultats de la classification, nous avons rajouté de nouveaux paramètres et testé leurs pertinences par une analyse en composantes principales. Ensuite nous avons testé un classifieur bayésien et un classifieur neuronal. Les résultats pratiques fournis par ces deux approches nous ont montré une complémentarité pour réaliser un bon diagnostic. En effet, le bayésien procure un bon score pour la classe "0" et le neuronal pour la classe "1".

Nous avons donc opté pour un schéma de classification combiné au moyen d'une règle *somme*. Les premiers résultats obtenus sont satisfaisants, et confirment ceux présentés par H. Loaiza, mais nous avons conscience qu'un ensemble

d'apprentissage plus large aurait été nécessaire. La non reconduction du contrat industriel avec Renault ne nous a pas permis de compléter notre base de données.

C'est essentiellement pour cette raison, que nous avons orienté nos recherches vers un autre problème fondamental qu'est la reconnaissance d'objets 3D de forme libre.

Chapitre 4

SYSTEMES DE RECONNAISSANCE D'OBJETS 3D DE FORME LIBRE

4.1 Introduction

Le problème de reconnaissance étudié dans ce chapitre est l'identification et la localisation d'objets 3D de formes libres à partir d'une image 2D de la scène où des occlusions partielles peuvent apparaître.

La reconnaissance par vision monoculaire passive consiste, à partir de connaissances sur l'aspect visuel d'objets et d'une image, à déterminer si un ou plusieurs de ces objets est présent dans la scène ainsi qu'à calculer la position et l'orientation de chacun de ces objets.

Cette définition générale concerne tous les systèmes de reconnaissance d'objets. Ce thème prend une part importante dans le domaine de recherche de la vision artificielle, et de nombreux systèmes, algorithmes et paradigmes ont été proposés durant ces dernières décennies [AGG 96], [PER 91]. Cependant, aucune solution universelle n'a pu être développée jusqu'à maintenant et seules des réponses partielles avec des réussites limitées ont été appliquées. D'ailleurs certains chercheurs avancent qu'il est impossible de concevoir un système de reconnaissance à large utilisation aussi performant qu'un système ciblé sur une situation spécifique.

La plupart des systèmes de reconnaissance performants développés jusqu'à présent ont utilisé des modèles d'objets polyédriques et sont basés sur l'appariement de primitives structurelles tels que les coins, les segments... La stratégie adoptée, généralement, est l'approche de reconnaissance par alignement [HUT 87] ou bien les approches par hypothèse et vérification [AYA 86], [DHO 89], [HOR 87a],

[LOW 87], [WHE 95]. Cependant la reconnaissance d'objets 3D de formes libres, avec la présence potentielle d'occlusions dans la scène, à partir d'une image d'intensité 2D, est un problème ouvert.

Dans ce chapitre, nous commencerons par présenter un état de l'art sur la reconnaissance d'objets 3D de formes libres en mettant l'accent sur les avantages et les limites de chacune d'entre elles. Nous présentons ensuite l'architecture générale de notre méthode basée sur la représentation en squelette des objets ainsi que l'intérêt qui en découle de l'utilisation de ce type de représentation.

4.2 Etat de l'art sur la reconnaissance d'objets 3D de forme libre

Dans cette partie, nous parlerons des différentes approches développées durant ces dernières années dans le domaine de la reconnaissance d'objets 3D de formes libres en présence d'occultation ou non et ce, à partir de leurs images d'intensité. Les approches qui utilisent les images de profondeurs sont omises volontairement car elles ne rentrent pas dans le cadre de notre problématique.

Les méthodes de reconnaissance d'objets de forme libre qui utilisent exclusivement des images de luminance peuvent être classées comme suit :

4.2.1 Méthodes basées sur l'apparence

Parmi les méthodes de reconnaissance qui utilisent les images d'intensité, les méthodes basées sur l'apparence "appearance-based approaches" sont de loin les plus populaires. Leur principe consiste à représenter chaque objet de la scène par un ensemble de ses images d'intensité prises à partir de plusieurs points de vue et avec toutes les directions d'illumination possibles. Chaque vue de l'objet est représentée par un point dans un espace multi-dimensionnel appelé espace propre "eigenspace". Les fonctions de base qui engendrent cet espace sont déterminées à partir d'un ensemble d'images de référence appelées images d'apprentissage. Pour reconnaître une vue quelconque de l'objet on la projette dans l'espace propre et on détermine, parmi les images d'apprentissage, celle qui s'en rapproche le plus.

Cette idée simple mais efficace a été largement exploitée par les chercheurs sous des formes différentes. Les premiers travaux dans le domaine de la reconnaissance basée sur l'apparence étaient orientés vers la reconnaissance faciale. Kirby et Sirovich [KIR 90] ont utilisé l'analyse en composantes principales (ACP) afin de déterminer un système de coordonnées qui prend le mieux en compte (en terme de compression) leurs images d'apprentissage. Les vecteurs de base du nouveau

système de coordonnées ont été appelés images propres "eigenpictures".

Soit l'ensemble X des images d'apprentissage x_i représentant toutes les apparences possibles de l'objet :

$$X = \{x_1, x_2, \dots, x_M\} \quad (4.1)$$

Chaque image d'apprentissage x_i est un vecteur colonne dont les éléments représentent les valeurs des N pixels de l'image d'intensité correspondante. Pour identifier correctement une apparence quelconque de l'objet, ce dernier doit être vu dans un ensemble de poses et sous un ensemble de conditions d'illumination qui sont supposées être identiques lors de la phase de reconnaissance. L'ACP est utilisée pour déterminer un ensemble de vecteurs orthogonaux qui vont former la base de l'espace propre engendré par X . Pour cela le vecteur moyen :

$$m = \frac{1}{M} \sum_{i=1}^M x_i \quad (4.2)$$

est soustrait de chaque image x_i de X pour produire l'ensemble centré X_c . X_c est alors utilisé pour former la matrice de covariance $N \times N$:

$$Q = X_c X_c^T$$

Les valeurs et vecteurs propres $\{(\lambda_i, v_i), i = 1 \dots N\}$ de la matrice Q sont ensuite déterminés. L'ensemble des N vecteurs propres forme la base orthogonale recherchée. La représentation en composantes principales est aussi utilisée pour réduire la dimension du problème. En effet, les images d'apprentissage peuvent être approximées dans un espace de dimension inférieure qui encode grossièrement mais efficacement l'apparence des objets. Pour cela on utilise les seuls vecteurs propres qui correspondent aux valeurs propres les plus importantes afin de caractériser les objets. Murase et Nayar [MUR 95] ont démontré qu'un sous espace de dimension 20 est suffisant pour caractériser l'apparence d'un objet pour le problème de la reconnaissance d'objets.

Soit $\{e_1, e_2, \dots, e_k\}$ l'ensemble des vecteurs propres qui correspondent aux k plus grandes valeurs propres. Les vecteurs centrés $(x_i - m)$ sont projetés dans le sous espace représenté par les fonctions de base $\{e_1, \dots, e_k\}$ pour générer des vecteurs réduits de dimensions k notés g_i . Une approximation de l'image x_j peut être construite en utilisant les fonctions de base e_i et le point projeté g_i , elle est donnée par :

$$\tilde{x}_j = (e_1, \dots, e_k) g_j + m \quad (4.3)$$

Malgré le fait que cette approximation utilise uniquement k vecteurs propres au lieu de $\min(N, M)$ (le rang de la matrice de covariance Q), les détails importants dans l'apparence de l'objet sont généralement préservés.

Turk et Pentland [TUR 91a], [TUR 91b] ont noté que l'utilisation de la reconnaissance basée sur l'apparence avec un petit nombre d'images d'apprentissage permettait de décrire et de reconstruire les visages d'une large tranche de la population. Leur algorithme a été testé sur une base de données de 16 sujets dont les visages ont été capturés en variant à chaque fois les conditions d'illumination, de pose et d'échelle (2500 images en tout). Ils ont constaté que leur système est plus sensible aux variations de la dimension des visages dans les images qu'aux variations de l'illumination et de la pose. Ils ont montré que cette méthode peut être utilisée pour reconnaître des surfaces complexes et sculptées (visages) en moins d'une seconde.

Murase et Nayar [MUR 95] ont étendu l'utilisation de la reconnaissance basée sur l'apparence pour des objets quelconques. En utilisant deux espaces propres, ils sont capables de reconnaître et de localiser les objets sous des conditions d'éclairage variables. L'espace propre universel U est construit à partir de l'ensemble X de toutes les images d'apprentissage, il est utilisé pour identifier un objet qui se trouve dans l'image de la scène, par contre l'espace propre de l'objet O_j est construit pour chaque objet j à partir de toutes les images d'apprentissage x_i qui contiennent cet objet, il est utilisé pour déterminer la pose de l'objet ainsi que les conditions d'éclairage. Pour chaque objet ils ont créé une variété qui permet d'interpoler les points g_i dans les espaces propres. Ces variétés sont construites grâce à un algorithme d'interpolation spline-cubique. Pour identifier un objet dans une image s , celle-ci est premièrement projetée dans l'espace propre universel :

$$g = \left\{ \begin{array}{c} e_1^T \\ \vdots \\ e_k^T \end{array} \right\} (s - m) \quad (4.4)$$

puis appariée avec l'objet ayant la variété la plus proche. Dans des travaux parallèles, Murase et Nayar [MUR 94] ont étudié plus en profondeur l'influence de l'éclairage sur leur méthode de reconnaissance. Le but de leur étude était de déterminer les paramètres de l'illumination qui permettaient de maximiser la différence en apparence entre les objets.

Munday et al [MUN 96] ont comparé la méthode de Murase et Nayar [MUR 95] avec deux approches référencées modèles. Ils ont conclu que le système référencé apparence possédait la plus grande vitesse de reconnaissance, mais d'un autre côté il fournissait un plus grand nombre de faux appariements. Ceci est dû à l'absence de la procédure de vérification dans les méthodes basées sur l'apparence ainsi qu'à leur sensibilité aux occultations et aux erreurs de segmentation. A l'inverse, les approches référencées modèles sont basées sur le principe même de la génération/vérification d'hypothèses, et de ce fait minimisent le nombre de

faux appariements.

Black et Jepson [BLA 98] ont montré que la projection I est le résultat d'une estimation au moindre carrés de l'image originelle x . Pour compenser la sensibilité bien connue des techniques des moindres carrés aux bruits, les auteurs ont utilisé des méthodes statistiques robustes dans le calcul de la projection g . Le but de leur étude était de développer un système de poursuite et de reconnaissance des gestes de la main. Toutefois, les méthodes basées sur l'apparence souffrent de leur manque d'efficacité pour traiter les cas où la scène contient plusieurs objets avec une possibilité d'occultation.

Huang et al [HUA 97] ont proposé de subdiviser les images en plusieurs régions. Ils exploitent les relations entre ces régions ainsi que leur apparences afin d'identifier aussi bien les objets de la scène que leur pose.

Camps et al [CAM 98] se sont basés sur les travaux de Huang [HUA 97] pour développer un système de reconnaissance qui intègre des bases de données hiérarchiques et une méthode d'appariement Bayésienne. L'inconvénient des approches "régions" est qu'elles doivent impérativement segmenter les images d'entrée en temps réel, ce qui limite la catégorie d'objets reconnaissables aux seuls objets pouvant être segmentés d'une manière fiable, ce qui est loin d'être le cas des objets de forme libre. Ohba et Ikeuchi [OHB 97] et [KRU 96] ont surmonté ce problème en développant une nouvelle technique basée sur l'apparence qui utilise des fenêtres locales définies à partir de l'apparence des objets.

Ohba et Ikeuchi [OHB 97] sont capables de traiter les cas de translation et d'occultation d'un objet de la scène en utilisant ce qu'ils ont appelé les fenêtres propres "eigenwindows". Les fenêtres propres encodent les informations sur l'apparence de l'objet dans de petites parties de l'image. Des mesures de la détectabilité, l'unicité ainsi que la précision ont été aussi développées pour les fenêtres propres. Ces mesures permettent d'écarter de l'ensemble d'apprentissage, lors de la phase de reconnaissance, les fenêtres qui sont difficiles à détecter, ne sont pas discriminantes ou bien sont sensibles aux bruits. Grâce à cette technique, les auteurs sont capables de reconnaître plusieurs objets qui ont subi des occultations dans la scène.

Leonardis et Bischof [LEO 96] se sont aussi intéressés au problème de l'occultation, leur méthode consistait à sélectionner d'une manière aléatoire un ensemble de points de l'image de la scène et leur correspondants dans l'espace engendré par la base des vecteurs propres $[e_1, \dots, e_k]$. Ils ont utilisé le paradigme de la génération/vérification d'hypothèses pour reconnaître les objets. Cette méthode a démontré qu'il était possible de reconstruire des portions non visibles des objets de la scène. Bischof et Leonardis [BIS 98] ont étendu leurs travaux pour traiter les problèmes du changement d'échelle "scaling" et de la translation des objets

dans la scène.

Rao [RAO 97] a appliqué l'apprentissage adaptatif aux vecteurs de base $[e_1, \dots, e_k]$ de l'espace propre dans les méthodes basées sur l'apparence. L'approche multi-vues dynamique est utilisée pour prédire les changements spatio-temporels dans l'apparence d'une séquence d'images. Cette prédiction permet de modifier les vecteurs propres en minimisant une fonction de coût basée sur le principe de la longueur minimale de la description "MDL : Minimum Description Length". Ceci permettait d'apprendre les données d'entrées sans pour autant avoir à mémoriser leurs détails, ce qui rendait le système de reconnaissance plus flexible.

4.2.2 Méthodes basées sur l'extraction de silhouettes

Les silhouettes d'objets ont été largement utilisées ces dernières années par la communauté de vision par ordinateurs pour caractériser les objets de formes libres. En effet, dans un environnement à éclairage contrôlé, la silhouette d'un objet peut être tout à fait utile pour déterminer son identité et sa pose.

Mokhtarian [MOK 95] a développé un système de reconnaissance basé sur l'utilisation des silhouettes fermées des objets. Le système est élaboré pour reconnaître des objets de forme libre en ayant seulement quelques vues de disponibles dans un environnement où un seul objet est présent. Pour cela, une source lumineuse contrôlable est utilisée pour illuminer l'objet de sorte à ce que la frontière entre le fond et l'objet soit facile à détecter, ce qui permet d'isoler l'objet par un simple seuillage. Les courbes qui correspondent aux frontières de l'objet (extraites par suivi de contour) sont alors représentées par leur courbures calculées dans l'espace échelle (CSS : Curvature Scale Space representation). L'appariement entre les courbes CSS est effectué en se basant sur la position des maximums de courbure. Pour les courbes de frontière convexes, la représentation CSS est lissée jusqu'à ce qu'il ne reste que quatre maximums de courbure, par contre pour les courbes concaves, elle utilise la totalité des maximums de courbure obtenus pour une échelle donnée. Durant la phase de reconnaissance, le rapport d'aspect de la silhouette de l'objet est utilisé en premier afin de pré-filtrer les appariements possibles entre la scène et le modèle avant de procéder à l'appariement proprement dit des silhouettes, ceci permet de réduire l'espace de recherche. Cette technique représente un moyen rapide et efficace pour appairer des caractéristiques grossières de la scène avec celles du modèle. Les appariements plausibles obtenus sont alors vérifiés par une méthode de reconstruction qui permet de mesurer l'erreur entre chacune des deux courbes appariées. Ce système est capable d'identifier les objets stockés dans la base de données.

Ponce et Kriegmann [PON 89] ont utilisé les contours générés à partir des

normales à la surface discontinues ainsi que les contours d'occultation pour reconnaître les objets 3D. Ils ont modélisé les objets par des surfaces paramétriques afin de générer les modèles de contours 3D à reconnaître dans les images d'intensité. Leur méthode consiste à déterminer les équations implicites des patches paramétriques, ensuite utiliser les intersections entre ces patches pour construire une équation implicite du contour. Un modèle de projection perspective faible "weak" est utilisé pour corrélérer les contours 3D avec les contours 2D extraits des images d'intensité. Les contours 2D sont découpés manuellement puis regroupés en plusieurs "clusters" représentant un seul objet. Ce système a été utilisé pour modéliser les surfaces de révolution. Leurs résultats expérimentaux ont donné des temps de reconnaissance rapides (30 secondes) et une localisation assez précise (erreur moyenne de l'ordre de 0.4-1.4 pixels).

Joshi et al [JOS 94], [JOS 97] ont développé une méthode de reconnaissance utilisant les courbes HOT (Higher Order Tangent) pour modéliser un objet lisse (smooth). Ces courbes sont identifiées à partir d'un ensemble de points sur la silhouette de l'objet dans l'image d'intensité. Les angles entre les droites tangentes aux points et le rapport des distances entre les points du contour sont utilisés pour construire un invariant géométrique à la pose de l'objet ainsi qu'au facteur d'échelle. Ce dernier est ensuite utilisé pour indexer une table de reconnaissance. Ils ont testé leur système de reconnaissance sur quatre objets (une courge, une poire, une banane et un leurre de canard) et ont démontré que la représentation par des courbes HOT permettait de retrouver dans tous les cas, l'identité et la pose de l'objet, après que la phase de vérification ait rejeté les faux appariements.

Chen et Stockman [CHE 98] ont utilisé les silhouettes avec les bords internes (internal edges) pour retrouver l'identité et la pose de 20 modèles d'objets de forme libre à partir d'images d'intensité. La courbe de la silhouette est combinée avec un invariant géométrique (dont le rôle est d'indexer la base de données) pour générer des hypothèses sur la pose et l'identité du modèle. Afin de tolérer les occultations, ils ont utilisé la représentation partielle de contour. Les attributs invariants des segments de courbe extraits à partir de cette représentation sont ensuite utilisés pour indexer les modèles dans la table de hachage. Cette méthode permet d'apparier des parties du modèle avec les parties observées dans l'image. Les appariements ainsi obtenus sont ensuite groupés dans des hypothèses consistantes basées sur l'identité de l'objet (associé à chaque partie) et sur une estimation approximative de la pose. Chaque hypothèse est ensuite vérifiée en comparant la carte des bords internes du modèle "model edges map" avec celle des bords observés. La phase de vérification produit aussi une estimation plus raffinée de la pose de l'objet. L'inclusion des bords internes dans la phase de

vérification améliore les performances du système en permettant de rejeter les fausses hypothèses sur l'identité et la pose de l'objet.

4.2.3 Méthodes basées sur le "shape from shading"

Comme son nom l'indique, il s'agit d'extraire une forme uniquement à partir des variations de lumières observées sur la surface d'un objet. En dépit de l'intérêt que portent les chercheurs au "shape from shading" (SFS) depuis plusieurs années, on trouve très peu de publications sur son utilisation dans les systèmes de reconnaissance d'objets 3D. Une des raisons principales de cet état de fait (omission) est l'absence d'algorithmes robustes capables de retrouver les détails fins et discriminants des surfaces des objets. Récemment Worthington et Hancock [WOR 01] ont étudié la possibilité d'utiliser le SFS dans la reconnaissance d'objets 3D. Ils ont démontré que les informations sur l'orientation locale de la surface obtenues en utilisant le SFS représentent des données d'entrée très utiles pour les systèmes de reconnaissance basés sur l'apparence. En effet, à partir des informations photométriques, ils extraient deux représentations de l'objet à savoir, la "needle-map" et l'index de la courbure locale de la forme (local curvature shape-index), puis ils examinent leurs performances relatives dans la reconnaissance d'objets 3D. Concrètement, ils ont développé deux stratégies de reconnaissance basées sur les histogrammes, la première utilise directement les normales à la surface, tandis que la seconde est basée sur l'index de la forme. Ils ont expérimenté ces deux stratégies de reconnaissance sur 20 objets de formes quelconques, et ont constaté que les deux représentations donnaient de bon résultats. Cependant, Ils ont trouvé que l'histogramme des normales à la surface était plus efficace que l'histogramme de l'index de la forme. Une étude de la sensibilité a révélé que la méthode "needle-map" offre une discrimination significative de la topologie différentielle de l'apparence de l'objet. Elle permet d'extraire des caractéristiques discriminantes de l'objet 3D à partir de ses images 2D.

Dernièrement Ababsa [ABA 02] a démontré dans sa thèse de doctorat que l'utilisation de la clinométrie d'un objet de forme libre, sous certaines conditions, est suffisant pour réaliser un système de recalage automatique simple et précis. Il a élaboré une approche, qu'il a appelée approche photoclinométrique. Cette technique exploite une seule image de luminance de l'objet à partir de laquelle elle extrait un invariant projectif de type photométrique (non géométrique). Cet invariant est par la suite utilisé pour apparier des zones de l'image de l'objet avec celles du modèle. Il a proposé une méthode originale basée sur ce qu'il appelle les aspects photométriques pour construire une base de données discriminante du modèle 3D de l'objet.

4.3 Présentation de l'approche développée

Cette étude de l'état de l'art nous a permis d'analyser de plus près l'architecture des systèmes de reconnaissance d'objets 3D de forme libre les plus populaires. Nous avons aussi constaté les problèmes ardues auxquels ils sont confrontés et qui n'ont pas encore été résolus d'une manière satisfaisante (tel que la complexité, l'occultation, l'efficacité, ...). Quelques unes de ces difficultés sont la construction de modèles d'objets complexes, l'extraction de primitives permettant une mise en correspondance correcte, et l'indisponibilité d'éléments saillants pour une estimation robuste de la pose. La difficulté de travailler dans une scène réelle (objet réel et lumière non contrôlée) rend la reconnaissance plus ardue à cause de la présence de différents bruits (ombre, ...), problème qui n'a pas été abordé dans la plupart des travaux sus-cités.

La construction de descripteurs de formes joue un rôle important dans la plupart des systèmes de vision par ordinateur. Une représentation de la forme pertinente et fiable doit satisfaire à plusieurs exigences ; telles que l'invariance, l'unicité, et la stabilité [MOK 92]. Si deux objets ont la même forme, alors leur représentation devrait être la même et devrait être invariante par rapport aux transformations isométriques (translation, rotation). L'unicité veut dire que si deux objets ont des formes différentes, ils devraient avoir des représentations différentes. La stabilité dénote le fait que si deux objets ont une petite différence de la forme, alors leur représentation devrait avoir une petite différence. Inversement, si deux représentations ont une petite différence, alors les objets qu'ils représentent devraient avoir aussi une petite différence de la forme. Par conséquent, une représentation stable veut dire une représentation qui est insensible au bruit. Le descripteur devrait représenter la forme d'un objet dans plusieurs niveaux d'abstraction et devrait contenir des informations aussi bien sur les régions que sur les contours de l'objet. Finalement, le descripteur de la forme devrait être déterminé efficacement.

La représentation des formes en squelette, introduite par Blum [BLU 67], est un excellent descripteur car il réunit la plupart de ces exigences. Il a la possibilité de conserver sous forme compacte un grand nombre d'informations topologiques et géométriques de la forme initiale. Un autre avantage qu'il faut signaler est le fait que les squelettes ont une structure de graphe qui permet d'utiliser les puissants outils issus de la théorie des graphes.

Le squelette d'un objet est une représentation de la forme dans une dimension inférieure. Pour un objet surfacique (2D), le squelette est un ensemble de lignes centrées dans la forme, par contre pour les objets volumiques il existe deux types de squelettes. Les squelettes surfaciques qui sont un ensemble de surfaces centrées

dans la forme et les squelettes curvilignes 3D qui sont un ensemble de lignes centrées dans la forme appelés aussi squelettes homotopiques.

L'utilisation des squelettes dans la reconnaissance de formes remonte à une quarantaine d'années, les premières applications étaient dédiées à la reconnaissance de caractères [ALC 69], [DEU 68] et depuis une littérature abondante existe dans ce domaine. Les plus récents sont les travaux Kégl et Krzyzak [KÉG 02] pour reconnaître des caractères. Ils utilisent ce qu'ils appellent courbes principales pour extraire un squelette robuste. Siddiqi et al ont résolu le problème d'appariement d'objets 2D en utilisant une représentation compacte de squelettes appelé "shock graph" [SID 99]. Dans [MAC 02a], [MAC 02b] Macrini et al ont associé la technique des "shock graph" avec une représentation en graphe d'aspects d'objets 3D, cela leur a permis d'effectuer une reconnaissance efficace 2D/3D.

Naf et al [NAF 97] ont utilisé des squelettes 3D déterminés à partir du diagramme de Voronoï pour représenter les données médicales volumétriques. Ces squelettes ont été utilisés pour analyser les anomalies dans des IRM du lobe temporel du cerveau humain.

Dans [HIL 01], une méthode de reconnaissance 3D/3D basée sur une représentation en MRG "multirésolution Reeb graph" a été présentée. Le Reeb graphe est un squelette 3D calculé en utilisant les méthodes continues, il est appelé aussi squelette géodésique. Le MRG est calculé pour chaque objet ensuite un algorithme d'isomorphisme de graphes est appliqué, pour effectuer l'appariement avec les MRGs d'une base de données. Dans [SUN 03] une méthode de reconnaissance 3D/3D a été décrite, les auteurs ont repris le même principe que [HIL 01] sauf qu'au lieu d'utiliser un Reeb graphe, ils calculent un squelette 3D homotopique avec une méthode d'amincissement. L'appariement des graphes utilisé est celui présenté dans [SID 99].

D'après les arguments énoncés ci-dessus, le formalisme des squelettes semble approprié à la modélisation des objets de forme libre. Nous nous proposons alors d'utiliser ce formalisme dans le cadre de la reconnaissance d'objets 3D de forme libre en vision monoculaire. Dans le paragraphe suivant nous présenterons le principe général de cette approche.

4.3.1 Méthode de reconnaissance par squelettes

L'approche que nous avons développée [MER 04a], [MER 04b] et [MER 04c], est basée sur l'utilisation du squelette 3D d'un modèle connu a priori et l'extraction en ligne du squelette 2D de l'image de l'objet vu par une caméra. Ceci consiste en une originalité car à notre connaissance, ce formalisme n'a pas été appliqué dans le contexte de la vision monoculaire.

Le principe général de notre méthode est illustré dans la figure 4.2. Nous pouvons établir 3 grandes phases nécessaires à une reconnaissance à base de modèles.

- Modélisation et extraction de primitives : recherche d'une représentation appropriée et détection de primitives dans l'image/modèle.
- Appariement : phase de mise en correspondance des primitives image et modèle.
- Localisation et vérification : phase de validation des appariements trouvés et estimation de la pose.

Modélisation et extraction de primitives

Pour avoir une mise en correspondance efficace, il est indispensable de représenter les attributs modèle et image selon une représentation proche. Dans notre approche le processus de squelettisation nous permet d'obtenir deux graphes nommés 3D et 2D. Le graphe 3D est obtenu à partir du squelette 3D homotopique qui est lui-même calculé à partir du modèle 3D, cette opération est exécutée off-line. Une opération de squelettisation on-line de l'image 2D de la scène (ligne 2 de la figure 4.1), nous permet de construire le graphe 2D (ligne 3 de la figure 4.1). Notre procédé code ces graphes de telle sorte que chaque graphe de squelette conserve en lui des informations topologique et géométrique de la forme initiale. En effet chaque nœud du graphe représente un élément du squelette c'est-à-dire un segment 2D pour le graphe 2D et un segment 3D pour le graphe 3D. Chaque arc dans le graphe représente une liaison entre les différents éléments du squelette. Dans le chapitre (5), les méthodes de squelettisation ainsi que les améliorations que nous avons apportées au cas 2D seront présentées et la procédure de codage des graphes sera détaillée dans le paragraphe (6.6.1).

Appariement

La reconnaissance revient donc à établir un isomorphisme entre le graphe 3D de la base de données et le graphe 2D obtenu à partir de l'image. Tout le problème revient donc à établir une bonne mesure de qualité sur l'appariement entre graphes. Dans notre application, cette tâche est difficile car cette mesure doit évaluer le degré de ressemblance entre structures similaires de deux sous-graphes, afin de gérer les problèmes d'occlusions, de rajout, d'ombrage... En conséquence, elle doit permettre de choisir, pour la phase de reconnaissance, le bon modèle de la base de données (ligne 4 de la figure 4.1), et ce, rapidement et aussi d'effectuer une mise en correspondance un à un des nœuds, pour pouvoir localiser et vérifier cette hypothèse (lignes 5 à 12 de la figure 4.1).

Après avoir étudié plusieurs algorithmes de mise en correspondance de graphes, nous avons implémenté une méthode d'isomorphisme qui utilise les signatures

```

1. SystèmeReconnaissance (image I, {graphe3D G2}) :
2.   S ← Squelettisation (I) ;
3.   G1 ← TransformationSqueletteGraphe (S) ;
4.   {hypothèses} ← Indexation (G1, {G2}) ;
5.   pour chaque h de {hypothèses} faire
6.     h' ← VérificationHypothèse (G1, h) ;
7.     si h' <> nul alors
8.       | SuperposerModèleSurImage ;
9.       | sortir ;
10.    sinon
11.      | h ← hypothèse-suivante ;
12.    afficher ("pas de modèle correspondant pour cette image") ;

```

Figure 4.1: Algorithme de reconnaissance de forme par squelettes

topologiques des nœuds. Cette méthode est issue des travaux de Siddiqi [SID 98], que nous avons par la suite améliorée pour l'adapter à notre problématique. Les détails de cet algorithme seront développés au paragraphe (6.6).

Projection et vérification

Le résultat de cet isomorphisme nous présente un appariement un-à-un des nœuds des deux squelettes (2D et 3D). Dans chaque nœud, sont stockés des primitives 2D et 3D (point et segment). Ainsi par une méthode classique de localisation qui sera exposée dans l'annexe D, on pourra estimer la position de l'objet et effectuer des vérifications.

4.4 Conclusion

Dans ce chapitre, nous avons vu que la reconnaissance d'un objet 3D de forme à partir d'une image est difficile, et cela pour plusieurs raisons à savoir : la difficulté d'extraire des primitives permettant une mise en correspondance correcte et surtout l'indisponibilité d'éléments saillants qui permettent une estimation robuste de la pose. Nous avons évoqué plusieurs techniques de reconnaissances basées essentiellement sur l'apparence, les silhouettes et sur le "shape from shading". Pour nous affranchir des problèmes cités ci-dessus nous avons opté pour une représentation en squelette des objets.

Associée à la technique d'isomorphisme de graphes, notre méthode permet d'effectuer une mise en correspondance entre le squelette 3D du modèle et le squelette 2D de l'objet et ainsi reconnaître et localiser l'objet en question.

Dans le chapitre suivant nous présenterons la première partie de notre système de reconnaissance, à savoir la modélisation des objets en squelettes. Nous parlerons des méthodes existantes et nous exposerons celles que nous avons développée.

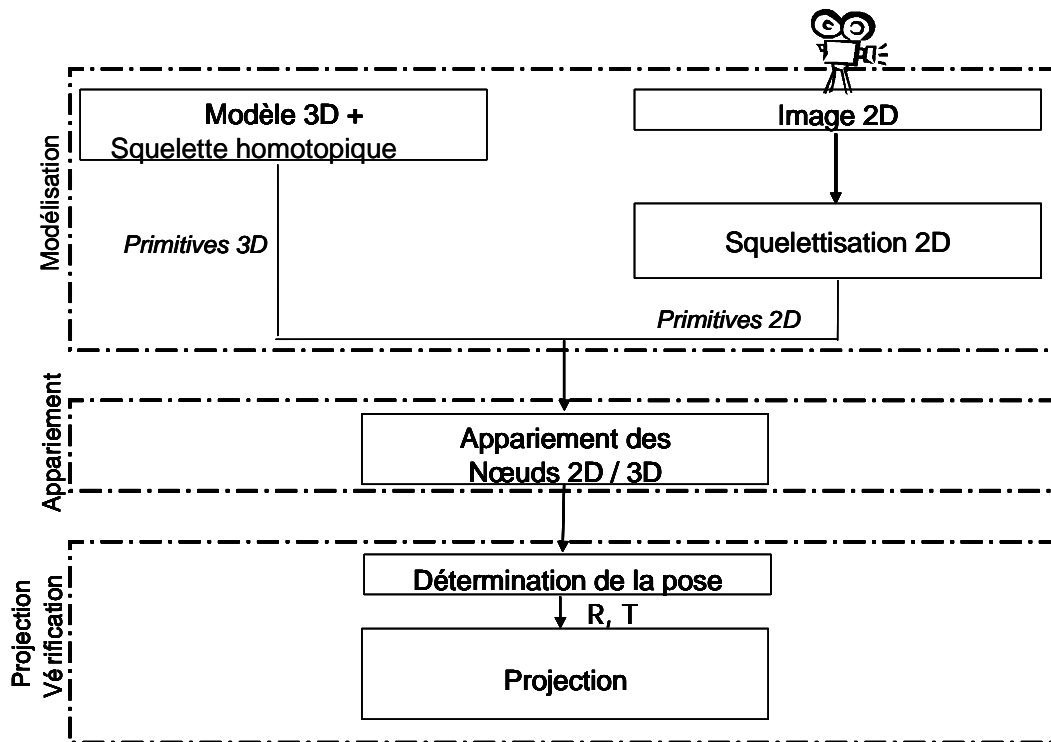


Figure 4.2: Reconnaissance d'objets 3D de forme libre par squelettes

Chapitre 5

SQUELETTISATION

5.1 Introduction

Décrire un objet par une représentation de type "squelette" constitue une étape importante dans un certain nombre d'applications relevant du domaine du traitement des images ou de la reconnaissance de formes. L'avantage du squelette est de préserver, en grande partie, la structure topologique des objets ainsi que les informations contenues dans leur contour.

La squelettisation constitue une partie essentielle de notre travail. En effet, deux types de squelettes seront calculés, le premier est un squelette homotopique 3D pour les modèles stockés dans la base de données et le second est un squelette 2D, calculé directement à partir de la scène. Pour cette raison, nous avons développé une nouvelle méthode de squelettisation 2D, afin d'apporter des améliorations du point de vue robustesse et localisation.

Dans ce chapitre, nous allons d'abord établir un état de l'art sur les différentes approches existantes. Nous commencerons par définir ce qu'est un squelette et présenter ses propriétés. Ensuite, nous passerons en revue les méthodes de squelettisation, nous présenterons à cette occasion les avantages et les lacunes de chacune d'entre elles. Dans la section (5.4), nous allons exposer notre méthode de squelettisation d'objet 2D que nous avons mise au point. Le squelette 2D obtenu est le résultat de la fusion de deux squelettes différents, l'un permet d'obtenir la précision et l'autre garantit la préservation de la topologie et la robustesse. Nous allons décrire en détails les étapes de construction de notre squelette et nous présenterons les résultats obtenus. Finalement, nous décrirons dans la section (5.5) la méthode implémentée pour générer le squelette curviligne 3D d'un objet.

5.2 Définitions et propriétés

Un squelette est une représentation géométrique d'un objet dans une dimension inférieure. Il permet de décrire d'une manière compacte les propriétés d'un objet, en particulier sa forme. Dans le plan, le squelette d'un objet est un ensemble de lignes passant en son milieu appelé axe médian "medial axis". Dans l'espace 3D, il existe deux types de squelettes : les squelettes surfaciques "medial surfaces" et les squelettes curvilignes. Les premiers sont constitués d'un ensemble de voxels de l'objet qui forment une surface d'épaisseur unité et les seconds sont constitués d'un ensemble de voxels de l'objet qui forment une courbe dans l'espace de largeur unité, souvent appelés squelettes homotopiques. C'est de ce type de squelettes dont il s'agit dans nos travaux.

Le concept de squelette a été introduit, pour la première fois, par H. Blum dans l'espace continu [BLU 64]. Il utilise le concept de feu de prairie, c'est-à-dire, des feux provenant des points de contour de l'objet et qui se propagent vers l'intérieur à vitesse constante. Le squelette est alors formé par les points où les fronts de ces feux s'intersectent. Ces points sont aussi appelés points d'extinction.

Une autre définition donnée par L. Calabi [CAL 65], qui considère le problème d'un point de vue topologique. Cette définition est basée sur le concept de boules maximales. Il définit le squelette d'un objet comme étant l'ensemble des centres de ses boules maximales. Une boule incluse dans un objet est dite maximale s'il n'existe pas d'autres boules incluses dans l'objet la contenant entièrement.

Les squelettes présentent quelques propriétés très intéressantes, qui seront exposées dans ce qui suit :

Invariance par translation et rotation

Le squelette est invariant par translation et rotation. Etant donné une translation ou une rotation g et un objet X . Nous avons $Sk(g(X)) = g(Sk(X))$. Avec $Sk(X)$ étant le squelette de l'objet X .

Réversibilité

A partir des points du squelette et des rayons des boules maximales, il est possible de reconstruire la forme. Ainsi la squelettisation est réversible à condition d'avoir mémorisé en chaque point p du squelette, le rayon $r(p)$ de la boule maximale centrée en p . La fonction r est appelée fonction d'étanchéité.

Structure de graphe

Sous certaines hypothèses de régularité, rappelées dans [SCH 93], il est possible de montrer que le squelette a une structure de graphe [RIL 64]. Ainsi, les

techniques issues de la théorie des graphes peuvent être appliquées directement aux objets, par exemple pour les problèmes de mise en correspondance.

Homotopie

La notion mathématique permettant de formaliser le concept d'objets topologiquement équivalents est le type d'homotopie. Deux objets ont même type d'homotopie s'ils sont topologiquement équivalents. C'est-à-dire, si nous pouvons passer de l'un à l'autre par une déformation continue. La condition essentielle est que les points de l'objet, qui étaient proches les uns des autres avant transformation, demeurent proches les uns des autres dans l'objet transformé.

Dans le plan, deux objets homotopes ont le même nombre de composantes connexes et pour chaque composante connexe le même nombre de trous. Cette propriété est fondamentale car elle garantit que l'objet et son squelette ont le même aspect et justifie l'utilisation du squelette comme descripteur de formes.

5.3 Squelettisation 2D

Depuis l'introduction du concept de squelette en tant que descripteur de formes, plusieurs algorithmes de squelettisation ont été proposés dans la littérature. Les différentes techniques de squelettisation peuvent être classées en deux catégories. Les méthodes discrètes, telles que l'amincissement, "le feu de prairie", les champs de potentiel et les cartes de distances. Et les méthodes continues, essentiellement, basées sur l'utilisation du diagramme de Voronoï.

Nous allons maintenant détailler ces différentes approches. Nous ferons ensuite un bilan où nous justifierons le choix effectué.

5.3.1 Algorithme d'amincissement

Ce type d'approche cherche à amincir l'objet tout en gardant sa topologie initiale [GON 90], [KON 89], [TSA 81]. Ce sont des algorithmes itératifs (par balayage successif des images ou par technique de suivi de contours). Ils partent du contour initial de l'objet, étudient la connexité de chaque point du contour dans un voisinage et enlèvent ceux dont la suppression n'affecte pas la topologie de l'objet.

Il existe aussi dans le cadre de la morphologie mathématique, des algorithmes d'amincissement basés sur l'érosion successive du contour de l'objet.

Ce type d'algorithmes est non invariant par rotation. Quant à la localisation obtenue, elle n'est pas correcte.

5.3.2 Simulation de la propagation de feu de prairie

La méthode présentée dans cette partie s'inspire directement d'une propriété des squelettes continus. Cette propriété est que le squelette peut s'interpréter comme le lieu d'extinction d'un front d'onde parti du contour et se propageant vers l'intérieur de la forme à vitesse constante et de façon isotrope. A l'endroit où plusieurs fronts d'ondes se rencontrent, leurs propagations s'arrêtent et donnent la position du squelette.

Pour calculer le squelette dans une image, il suffit donc de simuler la propagation d'une onde. Généralement, l'onde est modélisée par un contour discret qui se déplace selon des techniques de suivi de contour [ARC 81], [PAV 80].

Le fait de simuler un contour discret rend très difficile toute extension 3D. Néanmoins un algorithme procédant par un suivi de surface a été mis en œuvre par Marion-Poty et Miguet [MIG 94].

5.3.3 Squelette à partir de la carte des distances

Ces méthodes calculent le squelette à partir de l'axe médian après une phase de reconnections. Elles sont fondées sur le calcul de cartes de distances. Les points de l'axe médian sont les maxima locaux de la carte de distances. L'utilisation de la carte des distances pour le calcul du squelette est une idée très répandue. La carte des distances d'un objet bidimensionnel est une surface d'élévation où la hauteur pour chaque point de l'objet est égale à la distance de ce point au plus proche contour de l'objet. La projection des arêtes de cette surface sur le plan donne la structure du squelette de l'objet.

Plusieurs travaux ont été effectués pour essayer de détecter ces arêtes dans les cartes de distance discrètes. La plus utilisée est celle de Danielsson [DAN 80], qui est actuellement la plus proche de la fonction de distance euclidienne. D'autres métriques ont été utilisées, mais leur utilisation produit des squelettes très sensibles à la rotation.

Il existe dans la littérature plusieurs méthodes qui sont basées sur le même algorithme, avec quelques variations sur les trois aspects suivant :

- le choix de la métrique pour le calcul de distance,
- détection des maxima locaux,
- méthodes d'assemblage des arêtes du squelette.

5.3.4 Squelette à partir du diagramme de Voronoï

L'émergence de ces méthodes dites continues a été rendue possible grâce aux efforts menés en géométrie algorithmique pour le calcul du graphe de Voronoï et

aussi à l'amélioration de la puissance des ordinateurs.

Initialement le squelette à partir du diagramme de Voronoï était calculé pour les objets de formes polygonales. Le squelette d'une forme polygonale est un sous-ensemble du diagramme de Voronoï de cette forme [KIR 79]. Cette idée a été mise en œuvre par Lee [LEE 82].

Par la suite d'autres auteurs [ATT 94], [OGN 92] utilisent des approches qui calculent les squelettes à partir du diagramme de Voronoï d'un ensemble de points, cet ensemble étant un échantillonnage discret du contour (continu) de l'objet. En effet, Le squelette et le graphe de Voronoï sont deux notions étroitement liées. Pour se rendre compte, il suffit de remarquer que le graphe de Voronoï d'un ensemble fini de points E ($Vor(E)$), peut être interprété comme étant le squelette du complémentaire de E :

$$Sk(R^n - E) = Vor(E) \quad (5.1)$$

En faisant tendre E vers la frontière ∂X d'un objet X dans la formule précédente, il est possible d'obtenir l'inclusion suivante :

$$Sk(R^n - \partial X) = \lim Vor(E) \quad (5.2)$$

Cette formule suggère une stratégie simple pour approcher le squelette d'une forme : il suffit de choisir un sous graphe du graphe de Voronoï de points échantillonnant sa frontière.

Le problème majeur est le choix du sous graphe susceptible d'approcher au mieux le squelette. Il a été proposé d'approcher le squelette par :

- Les sommets de Voronoï inclus dans la forme [SCH 89],
- Les éléments de Voronoï inclus dans la forme [BRA 94],
- Les points de Voronoï inclus dans la forme [OGN 92],
- Le dual de la forme [BOI 92].

Ces méthodes possèdent de nombreux avantages. Le squelette obtenu a une structure de graphe qu'il est facile de parcourir et il existe des théorèmes de convergence [BRA 94], [SCH 89], ce qui n'est pas le cas des autres approches. Contrairement aux méthodes discrètes, les points du squelette connaissent les points du bord de l'objet qui leur sont plus proche et ils s'étendent facilement à l'espace 3D.

5.3.5 Bilan

Les algorithmes du domaine discret que nous venons de voir, sont très populaires, leur utilisation est fréquente car ils ont l'avantage d'être souvent simples, faciles

à mettre en œuvre et rapides en exécution. Mais en contre partie, les squelettes obtenus sont souvent sensibles aux rotations de l'image, la connexité n'est pas forcément conservée et nécessite souvent un post-traitement.

Actuellement, la meilleure façon d'obtenir des squelettes corrects est d'utiliser le diagramme de Voronoï. Les points de contour des objets sont un échantillonnage de la surface continue. A partir de cet échantillonnage, le diagramme de Voronoï de l'image est construit afin d'en extraire le squelette. L'utilisation d'une telle méthode a plusieurs avantages. Seuls les points de contour sont nécessaires, ce qui diminue considérablement la quantité de points de l'image. Le squelette obtenu est connecté et topologiquement équivalent, car la représentation des objets par leur contour conserve implicitement ces notions. Le dernier avantage est de ne pas avoir à utiliser la grille de coordonnées discrète, nous pouvons utiliser la distance euclidienne sans avoir besoin de l'approcher et par conséquent, le squelette n'est plus sensible aux rotations de l'image.

En plus du problème susmentionné relatif au choix de sous-graphe, le problème du domaine continu vient de la construction du diagramme de Voronoï. Les diagrammes présentés dans les articles sont toujours restreints à des germes non alignés et non cocycliques ou cocirculaires alors qu'on désire utiliser des images réelles où ces cas sont nombreux. Ce qui nous oblige à faire bouger les germes et cela entraîne une mauvaise localisation du squelette. De plus, les squelettes restent très sensibles au bruit souvent induit par l'échantillonnage et il est souvent nécessaire d'effectuer un traitement supplémentaire pour élaguer les branches indésirables.

Pour essayer de remédier à ces problèmes, nous proposons une méthode hybride que nous présenterons dans le paragraphe suivant. Dans cette méthode, nous avons combiné astucieusement deux techniques : celle basée sur la carte des distances et celle basée sur le diagramme de Voronoï, et ce afin de profiter des avantages de l'une et de l'autre.

5.4 Méthode hybride développée

Un processus de squelettisation correct doit respecter un certain nombre de conditions (paragraphe 5.2). Les méthodes existantes ne permettent pas de concilier toutes ces propriétés. Pour l'application envisagée, à savoir, la reconnaissance et localisation d'objets, nous devons impérativement obtenir un squelette qui est correctement centré dans la forme et surtout topologiquement équivalent à l'objet. La robustesse aux différents types de bruits est aussi requise car l'application se fera dans des conditions réelles. Afin de respecter ces conditions, nous avons

proposé une méthode hybride qui est une combinaison de deux méthodes, celle basée sur la carte des distances et celle basée sur le diagramme de Delaunay. Cette méthode nous permettra aussi d'obtenir une structure de graphe sans post-traitement [MER 04d] [MER 04e].

5.4.1 Algorithme

Dans un premier temps, à partir d'une image d'origine (figure 5.2), nous déterminons le diagramme de Delaunay (qui est le dual du Voronoï) à partir des points du contour échantillonnés. Pour ce faire, nous avons utilisé une méthode incrémentale qui sera décrite en annexe B. L'union des segments formés en joignant les centres des cercles circonscrits aux triangles, forme le premier squelette. Nous considérons dans cette étape les paires de triangles intérieures à la forme et qui sont adjacents par une arête. Cette procédure nous permet d'avoir un squelette continu $Vobj$ (ligne 2 de la figure 5.1) et de préserver l'homotopie (figure 5.3). Dans le but de ne garder que les éléments les plus significatifs, nous réalisons un processus d'élagage (lignes 4 à 7 de la figure 5.1) à l'aide d'un second squelette DMS (ligne 3 de la figure 5.1) calculé à partir de la carte de distance (figure 5.4). Pour chaque sommet des segments du squelette 1 ($Vobj$), si un cercle de rayon 3 centré en ce sommet intersecte un élément du squelette 2 (DMS) détecté par carte de distances, le segment est conservé (figures 5.5.a et 5.6), sinon il est éliminé (figures 5.5.b et 5.6).

```

1. Squelettisation (image I) :
2.   |  $Vobj \leftarrow \text{SqueletteDiagrammeDelaunay}(I)$  ;
3.   |  $DMS \leftarrow \text{SqueletteCarteDistance}(I)$  ;
4.   | Elagage ( $Vobj, DMS$ ) :
5.   |   | pour chaque segment  $S$  de  $Vobj$  faire
6.   |   |   | si {sommets} de  $S$  intersectent-éléments-de- $DMS$  alors
7.   |   |   |   |  $\text{SqueletteHybride} \leftarrow S$  ;
8.   |   | retourner  $\text{SqueletteHybride}$  ;

```

Figure 5.1: Algorithme de la méthode de squelettisation hybride

Le résultat de l'intersection de ces deux squelettes est illustré sur la figure 5.7.

Dans notre algorithme, la carte de distance de l'objet qui est une surface d'élévation, est calculée au moyen d'un algorithme séquentiel. Pour localiser le squelette, nous détectons les arêtes de cette surface par des opérateurs issus de

la géométrie différentielle. Les détails de cette étape seront présentés en annexe C.

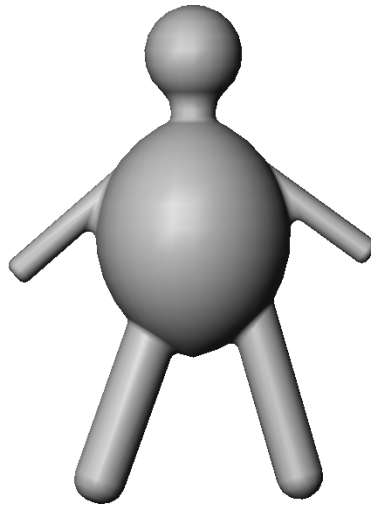


Figure 5.2: Image origine

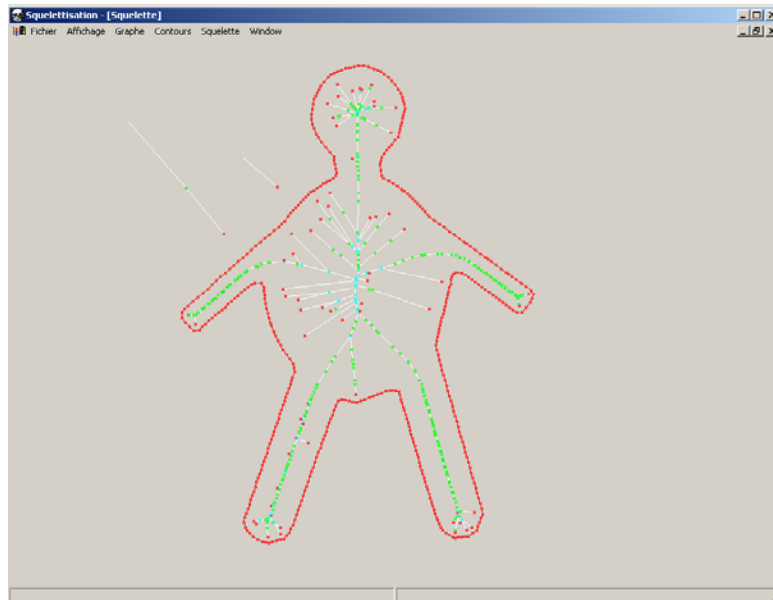


Figure 5.3: Squelette par diagramme de Delaunay

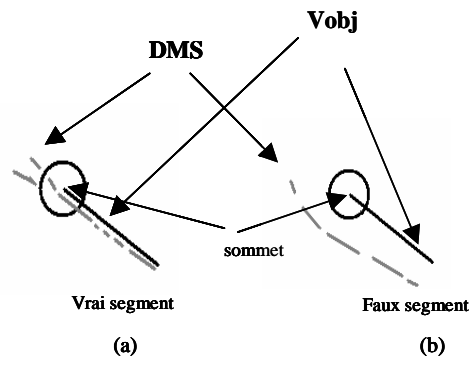


Figure 5.5: Elagage des segments

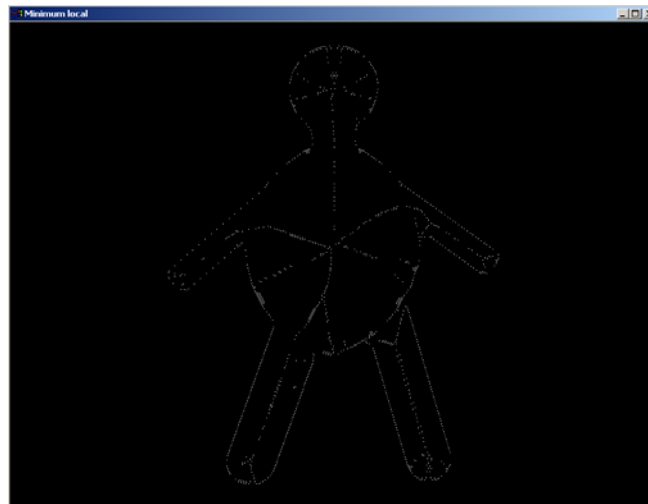


Figure 5.4: Squelette par carte de distance

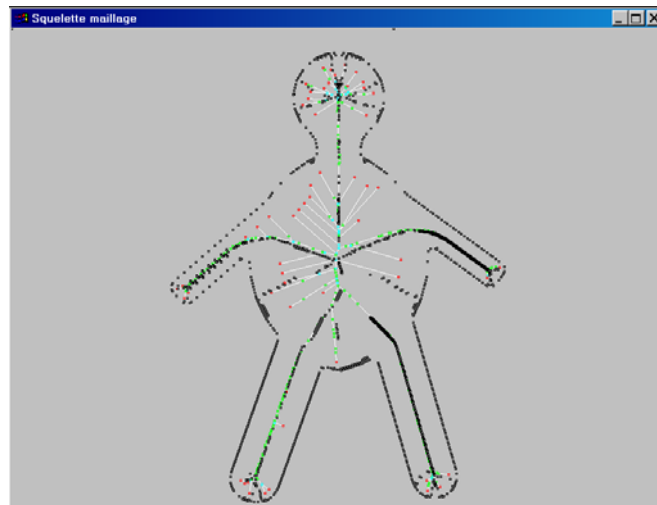


Figure 5.6: Squelette par Delaunay + carte de distance

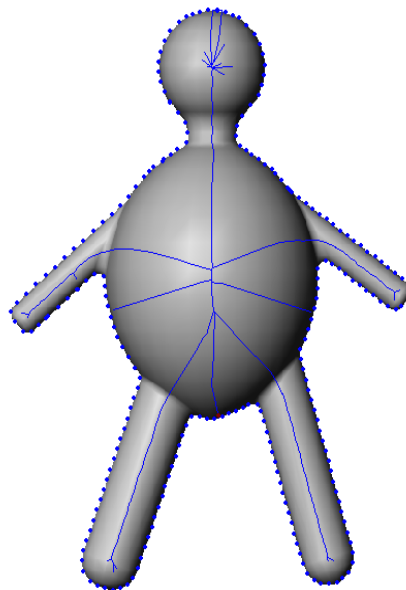


Figure 5.7: Squelette résultant

5.4.2 Évaluation de la méthode proposée

Il n'existe pas de méthodes d'évaluations d'algorithmes de squelettisation, ce qui explique en partie l'absence dans la littérature d'une telle étude. Tout au plus, nous pouvons évaluer les squelettes selon quelques critères subjectifs.

5.4.2.1 Robustesse

La figure 5.8 présente les squelettes obtenus par la méthode d'élagage automatique. Cette figure illustre la robustesse de notre méthode face à une série de formes où la distorsion de la frontière va en augmentant.

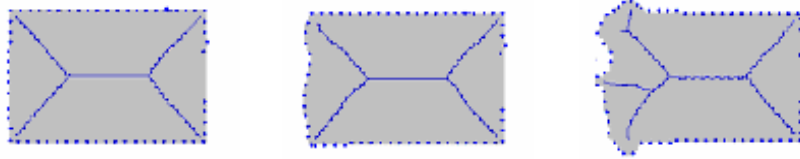


Figure 5.8: Test de robustesse

Même dans le cas de bords fortement dégradés, l'algorithme identifie correctement les points forts du squelette, en montrant une grande ressemblance avec la transformation de l'axe médian sur un rectangle idéal. Des résultats satisfaisants sont également obtenus si des trous ont été rajoutés dans la forme (figure 5.9.c). Ceci nous permet d'avoir une définition plus flexible pour la reconnaissance de formes.

5.4.2.2 Homotopie

L'objectif principal de cette approche est de pouvoir préserver l'homotopie (l'objet et son squelette sont topologiquement équivalents). Cette propriété est préservée par la structure même du diagramme de Voronoï qui construit un squelette continu, mais également par le fait que l'intersection avec le squelette de la carte de distance, le force à rester à l'intérieur de la forme.

Pour vérifier que notre algorithme préserve cette homotopie, nous avons effectué plusieurs tests.

Dans la figure 5.9, nous remarquons que malgré un mauvais échantillonnage, la structure du squelette est la même que celle de l'objet.

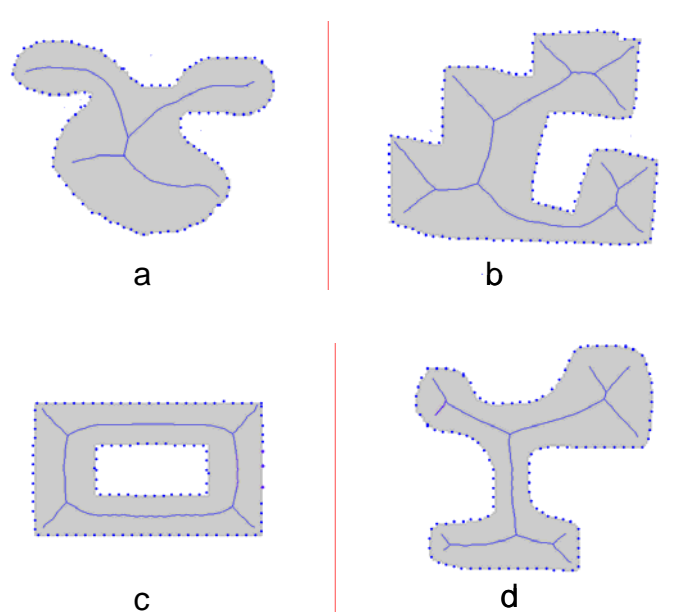


Figure 5.9: Test d'homotopie

5.4.2.3 Localisation

Pour vérifier que le squelette obtenu est bien centré dans la forme, nous avons utilisé des objets canoniques. En effet pour des objets simples tel que les rectangles, triangle, ..etc., nous pouvons déterminer analytiquement l'emplacement du squelette. Pour cela, nous avons utilisé un ovale, un rectangle, un serpent...

Sur toutes les différentes formes que nous avons étudiées, nous avons remarqué que la précision du squelette est fonction du pas d'échantillonnage. Nous illustrons ce problème sur la figure qui représente un ovale, le squelette analytique est représenté sur la figure 5.10. Dans le cas où le pas est de un point du contour sur deux ($1/2$), nous obtenons un squelette bien centré et les extrémités du squelette correspondent exactement au squelette théorique (à un pixel près) figure 5.11. Par contre, nous remarquons la présence de segments parasites. Lorsque le pas est de $1/5$ (figure 5.12), le squelette est toujours centré et les segments parasites disparaissent, mais les extrémités du squelette ne sont pas bien localisées. L'erreur est de 5 pixels d'un côté et de 3 de l'autre. Quand nous arrivons à un pas de $1/15$ (figure 5.13), le squelette n'est pas bien centré et des lignes brisées apparaissent.

Les mêmes constatations ont été observé sur d'autres figures. En résumé, il existe un compromis à faire entre la bonne localisation et la présence de segments parasites.

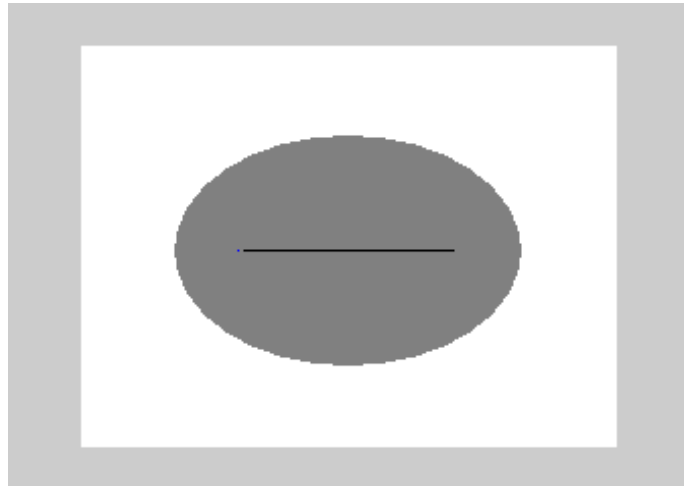


Figure 5.10: Squelette théorique

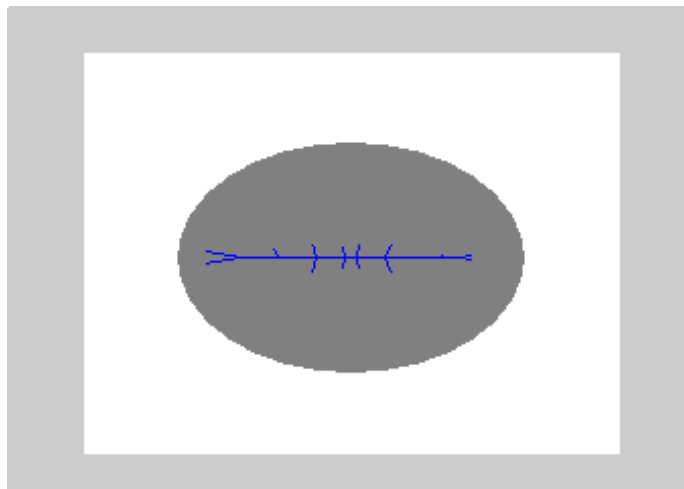


Figure 5.11: Squelette avec un pas de 1/2

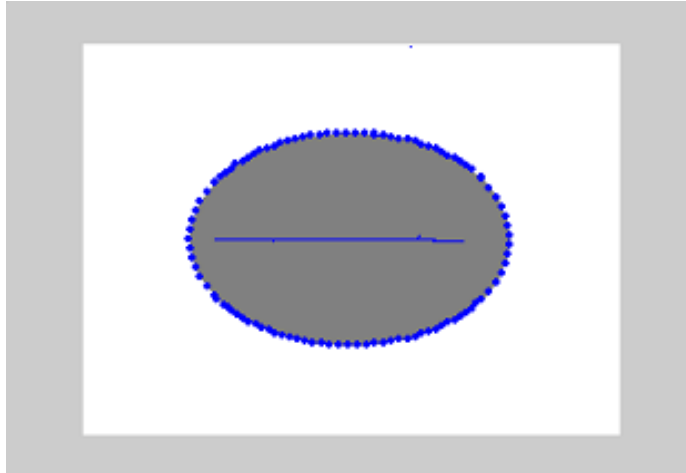


Figure 5.12: Squelette avec un pas de $1/5$

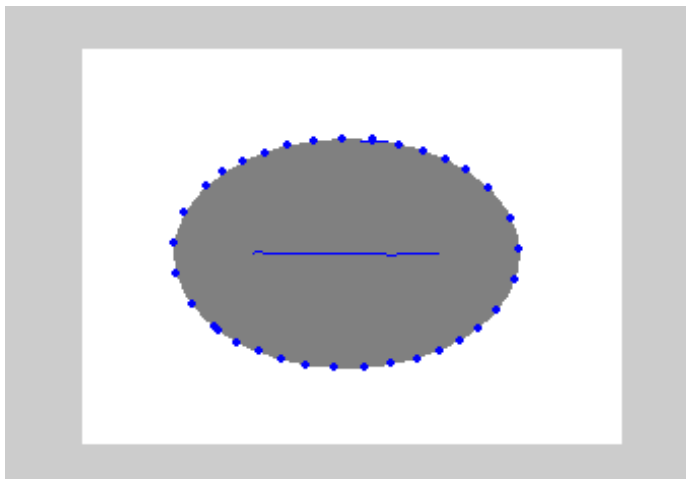


Figure 5.13: Squelette avec un pas de $1/15$

5.5 Squelette homotopique 3D

L'idée générale d'un squelette 3D est la même que pour le cas 2D. Il s'agit de représenter un volume à l'aide d'une structure mince mais ayant autant que possible les mêmes propriétés topologiques. Des méthodes semblables à celles utilisées pour le calcul du squelette dans le plan sont employées. Nous trouvons les méthodes par amincissement, celles basées sur le diagramme de Voronoï et celles basées sur les cartes de distance.

Avant de donner plus précisément un algorithme de calcul, il convient de préciser ce que nous espérons d'un algorithme de squelettisation 3D. En effet deux types de squelettes peuvent être obtenus ayant des propriétés très différentes. Des squelettes surfaciques et des squelettes curvilignes. Tout dépend en fait de l'utilisation envisagée du squelette. Dans notre application il s'agit d'effectuer une comparaison avec un squelette 2D. Dans ce cas notre squelette 3D doit impérativement être de même nature que le squelette 2D, c.à.d curviligne.

5.5.1 Algorithme d'amincissement

Dans cette section nous présenterons l'algorithme d'amincissement d'objets volumiques de Gagvani [GAG 99], que nous avons utilisé. Il est basé sur l'utilisation d'un paramètre d'amincissement "thinness parameter". Ce dernier permet de contrôler partout la densité du squelette et il induit une construction simple et efficace d'un squelette curviligne 3D.

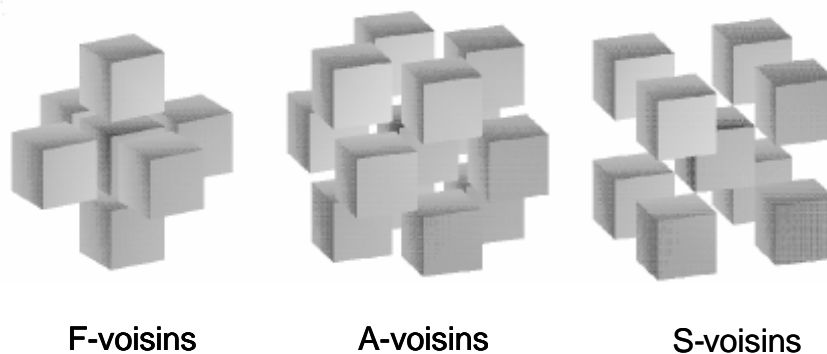


Figure 5.14: Voisins d'un voxel

Pour squelettiser un objet il faut d'abord calculer sa carte de distance, pour cela, plusieurs considérations sont à prendre en compte :

1. Le volume est uniformément échantillonné suivant les trois dimensions. Le voxel est le plus petit élément de cet échantillonnage.
2. Les voxels sont classés en deux types, les voxels-objets et les voxels-fonds.
3. Pour un voxel p , on définit les F-voisins (faces), les A-voisins (arêtes) et les S-voisins (sommets). Deux voxels sont respectivement des F-voisins, A-voisins, S-voisins s'ils ont respectivement une face, une arête ou un sommet en commun. Ce concept est illustré dans la figure 5.14. Cet ensemble forme les 26-voisins d'un voxel.

5.5.1.1 Calcul de la carte de distance 3D

La carte de distance peut être calculée en utilisant des masques de voisinage, basée sur l'idée qu'une distance globale dans un objet est approximée par la propagation de distances locales. Dans [BOR 96], Borgfors décrit une méthode "two-pass method" qui calcule une carte de distance pondérée en trois dimensions. Comme pour le cas 2D, cet algorithme consiste à effectuer deux passages sur le cuboïde total. Le premier passage commence à partir d'un coin et le second par son opposé par rapport à la diagonale. Nous fixons la distance entre tout couple de points, au coût du chemin minimal les rejoignant, formé des déplacements autorisés. L'ensemble des pondérations affectées aux déplacements autorisés sont nombreux, dans [THI 94], le détail de la théorie de ces distances de chanfrein est présenté.

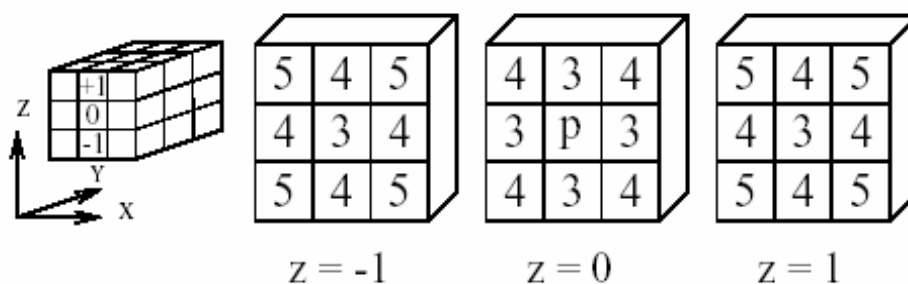


Figure 5.15: Masque de Chanfrein $d_{3,4,5}$

Soit S l'ensemble des voxels de l'objet, S' l'ensemble des voxels du fond et BV indique l'ensemble des voxels du contour. Pour déterminer la carte de distance 3D, nous utilisons une technique qui fait propager le contour vers l'intérieur, en attribuant les valeurs de distance aux voxels de l'objet qui sont voisins aux voxels du contour. Cet algorithme est illustré sur la figure 5.16.


```

1. Squelette3D :
2.   pour chaque voxel  $p$  dans  $S$  faire
3.      $D_p \leftarrow \infty$  ;
4.   pour chaque voxel  $p$  dans  $S$  à un voisin  $q$  dans  $\bar{S}$  faire
5.      $D_p \leftarrow (3 \text{ pour face } / 4 \text{ pour arête } / 5 \text{ pour sommet})$  ;
6.      $B_p \leftarrow B_p \cup \{p\}$  ;
7.   pour chaque voxel  $p$  dans  $B_p$  faire
8.     pour chaque voxel  $r$  dans  $S$  et tant que  $D_r$  est-Modifiable faire
9.       si  $r$  est-Voisin de  $p$  alors
10.         $D_r \leftarrow \min(D_r, D_p + (3 \text{ pour face } / 4 \text{ pour arête } / 5 \text{ pour sommet}))$  ;
11.         $B_r \leftarrow B_r \setminus \{p\}$  ;
12.         $B_r \leftarrow B_r \cup \{r\}$  ;

```

Figure 5.16: Algorithme de calcul de la carte de distance 3D

5.5.1.2 Extraction du squelette 3D

Une fois la carte de distance calculée, les voxels qui formeront le squelette doivent être identifiés.

Cette méthode utilise une heuristique qui compare la distance d'un voxel avec celles de ses 26-voisins. Si la distance de ce voxel est plus grande que la distance de ses 26-voisins, alors la sphère centrée en ce voxel inclut toutes les sphères centrées sur ses 26-voisins. Le paramètre d'amincissement détermine alors la taille d'une telle sphère qui devrait être centrée sur le voxel pour qu'il soit considéré comme étant important par rapport au contour.

Pour chaque voxel p , la moyenne MNT_p des distances de ses 26-voisins est calculée. Le paramètre d'amincissement TP est la différence de sa distance D_p avec MNT_p .

$$TP = D_p - MNT_p \quad (5.3)$$

pour chaque voxel p de l'objet, avec,

$$MNT_p = \frac{\sum_{i=1}^{26} D_{q_i}}{26} \quad (5.4)$$

avec $p, q_i \in$ voxels de l'objet et q_i est un 26-voisin de p .

Une petite valeur de TP indique que p est important pour le contour, si sa distance est légèrement plus grande que celle de ces voisins. Une grande valeur de TP suppose que p doit avoir une valeur plus grande que ces voisins et par conséquent, si TP augmente, le nombre de sphères couvrant l'objet s'amplifie, les bases théoriques de cette méthode sont exposées dans [GAG 99]. Ainsi le paramètre d'amincissement permet de représenter un objet volumique avec plusieurs niveaux de détails. Pour une valeur donnée du paramètre d'amincissement, on peut rapidement extraire les voxels dont les valeurs de distance sont plus grandes que celles de ces voisins. Nous calculons $D - MNT$, la différence entre la distance et la moyenne des distances des voisins pour chaque voxel. Les voxels sont rangés par ordre décroissant suivant la valeur de $D - MNT$. Pour un nombre donné de voxels n , on extrait les n premiers voxels de cette liste.

Nous avons remarqué que quand une petite valeur de TP est utilisée, les points se retrouvent sur une surface médiane (squelette surfacique). Lorsque TP augmente, les points qui restent sont ceux qui se trouvent sur l'axe médian.

L'avantage de cette approche est sa simplicité, et la possibilité de régler la finesse du squelette. Les figures 5.17 et 5.18 montrent le comportement de cette méthode sur des objets canoniques, le cube et le cylindre sont modélisés par 3DSmax. La figure 5.19 illustre le squelette homotopique d'un objet Cactus modélisé à l'aide de PovRay.

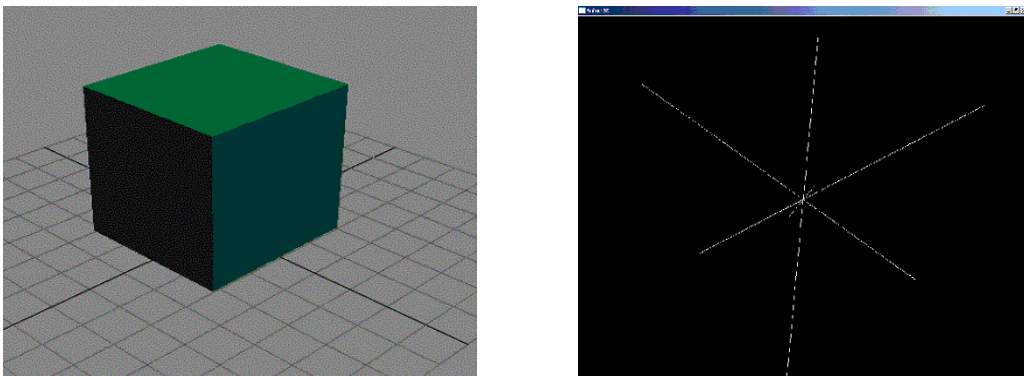


Figure 5.17: Cube et squelette curviligne

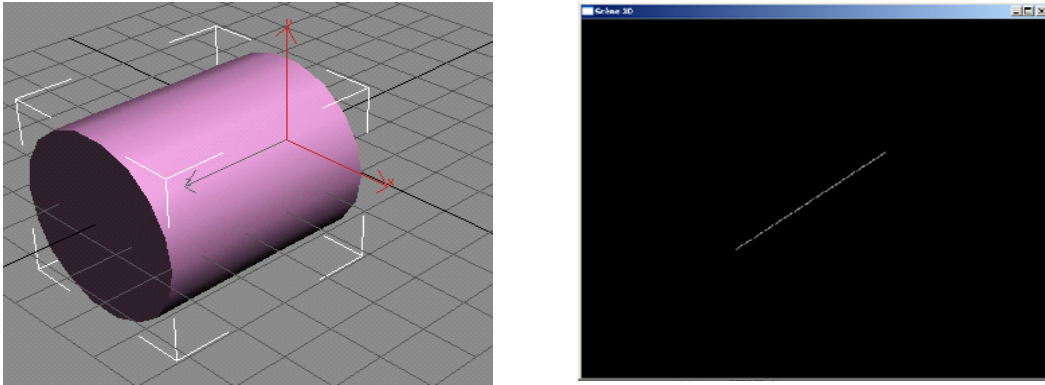


Figure 5.18: Cylindre et squelette curviligne

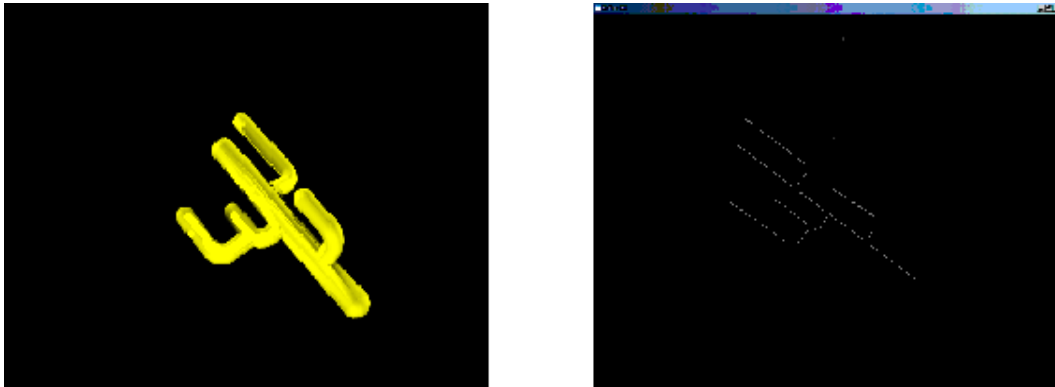


Figure 5.19: Cactus et squelette curviligne

Le squelette obtenu par cette méthode n'est pas exploitable comme tel. Un post-traitement est alors nécessaire pour avoir un squelette connexe et précis. Pour réaliser cette connexion, nous avons utilisé un puissant outil de modélisation qu'est SolidWorks.

La démarche est d'approximer le squelette 3D à l'aide de la méthode de Gagnani en ajustant le paramètre d'amincissement. Cette étape nous permet d'avoir un squelette fin et précis sans se soucier de la connexité. Par la suite à l'aide de SolidWorks, nous connectons manuellement les éléments du squelette. Nous rappelons que nos objets sont aussi modélisés à l'aide de ce logiciel et que toutes ces étapes sont exécutées off-line.

Pour compléter notre étude, nous avons comparé visuellement les squelettes 2D et 3D de plusieurs objets. Dans les figures 5.20 à 5.24, nous présentons des exemples de l'objet "Pantin" pris sur des points de vue différents. Les figures de gauche représentent le modèle 3D avec le squelette homotopique centré dans

la forme. Les figures du milieu représentent le contour de l'objet projeté ainsi que le squelette 2D brut. Les figures de droite représentent le squelette 2D après traitement (squelette hybride).

Comme nous pouvons le voir sur ces figures, les squelettes 2D et 3D sont topologiquement équivalents, homotopes à l'objet qu'ils représentent et ils ont les mêmes composantes connexes. Par contre le squelette 2D n'est pas la projection du squelette 3D. Ceci est dû au fait que les éléments centraux subissent des distorsions, qui est équivalent au problème d'auto-occlusion. Contrairement aux éléments proéminents, qui restent invariants par projection. Ce qui justifie leurs utilisations dans un processus de reconnaissance de formes 2D/3D. Cependant, il peut arriver qu'un ensemble connexe disparaisse du squelette 2D (figure 5.23). Par conséquent, nous pouvons les utiliser pour une mise en correspondance 2D/3D et la puissance des techniques d'isomorphisme de graphes peut remédier à ces problèmes.

5.6 Conclusion

Le but de cette recherche est de pouvoir étudier les possibilités d'extraire un squelette d'un objet. Dans la première partie de cette étude un survol des méthodes de squelettisation est présenté puis analysé.

Ensuite, afin de surmonter les limitations des méthodes basées sur la carte des distances ou de celles basées sur le squelette de Voronoï, une alternative est proposée qui exploite aussi bien les informations de la carte de distances que l'information de connectivité contenue dans le diagramme de Voronoï appliquée aux frontières échantillonnées de l'objet. L'intersection des deux squelettes garantit la connectivité complète, l'homotopie ainsi que l'invariance géométrique. Cet algorithme de squelettisation a été testé sur des images variées et les résultats expérimentaux ont été présentés et analysés. Cet algorithme est capable d'extraire un ensemble de lignes connectées correspondant au squelette désiré. Pour l'instant, le principal défaut de cette recherche est la dépendance du graphe à l'échantillonnage de la frontière. En ce qui nous concerne, les squelettes seront utilisés pour la reconnaissance et la localisation de l'objet par des méthodes d'appariement de graphes. Pour cela les propriétés du squelette qui nous intéressent sont l'homotopie et la bonne localisation, ainsi le problème de présence de segments parasites peut être contourné en utilisant une technique d'isomorphisme de graphes robuste qui sera présentée au chapitre suivant.

En dernier, nous avons présenté une méthode de squelettisation 3D [GAG 99].

Cette méthode convient à notre implémentation car elle permet de régler la finesse et la précision du squelette avec un paramètre. Rappelons qu'il s'agit d'un squelette curviligne et qu'il est calculé hors-ligne.

Maintenant que nous avons exposé la phase de modélisation en squelette des objets, nous passerons à l'étape de mise en correspondance. Dans le chapitre suivant, nous exposerons la méthode d'appariement de deux squelettes, issue des techniques de la théorie des graphes.

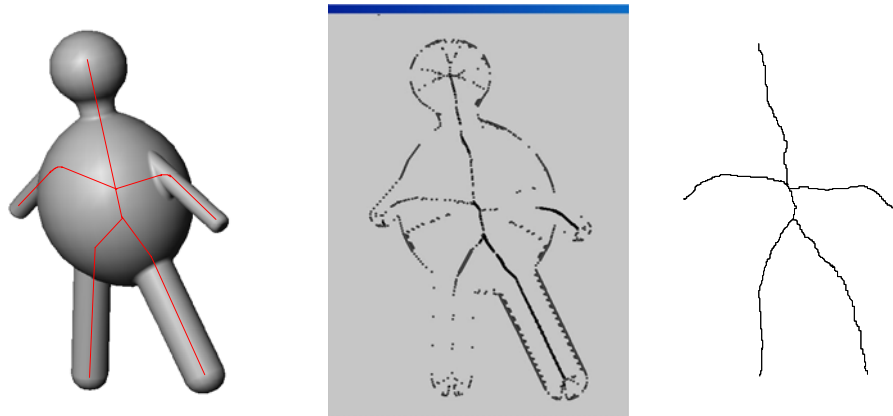


Figure 5.20: Squelettes 2D et 3D -1-

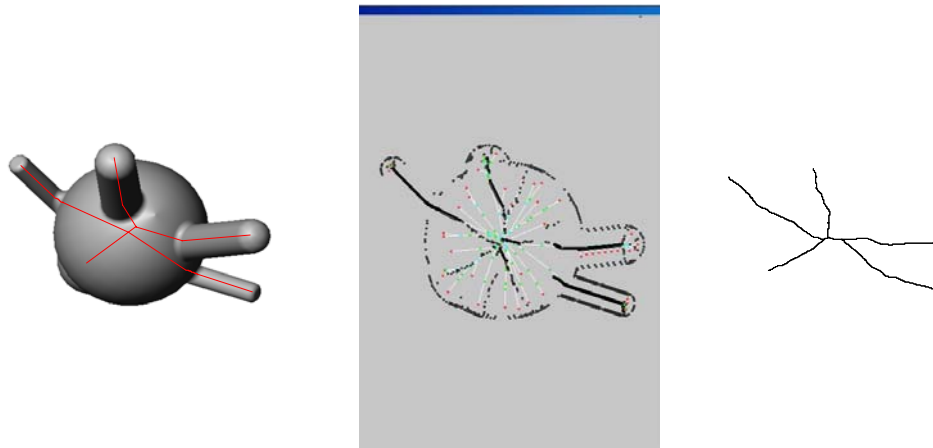


Figure 5.21: Squelettes 2D et 3D -2-

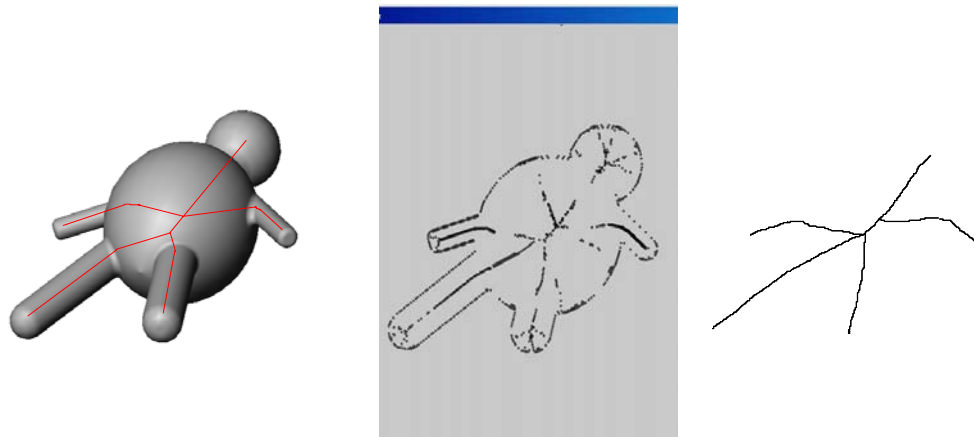


Figure 5.22: Squelettes 2D et 3D -3-

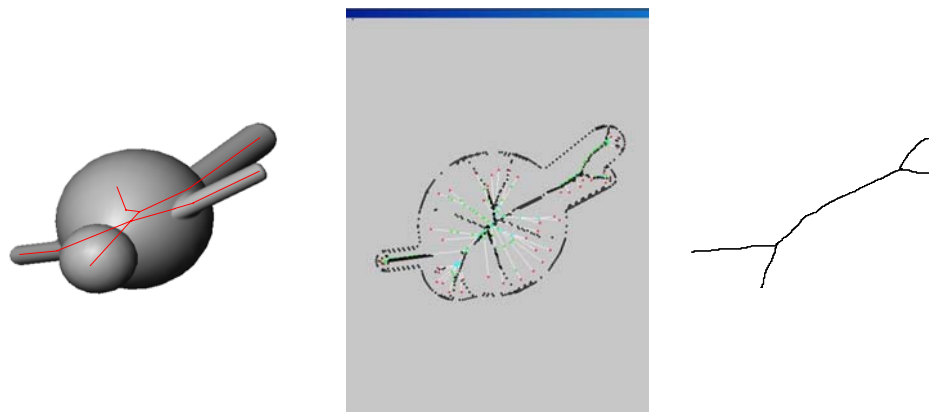


Figure 5.23: Squelettes 2D et 3D -4-

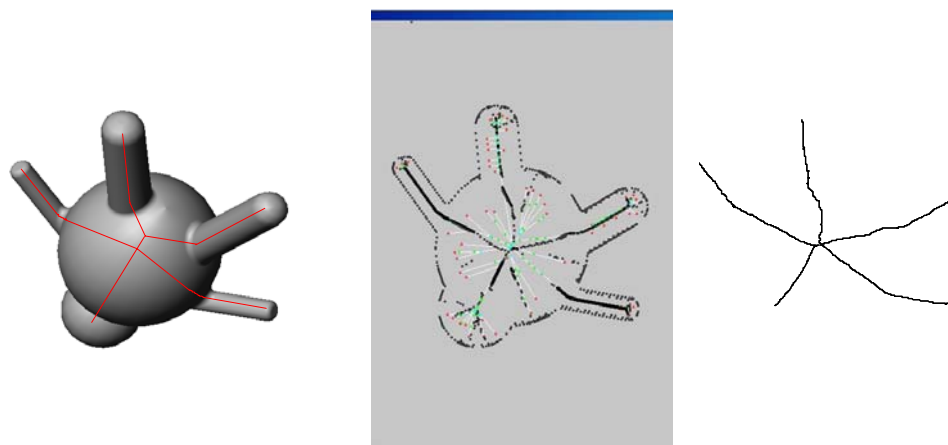


Figure 5.24: Squelettes 2D et 3D -5-

Chapitre 6

MISE EN CORRESPONDANCE DE GRAPHES

6.1 Introduction

Le but à atteindre est d'effectuer une mise en correspondance entre l'image 2D de l'objet et son modèle 3D de la base de données. Cela revient à faire un appariement entre le squelette 2D extrait de l'image et le squelette 3D du modèle. Etant donné que le squelette possède une structure de graphe, cette opération consistera à effectuer un isomorphisme entre deux graphes.

Dans le contexte de vision monoculaire de scènes réelles, cette mise en correspondance ne peut pas être complète. En effet, en raison des problèmes liés à la présence de bruits dans l'image ainsi qu'aux occlusions (et d'auto-occlusions), l'appariement consistera à effectuer une mise en correspondance entre une portion du squelette 2D et une portion du squelette 3D.

Cette section sera consacrée à l'étude des différentes techniques d'isomorphisme de graphes (sous-graphes) et à la mise en œuvre de l'une de ces méthodes.

Dans un premier temps, les différents algorithmes de mise en correspondance de graphes, utilisés en vision et en reconnaissance de formes, sont explorés. Dans un second temps, c'est la construction des graphes à partir des squelettes qui va être décrite et enfin nous exposerons en détails la méthode choisie.

Avant de commencer nous présenterons rapidement quelques définitions qui sont nécessaire à la compréhension du formalisme utilisé dans ce chapitre.

6.2 Terminologie et notations de base

Soit un graphe $G = (V, E)$, dans sa forme de base il est composé de nœuds et d'arêtes. Soit V l'ensemble des nœuds et $E \subset V \times V$ l'ensemble des arêtes du graphe G .

L'ordre d'un graphe G est défini par le nombre de nœuds de G et il est représenté par $|V|$, le nombre d'arcs par $|E|$.

Si deux nœuds dans G , nommés $u, v \in V$, sont connectés par une arête $e \in E$, ceci est noté par $e = (u, v)$ et les deux nœuds sont appelés *adjacents* ou *voisins*. Les arêtes sont appelées non orientées si elles ne possèdent aucune direction, et un graphe G ne contenant que ce type d'arêtes est dit non orienté. Quand tous les arcs ont des directions et par conséquent sont distinguables, le graphe est dit orienté. Habituellement, le terme arc est utilisé quand le graphe est orienté, et le terme arête est utilisé quand le graphe est non orienté. Dans notre exposé les graphes orientés sont utilisés, mais l'appariement de graphes peut aussi bien s'appliquer aux graphes non orientés. Un graphe orienté $G = (V, E)$ est dit complet, s'il existe toujours un arc $(u, u') \in E = V \times V$ entre deux nœuds u, u' dans le graphe.

Les nœuds et les arêtes d'un graphe peuvent contenir des informations. Quand cette information est une simple étiquette (i.e. un nom ou un nombre) le graphe est dit graphe *étiqueté*. Quand les nœuds et les arcs contiennent plus d'informations, ils sont appelés nœuds et arcs d'attributs et le graphe est dit graphe d'*attributs*.

6.3 Etat de l'art dans la littérature

Cette section est un tour d'horizon sur l'appariement de graphes. Toutes les références commentées correspondent aux méthodes et problèmes d'appariement de graphes. Bien que certaines d'entre elles ne correspondent pas aux types de problèmes que nous avons étudiés, elles permettent d'avoir une idée sur les différents sujets et groupes qui travaillent dans ce domaine. Le lecteur intéressé peut aussi consulter, [JOL 01] et [KRO 99], qui présentent des travaux dans les différents aspects discutés dans ce paragraphe.

6.3.1 Appariement par mesures de distance

Le graphe de distances (The graph edit distance) entre deux graphes est défini par le nombre de modifications que nous devons entreprendre pour arriver d'un graphe à un autre. La distance entre deux graphes est définie par la somme pondérée des coûts d'édition des opérations (insertion, suppression et changement d'étiquettes des nœuds et des arêtes) pour transformer un graphe en un autre. Le processus de mise en correspondance tend à déformer le modèle du graphe initial jusqu'à ce qu'il colle avec le modèle de l'autre graphe. La mise en correspondance est ainsi obtenue et le coût des opérations correspond à la mesure de sa qualité.

Une autre approche intéressante pour résoudre le problème d'appariement de graphes est la recherche de la clique maximale dans un graphe d'association. Des exemples de cette approche sont présentés dans [DOO 80], [MAS 01].

Un grand nombre d'applications de ces techniques peuvent être trouvées dans la littérature. Un exemple est présenté dans [HAR 99], où une méthode d'extraction et d'étiquetage du l'arbre artériel coronaire est proposée. Dans [GEU 99], l'appariement de graphes de distance est appliqué à la segmentation des tissus, en basant la caractérisation de tissus sur le rapport topographique entre les cellules. Le voisin de chaque cellule dans le tissu est modélisé par les distances aux cellules du contour, et la comparaison avec un exemple, ou le voisin du prototype, révèle la ressemblance topographique entre tissu et modèle. Dans [FOG 99], la reconnaissance des chiffres manuscrits à partir d'une base de données de caractères standard est résolue en utilisant cette méthode.

6.3.2 Appariement de graphes par les algorithmes génétiques

Le fait de formuler le problème complexe de mise en correspondance de graphes comme un problème d'optimisation combinatoire n'est pas nouveau, et plusieurs références utilisant différentes techniques dans ce domaine peuvent être trouvées dans la littérature. Les algorithmes génétiques ne sont qu'un exemple de ce type d'approches.

Certains travaux dans la littérature se sont concentrés initialement sur la recherche des différents types d'opérateurs de croisement et de mutation appropriés à l'appariement de graphes. Dans [KHO 02], les auteurs présentent une comparaison entre les différents opérateurs génétiques, et ils comparent les performances des algorithmes génétiques quand on utilise deux différents types de représentation individuelles. Dans [SIN 97], un algorithme évolué sans opérateurs de croisement est présenté dans le but d'obtenir une convergence rapide vers la solution. Les auteurs expliquent également la manière de paralléliser leur algorithme.

6.3.3 Appariement de graphes utilisant les techniques basées sur la théorie probabiliste

On peut trouver dans la littérature beaucoup de techniques qui appliquent la théorie probabiliste aux problèmes d'appariement de graphes. Les premiers travaux qui appliquent la théorie de probabilité pour l'appariement de graphes [KIT 93] emploient une approche itérative, qui utilise une méthode appelée la relaxation

probabiliste. Dans [WIL 99], une étude comparative de plusieurs stratégies discrètes et déterministes de recherches, pour la mise en correspondance de graphes, est présentée, en se basant sur la mesure Bayésienne antérieurement rapportée dans [WIL 97]. Malheureusement, tous ces travaux ne considèrent pas les dépendances entre plus de deux nœuds.

D'autres exemples de techniques basées sur la théorie probabiliste sont présentés dans ces références, abordant les sujets suivants : la distinction des régions faciales en utilisant le recuit simulé [HER 98], l'authentification multimodale des personnes utilisant la relaxation probabiliste [MAR 00], l'application de l'algorithme EM pour l'appariement inexact de graphes basée seulement sur leurs connectivités [LUO 01] et finalement, la relaxation probabiliste utilisée pour résoudre le problème d'appariement de graphes, quand ils sont formulés en une structure Bayésienne pour l'attribution d'étiquettes contextuelles [CHR 95].

6.3.4 Appariement de graphes utilisant les réseaux de neurones

Les réseaux de neurones ont aussi été largement appliqués aux problèmes d'appariement. Des types très différents de réseaux de neurones ont été testés afin d'essayer de trouver le plus convenable pour chaque type de problème. Les exemples de ces derniers sont : l'utilisation d'un regroupement de réseaux de neurones pour la reconnaissance automatique de sillons corticaux du cerveau humain dans les images IRM [RIV 02], l'utilisation d'un réseau de neurones Hopfield 3D pour la reconnaissance du langage des signes [HUA 98]. D'autres exemples d'utilisation de réseaux de neurones sont : les problèmes d'authentification des surfaces frontales [KOT 00] et la reconnaissance des caractères manuscrits Chinois [SUG 98].

6.4 Classification des problèmes d'appariement de graphes

Dans ce travail, nous avons considéré le problème de reconnaissances de formes, basées sur les modèles (model-based method), où le modèle et l'objet dans la scène sont représentés par des graphes (graphe du modèle et graphe de données, G_M , G_D respectivement). D'une manière intuitive la reconnaissance de l'objet correspond à un processus qui tend à associer "au mieux " les deux graphes, en mettant en correspondance des nœuds qui partagent les mêmes propriétés.

D'un point de vue plus formel on retrouve de nombreuses définitions de la mise en correspondance de graphes qui sont présentées ci-dessous.

6.4.1 Appariement exact et inexact de graphes

Les problèmes d'appariement exact et inexact de graphes sont appelés dans la littérature respectivement les problèmes d'appariement isomorphiques et homomorphiques de graphes.

Etant donné deux graphes, le graphe du modèle G_M et le graphe de données G_D , la procédure de les comparer implique la vérification de leur ressemblance. Généralement, on peut définir le problème d'appariement de graphes comme suit: étant donné deux graphes $G_M = (V_M, E_M)$ et $G_D = (V_D, E_D)$, avec $|V_M| = |V_D|$, le problème est de trouver une bijection $f : V_D \rightarrow V_M$ tel que si $(u, v) \in E_D$ alors $(f(u), f(v)) \in E_M$. Si une certaine correspondance f existe, on l'appelle un isomorphisme, et G_D est dit isomorphe à G_M . Ce type de problème est dit *appariement exact de graphes*.

Le terme "*inexact*" appliqué à certains problèmes de mise en correspondance de graphes veut dire qu'il n'est pas possible de trouver un isomorphisme entre les deux graphes. C'est le cas lorsque le nombre de nœuds est différent dans les deux graphes (modèle, données). Ceci peut résulter de l'aspect schématique du modèle et de la difficulté de segmenter correctement l'image en des entités significatives. Par conséquent, on ne peut pas s'attendre à un isomorphisme entre les deux graphes, et le problème de mise en correspondance de graphes ne consiste plus en la recherche d'un résultat exact de mise en correspondance, mais en la recherche du meilleur appariement entre eux. Cela mène à une classe de problèmes connue sous le nom d'*appariement inexact de graphes*. Dans ce cas, la mise en correspondance vise la recherche d'une correspondance non bijective entre le graphe de données et le graphe du modèle. Dans ce qui suit, on suppose que $|V_M| < |V_D|$.

L'intérêt de la mise en correspondance inexacte de graphes a augmenté récemment à cause de l'application de la vision en des domaines tels que la cartographie, la reconnaissance de caractères et la médecine. Dans ces domaines, la segmentation automatique des images résulte en une sur-segmentation et par conséquent, le graphe de données contient plus de nœuds que le graphe du modèle.

Dans le problème d'appariement inexact de graphes tels que $|V_M| < |V_D|$, le but est de trouver une application injective $f' : V_D \rightarrow V_M$ tel que si $(u, v) \in E_D$ alors $(f'(u), f'(v)) \in E_M$. Cela correspond à la recherche d'un petit graphe dans un autre plus grand. Un sous-type important de ces problèmes est *le problème d'appariement de sous-graphes*, dans lequel nous avons deux graphes $G = (V, E)$

et $G' = (V', E')$, tel que $V' \subset V$ et $E' \subseteq E$, dans ce cas le but est de trouver une correspondance $f' : V' \rightarrow V$ tel que si $(u, v) \in E'$ alors $(f'(u), f'(v)) \in E$. Si une telle correspondance existe, elle est appelée isomorphisme de sous-graphes.

6.4.2 Appariement de graphes en utilisant les nœuds factices (nuls)

Dans quelques cas de mise en correspondance de graphes inexact, le problème rencontré est de trouver une correspondance un-à-un des nœuds. Des exemples réels de ce type de problèmes de mise en correspondance de graphes sont présentés dans [FIN 98]. Ce cas est rencontré lorsque la procédure de sur-segmentation, utilisée dans la construction de graphe de données, est exécutée automatiquement et donc beaucoup de régions paraîtront comme des composants qui ne figurent pas dans le graphe du modèle.

Des exemples peuvent être trouvés dans le cas de la reconnaissance des structures dans les images de cerveau. L'atlas anatomique, celui que le graphe du modèle représente, est désigné par un cerveau sain. Cependant, si l'image de données, qui doit être reconnue, contient des pathologies comme les tumeurs, la procédure d'appariement de graphe est incapable de mettre en correspondance les nœuds d'une manière satisfaisante de n'importe quelle région de cerveau à reconnaître. Des exemples similaires peuvent être trouvés dans les images satellites, par exemple, lorsque le modèle de graphe est extrait d'une image obtenue depuis quelques années avant la photographie actuelle à analyser, et des nouvelles routes construites sont présentes.

Plus formellement, étant donné deux graphes $G_M = (V_M, E_M)$ et $G_D = (V_D, E_D)$, le problème consiste en la recherche d'un homomorphisme $h : V_D \rightarrow V_M \cup \phi$, tel que ϕ représente la valeur nulle, c'est-à-dire si pour un nœud $a \in V_D$, il existe $h(a) = \phi$, alors il n'existe pas de correspondance dans le graphe du modèle pour le nœud a dans le graphe de données. Cette valeur ϕ est connue dans la littérature comme étant le *nœud nul* ou le *nœud factice*.

Il est important de noter que l'utilisation de nœuds factices peut être considéré comme un nœud supplémentaire dans le graphe du modèle. Cependant dans certains problèmes, le cas opposé peut se produire : en utilisant le même exemple du cerveau humain, l'algorithme d'appariement de graphes doit faire face à une reconnaissance d'une image de données d'un patient qui a subi une lobotomie, et par conséquent le lobe n'existe pas dans l'image. Dans ce cas, le problème d'appariement de graphes ne doit pas trouver une correspondance dans la solution optimale de tous les modèles de nœuds, mais il doit utiliser les nœuds factices. Cet aspect est aussi commun dans d'autres problèmes de reconnaissance d'image tel

que le problème de reconnaissance d'images aériennes, dans lequel les anciennes constructions et usines laissent la place, dans la nouvelle photo, aux nouveaux parcs et routes.

6.4.3 Appariement de graphes nécessitant plus d'une correspondance par nœud

D'autres problèmes d'appariement de graphes nécessitent des mises en correspondance plusieurs à plusieurs.

Etant donné deux graphes $G_M = (V_M, E_M)$ et $G_D = (V_D, E_D)$, le problème consiste en la recherche d'un homomorphisme $f : V_D \rightarrow W$ tel que $W \in P(V_M) \setminus \{\phi\}$.

Ce type de problèmes est le plus difficile à résoudre. Comme la complexité de recherche du meilleur homomorphisme a beaucoup plus de combinaisons, l'espace de recherche de l'algorithme de mise en correspondance de graphes est beaucoup plus grand. Ce type de problèmes d'appariement de graphes a un sens quand l'étape de segmentation de l'image ne satisfait pas la condition de segmenter totalement toutes les régions, et donc, quelques régions de la segmentation automatique appartiennent à deux ou plus de modèles de régions.

Une difficulté majeure dans les problèmes d'appariement de graphe, permettant plus qu'un seul appariement par nœud de données, est la conception de la fonction d'énergie qui mesure la qualité de chaque homomorphisme possible. En outre, le nombre de nœuds modèles apparié à chaque nœud de données doit rester aussi bas que possible afin que l'algorithme d'appariement de graphes puisse obtenir des résultats concrets.

Par rapport à notre approche, notre problème d'appariement regroupe les trois catégories évoquées précédemment. Premièrement, nous avons à effectuer un appariement inexact de graphe car nous travaillerons sur des scènes réelles et la présence des bruits est inévitable. Deuxièmement, pour estimer la pose nous avons besoin d'apparier les primitives 2D de la scène et les primitives 3D du modèle, nous aurons alors à effectuer une mise en correspondance un-à-un des nœuds. Finalement, l'étape de mise en correspondance doit générer plusieurs hypothèses pour pouvoir effectuer des projections et des vérifications. Pour cette raison nous devons permettre à un nœud d'être apparié à plusieurs.

6.5 Complexité d'appariement de graphes

La mise en correspondance de graphes est considéré comme l'un des problèmes les plus complexe dans la reconnaissance d'objets [BIE 87]. Sa complexité est due à sa nature combinatoire. Comme les problèmes d'appariement présentés dans la section (6.4) sont de nature différente, nous allons analyser leur complexité séparément.

Les problèmes d'appariement exact de graphes ont été jusqu'à présent classifiés comme étant dans P ou $NP - complets$. Quelques articles dans la littérature essaient de prouver qu'elle est $NP - complet$ quand les deux graphes à apparier sont de types particulier ou bien dans le cas où ils possèdent quelques contraintes spécifiques [BAS 94], [GAR 79]. D'autre part, pour quelques types de graphes, ils ont démontré que la complexité du problème d'isomorphisme est polynomiale, par exemple pour l'isomorphisme des graphes planaires [HOP 74].

Le problème de l'isomorphisme de sous-graphes a été prouvé comme étant $NP - complet$ [GAR 79]. Cependant, certains types de graphes spécifiques peuvent avoir une complexité moins importante. Par exemple, le cas particulier, où le grand graphe est une forêt et le petit, qui doit être apparié, est un arbre, sa complexité est polynomiale [GAR 79], [REY 77]. Finalement, dans l'appariement inexact de graphes, la complexité a été prouvé dans [ABD 98] comme étant $NP - complet$.

6.6 Appariement de deux squelettes

L'étude de l'état de l'art nous indique qu'il n'existe pas d'algorithme parfait d'appariements de graphes, ce qui explique en partie le dynamisme des recherches dans ce domaine. Le choix de la méthode est souvent conditionné par l'application à laquelle la mise en correspondance est dédiée.

Dans notre situation, il s'agit de mettre en correspondance deux graphes représentant les squelettes. Le premier désigne l'objet dans la scène et le second l'objet dans une base de données. Pour cette application, nous avons besoin à la fois d'une mesure globale de similarité et d'une mise en correspondance nœud à nœud. La mesure globale nous permettra d'indexer rapidement la base de données et l'appariement un-à-un des nœuds, nous permet de trouver la pose de l'objet et implicitement valider l'étape de reconnaissance. L'introduction du bruit (rajout/suppression de nœuds) et/ou le problème d'occlusions, empêche l'existence de grands graphes isomorphiques. Par conséquent, nous avons besoin d'un algorithme de mise en correspondance qui pourra trouver des sous arbres isomorphiques dans ces conditions.

Pour accomplir ces tâches, nous avons choisi une représentation topologique des arbres invariante par rapport à des légères perturbations de la structure. Cette approche a été utilisée par Siddiqi pour l'appariement de deux shock-graphes [SID 98]. Nous avons procédé à des modifications pour rendre l'appariement plus robuste au niveau des nœuds terminaux et aussi pour réaliser la polygamie des nœuds, c'est à dire un nœud peut être apparié à plusieurs nœuds.

Dans ce paragraphe, nous commencerons par exposer la construction des graphes à partir du squelette et nous décrirons l'étape d'indexation de la base de données. Pour la mise en correspondance un-à-un des nœuds, nous présenterons la méthode d'appariement de Siddiqi et les modifications que nous lui avons apportées.

6.6.1 Construction du graphe à partir du squelette

Pour faciliter la recherche d'isomorphisme, les squelettes doivent être transformés en graphes acycliques orientés. Pour le squelette 3D cette procédure se fera hors ligne et le graphe sera stocké dans la base de données avec le modèle. Pour le squelette 2D, la construction du graphe se fera en ligne avec la construction du squelette. Durant ce processus, nous devons nous assurer que les informations sur la forme sont préservées. Nous devons aussi nous assurer que cette procédure est assez tolérante afin que des changements mineurs du squelette ne produisent pas des graphes radicalement différents.

En premier, nous générons un graphe acyclique non orienté (figure 6.1.b) à partir du squelette (figure 6.1.a). Chaque nœud dans le graphe du squelette représente un ensemble de points du squelette possédant la même distance au bord de l'objet, appelée poids. Les arcs représentent les chemins de connexion (figure 6.1.c). En appliquant un algorithme pour extraire l'arbre couvrant minimal (ACM), nous obtenons un arbre acyclique non orienté. Pour construire une structure hiérarchique, les arcs sont orientés des nœuds à poids forts vers les nœuds à poids faibles. Le graphe orienté (figures 6.1.c et 6.1.d) est ainsi créé dont le principe est similaire au concept de shock graphe [SHO 01]. La racine du graphe est le nœud qui a le plus fort poids. S'il existe plusieurs nœuds, ayant le poids le plus fort, celui qui possède plus de descendants directs sera choisi comme tel.

Rappelons que l'ACM est sensible aux variations de la distance au niveau des jonctions, ce qui pourrait engendrer une structure de connectivité incorrecte. Cependant la méthode d'appariement utilisée permet de s'accommoder à ces perturbations.

Pour chacun des nœuds du graphe des informations topologiques, géométriques

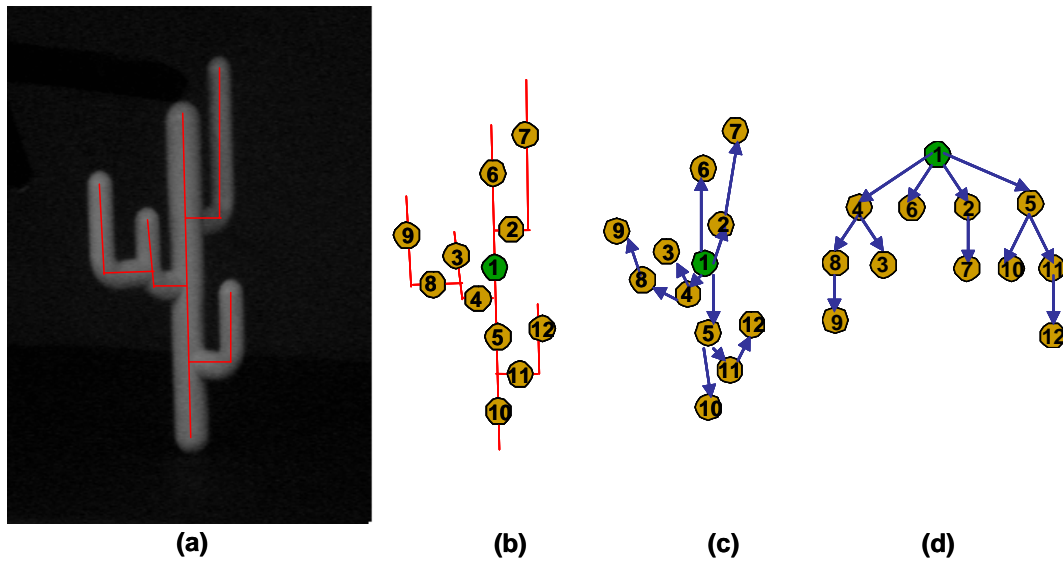


Figure 6.1: Construction du graphe à partir du squelette

et structurelles sont mémorisées :

- La signature topologique qui sera utilisée pour l'indexation. Elle correspond à une réalité structurelle de la composante de la forme codée par rapport à la forme globale. Elle est déterminée à l'aide de la configuration locale du graphe.
- Les coordonnées des extrémités des segments qui seront utilisées dans la phase localisation (2D pour le squelette 2D et 3D pour le squelette 3D).

La structure de données que nous avons choisie est de type redondant, c'est-à-dire, le même nœud peut apparaître plusieurs fois dans le fichier (le nombre d'apparitions = le nombre de ses descendants). Elle est pauvre en performance mémoire par contre les temps d'accès y sont rapides. Dans le cas où nous voulons chercher un voisin direct d'un nœud, nous pouvons accéder à l'information en temps constant. Et comme, dans notre application, nous avons des graphes de petite taille, la performance mémoire n'est pas une contrainte.

Par exemple, le graphe de la figure 6.1.d, sa structure de données aura la forme représentée au tableau 6.1.

La première colonne représente tous les nœuds du graphe qui ont au moins un descendant. Chaque nœud apparaît dans cette colonne autant de fois qu'il a de descendants directs. Chacun des nœuds de la deuxième colonne, représente le descendant direct de son correspondant de la première colonne.

Comme nous l'avons déjà mentionné, le nœud d'un graphe représente un ensemble de points connexes du squelette (2D ou 3D) que nous appelons segment. Par conséquent, un nœud possède deux points particuliers à savoir ses extrémités, il en existe deux types. Les extrémités qui sont des points de jonctions avec un autre segment du squelette et des points que nous appelons terminaux et qui représentent les limites du squelette.

La structure d'un nœud est alors représentée comme suit :

$$N = (\chi, P_1(x_1, y_1, z_1, \tau_1), P_2(x_2, y_2, z_2, \tau_2))$$

Tel que N est le nœud, χ est la signature topologique du nœud qui sera défini au paragraphe suivant, P_1 et P_2 représentent les extrémités du segment. Les composantes (x_1, y_1, z_1) et (x_2, y_2, z_2) sont les coordonnées des points P_1 et P_2 respectivement, et naturellement pour le graphe du squelette 2D, la composante z n'existe pas. Le paramètre τ indique si le point considéré est un point terminal, dans ce cas il prend la valeur de 1 sinon il est à 0.

1	4
1	6
1	2
1	5
2	7
4	3
4	8
8	9
5	10
5	11
11	12

Tableau 6.1: Structure d'un graphe

6.6.2 Caractérisation des squelettes par les valeurs propres

Pour décrire la topologie d'un arbre, on s'oriente vers le domaine des espaces propres des graphes. Nous rappelons que tout graphe peut être représenté par une matrice d'adjacence de $\{0, 1\}$, les valeurs 1 indiquent les nœuds adjacents dans le graphe (et 0 sur la diagonale). Les valeurs propres (VP) de la matrice d'adjacence d'un graphe (ou d'un arbre) stockent d'importantes propriétés structurelles du graphe (d'un arbre).

Pour exploiter les VP d'un arbre dans le but d'effectuer une mise en correspondance, Siddiqi a établi leurs stabilités par rapport à de légères perturbations topologiques dues essentiellement aux bruits, occlusions ou aux déformations. Il s'est basé sur le théorème suivant qui présente la relation entre les VP de deux arbres où l'un est un sous-arbre de l'autre.

Théorème 6.1 : *(Cvetkovic et al [CVE 97]) soit A une matrice symétrique avec comme valeurs propres $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$ et soit B une de ses principales sous-matrices. Si ses valeurs propres sont $v_1 \geq v_2 \geq \dots \geq v_m$, alors $\lambda_{n-m+i} \leq v_i \leq \lambda_i$ ($i = 1, \dots, m$).*

Cet important théorème, appelé théorème d'entrelacement, implique que s'il subsiste une légère différence entre A et B (dans le sens où l'un est un sous-arbre de l'autre) alors leurs valeurs propres sont légèrement différentes (proportionnellement).

Il a construit un descripteur à partir de la somme des VP pour chaque nœud plutôt que sur les VP de toute la matrice. De cette manière il obtient une signature dont la dimension n'est pas prohibitive et qui permet de coder l'ordre des nœuds dans l'arbre.

Explicitement, soit un arbre T de degré maximum $\Delta(T)$, et T_1, T_2, \dots, T_s les sous-arbres de sa racine. Pour chaque sous-arbre T_i , le degré de sa racine est $\delta(T_i)$. Pour déterminer cette signature, on calcule les VP de chaque matrice d'adjacence des sous-arbres T_i . Soit S_i la somme des $\delta(T_i)$ valeurs propres de T_i . Alors les éléments ordonnés S_i , deviennent les composants du vecteur χ de dimension $\Delta(T)$ appelé signature topologique et affecté à la racine de l'arbre figure 6.2. Si le nombre d'éléments S_i est inférieur à $\Delta(T)$, alors le vecteur est complété avec des 0. Cette procédure est répétée récursivement pour assigner un vecteur à chaque nœud de l'arbre.

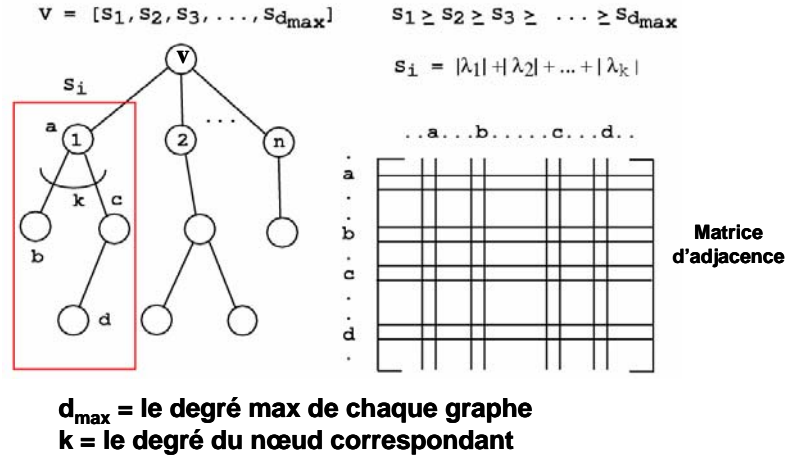


Figure 6.2: Calcul de la signature topologique

Pour bien illustrer ce que nous venons d'expliquer, nous allons l'appliquer pour déterminer la signature topologique du nœud 1 de la figure 6.1.d. Nous considérons le graphe T dont la racine est le nœud 1. Le degré maximum $\Delta(T) = 4$. T_1, T_2, T_3 et T_4 sont les sous graphes du graphe T , qui ont comme racine, respectivement, les nœuds 4, 6, 2 et 5. Les degrés respectifs de ces sous graphes sont $\delta(T_1) = 2$, $\delta(T_2) = 0$, $\delta(T_3) = 1$ et $\delta(T_4) = 2$.

Les matrices d'adjacences respectives de chaque sous graphe sont M_1 pour T_1 , M_2 pour T_2 , M_3 pour T_3 et M_4 pour T_4 telles que :

$$M_1 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}, M_2 = [0], M_3 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \text{ et } M_4 = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

Les valeurs propres de M_1 sont $VP_{M_1} = (-1.618 ; -0.618 ; 0.618 ; 1.618)$ dont la somme des valeurs absolues $S_1 = 4.472$, de la même manière $VP_{M_2} = (0)$ et $S_2 = 0$, $VP_{M_3} = (-1 ; 1)$ et $S_3 = 2$; et $VP_{M_4} = (-1.618 ; -0.618 ; 0.618 ; 1.618)$ et $S_4 = 4.472$. La signature topologique du nœud 1 est $[S_1, S_4, S_3, S_2]$, notée $\chi(1) = [4.472 ; 4.472 ; 2 ; 0]$.

6.6.3 Indexation du modèle dans la base de données

Pour la conception de notre algorithme de reconnaissance, nous commençons par une mesure de similarité globale pour indexer l'objet à reconnaître dans la base des modèles. Pour mesurer cette ressemblance nous calculons la distance

```

1. Indexation ( graphe2D  $G_1$ , {graphe3D  $G_2$ }) :
2.   | pour chaque nœud  $N$  de  $G_1$  faire
3.   |   |  $S_1 \leftarrow$  SignatureTopologique ( $N$ ) ;
4.   |   | pour chaque graphe  $G_2$  faire
5.   |   |   | pour chaque nœud  $M$  de  $G_2$  faire
6.   |   |   |   |  $S_2 \leftarrow$  SignatureTopologique ( $M$ ) ;
7.   |   |   |   |  $D \leftarrow$  DistanceEuclidienne ( $S_1, S_2$ ) ;
8.   |   | pour chaque couple ( $G_1, G_2$ ) faire
9.   |   |   | score  $\leftarrow$  DistanceMinimale (la racine  $R$  de  $G_1$ , un nœud  $M$  de  $G_2$ ) ;
10.  |   | {GraphesOrdonnés}  $\leftarrow$  OrdonnerGraphesOrdreCroissant (score) ;
11.  | retourner {GraphesOrdonnés} ;

```

Figure 6.3: Algorithme d'indexation

euclidienne entre les signatures topologiques de chaque nœuds du graphe 2D et de tous les nœuds du graphe 3D et ce, pour tous les graphes de la base (lignes 2 à 7 de la figure 6.3). Comme nous l'avons vu au paragraphe précédent, la signature topologique est une quantité facilement évaluable et suffisamment stable pour obtenir une indexation efficace et rapide. Un exemple de cette construction entre deux graphe, que nous appelons matrice d'appariement, est présentée au paragraphe suivant (tableaux 6.4 et 6.5).

Dans cette opération, nous allons attribuer un score, pour chaque couple (graphe 2D, graphe 3D), qui correspond à la distance minimale entre la racine du premier graphe et l'un des nœuds du second graphe (lignes 8 et 9 de la figure 6.3). Les hypothèses finales générées par cette procédure sont ordonnées par ordre croissant selon leur score et soumises à l'étape de vérification (ligne 10 de la figure 6.3). Notons que le couple dont le score est minimal représente l'hypothèse la plus probable pour que le modèle corresponde à l'objet de la scène.

Pour valider ou rejeter cette hypothèse, une analyse plus fine sera réalisée en amont de cette étape. Cette opération consiste à apparier le maximum de nœuds entre les deux graphes afin d'effectuer un recalage et ainsi de confirmer ou d'écarter le choix effectué. Cette phase est très importante car elle permet de gérer les problèmes cités précédemment. Elle est présentée en détail au paragraphe suivant.

6.6.4 Mise en correspondance un-à-un des nœuds

Comme cela a été mentionné au début de ce paragraphe, un isomorphisme large de sous-arbres ne peut pas exister entre le squelette de l'image et le squelette du

modèle, à cause des occlusions ou/et les bruits. La solution à ce problème consiste à trouver un appariement de cardinalité maximale et de poids minimum dans un graphe bipartite couvrant les nœuds des deux squelettes. Le graphe bipartite est un graphe où chaque arc est pondéré en fonction de la distance topologique (figure 6.6).

La formulation d'optimisation résultante est générale, c'est-à-dire qu'elle permet à chaque nœud d'un arbre de correspondre à n'importe quel nœud dans l'autre arbre. De cette façon il permet à des nœuds réels de correspondre à des nœuds bruit, cette formulation est donc faible puisque elle ne respecte pas l'ordre hiérarchique des nœuds.

Pour notre application la préservation de cet ordre est essentielle afin de donner un sens à la structure de la forme à retrouver. Pour résoudre ce problème, la solution était de contraindre la procédure d'appariement par graphe bipartite par un algorithme récursif qui cherchera la correspondance maximale des nœuds dans les sous-arbres.

Cet algorithme d'appariement de deux arbres G et H correspondant à deux squelettes a été proposé par [SID 98], inspiré de l'algorithme proposé par Reyner [REY 77]. L'algorithme retrouve récursivement des correspondances entre les nœuds des sous-arbres, il commence à partir de la racine de l'arbre, et procède vers le bas d'une manière descendante.

Contrairement aux traditionnelles méthodes de recherche en profondeur (depth-first) où le retour en arrière se fait statistiquement, cet algorithme recalcule efficacement les branches en chaque nœud, et choisit toujours la branche suivante pour descendre de la meilleure façon (best-first). Le grand avantage de cet algorithme est sa possibilité de mettre en correspondance deux arbres en présence de bruit (insertion et suppression de nœuds dans les sous-arbres).

Avant d'exposer cet algorithme quelques définitions sont présentées. Soit $G = (V_1, E_1)$ et $H = (V_2, E_2)$ les deux arbres à mettre en correspondance, avec $|V_1| = n_1$ et $|V_2| = n_2$. On définit d comme le degré maximum pour chacun des nœuds de G et de H , $d = \max(\delta(G), \delta(H))$.

Pour chaque nœud v , on définit $\chi(v) \in R^d$ comme l'unique vecteur propre appelé signature topologique introduit dans la section (6.6.2) et illustré sur la figure 6.2. Finalement, on définit $\Phi(G, H)$ l'ensemble final des nœuds mis en correspondance entre G et H et qui représente la solution de notre problème d'appariement (lignes 1 à 7 de la figure 6.4).

L'algorithme commence par former la matrice d'appariement $\Pi(G, H)$ de taille $n_1 \times n_2$, dont le (u, v) -ème élément a la valeur $\|\chi(u) - \chi(v)\|^2$ (ligne 3 de figure 6.5). Ensuite, on forme les arcs pondérés du graphe bipartite $\Gamma(V_1, V_2, E_T)$

```

1. Appariement-1-à-1 ( graphe G, graphe H ) :
2.    $\Phi \leftarrow \emptyset ;$ 
3.    $d \leftarrow \max (\delta (G), \delta (H)) ;$ 
4.   pour chaque nœud  $u$  de  $G$  et nœud  $v$  de  $H$  faire
5.     Calculer ( $\chi(u)$  et  $\chi(v)$ ) ;
6.   FormerAppariement (racine(G), racine(H)) ;
7.   retourner  $\Phi ;$ 

```

Figure 6.4: Algorithme d'appariement un-à-un des nœuds

```

1. FormerAppariement (u, v) :
2.   tant que le graphe  $G$  de racine  $u$  et le graphe  $H$  de racine  $v$  faire
3.     CalculerPoidsMatrice ( $\Pi(G_u, H_v) = |V_{G_u}| \times |V_{H_v}|$ ) ;
4.      $M \leftarrow$  CardinalitéMaximale( $\Pi(G_u, H_v)$ ) ;
5.      $(u_1, v_1) \leftarrow$  ScorePlusPetit ( $M$ ) ;
6.      $\Phi \leftarrow \Phi \cup \{(u_1, v_1)\} ;$ 
7.     FormerAppariement ( $u_1, v_1$ ) ;
8.      $G_u \leftarrow G_u \setminus \{x \mid x \text{ un nœud de } G_u \text{ et } (x, w) \text{ dans } \Phi \} ;$ 
9.      $H_v \leftarrow H_v \setminus \{y \mid y \text{ un nœud de } H_v \text{ et } (w, y) \text{ dans } \Phi \} ;$ 

```

Figure 6.5: Algorithme de recherche en profondeur

où les arcs sont pondérés à partir de la matrice $\Pi(G, H)$, on cherche la correspondance de cardinalité maximale et de poids minimum dans Γ (ligne 4 de la figure 6.5). Le résultat est un ensemble d'appariements de nœuds entre G et H appelé M_1 , qui peut être rangé dans un ordre décroissant de similarité (figure 6.7) (ligne 5 de la figure 6.5).

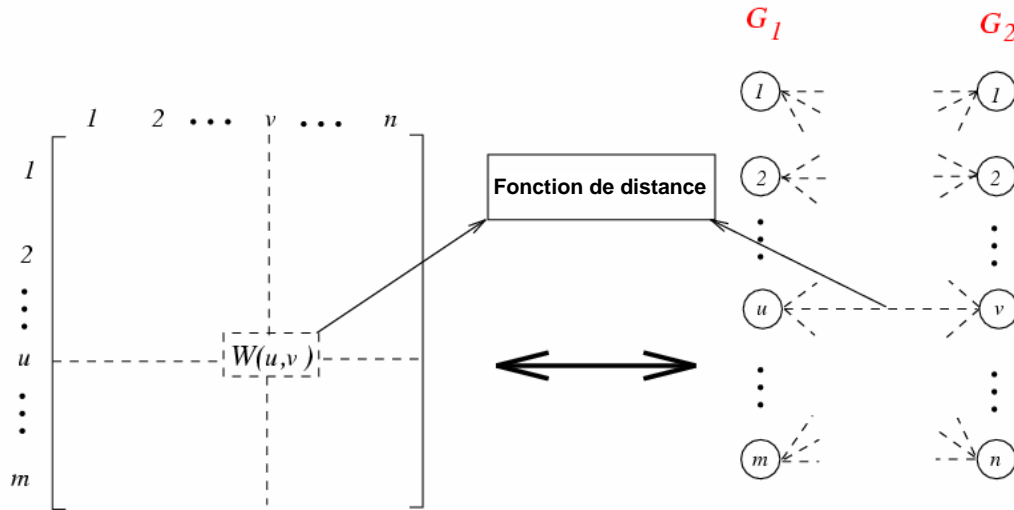


Figure 6.6: Graphe bipartite

A partir de M_1 , on choisit (u_1, v_1) comme la paire qui a le plus petit poids parmi toutes les paires dans M_1 . (u_1, v_1) est supprimé de la liste et il est rajouté à la solution $\Phi(G, H)$ (ligne 6 de la figure 6.5), et le reste de la liste est abandonné. A partir des sous-arbres G_{u_1} et H_{v_1} de G et H et de racines u_1 et v_1 respectivement, on forme la matrice $\Pi(G_{u_1}, H_{v_1})$ en utilisant la même procédure décrite précédemment. Une fois la matrice formée, on trouve une liste de correspondances M_2 dans le graphe bipartite défini à partir de la matrice pondérée $\Pi(G_{u_1}, H_{v_1})$, produisant une autre liste ordonnée de nœuds mis en correspondance. Cette procédure est appliquée récursivement à (u_2, v_2) (lignes 7 à 9 de la figure 6.5).

Ce processus récursif atteint finalement les feuilles de l'arbre, formant une liste de listes ordonnées de correspondances $\{M_1, \dots, M_k\}$. Dans l'étape i de retour-en-arrière, on enlève tous les sous-arbres des graphes G_i et H_i dont les racines participent dans une paire de $\Phi(G, H)$ (nous renforçons ainsi une mise en correspondance un-à-un des nœuds). Ensuite, comme dans les méthodes dites en profondeur-d'abord (depth-first), nous recalculons en premier M_i à partir des sous-arbres de racines u_i et v_i (sans prendre en compte les nœuds qui font déjà partie de la solution). Comme précédemment, on choisit une paire de nœuds de

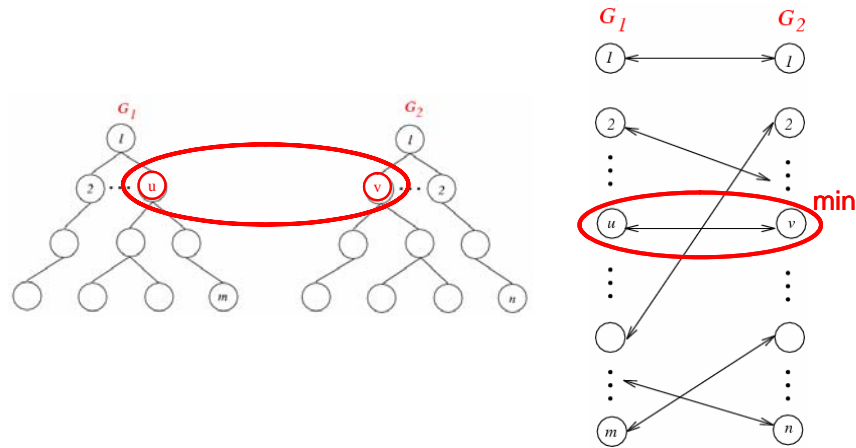


Figure 6.7: Noeuds de cardinalité maximale et de poids minimal

poids minimum, et on descend récursivement. Le traitement d'un nœud particulier se termine lorsque ces sous-arbres perdent tous leurs nœuds et se retrouvent dans l'ensemble des solutions $\Phi(G, H)$.

6.6.5 Problèmes et améliorations

Cependant subsistent un certain nombre de difficultés lors de l'application de la méthode de Siddiqi. Comme le vecteur des VP code les informations topologiques du graphe, quand nous sommes en présence d'un petit graphe, le vecteur assigné à chaque nœud ne conserve pas beaucoup d'informations. Le problème est plus évident pour les nœuds terminaux car les vecteurs de ces nœuds sont tous à 0. Un exemple est illustré sur la figure 6.8.

Les signatures topologiques pour chaque nœud des deux graphes sont représentées sur les tableaux 6.2 et 6.3.

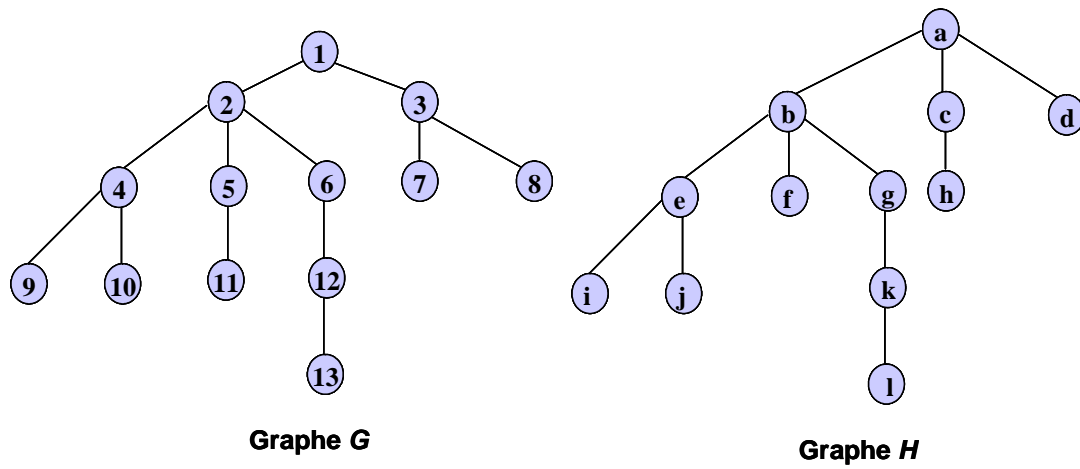


Figure 6.8: Exemple de mise en correspondance

Nœuds du graphe G	Signature topologique
1	[10.19; 2.82; 0]
2	[2.82; 2.82; 2]
3	[0; 0; 0]
4	[0; 0; 0]
5	[0; 0; 0]
6	[2; 0; 0]
7	[0; 0; 0]
8	[0; 0; 0]
9	[0; 0; 0]
10	[0; 0; 0]
11	[0; 0; 0]
12	[0; 0; 0]
13	[0; 0; 0]

Tableau 6.2: Signature topologique du graphe G

Nœuds du graphe H	Signature topologique
<i>a</i>	[8.64; 2; 0]
<i>b</i>	[2.82; 2.82; 0]
<i>c</i>	[0; 0; 0]
<i>d</i>	[0; 0; 0]
<i>e</i>	[0; 0; 0]
<i>f</i>	[0; 0; 0]
<i>g</i>	[2; 0; 0]
<i>h</i>	[0; 0; 0]
<i>i</i>	[0; 0; 0]
<i>j</i>	[0; 0; 0]
<i>k</i>	[0; 0; 0]
<i>l</i>	[0; 0; 0]

Tableau 6.3: Signature topologique du graphe H

$$\Pi(H, G) =$$

Nœuds	1	2	3	4	5	6	7
a	1.75	6.20	8.86	8.86	8.86	6.93	8.86
b	7.37	2	3.98	3.98	3.98	2.93	3.98
c	10.57	4.46	0	0	0	2	0
d	10.57	4.46	0	0	0	2	0
e	10.57	4.46	0	0	0	2	0
f	10.57	4.46	0	0	0	2	0
g	8.66	3.55	2	2	2	0	2
h	10.57	4.46	0	0	0	2	0
i	10.57	4.46	0	0	0	2	0
j	10.57	4.46	0	0	0	2	0
k	10.57	4.46	0	0	0	2	0
l	10.57	4.46	0	0	0	2	0

Tableau 6.4: Matrice d'appariement entre les graphes H et G

$$\Pi(H, G) =$$

Nœuds	8	9	10	11	12	13
a	8.86	8.86	8.86	8.86	8.86	8.86
b	3.98	3.98	3.98	3.98	3.98	3.98
c	0	0	0	0	0	0
d	0	0	0	0	0	0
e	0	0	0	0	0	0
f	0	0	0	0	0	0
g	2	2	2	2	2	2
h	0	0	0	0	0	0
i	0	0	0	0	0	0
j	0	0	0	0	0	0
k	0	0	0	0	0	0
l	0	0	0	0	0	0

Tableau 6.5: Matrice d'appariement entre les graphes H et G (suite)

Lorsque nous appliquons directement l'algorithme de Siddiqi sur ces deux graphes, nous trouvons dans la matrice de pondération $\Pi(G, H)$ du graphe bipartite plusieurs couples dont la distance est à 0. Le premier appariement trouvé est le couple $(6, g)$ car il est de poids minimal 0 et de cardinalité maximale 3, par une recherche en profondeur nous obtenons ensuite les couples $(12, k)$ et $(13, l)$.

Cependant dans l'étape de retour en arrière, nous obtenons plusieurs possibilités de couple à appairier. Exemple du nœud e du graphe H qui sera mis en correspondance soit avec le nœud 4, le 5, ou bien le 3 du graphe G car tous les vecteurs de ces nœuds ont la valeur $[0 ; 0 ; 0]$.

Or nous pouvons voir sur le schéma que cette solution n'est pas toujours la plus adaptée. En effet, il est préférable de sélectionner des grands groupes possédant de légères différences que des petits groupes de nœuds identiques. D'une part cela permet d'avoir des résultats plus cohérents, d'autre part cela limite le nombre de cas et rend donc plus simple le choix entre les différentes solutions possibles.

La solution souhaitée pour cet exemple est la mise en correspondance du nœud a du graphe H avec le nœud 1 du graphe G , hors cette possibilité ne peut pas être obtenue directement avec la méthode de Siddiqi. Le vecteur de a est $\chi(a) = [8.64 ; 2 ; 0]$ et celui de 1 est $\chi(1) = [10.19 ; 2.82 ; 0]$ par conséquent la distance calculée n'est pas minimale. La réponse que nous avons proposée à ce problème est sous la forme d'une pénalité qui favorise les grands groupes. Cette pénalité prend la forme :

$$- \alpha \times N_a \times N_b$$

où :

- α est un coefficient fixé de façon empirique
- N_a le nombre de nœuds du sous-graphe associé au nœud a
- N_b le nombre de nœuds du sous-graphe associé au nœud b

avec cette pénalité nous trouvons le résultat suivant :

$$(1 - a) (2 - b) (4 - e) (6 - g) (12 - k) (13 - l)$$

Par ailleurs nous avons également modifié l'algorithme afin d'obtenir plusieurs solutions au lieu d'une seule. En effet, là où l'algorithme de Siddiqi ne nous fournit qu'une solution considérée comme la meilleure, notre version modifiée génère tout un ensemble de solutions qui sont considérées comme équivalentes. A chaque fois que plusieurs couples (u, v) peuvent être choisis simultanément, pour chacun d'entre eux une nouvelle solution est générée en tenant compte de la solution partielle déjà calculée (la ligne 5 de la figure 6.5 devient : $\{(u,v)\} \leftarrow \text{ToutScorePlusPetit}(M_1)$)

Dans l'exemple précédent le nœud 3 du graphe G et les nœuds c et d du graphe H ont une signature topologique nulle. Les nœuds 3 et c sont appariés puisqu'ils possèdent le plus de descendants. Il faut maintenant choisir entre associer le nœud h au nœud 7 ou au nœud 8. Là où avec la méthode de Siddiqi une seule solution est choisie parmi les deux, notre méthode crée deux solutions indépendantes en conservant l'appariement $3 - c$ qui avait déjà été choisi.

De là un même appariement peut se retrouver dans un grand nombre de solutions puisque les solutions générées conservent les appariements préalablement choisis. Nous avons donc choisi de nous en servir comme un indice de cohérence pour nos appariements, un appariement choisi de nombreuses fois apparaissant comme bien plus probable qu'un autre choisi une seule fois. Au final, pour l'exemple précédent nous obtenons le résultat suivant :

Couples	$(1, a)$	$(2, b)$	$(4, e)$	$(6, g)$	$(12, k)$	$(13, l)$	$(5, f)$	$(11, f)$
indice	100%	100%	100%	100%	100%	100%	50%	50%

Couples	$(9, i)$	$(9, j)$	$(10, i)$	$(10, j)$	$(3, c)$	$(7, h)$	$(8, h)$
indice	50%	50%	50%	50%	100%	50%	50%

6.7 Conclusion

Dans ce chapitre nous avons exposé les principaux algorithmes d'isomorphisme de graphes. Nous avons aussi présenté leurs applications dans le domaine de la vision et de la reconnaissance de forme.

Pour résoudre notre problème de mise en correspondance de squelettes, nous avons opté pour une méthode de graphe bipartite que nous avons modifiée et adaptée à notre approche. La mesure de similarité utilisée est dérivée de la représentation en valeurs propres des graphes. Pour chaque nœuds correspond un vecteur nommé signature topologique. Ce dernier est utilisé premièrement pour l'indexation du squelette objet dans la base de données qui correspond à la phase de reconnaissance. En deuxième lieu, il est utilisé pour une mise en correspondance un-à-un des nœuds, ce qui nous permet d'appliquer un algorithme de localisation. Cette méthode possède la faculté de gérer le rajout et la suppression de nœuds. Par conséquent notre système de vision pourra détecter et reconnaître des objets de formes libres dans des scènes réelles en présence de bruits et d'occlusions.

Dans le chapitre suivant, nous proposons une étude expérimentale permettant d'apprécier l'efficacité de notre système de reconnaissance et d'en déduire les limitations.

Chapitre 7

RÉSULTATS ET APPLICATIONS

7.1 Introduction

Dans le chapitre précédent, nous avons vu que dans la phase d'appariement plusieurs hypothèses peuvent être générées à cause de la présence des différents bruits.

Afin de vérifier la validité d'une hypothèse, nous allons estimer la transformation rigide entre le modèle reconnu (supposé) et le repère de l'affichage. La projection du modèle sur l'image nous permet alors de valider ou de rejeter cette hypothèse.

Dans la première partie de ce chapitre, nous commencerons par présenter la méthode de vérification des hypothèses utilisé. Dans la seconde partie, nous étudierons les performances de notre méthode de reconnaissance appliquée sur des scènes réelles et dans des conditions diverses. Finalement, nous montrerons l'adaptation de notre concept au suivi d'objets de formes libre.

7.2 Vérification des hypothèses

Nous commençons par vérifier la première hypothèse générée dans l'étape d'indexation. Par la suite, on estime la pose à partir des différents appariements générés dans la phase de mise en correspondance un-à-un des nœuds (lignes 2 à 4 de la figure 7.1, Algo de la figure 6.3). Chaque couple de nœuds que nous avons apparié correspond à un couple de segments, l'un appartenant au squelette 2D, l'autre au squelette 3D curviligne. Cependant pour la localisation, seuls les points terminaux du squelette s'avèrent être utilisables, les autres points du squelette 2D pouvant connaître d'importantes variations. Or un nœud terminal du graphe correspond toujours à un segment comprenant un point terminal du squelette par

```

1. VérificationHypothèse (graphe2D G1, hypothèse h) :
2.   |  {{{(u, v)}}} ← Appariement-1-à-1 (G1, h) ;
3.   |  pour chaque {(u, v)} de {{{(u, v)}}} faire
4.   |  |  EstimationPose ;
5.   |  |  erreur ← CalculErreurProjection ;
6.   |  |  si (erreur < seuil) alors
7.   |  |  |  retourner h ;
8.   |  retourner nul ;

```

Figure 7.1: Algorithme de vérification des hypothèses

construction. Il est donc aisé de passer de nos couples de nœud à des couples comprenant une extrémité du squelette 2D et une extrémité du squelette 3D.

Afin de vérifier la validité d'une hypothèse, on projette les primitives du modèle 3D de l'objet sur le plan image. Si les primitives projetées ne coïncident pas, selon un seuil de tolérance fixé, avec les primitives images qui leur correspondent alors cette hypothèse est rejetée (lignes 5 à 8 de la figure 7.1) et on passe alors à l'hypothèse suivante et ainsi de suite. C'est-à-dire, si les points du squelette 3D projetés sur l'image, ne coïncident pas avec les points du squelette 2D, alors la reconnaissance n'est pas valide.

Pour estimer la transformation rigide nécessaire pour le recalage, nous utilisons les nœuds (points) appariés dont l'indice de cohérence est élevé (voir paragraphe 6.6.4). Si l'erreur de projection n'est pas satisfaisante, nous passons alors aux autres points et nous sélectionnons la transformation qui présente une erreur minimale. Ce module de vérification est basé sur la mesure de cette erreur (7.1) qui est la distance dans l'image (en pixel) entre, à chaque fois, un point du squelette 2D et un point du squelette 3D projeté dans l'image.

$$erreur = \sqrt{\frac{1}{N} \sum_{i=1}^N ((\hat{u}_i - u_i)^2 + (\hat{v}_i - v_i)^2)} \quad (7.1)$$

où : N est le nombre de points d'appariement,
 (u_i, v_i) les coordonnées d'un point image,
 (\hat{u}_i, \hat{v}_i) estimés des coordonnées du point de l'objet après recalage et projection sur le plan image.

Le formalisme mathématique du recalage 3D appliqué dans la vérification des hypothèses est décrit dans l'annexe D.

7.3 Evaluation des performances de la reconnaissance

La validation de notre système de reconnaissance nécessite de tester son efficacité et sa robustesse par sa mise en oeuvre dans différentes situations. Celle-ci inclut une étude du comportement de la méthode dans les cas d'occlusions, d'auto-occlusions, ...

L'évaluation que nous proposons a été réalisée dans des conditions réelles pour donner plus de crédibilité à notre méthode. Dans ces conditions, nous allons essayer de reconnaître un objet "Cactus" dans une base de données. Pour des raisons didactiques, nous avons choisi cet objet car on retrouve une analogie trivial entre sa forme et le graphe de son squelette. Nous commençons par décrire les conditions expérimentales et ensuite nous présentons les différentes situations traitées pour valider notre système de reconnaissance.

7.3.1 Conditions expérimentales

Pour implémenter nos algorithmes, nous avons utilisé, Matlab pour la partie on-line. Pour la squelettisation 3D (la partie off-line), nous avons utilisé C++ et SolidWorks. Ces développements ont été effectués sur un PC possédant un processeur Pentium III de 833 Mhz et de 512 Mo de RAM. Afin d'acquérir les images de la scène, nous avons utilisé une caméra CCD de type Sony avec une focale de 16 mm, ainsi qu'une carte d'acquisition de type Meteor II. Les images obtenues sont monochromes de résolution $768 \times 576 \text{pixel}^2$.

Pour modéliser l'environnement, nous avons calibré la caméra manuellement avec une mire de 128 points. Nous avons utilisé la méthode de Faugéras Toscani [FAU 93] basée sur le modèle sténopé de la caméra.

L'objet "Cactus" de taille $\simeq 170 \times 80 \times 40$ mm est placé sur un banc à une distance d'environ 1 m de la caméra.

7.3.2 Différentes situations étudiées

Nous disposons d'un seul objet à reconnaître dans un ensemble de modèles 3D, néanmoins il sera confronté à plusieurs configurations.

La base de données est constituée de 10 modèles de formes libres et de complexités variables, l'ensemble de cette base de données est représenté sur la figure 7.18. Pour générer ces modèles, nous avons utilisé un logiciel de modélisation de pièces mécaniques "SolidWoks". A chaque modèle est associé un squelette 3D et un graphe lui correspondant. Dans chaque nœud du graphe, nous stockons sa

signature topologique ainsi que les coordonnées 3D des extrémités des segments du squelette.

Dans les paragraphes suivants nous traiterons et nous étudierons les différentes situations de reconnaissances que nous trouvons sur des scènes réelles.

7.3.2.1 Objet unique -cas idéal-

Dans un premier temps, pour bien expliquer notre méthode de reconnaissance, nous allons détailler son comportement sur un cas idéal, représenté dans la figure 7.2. Dans cet exemple, l'objet est seul et tous ses éléments sont présents dans l'image.

La figure 7.2 montre l'objet cactus vu par la caméra, l'opération de squelettisation 2D donne le squelette représenté sur la figure 7.3. Sur cette figure, les croix rouges représentent le squelette réel et les segments en vert indiquent les nœuds après l'opération de transformation du squelette en graphe. Pour cet exemple, le graphe 2D résultant de l'image (figure 7.4) est identique au graphe 3D, car il n'y a ni suppression ni rajout de nœuds.

Les scores obtenus lors de la procédure d'indexation sont représentés sur le tableau 7.1. Ils sont ordonnés par ordre croissant ce qui correspond à un rangement par ordre décroissant de similarité.

Modèles	Cactus 1	Cactus 2	Cactus 3	Avion	Chromo
Scores	0	0	2,472	3.2734	3.4959
Modèles	Poisson	Pantin	Chaise	Tortue	Coque
Scores	3.7535	4.0276	4.4723	5.8401	6.6331

Tableau 7.1: Table d'indexation de l'objet

Nous obtenons deux scores identiques qui sont à 0, pour le cactus 1 et le cactus 2, ce qui est normal car les deux sont topologiquement équivalents (voir figure 7.18). L'étape sous-jacente de vérification va permettre de sélectionner l'un des deux modèles (Algo de la figure 7.1).

Le premier modèle sur la liste est introduit à l'entrée de l'algorithme d'appariement un-à-un des nœuds (ligne 2 de la figure 7.1, Algo de la figure 6.4). Cet algorithme nous fournit deux configurations de correspondances représentés sur les tableaux 7.2 et 7.3.

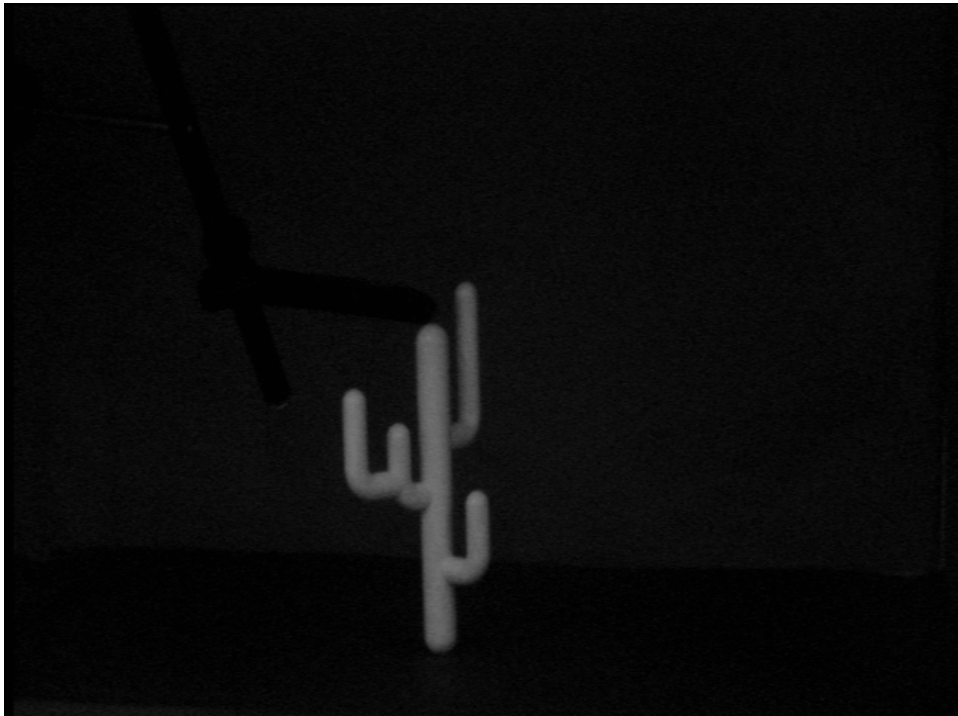


Figure 7.2: Objet Cactus

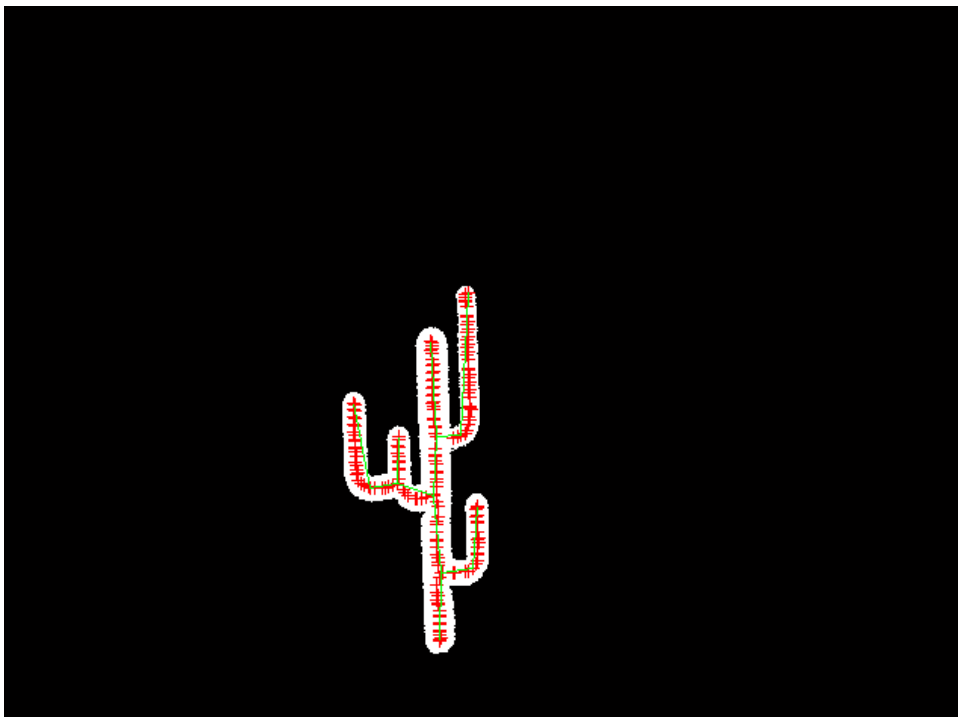


Figure 7.3: Squelette 2D de l'objet Cactus

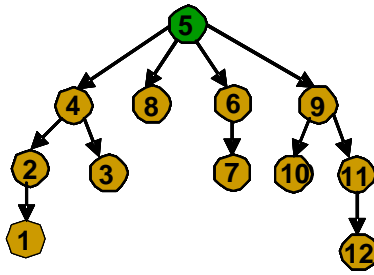


Figure 7.4: Graphe du squelette 2D

Nœuds du graphe 3D	1	2	3	4	5	6
Nœuds du graphe 2D	5	9	8	6	4	11
Nœuds du graphe 3D	7	8	9	10	11	12
Nœuds du graphe 2D	10	7	3	2	12	1

Tableau 7.2: Premier ensemble d'appariements

Nœuds du graphe 3D	1	2	3	4	5	6
Nœuds du graphe 2D	5	4	8	6	9	2
Nœuds du graphe 3D	7	8	9	10	11	12
Nœuds du graphe 2D	3	7	10	11	1	12

Tableau 7.3: Second ensemble d'appariements

Le premier ensemble d'appariement nous donne une erreur pixelique après projection du modèle de 161.76 pixels (figure 7.5), qui est largement supérieure au seuil que nous avons fixé à 10 pixels. Avec le second groupe d'appariement l'erreur obtenu est de 4.3067 pixels. Nous voyons clairement sur la figure 7.6 que la projection du modèle sur l'image est presque parfaite. De là, nous pouvons déduire que l'objet présent dans la scène est bien le cactus 1.

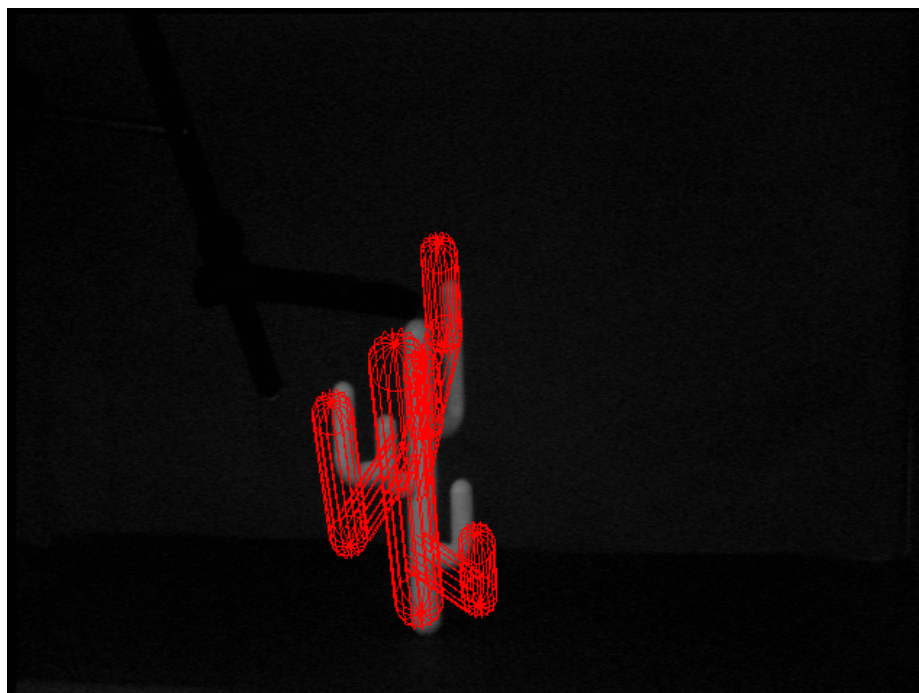


Figure 7.5: Projection du modèle à partir du premier ensemble d'appariements

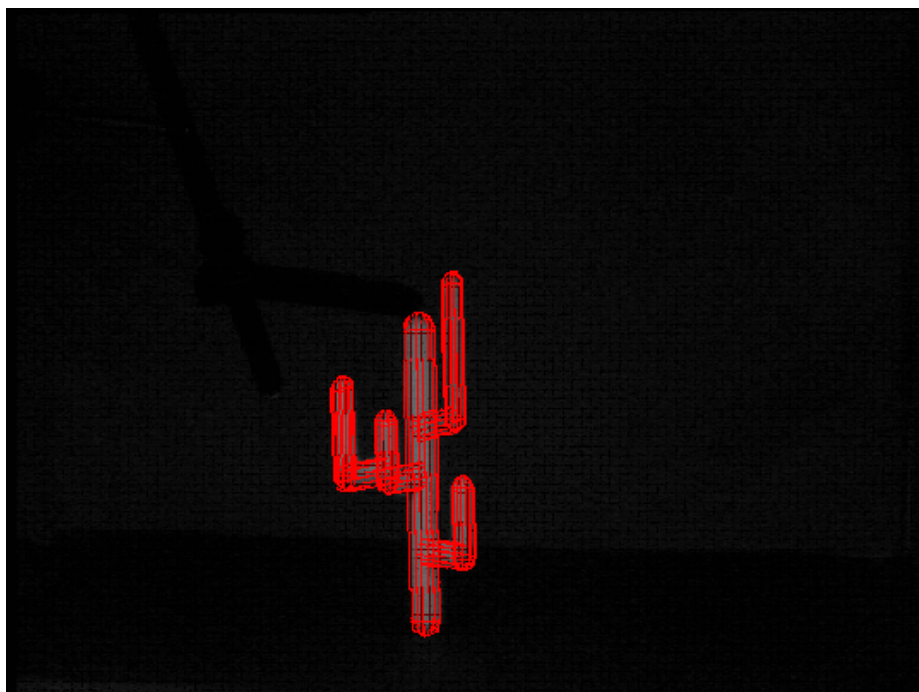


Figure 7.6: Projection du modèle à partir du second ensemble d'appariements

Il faut remarquer que la première hypothèse, fournie lors de l'indexation, est aléatoire, c'est-à-dire on aurait pu avoir le cactus 2 comme première hypothèse. Dans cette éventualité, la méthode de projection-vérification est appliquée, aux hypothèses une-à-une, sur les différents ensembles d'appariement. Les erreurs de projection trouvées sont alors largement supérieures à ceux du cactus 1, par conséquent, le cactus 2 sera écarté de la liste, et on passera alors au cactus 1 dont le résultat est présenté ci-dessus. A titre indicatif, nous trouvons deux ensembles d'appariements entre le squelette de l'image et le squelette du cactus 2. L'erreur obtenue pour le premier groupe est de 187, 21 pixels et pour le second groupe est de 476, 37 pixels.

7.3.2.2 Objet unique avec auto-occlusions

Nous avons détaillé ci-dessus notre méthode de reconnaissance. Dans ce qui suit, nous allons présenter son comportement selon différents aspects et principalement avec des auto-occlusions.

La première série d'images (7.7, 7.8, 7.9) représente quelques résultats en fonction de la position de l'objet dans l'espace. Sur chaque ligne nous présentons respectivement de la gauche vers la droite, l'objet réel, son squelette 2D et finalement le modèle 3D, reconnu par notre algorithme et superposé sur l'image réelle.

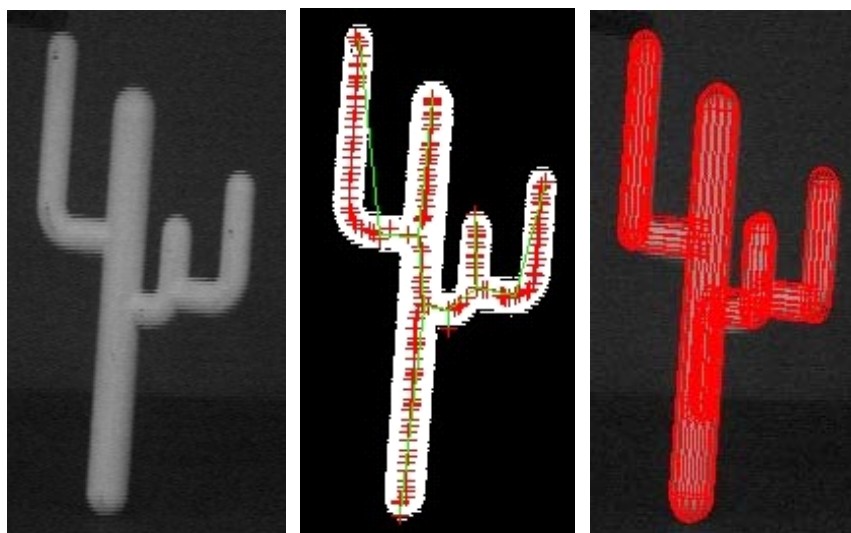


Figure 7.7: Résultat de la reconnaissance sur l'aspect -1-

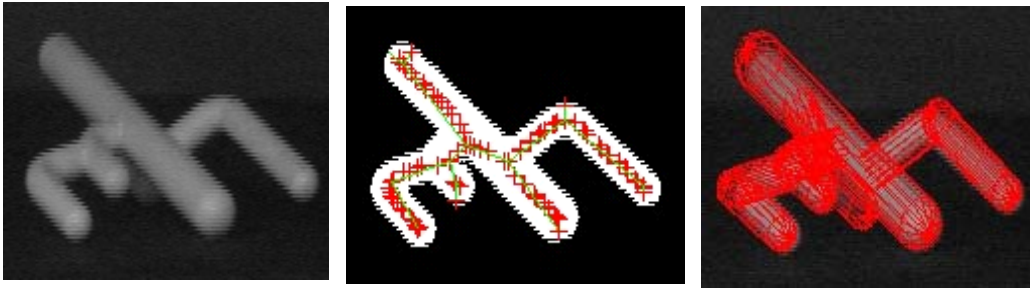


Figure 7.9: Résultat de la reconnaissance sur l'aspect -3-

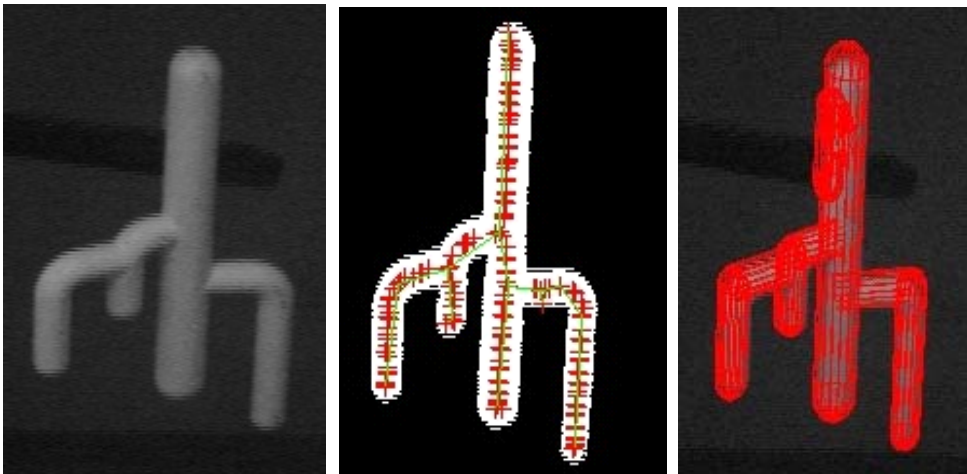


Figure 7.8: Résultat de la reconnaissance sur l'aspect -2-

Dans la phase d'indexation, ces trois aspects donnent le même résultat, il est représenté sur le tableau 7.4. La première hypothèse qui ressort est le cactus 3, cependant elle est rapidement écartée dans l'étape de vérification car l'erreur minimum lors de la projection est de 89,785 pixels. Nous rappelons que le seuil que nous avons choisi est de 10 pixels. Quand on passe à la deuxième hypothèse qui représente le cactus 1, nous obtenons des erreurs acceptables (voir tableau 7.5) pour chacun des aspects. Par conséquent, nous pouvons prétendre avoir reconnu l'objet présent dans la scène pour ces différents points de vue.

Modèles	Cactus 3	Cactus 1	Cactus 2	Chromo	Pantin
Scores	2	3.179	3.179	3.179	3.756
Modèles	Chaise	Avion	Poisson	Coque	Tortue
Scores	4.229	4.376	4.402	4.898	7.356

Tableau 7.4: Table d'indexation de l'objet pour les aspects 1, 2 et 3

N° aspect	aspect -1-	aspect -2-	aspect -3-
Erreur de projection	5.439	4.373	7.138

Tableau 7.5: Erreurs de projections

Nous avons testé notre méthode sur plusieurs aspects, nous remarquons que pour la plupart des images, l'algorithme reconnaît bien l'objet. Cependant il subsiste des vues (cas extrêmes) où notre système de reconnaissance ne fonctionne pas, un exemple est illustré sur la figure 7.10.

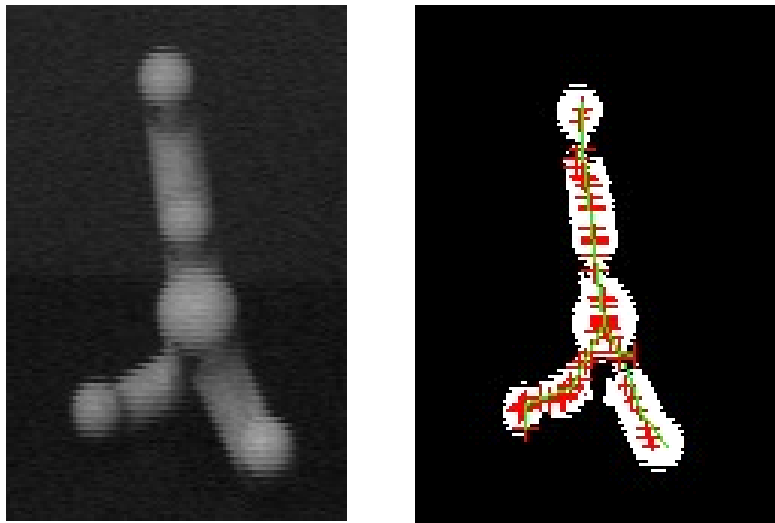


Figure 7.10: Résultat de la reconnaissance sur l'aspect -4- : cas dégénéré

Cette imperfection qui représente le principal défaut de notre méthode, est due à la nature même du squelette. En effet, il existe des images pour lesquels le squelette est dégénéré, par exemple, le squelette d'un disque ou d'une boule est un point centré sur cette forme. Nous pouvons citer aussi un autre exemple de ce type de problème, le cas du squelette d'une bouteille vue d'en haut, qui est aussi un point.

Pour revenir à notre objet cactus de la figure 7.10, le graphe correspondant à ce squelette est constitué de 3 nœuds (une racine et deux descendants). Le résultat de l'étape d'indexation est représenté sur le tableau 7.6, nous voyons clairement que ce premier classement est très loin de la réalité et nous pouvons même pas effectuer de vérifications car cette phase nécessite au minimum quatre points non coplanaires.

Modèles	Coque	Chromo	Pantin	Cactus 3	Poisson
Scores	0	3.464	4	5.291	5.656
Modèles	Avion	Cactus 1	Cactus 2	Chaise	Tortue
Scores	6.324	6.633	6.633	7.974	9.38

Tableau 7.6: Table d'indexation de l'objet pour l'aspect 4

7.3.2.3 Objet unique avec occlusions

La deuxième série de tests concerne la robustesse aux occlusions. Nous avons remarqué qu'à cause de l'occlusion, nous rencontrons deux types de problèmes. Le premier type concerne la suppression des nœuds, ce problème se présente lorsque l'objet parasite et notre objet de référence sont fortement contrastés (figure 7.11). Le deuxième type de problèmes est le rajout des nœuds, et ce, lorsque l'objet à reconnaître est caché avec un autre objet de même texture et de même luminance que le premier (figure 7.12 et figure 7.13).

Pour le premier type d'occlusions, notre système est robuste car il arrive à apparier facilement le graphe 2D avec un sous-graphe graphe 3D. Il suffit d'avoir quatre bons appariements non coplanaires pour effectuer la validation, dans l'exemple de la figure 7.11, nous en avons 5.

Pour le deuxième type d'occlusion, lorsque le rajout des nœuds se fait au niveau des feuilles du graphe sans bruit (figure 7.12), notre méthode fonctionne parfaitement. Presque comme pour le premier cas, il arrive à mettre en correspondance le graphe 3D avec un sous graphe 2D.

Par contre lorsque le rajout des nœuds change la topologie du graphe 2D, la reconnaissance échoue (figure 7.13). Ce changement de topologie arrive lorsque un nœud bruit est pris comme racine dans la phase de transformation en graphe. Ceci constitue la seconde limitation de notre système de reconnaissance.

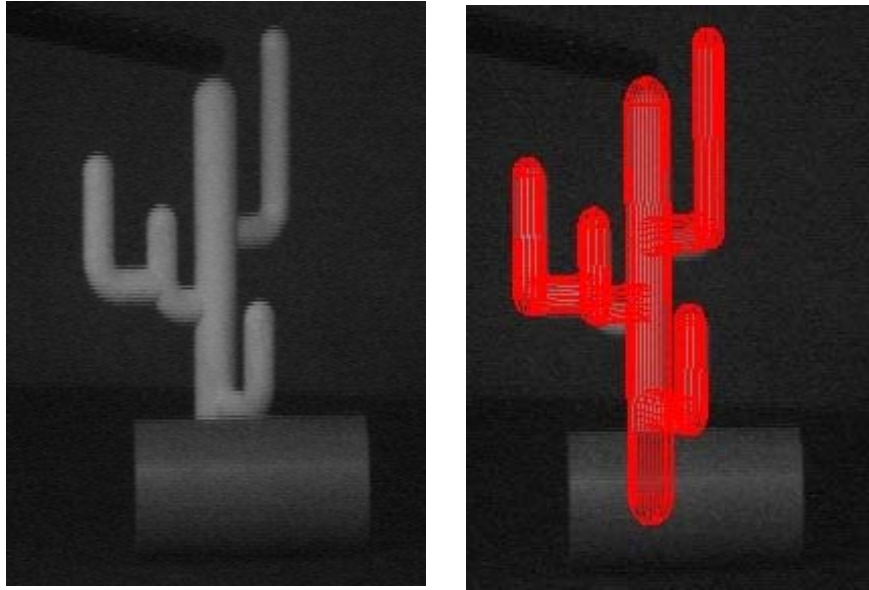


Figure 7.11: Reconnaissance dans le cas d'occlusions -1-

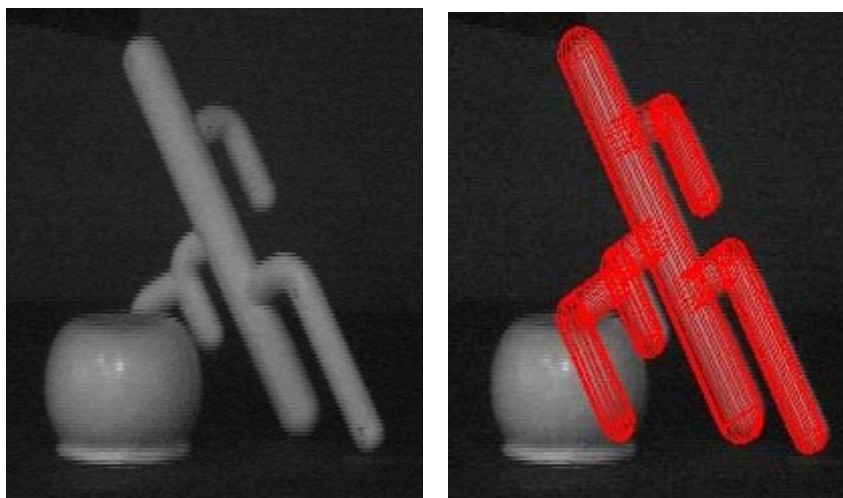


Figure 7.12: Reconnaissance dans le cas d'occlusions -2-

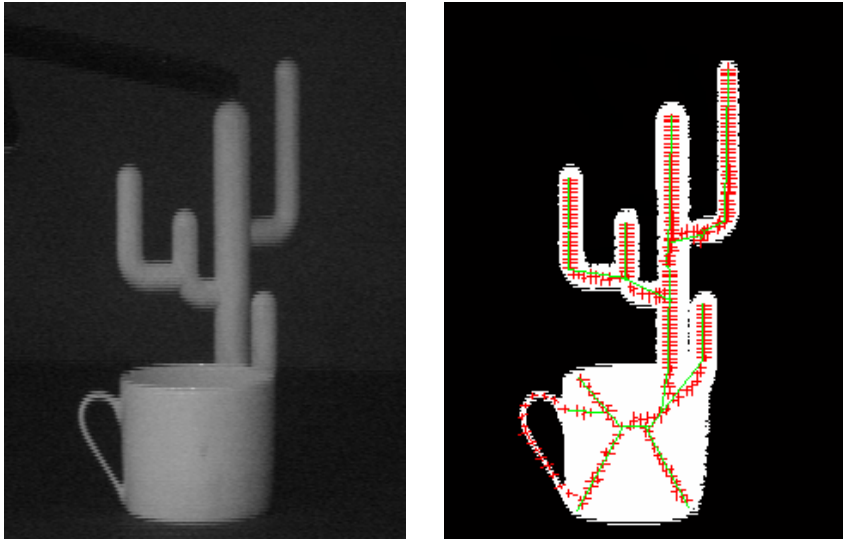


Figure 7.13: Reconnaissance dans le cas d'occlusions -3-

7.3.2.4 Plusieurs objets dans la scène

En toute rigueur, la reconnaissance d'un objet doit s'appuyer sur un ensemble de modèles 3D et la présence de plusieurs objets dans l'image.

Nous avons exploré la situation où nous disposons du modèle 3D de la cible à reconnaître parmi plusieurs objets présents dans l'image. Un exemple est illustré dans la figure 7.14, le cactus est posé à côté de plusieurs objets de formes différentes. Dans cet exemple, la partie indexation de notre méthode distingue facilement l'objet "cactus" des autres objets. Les squelettes 2D correspondants aux autres objets sont très différents des squelettes 3D de la base de données. La superposition n'est pas parfaite, on remarque un léger décalage du modèle vers le haut. La reconnaissance est tout de même validée car l'erreur de superposition est de 8,731 pixels.

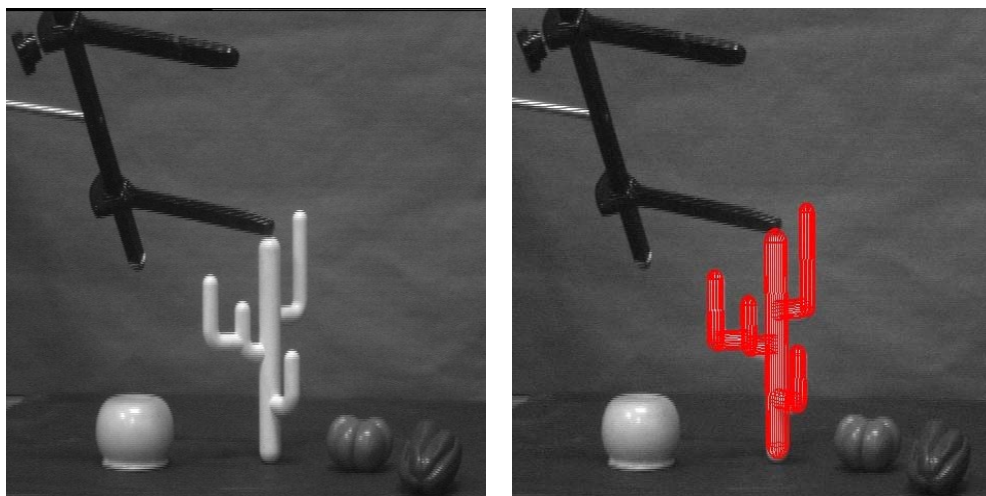


Figure 7.14: Reconnaissance dans une scène à plusieurs objets

7.4 Application au suivi d'objets de forme libre

Vu les résultats intéressants obtenus par le formalisme des squelettes pour la reconnaissance de formes, nous avons exploré son application au suivi d'objets.

Le suivi d'objets est un aspect important de la vision artificielle. Cela reste un problème difficile qui a une grande place dans des tâches comme la mesure du mouvement ou de l'asservissement visuel et ce, pour des applications allant de l'imagerie médicale à la sécurité en passant par l'analyse du trafic routier.

7.4.1 Principe du suivi

La plupart des algorithmes permettant une estimation de la pose 3D à partir d'une séquence d'images 2D sont basés sur le même principe, illustré par la figure 7.15.

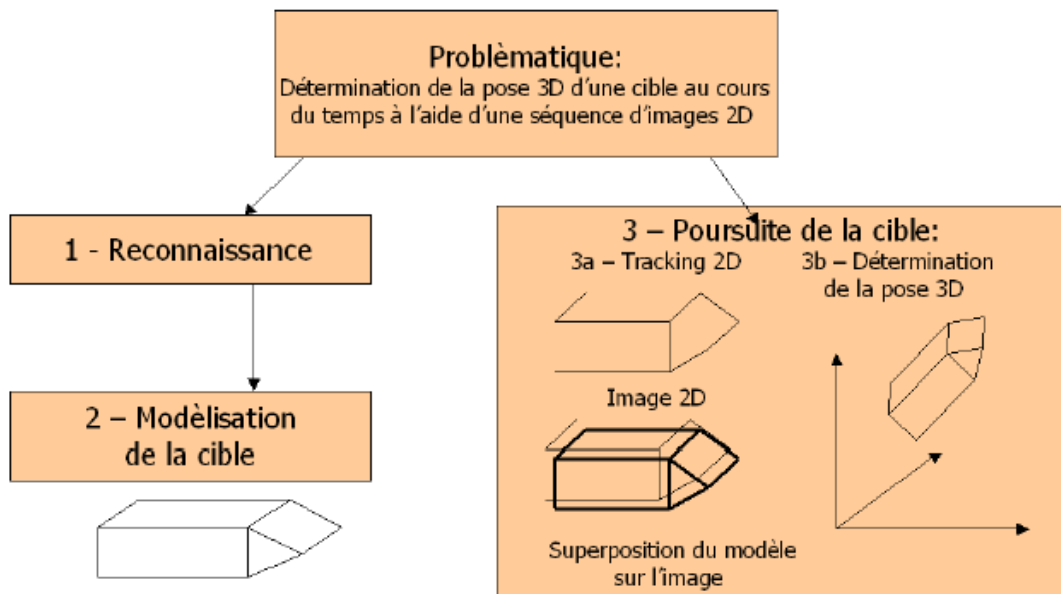


Figure 7.15: Principe général du suivi

Une première phase consiste donc à reconnaître le type de cible pour choisir dans la base de données le modèle adapté (initialisation). La seconde phase, qui est elle réalisée au cours du traitement, consiste à effectuer la poursuite de la cible (suivi). Une pose initiale de l'objet est connue soit par un processus d'initialisation au début de la poursuite, soit par un processus de prédiction utilisant les résultats obtenus dans l'image précédente. Cette pose initiale permet alors de projeter le modèle de la cible dans le plan image. A partir de cette projection il est possible de savoir si la pose est bonne ou non. Une mesure de comparaison entre l'image 2D et le modèle projeté est alors utilisée pour effectuer un ajustement de la pose.

Le principe général du suivi est commun à toutes les méthodes, cependant elles varient très largement en fonction des techniques d'appariements ou des types de prédiction utilisés. Parmi les différentes méthodes de suivi existantes, nous citerons celle de Lowe [LOW 92] [LOW 87] de Gennery [GEN 82] [GEN 92] et de Harris [HAR 90] qu'on peut qualifier de génériques.

Ces trois méthodes de suivi offrent de très bonnes performances, avec des algorithmes de complexités différentes. Il est important de remarquer que malgré le fait que ces algorithmes datent tous de plus de 10 ans, ils restent encore utilisés à l'heure actuelle, ce qui démontre bien leur efficacité. Cependant ces méthodes sont développées pour les objets de forme polyédrique. Ceux-ci possèdent des arêtes

saillantes qui facilitent grandement l'isolation des points d'intérêt sur l'image. De ce fait les méthodes que nous venons de citer ici, ne sont pas applicables directement au suivi d'objets de forme libre.

De même que pour la reconnaissance, nous avons utilisé le concept de squelettes, afin de pouvoir exploiter les avantages des méthodes cités ci-dessus. Plus précisément, nous avons adapté la méthode de Lowe pour faire le suivi du squelette 2D à l'aide du squelette 3D et par conséquent de suivre un objet 3D dans une séquence d'images 2D. Les points d'intérêts utilisés sont les extrémités des deux squelettes.

7.4.2 Suivi d'objets 3D par squelettes

L'étape d'initialisation correspond au calcul de la pose de la cible sans connaissance a priori de la scène. Pour nous, elle est effectuée par le système décrit au chapitre (4). Dès lors que la position initiale de notre objet est connue, il est possible de débiter le suivi.

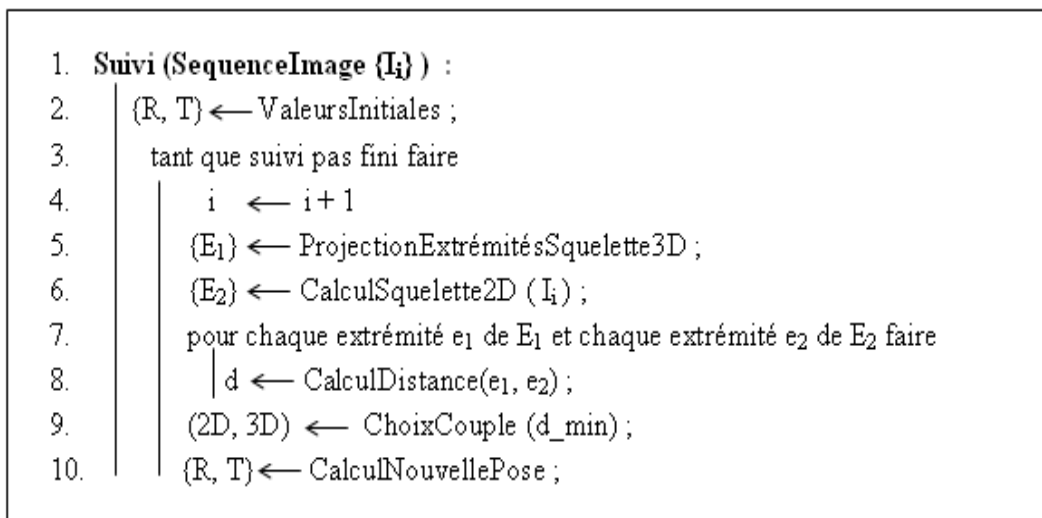


Figure 7.16: Algorithme de suivi par les squelettes

La position originale utilisée est celle qui a été calculée sur la première image durant la phase d'initialisation (ligne 2 de la figure 7.16). Le squelette 3D est alors projeté sur le plan image, à partir de cette position, grâce aux matrices de rotation et de translation calculées précédemment (ligne 5 de la figure 7.16). L'algorithme se base sur le fait qu'entre deux images de la vidéo, les déplacements sont très faibles. Ainsi les positions des points du squelette 2D seront très proches d'une image à l'autre.

Notons p_{ij} les positions des extrémités j du squelette 2D créé à l'image i , numérotées de 1 à n_i (leur nombre pouvant varier en fonction de l'image) et q_k les positions des projections sur le plan image à l'aide des matrices de rotation et de translation, respectivement R et T , des extrémités du squelette 3D.

L'initialisation est supposée correctement faite, donc lors de la projection de notre squelette 3D, les points q_k qui ont servi au calcul de la position, se trouvent à proximité des points p_{ij} qui leurs avaient été appariés. A l'image suivante (numérotée $i + 1$), les points $p_{(i+1)j}$ sont donc différents des points p_{ij} , toutefois, le déplacement de l'objet étant faible, la position du squelette 2D varie peu. Ainsi les points q_k qui par construction même étaient très proches de certains points p_{ij} , se retrouvent de fait proches de certains points $p_{(i+1)j}$. Le déplacement de notre objet étant très faible, l'ancien et le nouveau squelette 2D sont très proches l'un de l'autre par conséquent leurs extrémités le sont aussi.

En mesurant les écarts en pixels qui séparent les points $p_{(i+1)j}$ et les points q_k , et en les classant par ordre croissant, nous pouvons former un certain nombre de couples $(p_{(i+1)j}, q_k)$. Ainsi, alors que durant l'initialisation, l'appariement entre les extrémités des squelettes 2D et 3D nécessitait de lourds calculs du fait que la position était inconnue, il est ici quasiment immédiat en considérant les points 3D à l'origine des q_k (lignes 6 à 9 de la figure 7.16)

L'algorithme d'itération orthogonale que nous avons précédemment utilisé lors de l'initialisation, est appliqué sur les couples $(Point2D, Point3D)$, les dits points 3D étant obtenus en remplaçant les q_k par les points originels du squelette 3D. Cela nous fournit une nouvelle valeur pour R et T , laquelle pourra être utilisée pour effectuer à nouveau des calculs sur l'image suivante (ligne 10 de la figure 7.16).

7.4.3 Résultats du suivi par les squelettes

Le suivi a été développé en C++ du fait de l'importance de pouvoir réaliser les calculs en temps réel. Le passage de la phase d'initialisation, réalisée sous Matlab, au suivi se fait via un fichier spécifique. Ce fichier contient les matrices de rotation et de translation correspondant à la position calculée, ainsi que le modèle du squelette 3D.

Afin de valider nos hypothèses, les tests ont été réalisés sur des images de synthèse. Car nous n'avons pas eu le temps de développer les outils nécessaires pour effectuer les prétraitements sur les images réelles.

Les vidéos ont été créées à l'aide des logiciels POV-Ray en ce qui concerne la création d'images et le logiciel Virtual Dub qui nous a permis de réaliser le montage.

En ce qui concerne la lecture de vidéo, nous utilisons la librairie OpenCV développée par Intel. Son principal avantage est d'intégrer également un certain nombre de fonctions spécialisées dans le traitement d'images.

La squelettisation est au cœur de notre application, puisque c'est au cours de cette opération que seront extraits les points qui nous serviront à effectuer le suivi. Contrairement à la partie initialisation, où nous avons mis l'accent sur l'extraction d'un squelette robuste par une méthode hybride, pour la partie suivi, nous avons privilégié l'aspect temps de calcul. Pour cette raison le squelette est calculé à partir du diagramme de Delaunay décrit dans l'annexe B. Cette méthode est particulièrement rapide, pour une vidéo à 15 images par secondes, l'extraction du squelette est effectuée en temps réel. Celle-ci bien entendu montre sa limite pour les images réelles. Il s'agit de l'une des raisons pour lesquelles nous nous sommes ici limités au suivi d'objets sur des images de synthèses.

Les expériences présentées considèrent un cactus comme objet d'intérêt. Précisons que malgré le fait que l'objet utilisé soit virtuel, son suivi dans une séquence d'images est rendu difficile par la présence d'ombres, de reflets et d'auto-occlusions. Par ailleurs l'étape d'initialisation est faite manuellement.

La figure 7.17 montre le résultat du suivi du cactus sur une séquence de 68 images. Dans ce cas, le suivi est réalisé à la cadence de 15 Hz. Pour rendre le suivi plus compliqué, nous avons simulé un mouvement complexe avec une forte rotation autour de l'axe z (axe optique de la caméra). La tâche devient alors difficile à cause de la disparition d'une partie de l'objet. Le suivi est effectué par un modèle filaire (en magenta), identique à celui utilisé dans la partie reconnaissance. Les points d'intérêt représentant les extrémités des squelettes sont en rouge pour les points 3D, et en vert pour les points 2D.

Dans l'ensemble, le suivi s'effectue correctement, il arrive même parfois que le modèle lâche l'objet et le rattrape dans la séquence suivante (image 2). Cependant dans quelques vidéos et après quelques images, l'algorithme n'est plus capable de suivre efficacement l'objet. Cela est principalement dû à l'absence d'un nombre suffisant d'appariements ou d'une mauvaise squelettisation, les paramètres de pose 3D sont alors mal calculés. A l'instar de la méthode de Lowe, elle reste inefficace pour de grands déplacements car elle ne possède pas d'étape de prédiction.

7.5 Conclusion

Dans ce chapitre, nous avons présenté la dernière étape de notre système de reconnaissance. Elle est basée sur un système de projection et de vérification des

différentes hypothèses générées dans la phase d'appariement. Pour l'estimation de la pose et le calcul de l'erreur de projection, nous avons utilisé l'algorithme d'itération orthogonale.

Nous avons par la suite détaillé le comportement de notre méthode de reconnaissance sur un exemple, et nous avons analysé l'ensemble des résultats obtenus dans différentes situations (sans contrôle de la scène). Nous avons ainsi montré sa robustesse par rapport aux occlusions et aux auto-occlusions et son efficacité en présence d'autres objets.

Bien que les résultats expérimentaux nous aient montré l'efficacité de notre système sur l'objet de cactus, elle reste néanmoins limitée pour des objets du type coque. Car l'utilisation de la signature topologique reste insuffisante pour indexer et apparier les graphes correspondants (tous les nœuds sont à 0). Nous pouvons déduire que notre méthode fonctionnera parfaitement sur des objets dont la forme est complexe. L'analyse de ces résultats, nous a également montré les deux problèmes principaux de notre méthode. Le premier est le cas où le squelette 2D est dégénéré, le second problème est la sensibilité au choix de la racine lors de la construction du graphe.

L'adaptation de notre méthode de reconnaissance par squelettes au suivi d'objet nous a donné des résultats prometteurs et justifie le bien fondé de notre approche. Cependant pour cette partie, il reste à insérer plusieurs outils de traitement d'images rapides, afin de l'utiliser pour des séquences d'images réelles. Il reste aussi à intégrer une étape de prédictions, nous envisageons par exemple d'intégrer l'étape d'initialisation dans le suivi qui permettra de suivre l'objet dans le cas de mouvements saccadés.

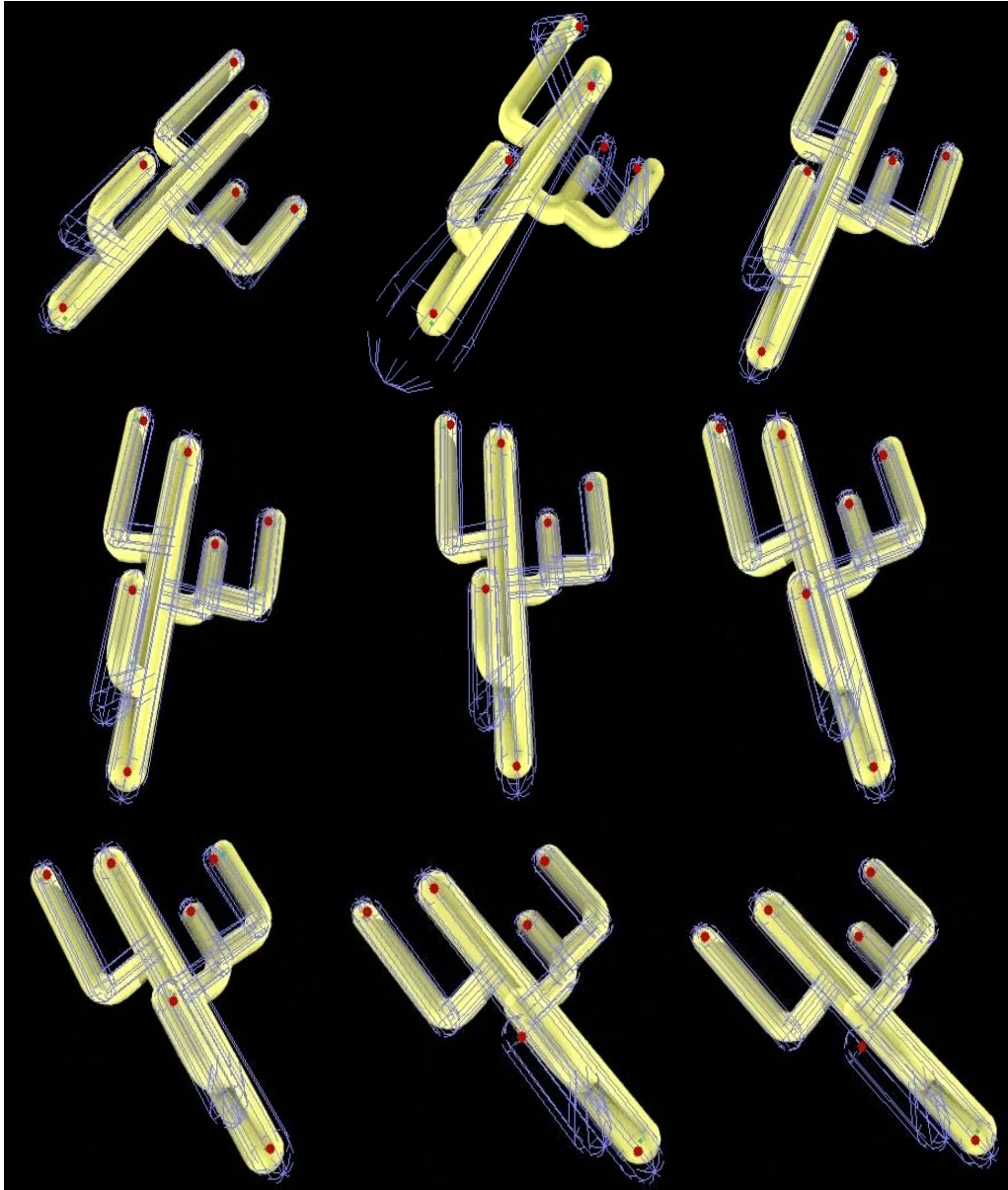


Figure 7.17: Séquence de suivi d'un objet

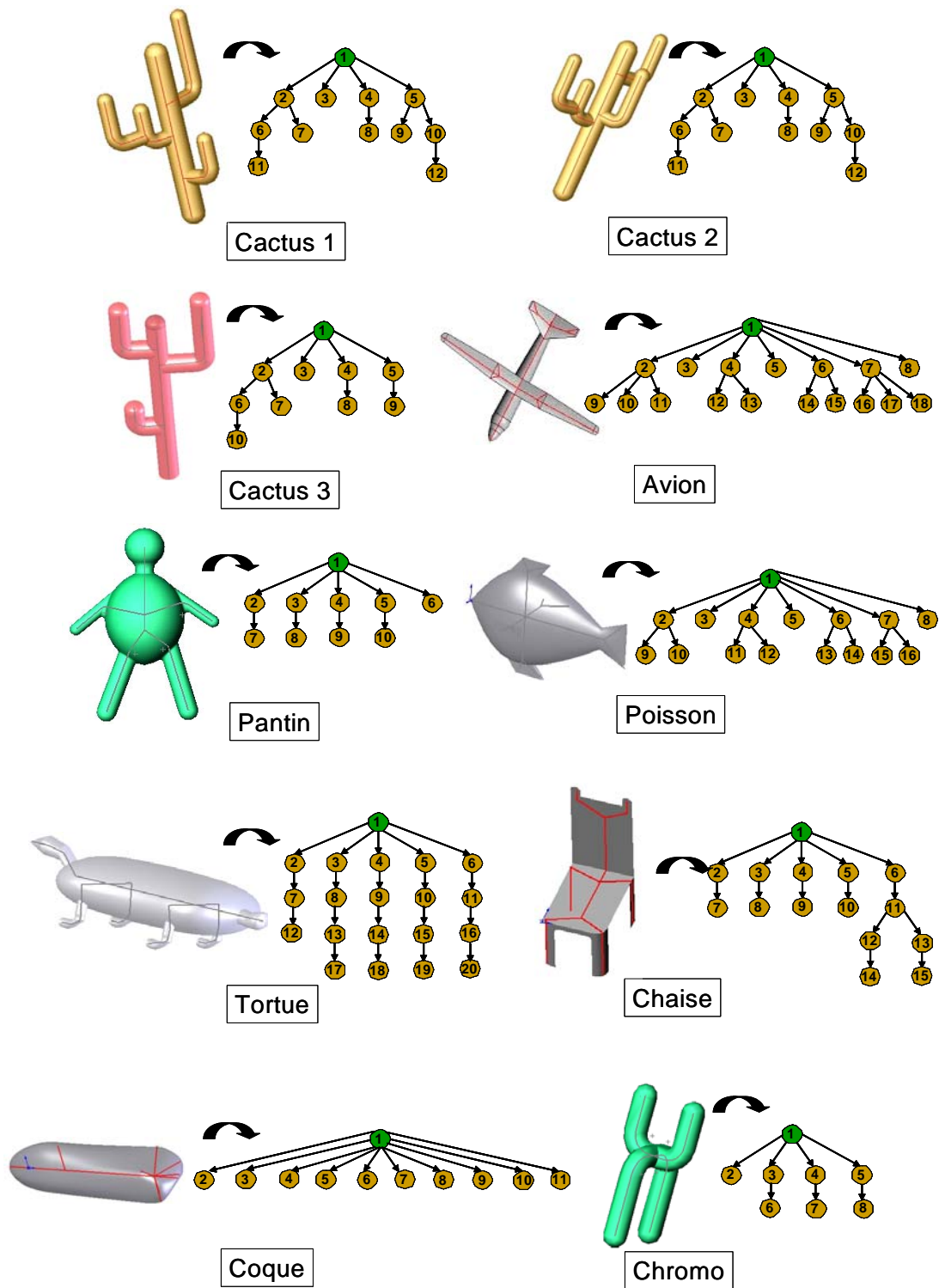


Figure 7.18: Base de données de modèles

Conclusion générale

Dans le cadre de cette thèse nous nous sommes intéressés successivement :

- à l'inspection automatique d'objets comportant des surfaces complexes ;
- à la reconnaissance d'objets 3D de formes libres.

Inspection automatique

Nous avons présenté dans cette première partie une méthode permettant la reconnaissance automatique de défauts ou de bougés dans le cadre d'une assistance au diagnostic de machines à usinage rapide.

En fonction des données contenues dans le cahier de charge, nous avons mis au point une chaîne de traitement complète pour effectuer cette tâche. Le principe de la méthode proposée consiste à extraire des paramètres sur une image, puis à détecter d'éventuels défauts, par comparaison de ces paramètres avec ceux d'une image de référence. Elle est composée principalement de deux parties, une partie traitement d'images et une partie classification supervisée.

Avant de choisir les algorithmes à intégrer dans la partie traitement d'images, nous avons effectué un état de l'art et nous avons choisi celles qui présentaient un bon compromis entre l'efficacité et le temps de calcul. Nous avons aussi comparé quelques unes de ces méthodes et étudié leurs comportements. Nous avons alors retenu la méthode d'Alvarez pour la restauration d'images et le filtre de Deriche et Harris pour les contours et les points de forte courbure respectivement.

Dans la partie classification, nous avons d'abord étudié la pertinence et la corrélation des paramètres extraits des deux images. Ensuite nous avons utilisé une méthode empirique pour la classification. Dans le but d'avoir un diagnostic robuste et fiable, nous avons testé un classifieur Bayésien et un autre basé sur un réseau de neurones. Finalement, nous avons opté pour un classifieur combiné avec une règle somme qui nous donne le meilleur résultat de classification. Les tests ont été effectués sur des images issues du site industriel, les taux de classifications obtenus sont satisfaisants et nous pensons qu'avec une base de données exhaustive, on aurait pu les améliorer d'avantage. Enfin sachant qu'un diagnostic

sans erreur est non réaliste, nous pouvons dire que nous avons répondu en grande partie au cahier des charges établi par le partenaire industriel. Le système que nous avons développé est un prototype complet et il ne reste que son transfert sur le site industriel pour qu'il soit opérationnel.

Reconnaissance d'objets 3D de forme libre

Nous avons étudié dans la deuxième partie de la thèse un problème important en vision, celui de la reconnaissance 3D d'objets de forme libre en vision monoculaire. Ce problème est d'autant plus important qu'il permet d'apporter une aide à la commande et à la perception d'une scène complexe, à un opérateur en situation de téléopération, en cas de vision indirecte.

Après un état de l'art qui nous a permis de constater l'absence d'une approche satisfaisante pour les objets de forme libre, nous avons présenté une méthode originale qui permet à l'aide d'une représentation en squelettes, de reconnaître ce type d'objets.

Afin de mettre en œuvre notre approche, nous nous sommes attachés au schéma traditionnel de la reconnaissance d'objets en vision. Ce schéma est composé de trois grandes parties, la première est la modélisation et l'extraction de primitives. Dans cette partie, nous appliquons un processus de squelettisation nous permettant d'obtenir deux graphes, un graphe 3D et un autre 2D. Le graphe 3D est obtenu à partir du squelette 3D homotopique et le graphe 2D est construit à partir du squelette 2D.

La squelettisation étant une étape importante dans notre système, nous avons alors étudié les méthodes existantes et nous avons développé une méthode hybride de squelettisation. Cette technique nous a permis de construire des squelettes robustes et ainsi de faciliter l'étape de mise en correspondance.

La seconde partie de la reconnaissance d'objets est l'appariement. Dans notre approche, cette partie revient à faire un isomorphisme entre le graphe 3D de la base de données et le graphe 2D obtenu à partir de l'image. Dans notre approche, cette tâche est difficile car cette mesure doit évaluer le degré de ressemblance entre structures similaires de deux sous-graphes, afin de gérer le problème d'occlusions, ... Après avoir étudié plusieurs algorithmes de mise en correspondance de graphes, nous avons adapté la méthode d'isomorphisme de Siddiqi qui utilise les signatures topologiques des nœuds. Cette méthode, nous permet d'indexer une base de donnée et d'apparier un-à-un les nœuds des deux graphes.

Après la phase d'appariement, l'algorithme nous propose généralement plusieurs hypothèses. La troisième et dernière partie consiste alors à vérifier les différentes

hypothèses et à en choisir la meilleure par un processus classique de projection/vérification.

Afin de démontrer l'efficacité de cette approche, nous l'avons testé sur une série d'images réelles représentant un modèle dans des poses différentes. Nous avons analysé l'ensemble des résultats obtenu dans différentes situations. Nous avons ainsi démontré sa robustesse par rapport aux occlusions et aux auto-occlusions et son efficacité en présence d'autres objets. Ainsi, les résultats obtenus sont très satisfaisants et nous confortent dans notre approche.

Nous avons également proposé un algorithme de suivi d'objets 3D de forme libre par les squelettes. Nous avons effectué des tests sur des images de synthèse et les résultats obtenus nous encouragent à poursuivre la recherche dans cette voie.

Cela dit, notre approche reste néanmoins limitée pour des objets dont le graphe correspondant n'est pas profond. Dans ce cas, l'utilisation de la signature topologique reste insuffisante pour indexer et apparier ce genre de graphe. L'analyse de ces résultats, nous a également montré les deux problèmes principaux de notre méthode. Le premier est le cas où le squelette 2D est dégénéré, le second problème est la sensibilité au choix de la racine lors de la construction du graphe.

Nous pouvons tous de même affirmer que l'approche basée sur les squelettes que nous proposons est pratique et efficace pour ces nombreuses raisons :

- L'appariement partiel d'objets : contrairement à une mesure globale de la forme, la mise en correspondance par squelette peut s'accommoder à la mise en correspondance partielle d'objets, c-à-d. l'objet à apparier peut être trouvé dans une partie d'un objet plus grand, ou vice versa.
- L'indexation : L'utilisation de l'indexation, basée sur la distance entre squelettes 2D et 3D, permet restreindre l'espace de recherche pendant le processus de reconnaissance dans la base de données.
- La visualisation : Le squelette peut être utilisé pour superposer un objet sur un autre et visualiser le résultat. Ceci est fort utile dans un système de réalité augmentée, notamment pour l'obtention de la vue par transparence.
- L'intuitivité : Le squelette est une représentation intuitive de la forme et peut être compris par n'importe quel utilisateur. En effet, l'aspect synthétique d'un squelette peut apporter une assistance à la perception de scènes dans une interface homme-machine.
- L'articulation : Le squelette peut être utilisé pour l'appariement d'objets articulés, parce que la topologie du squelette ne change pas pour un mouvement articulé.

Parmi les perspectives immédiates de ces travaux nous pouvons bien sûr citer la validation de l'approche sur d'autres objets réels de forme libre. Ainsi que son intégration dans un système de téléopération. Nous avons déjà démontré la faisabilité d'une telle approche sur un objet réel, néanmoins, son intégration dans un système de téléopération peut nous permettre de quantifier tant en précision qu'en qualité la reconnaissance ainsi obtenue.

Notre approche est limitée quant à la reconnaissance d'objets de type "coque". Pour cela, nous envisageons améliorer l'appariement en utilisant d'autres techniques d'isomorphismes de graphes.

Une autre alternative de notre approche pourra avoir lieu en lui intégrant les graphes d'aspects. Et ce, pour traiter les cas de squelettes dégénérés. Nous pourrions intégrer facilement dans la base de données, pour chaque modèle, un ensemble limité d'aspects dont le squelette 2D est dégénéré. Nous pourrions ainsi traiter ce genre d'ambiguïtés.

Cependant pour le suivi d'objets, il reste à insérer plusieurs outils de traitement d'images rapides afin de l'utiliser pour des séquences d'images réelles. On peut noter que certains maillons de la chaîne de traitement ont été volontairement réduits à leur plus simple expression. La squelettisation 2D mérite néanmoins une attention particulière et son amélioration (rapidité et robustesse) est nécessaire pour un suivi efficace.

Finalement, nous envisageons d'étendre l'utilisation de nos méthodes de reconnaissance et de suivi aux objets articulés qui est une suite logique à nos travaux.

Annexes

Annexe A

Analyse Multi-échelles

Nous rappelons ici quelques définitions relatives à l'analyse multi-échelles. Puis nous présenterons le schéma numérique de résolution de l'analyse euclidienne (voir [ALV 96] pour plus de détails).

A.1 Généralités

L'analyse multi-échelles d'une image $I_0(x, y)$ consiste en la génération d'images $I(x, y, t)$ où chaque image nouvelle apparaît comme une version lissée, à l'échelle t , de l'image initiale $I_0(x, y)$.

On peut écrire (A.1) :

$$\begin{cases} I(x, y, t) = T_t(I_0)(x, y) \\ I(x, y, 0) = I_0(x, y) \end{cases} \quad (\text{A.1})$$

L'axiomatisation de l'analyse multi-échelles est alors déterminée par la famille de transformations $(T_t)_{t \geq 0}$ dans laquelle l'image $I(x, y, t)$ est l'analyse de $I_0(x, y)$ à l'échelle t .

Cet opérateur (T_t) doit vérifier, selon l'analyse étudiée, les axiomes suivants :

A.1.1 Axiome de la structure pyramidale

L'image à une échelle $t + h$ avec $h > 0$ peut être obtenue à partir de la version à l'échelle précédente t , sans passer par l'image initiale. Cet axiome s'énonce comme suit (A.2) :

$$T_{t+h}(I) = T_{t,h} \circ T_t(I) \quad (\text{A.2})$$

avec $T_0 = Id$ et $T_{t,t}(I) = I$.

$T_{t,t+h}$ est un filtre de transition.

A.1.2 Axiome de comparaison locale

L'analyse ne doit pas introduire dans les images lissées des détails qui n'y étaient pas aux échelles précédentes. L'énoncé de l'axiome est le suivant (A.3) :

Si pour tout point (x, y) dans un voisinage d'un autre point (x_0, y_0) on a : $I(x, y) > v(x, y)$ (égalité stricte, sauf éventuellement en (x_0, y_0)), alors on doit avoir :

$$T_{t,t+h}(I)(x_0, y_0) \geq T_{t,t+h}(v)(x_0, y_0) \text{ pour } h \text{ assez petit} \quad (\text{A.3})$$

A.1.3 Axiome de régularité

Si I est une forme quadratique au voisinage d'un point m_0 de coordonnées (x_0, y_0) , c'est-à-dire :

$$I(x, y) = a + p^t(m - m_0) + \frac{1}{2}(m - m_0)^t A(m - m_0) \text{ si } \|m - m_0\| < e \quad (\text{A.4})$$

où $m = (x, y)$, alors, quand h tend vers 0, la valeur de $T_{t,t+h}(I)(x_0, y_0)$ ne doit dépendre que des valeurs de a, p, A .

A.1.4 Invariance morphologique

L'analyse doit commuter avec toute redistribution croissante $g(\cdot)$ des niveaux de gris. Ceci traduit le fait que seule la notion d'**isophote** est importante lors de cette analyse, d'où sa référence à la morphologie mathématique.

Soit une fonction croissante $g : R \rightarrow R$, alors:

$$T_{t,t+h}(g \circ I) = g \circ T_{t,t+h}(I) \quad (\text{A.5})$$

Cette invariance morphologique (A.5) est dite forte si cet axiome est respecté pour toute fonction $g(\cdot)$, croissante ou décroissante.

A.1.5 Invariance euclidienne ou affine

L'analyse doit être invariante pour toute transformation $F_{A,B} : \mathfrak{R}^2 \rightarrow \mathfrak{R}^2$, définie par $F_{A,b}(m) = (Am + b)$, où $n \in \mathfrak{R}^2$, A est une matrice $A \in \mathfrak{R}^2 \times \mathfrak{R}^2$ non singulière et $b \in \mathfrak{R}^2$ (A.6) :

$$T_{t,t+h}(I) \circ F_{A,B} = T_{t,t+h}(I \circ F_{A,B}) \quad (\text{A.6})$$

Si A est une isométrie, notons AI la fonction définie par $AI(x) = I(Ax)$. Alors l'invariance euclidienne s'exprime par (A.7) :

$$AT_{t,t+h} = T_{t,t+h}A \quad (\text{A.7})$$

A.1.6 Linéarité

L'opérateur est linéaire par rapport à ses arguments (A.8) :

$$T_{t,t+h}(aI + bv) = aT_{t,t+h}(I) + bT_{t,t+h}(v) \quad (\text{A.8})$$

A.2 Théorèmes généraux

Si une analyse multi-échelles $T(I)$ satisfait les axiomes de structure pyramidale, de comparaison locale et de régularité, alors la solution $I(x, y, t) = T_t(I)(x, y)$ est une solution de viscosité de l'EDP suivante (A.9) :

$$\frac{\partial I}{\partial t} = F(H(I), \nabla I, I, x, y, t) \quad (\text{A.9})$$

avec $I(x, y, 0) = I_0(x, y)$, où $H(I)$ et ∇I désignent respectivement le Hessien et le gradient de I et où $F(H(I), \nabla I, I, x, y, t)$ est la fonction définie dans le principe de régularité.

Théorème A.1 *Si une analyse multi échelles satisfait aux axiomes de linéarité, d'invariances euclidienne, de structure pyramidale et de comparaison locale, alors la solution est une solution de l'EDP suivante :*

$$\frac{\partial I}{\partial t} = \Delta I(x, y, t) \quad (\text{A.10})$$

avec $I(x, y, 0) = I_0(x, y)$

Cette équation, plus connue sous le nom d'équation de la chaleur, correspond au cas où la fonction F définie dans l'axiome de régularité est égale à la trace du Hessien $H(I)$. La solution de cette EDP s'obtient par la convolution de l'image initiale $I_0(x, y)$ avec l'opérateur Gaussien.

Théorème A.2 *Si une analyse multi échelles satisfait aux axiomes de régularité, d'invariances euclidienne et morphologique, de structure pyramidale et de comparaison locale, alors la solution est une solution de l'EDP suivante :*

$$\frac{\partial I}{\partial t} = |\nabla I| G \left(\operatorname{div} \left(\frac{\nabla I}{|\nabla I|} \right), t \right) \quad (\text{A.11})$$

avec $I(x, y, 0) = I_0(x, y)$

où $\operatorname{div} \left(\frac{\nabla I}{|\nabla I|} \right) = \frac{I_y^2 I_{xx} - 2I_x I_y I_{xy} + I_x^2 I_{yy}}{(I_x^2 + I_y^2)^{3/2}}$ désigne la courbure en chaque point (x, y, t) de la ligne de niveau définie par $I(x, y, t) = C$ et où la fonction $G(s, t)$ est une fonction non décroissante par rapport à la variable réelle s .

Un autre cas particulier intéressant cette fonction est celui où $G(s, t) = s.t$:

$$\frac{\partial I}{\partial t} = t |\nabla I| \operatorname{div} \left(\frac{\nabla I}{|\nabla I|} \right) \quad (\text{A.12})$$

avec $I(x, y, 0) = I_0(x, y)$

Cette équation correspond à une diffusion anisotrope dans la direction des lignes de niveaux de l'image, il s'agit d'une analyse morphologique euclidienne, elle se réécrit :

$$\frac{\partial I}{\partial t} = t \frac{I_y^2 I_{xx} - 2I_x I_y I_{xy} + I_x^2 I_{yy}}{I_x^2 + I_y^2} \quad (\text{A.13})$$

avec $I(x, y, 0) = I_0(x, y)$

A.3 Rappel théorique et implémentation

D'après (A.13) :

$$I_t = t \frac{I_y^2 I_{xx} - 2I_x I_y I_{xy} + I_x^2 I_{yy}}{I_x^2 + I_y^2} \quad (\text{A.14})$$

Posons :

$$F(I) = I_y^2 I_{xx} - 2I_x I_y I_{xy} + I_x^2 I_{yy} \quad (\text{A.15})$$

alors

$$\frac{\partial I}{\partial t} = t \frac{F(I)}{I_x^2 + I_y^2} \quad (\text{A.16})$$

En faisant le changement d'échelle

$$t_1 = \left(\frac{1}{2} \right) \times t^2 \quad (\text{A.17})$$

l'équation devient :

$$\frac{\partial I}{\partial t_1} = \frac{F(I)}{I_x^2 + I_y^2} \quad (\text{A.18})$$

Lorsque le temps est discrétisé, la récurrence suivante est dégagée :

$$I_{i,j}^{n+1} = I_{i,j}^n + dt \times \frac{F(I_{i,j}^n)}{(I_{i,j}^n)_x^2 + (I_{i,j}^n)_y^2} \quad (\text{A.19})$$

L'équation (A.19) peut être implémentée de manière récursive selon deux schémas numériques différents pour l'estimation de l'opérateur non-linéaire F . Ils sont appelés les schémas numériques 8 voisins et 6 voisins. Pour notre application, nous avons utilisé le schéma présenté ci-dessous.

Soit $I_{i,j}$ un pixel central. Ce schéma fait intervenir ses 8 premiers voisins de la manière suivante :

$$F(I) = \frac{1}{h^2} (-4\lambda_0 I_{i,j} + \lambda_1 (I_{i,j+1} + I_{i,j-1})) + \lambda_2 (I_{i-1,j} + I_{i+1,j}) + \lambda_3 (I_{i-1,j+1} + I_{i-1,j-1}) + \lambda_4 (I_{i+1,j+1} + I_{i-1,j-1}) + O(h) \quad (\text{A.20})$$

Soit le masque suivant, au pixel $I_{i,j}$, en considérant que l'indice i désigne la ligne et j la colonne, et que le repère est orienté ainsi : origine en haut à gauche, les i croissants de haut en bas, les j croissants de gauche à droite.

$I_{i-1,j-1}$	$I_{i-1,j}$	$I_{i-1,j+1}$
$I_{i,j-1}$	$I_{i,j}$	$I_{i,j+1}$
$I_{i+1,j-1}$	$I_{i+1,j}$	$I_{i+1,j+1}$

λ_4	λ_2	λ_3
λ_1	$4\lambda_0$	λ_1
λ_3	λ_2	λ_4

Sachant que, d'après les développements de Taylor, on a :

$$I(t+k, x, y) = I + ktI_t + O(k^2)$$

$$I(t-k, x, y) = I - ktI_t + O(k^2)$$

$$I(t, x+h, y) = I - hI_x + \frac{h}{2}I_{xx} + O(h^3)$$

$$I(t, x, y+l) = I + lI_y + \frac{l^2}{2}I_{yy} + O(l^3)$$

$$I(t, x, y-l) = I - lI_y + \frac{l^2}{2}I_{yy} + O(l^3)$$

$$I(t, x+h, y+l) = I + hI_x + lI_y + \frac{1}{2}(h^2I_{xx} + 2hlI_{xy} + l^2I_{yy}) + O(l^3 + h^3)$$

$$I(t, x - h, y - l) = I - hI_x - lI_y + \frac{1}{2}(h^2I_{xx} + 2hllI_{xy} + l^2I_{yy}) + O(l^3 + h^3)$$

$$I(t, x + h, y - l) = I + hI_x - lI_y + \frac{1}{2}(h^2I_{xx} - 2hllI_{xy} + l^2I_{yy}) + O(l^3 + h^3)$$

$$I(t, x - h, y + l) = I - hI_x + lI_y + \frac{1}{2}(h^2I_{xx} - 2hllI_{xy} + l^2I_{yy}) + O(l^3 + h^3)$$

En considérant $l = h = 1$:

$$\begin{aligned} F(I) = & I_{i,j}(-4\lambda_0 + 2\lambda_1 + 2\lambda_2 + 2\lambda_3 + 2\lambda_4) + I_{xx}(\lambda_2 + \lambda_3 + \lambda_4) + I_{yy}(\lambda_1 + \lambda_3 + \lambda_4) \\ & + 2I_{xy}(-\lambda_3 + \lambda_4) \end{aligned} \quad (\text{A.21})$$

En identifiant avec l'expression de $F(I)$ dans (A.15), on en déduit les coefficients :

$$\begin{cases} \lambda_1 = 2\lambda_0 - I_y^2 \\ \lambda_2 = 2\lambda_0 - I_x^2 \\ \lambda_3 = -\lambda_0 + \frac{1}{2}(I_x^2 + I_y^2 + I_xI_y) \\ \lambda_4 = -\lambda_0 + \frac{1}{2}(I_x^2 + I_y^2 - I_xI_y) \end{cases} \quad (\text{A.22})$$

où λ_0 est un paramètre libre.

Plusieurs choix sont possibles pour λ_0 , pour tenir compte des critères de stabilité et d'invariances du schéma numérique. Dans [ALV 96], Alvarez a présenté les différentes possibilités dans le choix de ce paramètre, ainsi que les masques qui en sont déduits pour l'approximation numérique de l'opérateur $F(I_{ij}^n)$.

Pour notre implémentations nous avons choisi $\lambda_0 = 0.25 \times (I_x^2 + I_y^2)$. Cette approche résulte de la minimisation de l'erreur résiduelle dans le développement de Taylor de l'opérateur $F(I)$. Le masque qui lui est associé est :

$$\begin{bmatrix} \frac{1}{4}(I_x^2 + I_y^2) - \frac{1}{2}I_xI_y & \frac{1}{2}(-I_x^2 + I_y^2) & \frac{1}{4}(I_x^2 + I_y^2) - \frac{1}{2}I_xI_y \\ \frac{1}{2}(I_x^2 - I_y^2) & -(I_x^2 + I_y^2) & \frac{1}{2}(I_x^2 - I_y^2) \\ \frac{1}{4}(I_x^2 + I_y^2) - \frac{1}{2}I_xI_y & \frac{1}{2}(-I_x^2 + I_y^2) & \frac{1}{4}(I_x^2 + I_y^2) - \frac{1}{2}I_xI_y \end{bmatrix} \quad (\text{A.23})$$

Annexe B

Triangulation de Delaunay

Dans cette annexe, nous rappelons les définitions et propriétés du diagramme de Delaunay. Parallèlement, nous introduisons la notion du diagramme de Voronoi, qui constitue l'élément principal à partir duquel sont construites les différentes approximations du squelette continu. Par la suite, nous parlerons sommairement des méthodes de construction du diagramme de Delaunay et de son application pour l'extraction du squelette.

B.1 Diagramme de Voronoi

Soit E un nuage de points $S_i, i = 1, \dots, m$, en n dimensions, la construction de la triangulation de Delaunay de l'enveloppe convexe de ce nuage peut s'établir en utilisant le fait que celle-ci est le dual du diagramme de Voronoi construit sur E .

Le diagramme de Voronoi est l'ensemble des cellules ou polytopes, V_i , définis par :

$$V_i = \{S \text{ tel que } d(S, S_i) \leq d(S, S_k), \forall k \neq i\} \quad (\text{B.1})$$

Où $d(,)$ est la distance entre deux points, distance induite par la métrique euclidienne. Une cellule, V_i , est donc le lieu des points plus proches de S_i que de tout autre point de E . Les cellules V_i sont des polygones fermés (en 2D) convexes, et recouvrent l'espace sans chevauchement. Un exemple de diagramme de Voronoi de points en métrique euclidienne est présenté sur la figure B.1.

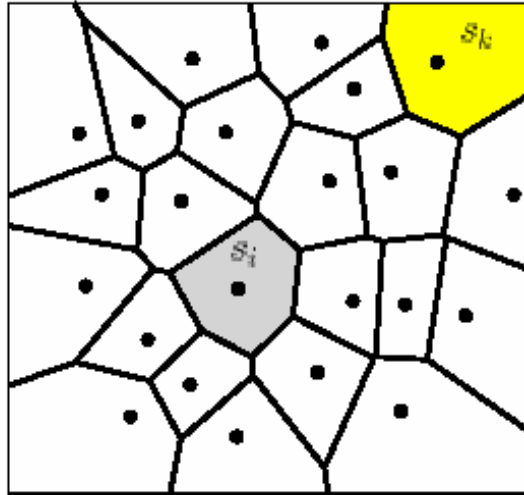


Figure B.1: Diagramme de Voronoï de points dans le plan.

B.2 Delaunay, dual de Voronoï

Par définition, chaque cellule V_i est non vide et est associée à un point S de E . À partir de ces V_i on peut construire le dual qui est la triangulation de Delaunay.

En deux dimensions, les côtés des cellules sont à mi-chemin des deux points qu'ils séparent et par suite ne sont autres que les médiatrices des arêtes de la triangulation. Autrement dit, si on joint les sommets S appartenant à deux cellules adjacentes, on obtient cette triangulation. En deux dimensions (ou trois), s'il n'existe pas de points cocycliques (cosphériques) alors le dual des cellules de Voronoï associées au nuage de points donnés, est la triangulation de Delaunay de l'enveloppe convexe de ce nuage (figure B.2). Cette triangulation est formée de simplexes (triangle en 2D et tétraèdre en 3D) définis de la façon suivante.

Définition B.1 Soit E un ensemble fini de points de l'espace R^n . On appelle *simplexe de Delaunay*, tout simplexe dont la boule circonscrite B a un intérieur vide de points de E . B est appelée *boule de Delaunay*.

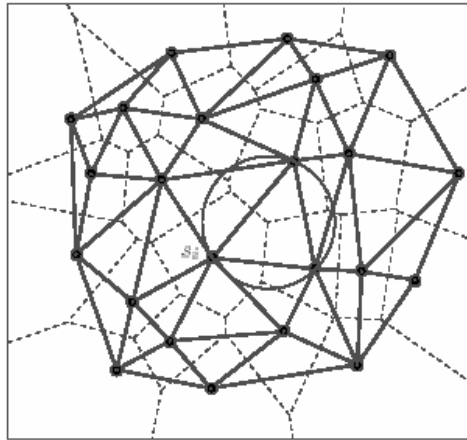


Figure B.2: Dualité Delaunay/Voronoi.

B.2.1 Algorithmes de construction

Les algorithmes de construction de ces structures géométriques sont divers et variés. Nous noterons néanmoins trois classes principales de méthodes (méthodes incrémentales, méthodes "divide and conquer", et méthodes de balayage). Des informations supplémentaires sont disponibles dans [GEO 97] pour les triangulations.

B.2.1.1 Algorithmes "Divide and Conquer"

Les algorithmes du type "diviser pour régner" consistent à diviser le problème général en sous-problèmes de plus petites tailles [SHA 75]. La division se fait de façon récursive jusqu' à obtenir des problèmes simples à résoudre (lorsqu'il ne reste plus que trois points par exemple). Chaque sous-problème est traité de façon indépendante, et une étape de fusion est nécessaire pour unifier les sous-problèmes. Le point crucial de ce type de méthodes est la partie fusion du processus.

B.2.1.2 Algorithmes de balayage

Ces algorithmes construisent les structures géométriques en balayant l'espace des données par une droite ou un plan suivant un axe privilégié, et mettent à jour la structure à chaque fois qu'un point est rencontré. Fortune [FOR 87] a

adapté cette méthode pour calculer des diagrammes de Voronoï dans le plan. Cet algorithme n'a pas été généralisé au cas tridimensionnel.

B.2.1.3 Algorithmes incrémentaux

Les algorithmes incrémentaux consistent à insérer les sites S , les uns après les autres, et à modifier la structure à chaque itération. La modification ne se fait que de façon locale. Pour la triangulation de Delaunay incrémentale, lors de l'insertion d'un point S_i , seuls les triangles dit "en conflit" avec S_i sont modifiés.

Définition B.2 *On dira qu'un triangle T est en conflit avec un site S_i , si S_i se trouve à l'intérieur du cercle circonscrit à T .*

Soit DT_t la triangulation de Delaunay à l'instant t . Lorsqu'on insère un point S_i à l'étape $t + 1$, on recherche dans DT_t l'ensemble des triangles en conflit avec S_i . Celui-ci forme une région connexe $R_t(S_i)$ contenant S_i . (figure B.3).

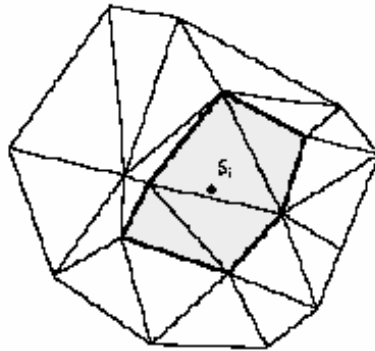


Figure B.3: Simplexes de Delaunay en conflit avec S_i

La région $R_t(S_i)$ est la seule à être modifiée dans la nouvelle triangulation DT_{t+1} : on forme les nouveaux triangles à partir d'une face sur le bord de $R_t(S_i)$ et du point S_i . (figure B.4).

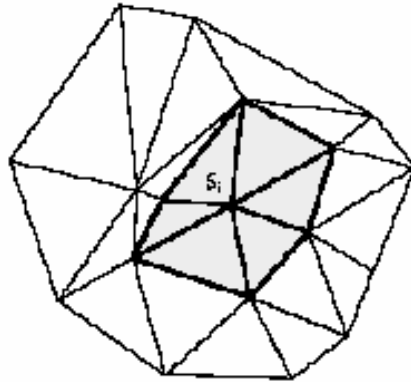


Figure B.4: Nouvelle triangulation

B.3 Squelette et triangulation de Delaunay

Historiquement, les méthodes continues utilisent le diagramme de Voronoi pour extraire les squelettes [ATT 95]. Une conséquence directe de la dualité (Voronoi-Delaunay) est que la construction du squelette à partir du diagramme de Voronoi, peut se déduire de la triangulation de Delaunay.

Pour nos applications (squelettisation hybride et suivi), nous avons commencé par utiliser le diagramme de Voronoi. Par la suite, nous avons remarqué que parcourir une triangulation est souvent plus simple que de parcourir un complexe quelconque. En effet les simplexes d'une triangulation ont un nombre fixe de voisins (trois en 2D et quatre en 3D) alors que pour un complexe général le nombre de voisins est variable. Nous avons alors basculer vers le diagramme de Delaunay, calculé par la méthode incrémentale décrite ci-dessus. Le squelette est alors déduit de la façon suivante.

B.3.1 Construction du squelette

Soit X un domaine de R^n , le problème est de trouver sa ligne médiane ($n = 2$) qui constitue le squelette de X . Pour nous le domaine X (objet) est connu via une discrétisation ∂X de sa frontière en ligne polygonale (obtenu par détection de contours). On suppose que ∂X est discrétisée par un ensemble d'arêtes, alors, on a le théorème suivant.

Théorème B.3 *Si h tend vers 0, où h est la longueur de la plus grande arête de la discrétisation de la frontière ∂X du domaine X et si DT est la triangulation de Delaunay vide de X , alors l'union des segments formés en joignant les centres de cercles circonscrits aux éléments de DT , en considérant les paires d'éléments adjacents par une arête, tend vers le squelette de X .*

D'un point de vue pratique, la discrétisation de la frontière n'est pas infiniment fine et la construction précédente n'est pas réaliste ($h \rightarrow 0$).

Pour la partie squelettisation hybride, nous avons utilisé cette technique en choisissant h aussi fin que possible. Nous obtenons un squelette avec des segments parasites que l'opération d'élagage élimine facilement (paragraphe 5.4).

Pour la partie suivi, nous avons utilisé une idée voisine qui consiste en l'analyse par adjacence des éléments de la triangulation DT . Ceci permet de construire une suite de segments de la façon suivante :

- (1) si un élément comprend une arête de la frontière a , on forme le segment dont les extrémités sont les milieux des arêtes autres que a ,
- (2) si un élément comprend pas d'arêtes frontières a_1 et a_2 , on forme le segment joignant l'extrémité commune à a_1 et a_2 et le milieu de la troisième arête
- (3) si un élément ne comprend pas d'arêtes frontières (ou en comprend trois), on construit G le barycentre du triangle et on forme les trois segments joignant G aux milieux des trois arêtes.

L'union des segments ainsi définis forme une approximation du squelette qui est rapide et facile à obtenir.

Annexe C

Squelette par Carte de Distance

Dans cette annexe, nous allons rappeler quelques notions relatives à la carte de distance, qui nous ont été utiles à la compréhension des différents problèmes relatifs au domaine discret. Nous parlerons des différentes façons d'approximer la carte de distance et ensuite de la manière d'extraire le squelette. Nous rappelons que nous avons utilisé ce type de squelette dans notre méthode de squelettisation hybride.

C.1 Calcul de la carte de distance

C.1.1 Principe

Soit X l'objet dont nous voulons extraire le squelette. Dans un espace discret n -dimensionnel, X est défini comme un sous-ensemble fini de Z^n . Le complément de X (le fond de l'image) est noté \bar{X} .

La carte de distance ρ de est défini par :

$$Z^n \rightarrow R^+$$

$$M \mapsto \rho(M) = d(M, \bar{X}) = \inf_{p \in \bar{X}} d(M, p) \quad (\text{C.1})$$

où d est une fonction de distance quelconque, classiquement la distance euclidienne.

La projection du point M en \bar{X} est le sous ensemble \bar{X} défini par :

$$M \mapsto \Pi(M) = \{p \in \bar{X} / d(M, p) = \rho(M)\} \quad (\text{C.2})$$

Rappelant que nous sommes dans un espace discret et pour obtenir des résultats invariants par rapport aux transformations isométriques, il faut utilisé la

distance euclidienne. Le calcul exact de la distance euclidienne est théoriquement possible. Toutefois, son coût, en terme de temps de calcul, est élevé. Aussi, nous allons présenter maintenant deux méthodes d'approximation de la distance euclidienne.

C.1.2 Méthode de Danielsson

Pour obtenir une carte de distance représentant fidèlement la distance euclidienne, Danielsson [DAN 80] a mis au point une méthode qui associe à chaque pixel de l'objet X , non pas une distance, mais un vecteur qui pointe vers un des pixels le plus proche du fond \bar{X} . Lors du calcul de la carte de distance, on propage alors des vecteurs et non pas des distances. Ceci impose de travailler avec deux images de composantes au lieu d'une seule image de distance.

C.1.3 Distance du chanfrein

Nous présentons ici le type de distance que nous avons utilisé.

La distance du chanfrein est très utilisée grâce à son efficacité et sa simplicité de mise en œuvre [BOR 86], [MON 68]. On définit sur une trame choisie un ensemble de déplacements élémentaires correspondants aux arcs du graphe dont les points de la trame sont les nœuds. On associe à chaque type de déplacement élémentaire un coût correspondant à une distance locale. La complexité de l'algorithme de propagation de distance est proportionnelle à la taille de l'image et au nombre de déplacements élémentaires. Les métriques liées aux connexités classiques (4 et 8, etc) font partie de la classe de métriques que cette approche permet d'employer.

Pour déterminer nos cartes de distances, nous avons utilisé l'algorithme séquentiel avec les deux masques de la figure C.1, cela nous permet d'avoir ces dernières avec seulement deux passages sur l'image indépendamment de la taille des objets.

C.1.3.1 Algorithme Séquentiel

On découpe le masque en deux demi-masques symétriques par rapport à 0 (figure C.1). Le demi-masque avant contient les points antérieurs au milieu 0 dans le sens du balayage. Ses n points ont les coordonnées (x_i, y_i) et un poids ω_i . Le demi-masque arrière est donc constitué des n points $(-x_i, -y_i)$ de poids ω_i . Aucun des deux demi-masques ne contient le point 0. Pour avoir la carte de distances on effectue deux passages, un dans le sens vidéo et le second dans le sens vidéo inverse.

Parcours avant : de haut en bas et de gauche à droite.

$$\rho(x, y) = \min\{\rho(x + x_i, y + y_i) + \omega_i\}$$

Parcours arrière : de bas en haut et de droite à gauche

$$\rho(x, y) = \min\{\rho(x, y), \rho(x - x_i, y - y_i) + \omega_i\}$$

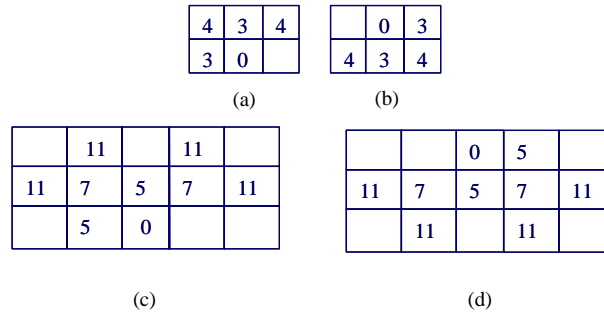
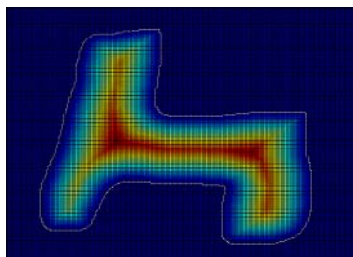


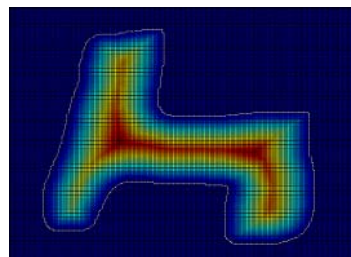
Figure C.1: Masques séquentiels d_{34} et d_{5711} : avant (a, c) et arrière (b, d)



(a): Objet



(b): Avec le masque d_{34}



(c): Avec le masque d_{5711}

Figure C.2: Carte de distance

Les figures C.2.b et C.2.c représentent les cartes de distances de l'objet de la figure C.2.a.

C.2 Détection du squelette

Nous parlerons maintenant des méthodes de détection des arêtes de la carte de distance associée à un objet, utilisées pour localiser le squelette.

C.2.1 Centre des boules maximales

La première caractérisation du squelette de X se base sur le concept de boules maximales de X , le squelette $Sk(X)$, étant l'ensemble de leurs centres.

Une boule $B_r(M)$ est définie par son rayon r et son centre M . On dit qu'une boule $B_r(M)$ de X est maximale si et seulement si :

$$B_r(M) \subset X$$

et

$$\nexists (r', M') \neq (r, M) \quad \text{telque} \quad B_{r'}(M') \subset X \text{ et } B_r(M) \subset B_{r'}(M') \quad (\text{C.3})$$

Calabi [CAL 65] prouve que si $\text{card}(\Pi(M)) \geq 2$ alors le point M est le centre d'une boule maximale.

Cette méthode ne donne pas de bons résultats pour la squelettisation. Elle détecte des squelettes non convexes, car on ne peut pas détecter les centres des boules maximales qui ne sont pas placés au milieu d'un pixel, à cause de l'approximation d'une boule dans une trame discrète.

C.2.2 Approche gradient

Les inconvénients de la méthode précédente, dus à la discrétisation de l'espace, peuvent être partiellement évités avec une autre définition du squelette. En effet le squelette est aussi le lieu des discontinuités de la dérivée de la carte de distance ρ .

Ces discontinuités sont les maxima locaux de la carte de distance dans la direction de son gradient (figure C.3).

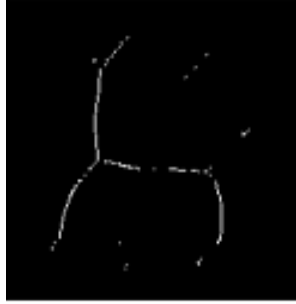


Figure C.3: Squelette par le gradient

C.2.3 Détection du squelette par géométrie différentielle

La carte de distance peut être exprimée à l'aide d'une surface d'élévation. Pour localiser le squelette, on détecte les arêtes de cette surface par des opérateurs issus de géométrie différentielle.

Soit la surface d'élévation :

$$z = \rho(x, y) \quad (\text{C.4})$$

Si nous considérons que cette surface est différentiable, nous pouvons calculer certains paramètres différentiels :

- La courbure gaussienne.

$$K = \frac{\rho_{xx}\rho_{yy} - \rho_{xy}^2}{(1 + \rho_x^2 + \rho_y^2)^2} \quad (\text{C.5})$$

- La courbure moyenne.

$$2H = \frac{(1 + \rho_x^2)\rho_{yy} - \rho_x\rho_y\rho_{xy} + (1 + \rho_y^2)\rho_{xx}}{(1 + \rho_x^2 + \rho_y^2)^{3/2}} \quad (\text{C.6})$$

Les courbures moyenne et gaussienne prennent des valeurs caractéristiques le long des arêtes de la surface, c'est-à-dire des petites valeurs par rapport aux valeurs des autres régions de la surface. Ceci est dû au fait que la valeur d'une des courbures principales est nulle ou presque le long d'une ligne de crête.

- L'opérateur ρ_{vv} .

Van Den Elsen a proposé dans [ELS 92], un opérateur qui est en réalité la deuxième dérivée dans la direction orthogonale au gradient local de ρ .

Il est donné par l'expression suivante :

$$\rho_{vv} = \frac{\rho_x^2 \rho_{yy} - 2\rho_x \rho_y \rho_{xy} + \rho_y^2 \rho_{xx}}{\rho_x^2 + \rho_y^2} \quad (\text{C.7})$$

Le long des arêtes, cet opérateur prend des valeurs négatives relativement grandes, partout ailleurs sur la surface il prend des valeurs positives relativement grandes.

Nous avons testé ces trois opérateurs et le meilleur résultat est obtenu en utilisant la courbure moyenne (figure C.4).

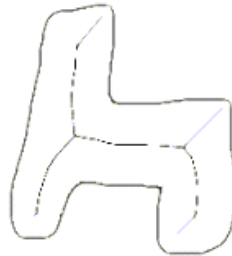


Figure C.4: Squelette par Courbure moyenne

Annexe D

Recalage 3D

L'objectif du recalage 3D est de déterminer la transformation géométrique rigide (translation et rotation) permettant d'amener l'objet de sa position initiale à sa position courante dans le repère de la scène. Classiquement il s'agit de calculer la position de l'objet soit dans le repère associé au capteur soit dans le repère associé à la scène.

Le recalage d'objets suppose la connaissance d'un nombre suffisant d'appariements entre les primitives de la scène ainsi que du modèle 3D géométrique associé à l'objet. De plus, dans le cas où l'on utilise des images d'intensité, nous avons aussi besoin de connaître le modèle de projection perspective de la caméra. Ce modèle est déterminé par le processus de calibration de la caméra. Il rend possible, connaissant les coordonnées d'un point 3D de la scène, le calcul des coordonnées 2D associées dans le plan image de la caméra.

Notre méthode permet d'apparier des points 2D de l'image (points du squelette 2D) avec des points 3D du modèle (points du squelette 3D). Nous allons donc formuler le problème de recalage 3D de telle manière que la transformation recherchée soit exprimée en fonction de ces données.

D.1 Définition du problème

Soit un ensemble de points de référence 3D non colinéaires $p_i = (x_i, y_i, z_i)^t$, $i = 1, \dots, n$ $n \geq 3$ exprimés dans le référentiel propre de l'objet (R_O), les points 3D qui leur correspondent dans le référentiel de la caméra $q_i = (x'_i, y'_i, z'_i)^t$ sont définis par une transformation rigide :

$$q_i = R.p_i + T$$

où

$$R = \begin{pmatrix} r_1^t \\ r_2^t \\ r_3^t \end{pmatrix} \text{ tel que } R^t R = I \quad \text{et} \quad T = \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix} \quad (\text{D.1})$$

sont la matrice de rotation et le vecteur de translation, respectivement. Dans notre cas, les points p_i correspondent aux extrémités du squelette 3D homotopique du modèle retenu. La condition $R^t R = I$ est la condition d'orthonormalité de la matrice de rotation R .

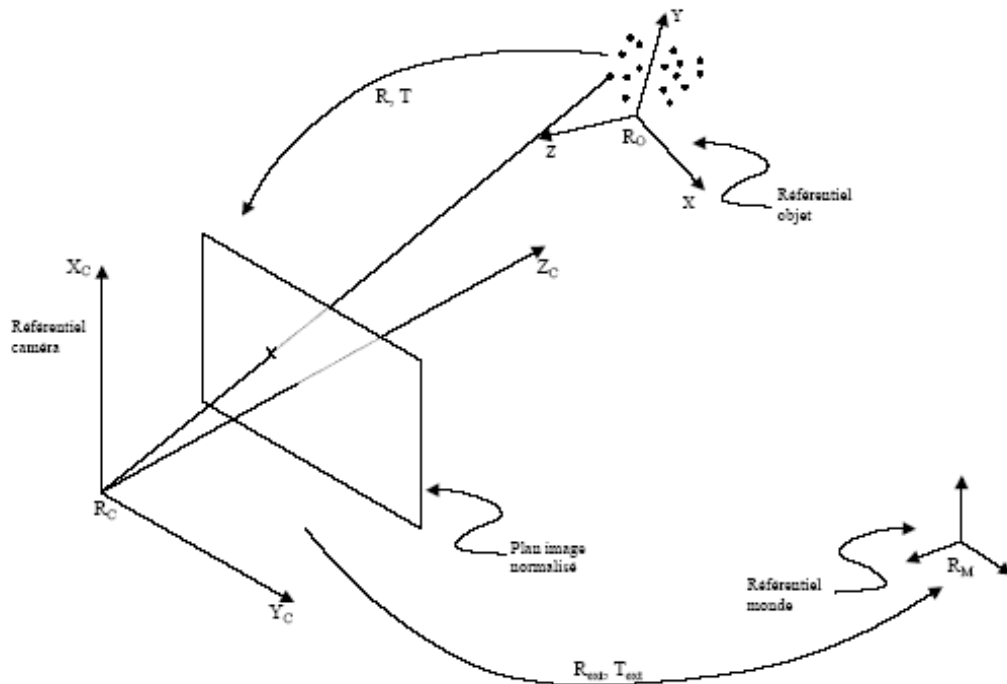


Figure D.1: Les différents repères mis en jeu dans le problème de recalage 3D

La méthode de recalage 3D que nous avons élaborée fait intervenir trois repères principaux (figure D.1) :

- (R_C) le repère lié à la caméra.
- (R_O) le repère lié à l'objet représentant la position courante. Dans notre cas il correspond au repère centré sur le modèle sélectionné
- (R_M) le repère monde (de la scène).

Le référentiel de la caméra est choisi de telle sorte que le centre de projection de la caméra soit à l'origine et que l'axe optique soit orienté dans la direction des z positifs. Les points de référence sont projetés dans le plan $z=1$, appelé plan image normalisé, et décrits dans le repère caméra.

Soit $m_i = (u_i, v_i, 1)^t$ les points image qui correspondent aux projections de p_i dans le plan image normalisé (figure D.1).

Sous l'hypothèse du modèle pinhole idéal, les points m_i, q_i , et le centre de projection sont colinéaires. Cette propriété est exprimée par l'équation suivante :

$$u_i = \frac{r_1^t p_i + t_x}{r_3^t p_i + t_z} \quad (\text{D.2})$$

$$v_i = \frac{r_2^t p_i + t_y}{r_3^t p_i + t_z} \quad (\text{D.3})$$

ou bien

$$m_i = \frac{1}{r_3^t p_i + t_z} (R p_i + T) \quad (\text{D.4})$$

l'équation (D.4) est appelée "équation de colinéarité". Une autre manière d'exprimer cette colinéarité est d'utiliser le fait que la projection orthogonale de sur la ligne de vue définie par le point doit être égale au point lui même. Ce fait est exprimé par l'équation :

$$R \cdot p_i + T = V_i (R \cdot p_i + T) \quad (\text{D.5})$$

où

$$V_i = \frac{m_i \cdot m_i^t}{m_i^t \cdot m_i} \quad (\text{D.6})$$

représente la matrice de projection de la ligne de vue qui, lorsqu'elle est appliquée à un point de la scène, le projette orthogonalement sur la ligne de vue définie par le point image m_i . Ainsi, l'équation (D.4) représente l'équation de colinéarité dans l'espace image tandis que l'équation (D.5) représente l'équation de colinéarité mais dans l'espace objet. Le recalage 3D consiste à développer une méthode qui permet de trouver la transformation rigide (R, T) minimisant une certaine forme d'accumulation d'erreurs (tel que la sommation des carrés des erreurs) de l'une des deux équations de colinéarité (figure D.2).

Dans les méthodes classiques de recalage 3D, la fonction objective à minimiser est donnée par :

$$Q = \sum_{i=1}^n \left[\left(\hat{u}_i - \frac{r_1^t \cdot p_i + t_x}{r_3^t \cdot p_i + t_z} \right)^2 + \left(\hat{v}_i - \frac{r_2^t \cdot p_i + t_y}{r_3^t \cdot p_i + t_z} \right)^2 \right] \quad (\text{D.7})$$

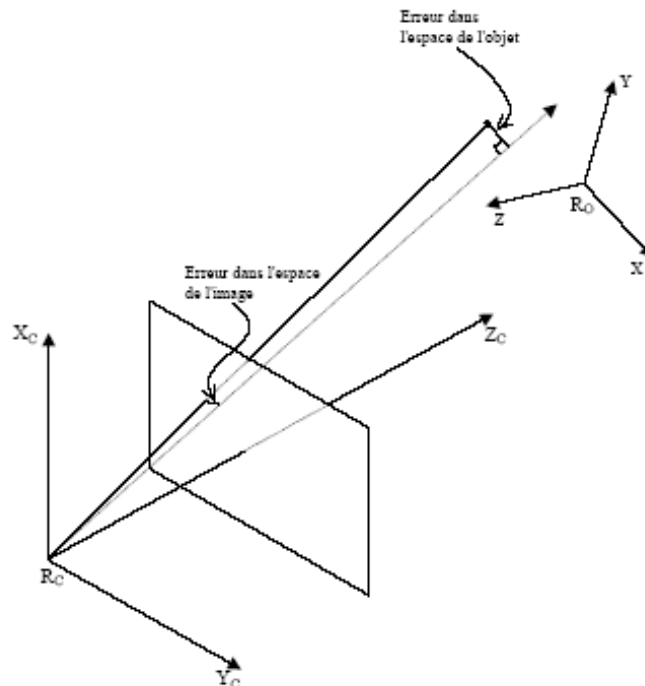


Figure D.2: Les erreurs de colinéarité dans l'espace image et l'espace objet

où $\hat{m}_i = (\hat{u}_i, \hat{v}_i, 1)^t$ sont les points image observés. La matrice de rotation R est généralement représentée par les angles d'Euler. Notons au passage que la minimisation dans ce cas est basée sur la colinéarité dans l'espace image. Deux méthodes sont généralement utilisées pour résoudre ce type de problème, en l'occurrence la méthode de Gauss-Newton et la méthode de Levenberg-Marquardt [LOW 91].

En ce qui nous concerne, nous avons adopté la méthode élaborée par [LU 00] et al qu'ils ont appelée algorithme d'itération orthogonale ("Orthogonal Iteration -OI- algorithm"). A l'inverse des méthodes classiques qui sont utilisées pour résoudre les problèmes d'optimisation d'une manière générale, l'algorithme d'itération orthogonale exploite judicieusement la structure spécifique du problème de recalage 3D. De plus, il donne des résultats très précis et converge globalement d'une manière assez rapide ce qui fait de lui un outil très intéressant pour les applications temps réel. Cet algorithme est détaillé dans la section suivante.

D.2 Algorithme d'itération orthogonale (IO)

Pour estimer la pose de l'objet, cet algorithme utilise une fonction d'erreur appropriée définie dans l'espace de l'objet qu'il réécrit sous une forme particulière

afin qu'elle admette une itération basée sur la solution du problème classique de l'estimation de la pose 3D/3D appelé problème de l'orientation absolue ("absolute orientation problem"). Comme l'algorithme de l'IO dépend entièrement de l'orientation absolue, nous allons tout d'abord décrire ce problème ainsi que la solution proposée pour le résoudre.

D.2.1 Problème de l'orientation absolue

Le problème de l'orientation absolue peut être posé comme suit : Supposons que les coordonnées 3D des points q_i définis dans l'espace caméra puissent être reconstruits physiquement (par exemple, par un scanner) ou par calcul (en utilisant la stéréovision), alors pour chaque point observé, nous pouvons écrire :

$$q_i = R.p_i + T$$

Le calcul de l'orientation absolue revient à déterminer R et T à partir des paires d'appariement (q_i, p_i) . Avec au moins 3 points de référence non colinéaires, R et T peuvent être obtenues comme une solution du problème des moindres carrés suivant :

$$\min_{R,T} \sum_{i=1}^n \|R.p_i + T - q_i\|^2 \quad \text{tel que } R.R^t = I \quad (\text{D.8})$$

Ce problème des moindres carrés avec contrainte (la contrainte d'orthogonalité de la matrice de rotation) peut être résolu en utilisant les quaternions [HOR 87b] ou bien la décomposition en valeurs singulières (SVD) [ARU 87]. Nous allons détailler ci dessous, la solution par SVD que nous allons utiliser par la suite dans l'algorithme IO.

Soient $\{p_i\}$ et $\{q_i\}$ les deux ensembles de points mis en correspondance, et soit :

$$\bar{p} = \frac{1}{n} \cdot \sum_{i=1}^n p_i \quad \bar{q} = \frac{1}{n} \cdot \sum_{i=1}^n q_i \quad (\text{D.9})$$

leurs centres de gravité respectivement. On définit alors les points :

$$p'_i = p_i - \bar{p} \quad q'_i = q_i - \bar{q} \quad (\text{D.10})$$

et la matrice

$$M = \sum_{i=1}^n q'_i p_i{}^t \quad (\text{D.11})$$

d'inter-covariance entre les points $\{p_i\}$ et les points $\{q_i\}$. Il est démontré dans [HOR 88] que si R^* et T^* minimisent l'équation (D.8) alors elles satisferont :

$$R^* = \arg \max_R \text{trace}(R^t M) \quad (\text{D.12})$$

$$T^* = \bar{q} - R^* \cdot \bar{p} \quad (\text{D.13})$$

Soit (U, Σ, V) la décomposition en valeurs singulières de la matrice M , tel que $U^t M V = \Sigma$, alors, la solution de l'équation (D.8) est donnée par :

$$R^* = V U^t \quad (\text{D.14})$$

Notons que la translation optimale est entièrement définie par la rotation optimale et que les informations nécessaires pour trouver cette rotation sont contenues dans la matrice M définie par l'équation (D.11). Ainsi, seule la position relative des points 3D par rapport à leurs centres de gravité est significative dans la détermination de la matrice de rotation optimale.

D.2.2 Algorithme

L'algorithme "IO" traite le problème du recalage 3D en utilisant le vecteur erreur de colinéarité défini dans l'espace de l'objet :

$$e_i = (I - \hat{V}_i)(R \cdot p_i + T) \quad (\text{D.15})$$

où \hat{V}_i est la matrice de projection de la ligne de vue observée correspondant au point \hat{m}_i , elle est définie par :

$$\hat{V}_i = \frac{\hat{m}_i \cdot \hat{m}_i^t}{\hat{m}_i^t \cdot \hat{m}_i} \quad (\text{D.16})$$

Nous cherchons à déterminer R et T qui minimisent la somme des erreurs quadratiques suivantes :

$$E(R, T) = \sum_{i=1}^n \|e_i\|^2 = \sum_{i=1}^n \left\| (I - \hat{V}_i)(R \cdot p_i + T) \right\|^2 \quad (\text{D.17})$$

Notons que toutes les informations fournies par les points image observés $\{\hat{m}_i\}$ sont incluses dans l'ensemble des matrices de projection $\{\hat{V}_i\}$. Puisque la fonction objective E est quadratique en T , pour une rotation R donnée, la valeur optimale de T peut être calculée d'une manière approchée par :

$$T(R) = \frac{1}{n} \left(I - \frac{1}{n} \sum_{j=1}^n \hat{V}_j \right)^{-1} \sum_{j=1}^n (\hat{V}_j - I) R \cdot p_j \quad (\text{D.18})$$

Ayant la translation optimale en fonction de R et en définissant :

$$q_i(R) = \hat{V}_i.(R.p_i + T(R)) \quad \text{et} \quad \bar{q}(R) = \frac{1}{n} \sum_{i=1}^n q_i(R) \quad (\text{D.19})$$

l'équation (D.17) peut être réécrite comme suit :

$$E(R) = \sum_{i=1}^n \|R.p_i + T(R) - q_i(R)\|^2 \quad (\text{D.20})$$

cette équation ressemble à l'équation (D.8) qui définit le problème de l'orientation absolue. Malheureusement, dans ce cas on ne peut pas déterminer la matrice R directement à partir d'une simple estimation de la matrice d'inter-covariance entre $\{p_i\}$ et $\{q_i(R)\}$, car la matrice :

$$M(R) = \sum_{i=1}^n q'_i(R).p_i^t \quad \text{où} \quad p'_i = p_i - \bar{p} \quad \text{et} \quad q'_i(R) = q_i(R) - \bar{q}(R) \quad (\text{D.21})$$

dépend de R , est déterminée en résolvant le problème de l'orientation absolue suivant:

$$R^{(k+1)} = \arg \min_R \sum_{i=1}^n \left\| R.p_i + T - \hat{V}_i.q_i^{(k)} \right\|^2 = \arg \max tr(R^t M(R^{(k)})) \quad (\text{D.22})$$

où l'ensemble de points $\hat{V}_i.q_i^{(k)}$ est considéré comme une hypothèse de l'ensemble de points q_i de la scène dans l'équation (D.8). Sous cette forme, la solution pour $R^{(k+1)}$ est donnée par l'équation (D.14). Nous calculons ensuite l'estimée suivante ($k+1$) de la translation en utilisant l'équation (D.18), telle que :

$$T^{(k+1)} = T(R^{(k+1)}) \quad (\text{D.23})$$

et nous répétons le processus. La solution optimale R^* est obtenue lorsque l'équation (D.22) reste inchangée, en d'autres termes, lorsque R^* satisfait :

$$R^* = \arg \min_R \sum_{i=1}^n \left\| R.p_i + T - \hat{V}_i(R^*.p_i + T(R^*)) \right\|^2 \quad (\text{D.24})$$

L'avantage de cet algorithme est qu'il converge globalement vers l'optimum de l'équation (D.22) pour n'importe quel ensemble de points observés et pour tout point de départ $R^{(0)}$. La seule contrainte sur $R^{(0)}$ est qu'elle ne doit pas placer l'objet observé derrière la caméra.

Finalement, la transformation rigide qu'a subi l'objet pour passer de sa position initiale à sa position courante dans le référentiel monde est donnée par :

$$A = A_{ext}^{-1} \cdot A_{opt} \quad (D.25)$$

où

$$A_{ext} = \begin{pmatrix} R_{ext} & T_{ext} \\ 0 & 1 \end{pmatrix} \quad et \quad A_{opt} = \begin{pmatrix} R^* & T^* \\ 0 & 1 \end{pmatrix} \quad (D.26)$$

La matrice A_{ext} représente les paramètres extrinsèques de la caméra qui correspondent à la transformation rigide entre le repère monde et le repère caméra. Pour cet algorithme, il est nécessaire d'avoir au minimum quatre couples (points 2D , points 3D) non coplanaires pour retrouver les trois rotations et la translation permettant d'amener le modèle de l'objet dans le repère lié à la caméra.

Références bibliographiques

- [ABA 02] ABABSA F., Recalage Automatique D'objets 3D de Forme Libre En Utilisant la Clinométrie, *Thèse de Doctorat, Université D'Evry Val D'Essonne*, 2002.
- [ABD 98] ABDULKADER A. M., Parallel Algorithms for Labelled Graph Matching, *PhD thesis, Colorado School of Mines*, 1998.
- [AGG 96] AGGARWAL J., GHOSH J., NAIR D. et TAHA I., A Comparative Study of Three Paradigms for Object Recognition Bayesian Statistics, Neural Networks and Expert Systems, *Advances in Image Understanding, K.Bowyer and N.Ahuja, Eds, IEEE Computer Society Press*, 241-262, 1996.
- [ALC 69] ALCORN T. et HOGGAR C., Preprocessing of Data for Character Recognition, *Marconi Rev*, 32, 61-81, 1969.
- [ALL 98] ALLAN S. A., Reconnaissance d'Environnement et Navigation Réactive D'un Robot Mobile Autonome Par Réseaux de Neurones, *Thèse Université d'Evry Val d'Essonne*, 1998.
- [ALV 92] ALVAREZ L., GUICHARD F., LIONS P.-L. et MOREL J.-M., Axioms and Fundamental Equations of Images Processing, *CEREMADE, Univ. Paris-Dauphine, Rapport N° 9231*, 1992.
- [ALV 94] ALVAREZ L. et MOREL J., Formalization and Computational Aspects of ImageAnalysis, *Acta Numerica*, 1-59, 1994.
- [ALV 96] ALVAREZ L., Images and PDE's, *ICAOS'96, 12th International Conference on Analysis and Optimization of Systems*, Juin 1996.
- [ANA 87] ANADAN P., Measuring Visual Motion From Image Sequence, *Phd Dissertation and COINS Technical Report 87-21, University of Massachusetts, Amherst*, 1987.
- [ARC 81] ARCELLI C., Pattern Thinning by Contour Tracing, *Computer Graphics and Image Processing*, 17, 130-144, 1981.
- [ARU 87] ARUN K. S., HUANG T. S. et BLOSTEIN S. D., Least-Square Fitting of Two 3D Points Sets, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9, 698-700, 1987.
- [ASA 86] ASADA H. et BRADY M., The Curvature Primal Sketch, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8, 1, 2-14, 1986.
- [ASC 92] ASCHWANDEN P. et GUGGENBÜHL W., Experimental Results from a

- Comparative Study on Correlation-Type Registration Algorithms, *Robust Computer Vision, Förstner and Ruwiedel, Wichmann Publisher*, 268-289, 1992.
- [ATT 94] ATTALI D., BERTOLINO P. et MONTANVERT A., Using Polyballs to Approximate Shapes and Skeletons, *In 12th International Conference on Pattern Recognition*, 626-628, Jérusalem, Israël, October 1994.
- [ATT 95] ATTALI D., Squelettes et Graphes de Voronoi 2D et 3D, *Thèse de doctorat, Université Joseph Fournier Grenoble I*, 1995.
- [AYA 86] AYACHE N. et FAUGERAS O., HYPER: A New Approach for the Recognition and Positioning of Two-Dimensional Objects, *IEEE Trans. On Pattern Analysis and Machine Intelligence*, 8, 1, 44-54, 1986.
- [BAR 96] BARAT C., Modélisation D'un Système de Relevé Télémétrique Laser 3D Application À la Robotique, *Thèse de doctorat, Université d'Evry Val d'Essonne*, 1996.
- [BAS 94] BASIN D. A., A Term Equality Problem Equivalent to Graph Isomorphism, *Information Processing Letters*, 54, 61-66, 1994.
- [BEA 78] BEAUDET P., Rotational Invariant Image Operators, *Int. Conf. Pattern Recognition*, 579-583., 1978.
- [BIE 87] BIENENSTOCK E. et DER MALSBURG C. V., A Neural Network for Invariant Pattern Recognition, *Europhysics Letters*, 4, 1, 121-126, 1987.
- [BIS 95] BISHOP C., Neural Networks for Pattern Recognition, *Clarendon Press, Oxford*, 1995.
- [BIS 98] BISCHOF H. et LEONARDIS A., Robust Recognition of Scaled Eigen-images Through a Hierarchical Approach, *In Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 664-670, Santa Barabara, California, June 1998.
- [BLA 87] BLAKE A. et ZISSERMAN A., Visual Reconstruction, *MIT Press, Cambridge-MA*, 1987.
- [BLA 98] BLACK M. J. et JEPSON A. D., Eigen Tracking : Robust Matching and Tracking of Articulated Objects Using a View-Based Representation, *International Journal of Computer Vision*, 26, 1, 63-84, 1998.
- [BLU 64] BLUM H., A Transformation for Extracting New Descriptors of Shape, *Symposium on Models for the Perception of Speech and Visual Form*, November 1964.
- [BLU 67] BLUM H., A Transformation for Extracting New Descriptors of Shape, in Models for the Perception of Speech and Visual Form, (*W. Wathen-Dunn, ed.*), *Cambridge MA: MIT Press*, 1967.
- [BOI 92] BOISSONNAT J. D. et GEIGER B., Three Dimensional Reconstruction

- of Complex Shapes Based on the Delaunay Triangulation, *Technical Report No. 1697, INRIA*, May 1992.
- [BOR 86] BORGEFORS G., Distance Transformations in Digital Images, *Vision, Graphics, and Image Processing*, 34, 344-371, 1986.
- [BOR 96] BORGEFORS G., On Digital Distance Transforms in Three Dimensions, *Computer Vision and Image Understanding*, 64, 3, 368-376, November 1996.
- [BRA 94] BRANDT J. W., Convergence and Continuity Criteria for Discrete Approximations of the Continuous Planar Skeletons, *CVGIP : Image Understanding*, 59, 1, 116-124, 1994.
- [CAL 65] CALABI L., A Study of the Skeleton of Plane Figures, *Technical Report 60429, Sr-2, Parke Mathematical Laboratories*, December 1965.
- [CAM 98] CAMPS O. I., HUANG C. et KANUNGO T., Hierarchical Organization of Appearancebased Parts and Relations for Object Recognition, *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 685-691, Santa Barbara, California, June 1998.
- [CAN 86] CANNY J., A Computational Approach to Edge Detection, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8, 6, 679-698, 1986.
- [CEL 89] CELEUX G., DIDAY E. et AL, Classification Automatique Des Données, *Editorial Dunod, Paris*, 1989.
- [CEL 90] CELEUX G., Analyse Discriminante sur Variables Continues, *Éditeur Scientifique, INRIA, France.*, 1990.
- [CHE 96] CHEN F. et SU C., Vision - Based Automated Inspection System in Computer Integrated Manufacturing., *International Journal of Advanced Manufacturing Technology*, 11, 3, 206-213, 1996.
- [CHE 98] CHEN J. L. et STOCKMAN G. C., 3D Free-Form Object Recognition Using Indexing by Contour Features, *Computer Vision and Image Understanding*, 71, 3, 334-353, 1998.
- [CHR 95] CHRISTMAS W. J., KITTLER J. et PETROU M., Structural Matching in Computer Vision Using Probabilistic Relaxation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17, 8, 749-764, 1995.
- [COC 95] COCQUEREZ J.-P. et PHILIPP S., Analyse D'images : Filtrage et Segmentation, *Masson, Paris, France*, 1995.
- [CVE 97] CVETKOVI'Ć D., ROWLINSON P. et SIMI'Ć S., Eigenspaces of Graphs, *Cambridge University Press, Cambridge, United Kingdom*, 1997.
- [DAN 80] DANIELSSON P., Euclidean Distance Mapping, *Computer Graphics and Image Processings*, 14, 227-248, 1980.

- [DER 87] DERICHE R., Using Canny's Criteria to Derive a Recursively Implemented Optimal Edge Detector, *International Journal of Computer Vision*, 1, 2, 167-187, 1987.
- [DER 93] DERICHE R. et GIRAUDON G., A Computational Approach for Corner and Vertex Detection, *International Journal of Computer Vision*, 10, 101-124, 1993.
- [DER 95] DERICHE R. et FAUGERAS O., Les EDP En Traitement D'images et Vision Par Ordinateur, *Rapport de Recherche INRIA*, , 2697, Novembre 1995.
- [DER 96] DERICHE R., Techniques d'Extraction de Contours, *Technical Report, INRIA Sophia-Antipolis*, 1996.
- [DEU 68] DEUTSCH E., Preprocessing for Character Recognition, *.IEE NPL Comf. Pattern Recognition*, 179-190, 1968.
- [DEV 82] DEVIJVER P. et KITTLER J., Pattern Recognition: A Statistical Approach, *Prentice Hall, Englewood Cliffs*, 1982.
- [DHO 89] DHOME M., RICHTIN M., LAPRESTE J. et RIVES G., Determination of the Attitude of 3D Objects from a Single Perspective View, *IEEE Trans. On Pattern Analysis and Machine Intelligence*, 11, 12, 1265-1278, 1989.
- [DID 82] DIDAY E., LUMAIRE J., POUGET J. et TESTU F., *Eléments D'analyse de Données.*, Dunod , 82.
- [DOO 80] DOOB M., CVETKOVIC D. et SACHS H., Spectra of Graphs: Theory and Application, *Academic Press*, 1980.
- [DRE 82] DRECHLER L. et NAGEL H., On Selection of Critical Points and Local Curvature Extrema of Region Boundaries for Interframe Matching., *International Conference on Pattern Recognition*, 542-544, 1982.
- [EDI 95] EDINBAROUGH R. A., NAINAR R. A. et RADHAKRISHNAN P., Automated Visual Inspection of Geometric Tolerance in Circular Components., *International Journal of Advanced Manufacturing Technology*, 10, 6, 367-373, 1995.
- [ELS 92] ELSER P. V. D., MAINTZ J. A., POL E. et VIERGEVER M., Image Fusion Using Geometrical Features, *In Visualization in Biomedical Computing, Volume 1808 of SPIE Proceedings Series*, 172-186, 1992.
- [FAU 93] FAUGERAS O., Three Dimensional Computer Vision: A Geometric Viewpoint, *MIT Press*, 1993.
- [FIN 98] FINCH A. W., WILSON R. C. et HANCOCK E. R., Symbolic Graph Matching with the EM Algorithm., *Pattern Recognition*, 31, 11, 1777-1790, 1998.

- [FOG 99] FOGGIA P., SANSONE C., TORTORELLA F. et VENTO. M., Definition and Validation of a Distance Measure Between Structural Primitives., *Pattern Analysis and Applications*, 2, 3, 215-227, 1999.
- [FOR 87] FORTUNE S., A Sweepline Algorithm for Voronoi Diagrams, *Algorithmica* 2, 153-174, 1987.
- [GAG 99] GAGVANI N. et SILVER D., Parameter Controlled Volume Thinning, *Parameter Controlled Volume Thinning*, 61, 3, 149-164, May 1999.
- [GAR 79] GAREY M. et JOHNSON D., Computers and Intractability : A Guide to the Theory of NP-Completeness., *W. H. Freeman, New-York*, 1979.
- [GEM 84] GEMAN S. et GEMAN D., Stochastique Relaxation, Gibbs Distributions and Bayesian Restoration of Images, *IEEE Trans. Pattern Analysis, Machine Intel.*, 6, 721-741, November 1984.
- [GEN 82] GENNERY D., Tracking Known Three-Dimensional Objects, *Conf. American Association of Artificial Intelligence*, 13-17, 1982.
- [GEN 92] GENNERY D., Visual Tracking of Known Three-Dimensional Objects, *International Journal of Computer Vision*, 8, 243-270, 1992.
- [GEO 97] GEORGE P. L. et BOROUCAKI H., *Triangulation de Delaunay et Maillage : Applications Aux Éléments Finis*, Hermes, paris , 1997.
- [GEU 99] GEUSEBROEK J., SMEULDERS A., CORNELISSEN F. et GEERTS H., Segmentation of Tissue Architecture by Distance Graph Matching, *Cytometry*, 35, 1, 11-22, Jan 1999.
- [GON 90] GONG W. et BERTRAND G., A Simple Parallel 3-D Thinning Algorithm, *In 10th International Conference on Pattern Recognition*, Atlantic City, June 1990.
- [GOU 00] GOUET V., MONTESINOS P., PELÉ D. et DERICHE R., Evaluation Des Détecteurs de Points D'intérêts Pour la Couleur, *Rapport de Recherche, EMA/LGI2P, INRIA, CCETT, France*, 2000.
- [HAR 88] HARRIS C. et STEPHENS M., A Combined Corner and Edge Detector, *Proc 4th Alvey Vision Conf. Manchester, England*, 189-192, August 1988.
- [HAR 90] HARRIS C. et STENNET. C., Rapid, a Video Rate Object Tracker, *British Machine Vision Conference.*, 73-77, 1990.
- [HAR 99] HARIS K., EFSTRATIADIS S., MAGLAVERAS N., PAPPAS C., GOURASSAS J. et LOURIDAS G., Modelbased Morphological Segmentation and Labeling of Coronary Angiograms, *IEEE Transactions on Medical Imaging*, 18, 10, 1003-1015, Oct 1999.
- [HER 98] HERPERS R. et SOMMER G., Discrimination of Facial Regions Based on Dynamic Grids of Point Representations, *International Journal of Pattern Recognition and Artificial Intelligence*, 12, 4, 381-405, Jun 1998.
- [HIL 01] HILAGA M., SHINAGAWA Y., KOHMURA T. et KUNII T., Topology

- Matching for Fully Automatic Similarity Estimation of 3D Shapes, *ACM-SIGGRAPH 2001 Proceedings*, August 2001.
- [HIR 94] HIRSCHAUER P., LARZABAL P. et CLERGEOT. H., Parametric Estimation with Neural Networks: Taking the Non - Linearity Into Account for Backpropagation Initialization and Network Size., *SPRANN'94*, 540-545, Lille, 1994.
- [HOP 74] HOPCROFT J. E. et WONG J. K., Linear Time Algorithm for Isomorphism of Planar Graphs, *In Sixth ACM Symposium on Theory of Computing*, 1974.
- [HOR 87a] HORAND R., New Methods for Matching 3D Objects with Single Perspective Views, *IEEE Trans. On Pattern Analysis And Machine Intelligence*, 9, 3, 401-412, 1987.
- [HOR 87b] HORN B. K. P., Closed-Form Solution of Absolute Orientation Using Unit Quaternion, *Journal of Optical Soc. Am.*, 4, 629-642., 1987.
- [HOR 88] HORN B. K. P., HILDEN H. M. et NEGAHDARIPOUR S., Closed-Form Solution of Absolute Orientation Using Orthonormal Matrices, *Journal of Optical Soc. Am.*, 5, 127-135., 1988.
- [HOR 89] HORNIK K., M.STINCHCOMBE et WHITE H., Multilayer Feedforward Networks are Universal Approximators, *Neural Networks*, 2, 359-366, 1989.
- [HOR 90] HORAUD R., SKODAS T. et VEILLON F., Finding Geometric and Relation Structures in an Image., *Proceedings of the 1st European Conference on Computer Vision, Antibes France , Lecture Notes in Computer Science*, 374-384, April 1990, Springer-Verlag.
- [HOR 95] HORAUD R. et MONGA O., Vision Par Ordinateur, Outils Fondamentaux, *Hermes, Paris, France*, Juin 1995.
- [HUA 97] HUANG C., CAMPS O. I. et KANUNGO. T., Object Recognition Using Appearance Based Parts and Relations, *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 877-883, San Juan, Puerto Rico, June 1997.
- [HUA 98] HUANG C. et HUANG W., Sign Language Recognition Using Model-Based Tracking and a 3D Hopfield Neural Network, *Machine Vision and Applications*, 10, 5-6, 292-307, Apr 1998.
- [HUT 87] HUTTENLOCHER D. P. et ULLMAN. S., Object Recognition Using Alignment, *Proceedings First Int. Conf. On Computer Vision*, 102-111, 1987.
- [JOL 98] JOLION J., Multiresolution Contrast Based Detection of Interest Points, *Research report, INSA Lyon, France*, 1998.
- [JOL 01] JOLION J.-M., KROPATSCH W. G. et VENTO. M., Graph Based Representations, *In Third IAPR Int. Workshop on Graph Based Representations*, Italy 2001, ISBN: 88-7146-579-2.

- [JOS 94] JOSHI S., PONCE J., VIJAYAKUMAR B. et KRIEGMAN D. J., HOT Curves for Modeling and Recognition of Smooth Curved 3D Objects, *IEEE Conference on Computer Vision and Pattern Recognition*, 876-880, 1994.
- [JOS 97] JOSHI T., VIJAYAKUMAR B., KRIEGMAN D. et PONCE J., HOT Curves for Modeling and Recognition of Smooth Curved 3D Objects, *Image and Vision Computing*, 15, 479-498, 1997.
- [KÉG 02] KÉGL B. et KRZYŻAK A., Piecewise Linear Skeletonization Using Principal Curves, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 24, 1, 59-74, September 2002.
- [KHO 02] KHOO K. et SUGANTHAN. P., Evaluation of Genetic Operators and Solution Representations for Shape Recognition by Genetic Algorithms., *Pattern Recognition Letters*, 23, 13, 1589-1597, Nov 2002.
- [KIR 79] KIRKPATRICK D., Efficient Computation of Continuous Skeletons, *IEEE 20th Annual Symposium on Foundations of Computer Science*, 18-27, 1979.
- [KIR 90] KIRBY M. et SIROVICH L., Application of the Karhunen Loeve Procedure for the Characterization of Human Faces, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 12, 1, 103-108, January 1990.
- [KIT 82] KITCHEN L. et ROSENFELD A., Gray-Level Corner Detection, *Pattern Recognition Letters*, 95-102, 1982.
- [KIT 93] KITTLER J., CHRISTMAS W. J. et PETROU M., Probabilistic Relaxation for Matching Problems in Computer Vision, *IEEE Proceedings of the International Conference on Computer Vision (ICCV93)*, 666-673, 1993.
- [KIT 98] KITTLER J., HATEF M., DUIN R. et MATAS J., On Combining Classifiers, *IEEE Trans. on Pattern Analysis And Machine Intelligence*, 226-238, 1998.
- [KOE 84] KOENDERINK J., The Structure of Images, *Biol. Cybern*, 363-370, 1984.
- [KON 89] KONG T. et ROSENFELD A., Digital Topology: Introduction and Survey, *Computer Vision, Graphics, and Image Processing*, 48, 357-393, 1989.
- [KOT 00] KOTROPOULOS C., TEFAS A. et PITAS. I., Frontal Face Authentication Using Morphological Elastic Graph Matching, *IEEE Transactions on Image Processing*, 9, 4, 555-560, Apr 2000.
- [KRO 99] KROPATSCH W. G. et JOLION J.-M., Graph Based Representations, *In Second IAPR Int. Workshop on Graph Based Representations, Osterreichische Computer Gesellschaft, band 126 (isbn: 3-85403-126-2)*, 1999.
- [KRU 96] KRUMM J., Eigenfeatures for Planar Pose Measurement of Partially Occluded Objects, *In Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 55-60, San Francisco, California June 1996.

- [LEE 82] LEE D., Medial Axis Transformation of a Planar Shape, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 4, 363-369, 1982.
- [LEO 96] LEONARDIS A. et BISCHOF H., Dealing with Occlusions in the Eigenspace Approach, *In Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 453-458, San Francisco, California, June 1996.
- [LOA 99] LOAIZA H., Capteur de Vision Stéréoscopique Multiconfiguration Pour la Localisation D'un Robot Mobile, *Thèse Université d'Evry Val d'Essonne*, 1999.
- [LOW 87] LOWE D. G., Three-Dimensional Object Recognition from Single Two-Dimensional Images, *Artificial Intelligence*, 31, 3, 355-395, 1987.
- [LOW 91] LOWE D. G., Fitting Parametrized Three-Dimensional Models to Images, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13, 5, 441- 450, 1991.
- [LOW 92] LOWE D. G., Robust Model-Based Motion Tracking Through the Integration of Search and Estimation, 8, 113-122, 1992.
- [LU 00] LU C., HAGER G. D. et MJOLSNESS E., Globally Convergent Pose Estimation from Video, *Transactions on Pattern Analysis and Machine Intelligence*, 22, 610-622, 2000.
- [LUO 01] LUO B. et HANCOCK E. R., A Robust Eigendecomposition Framework for Inexact Graph Matching, *E. Ardizzone and V. Di Gesu, Editors, Proceedings 11th International Conference on Image Analysis and Processing*, 465-470, 2001.
- [MAC 02a] MACRINI D., SHOKOUFANDEH A., DICKINSON S., SIDDIQI K. et ZUCKER S., Spectral Methods for View-Based 3-D Object Recognition Using Silhouettes, *Proceedings, Joint IAPR International Workshop on Syntactical and Structural Pattern Recognition*, Windsor, ON, August 2002.
- [MAC 02b] MACRINI D., SHOKOUFANDEH A., DICKINSON S., SIDDIQI K. et ZUCKER S., View-Based 3-D Object Recognition Using Shock Graphs, *Proceedings 16 Th International Conference on Pattern Recognition*, 3, Quebec, August 2002.
- [MAR 80] MARR D. et HILDRETH E., Theory of Edge Detection, *Proceedings of Royal Society of London*, Volume 207, 181-217, 1980.
- [MAR 00] MARIANI R., Face Learning Using a Sequence of Images, *International Journal of Pattern Recognition and Artificial Intelligence*, 14, 5, 631-648, Aug 2000.
- [MAS 01] MASSARO A. et PELILLO M., A Linear Complementarity Approach to Graph Matching, *In Proc. Of GbRŠ01, 3rd IAPR TC15 Int. Workshop on Graph Based Representations*,, 160-169, 2001, ISBN: 88 7146 579-2.
- [MED 87] MEDIONI G. et YSUMOTO Y., Detection and Curve Representation

- Using Cubic B-Splines., *Computer Vision, Graphics and Image Processing*, Volume 39, 267-278., 1987.
- [MER 01] MERAD D., TRIBOULET J., CHEDEBOIS A. et MALLEM M., Diagnostic Assistance Using Oriented Vision System for High Velocity Tooling Machines, *IEEE ETFA 2001 8th International Conference on Emerging Technologies and Factory Automation*, 2, 791-794, Oct., 15-18 2001, Antibes, France.
- [MER 03a] MERAD D., LELANDAIS S., MALLEM M. et TRIBOULET J., Adaptive Vision System for High Velocity Tooling Machines, *ISSPA 2003, International Symposium on Signal Processing and its Applications*, 1-4 July 2003, Paris, France.
- [MER 03b] MERAD D., LELANDAIS S., TRIBOULET J. et MALLEM M., Automatic Object Recognition for an Augmented Reality Vision System, *Informatica*, March 17-22 2003.
- [MER 04a] MERAD D., KHEZAMI N., MALLEM M. et OTMANE S., A Skeleton Based Method for Efficient 3D Object Localization: Application to Teleoperation, *1st International Conference on Informatics in Control, Automation and Robotics (ICINCO 2004)*, 25-28 Août 2004, Setubal (Portugal).
- [MER 04b] MERAD D., KHEZAMI N., OTMANE S. et MALLEM M., A 3D Free Form Object Localization Using Skeletons: Application to Teleoperation, *16th European Conference on Artificial Intelligence (ECAI 2004)*, Août 2004.
- [MER 04c] MERAD D., LELANDAIS M. M. A. et TRIBOULET J., A Skeleton Based Method for 3D Free-Form Object Recognition, *International Conference on Intelligent Manipulation and Grasping*, July 1-2 2004., Genova - Italy.
- [MER 04d] MERAD D., MALLEM M. et LELANDAIS S., Skeletonization of Two-Dimensional Regions Using Hybrid Method, *The 12th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision.*, February 2 - 6 2004, Plzen , Czech Republic.
- [MER 04e] MERAD D., TRIBOULET J., LELANDAIS S. et MALLEM M., Méthode Robuste de Squelettisation Par Carte de Distances et Diagramme de Voronoi, *SETIT*, 15-20 Mars Sousse, Tunisie 2004.,
- [MIG 94] MIGUET S. et MARION-POTY V., A New 2-D and 3-D Thinning Algorithm Based on Successive Border Generations, *In 4th Discrete Geometry for Computer Imagery*, 195-206, Grenoble, September 1994.
- [MIL 93] MILGRAM M., *Reconnaissance Des Formes: Méthodes Numériques et Connexionnistes*, Armand colin , 1993.
- [MOD 92] MODAYUR B., SHAPIRO L. et HARALICK R., Visual Inspection of

- Machined Parts, *Proceedings of 1992 IEEE Computer Society Conference On Computer Vision and Pattern Recognition*, 393-398, 1992.
- [MOG 96] MOGANTI M., ERCAL F., DAGLI C. et S.TSUNEKAWA, Automatic PCB Inspection Algorithms : A Survey, *Computer Vision and Image Understanding*, 63, 2, 287-313, 1996.
- [MOK 92] MOKHTARIAN F. et MACKWORTH A. K., A Theory of Multiscale, Curvature-Based Shape Representation for Planar Curves, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14, 8, 789-805, 1992.
- [MOK 95] MOKHTARIAN F., Silhouette-Based Isolated Object Recognition Through Curvature Scale Space, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17, 5, 539-544, 1995.
- [MON 68] MONTANARI U., A Method for Obtaining Skeletons Using Quasi-Euclidean Distance, *JACM*, 15, 600-624, 1968.
- [MOR 77] MORAVEC H., Towards Automatic Visual Obstacle Avoidance", *Proc Int. Joint Conf. Artificial Intelligence, Cambridge, UK*, August 1977.
- [MOR 80] MORAVEC H. P., Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover, *Phd Thesis, Stanford University*, 1980.
- [MOR 96] MORON V., Mise En Correspondance de Données 3 D Avec un Modèle CAO: Application À L'inspection Automatique, *Thèse de Doctorat À L'INSA de Lyon*, 9 Décembre 1996.
- [MUN 96] MUNDY J., LIU A., PILLOW N., ZISSERMAN A., ABDALLAH S., UTCKE S., NAYAR S. et ROTHWELL C., An Experimental Comparison of Appearance and Geometric Model Based Recognition, *J. Ponce, A. Zisserman, and M. Hebert, Editors, Object Representation in Computer Vision II*, 247-269, SpringerVerlag, Cambridge, U.K 1996.
- [MUR 94] MURASE H. et NAYAR S. K., Illumination Planning for Object Recognition Using Parametric Eigenspaces, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 16, 12, 1219-1227, 1994.
- [MUR 95] MURASE H. et NAYAR S. K., Visual Learning and Recognition of 3D Objects from Appearance, *International Journal of Computer Vision*, 14, 5-24, 1995.
- [NAF 97] NAF M., SZEKELY G., KIKINIS R. et ET O. KUBLER M. E. S., Voronoï Skeletons and their Usage for the Characterization and Recognition of 3D Organ Shape, *Computer Vision and Image Understanding*, 66, 2, 147-161, 1997.
- [NIE 94] NIESSEN W., ROMERY B. T. H., FLORACK L., SALDEN A. et VERGEVER. M., Nonlinear Diffusion of Scalar Images Using Well-Posed Differential Operators., *IEEE, Editor, IEEE Conf. On Comp. Vision and Pattern Recognition*, 92-120, June 1994., Seattle.

- [NOB 88] NOBLE J., Finding Corners, *Image and Vision Computing*, Vol 6, 121-128., May 1988.
- [OGN 92] OGNIWICZ R. et ILG M., Voronoï Skeletons: Theory and Applications, *In Computer vision and Pattern Recognition*, 63-69, 1992, IEEE Computer Society Press.
- [OHB 97] OHBA K. et IKEUCHI K., Detectability, Uniqueness, and Reliability of Eigen Windows for Stable Verification of Partially Occluded Objects, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 19, 9, 1043-1048, September 1997.
- [PAV 77] PAVLIDIS T., Structural Pattern Recognition, *Springer-Verlag, New-York*, 1977.
- [PAV 80] PAVLIDIS T., A Thinning Algorithm for Discrete Binary Images, *Computer Graphics and Image Processing*, 13, 142-157, 1980.
- [PER 90] PERONA P. et MALIK J., Scale-Space and Edge Detection Using Anisotropic Diffusion, *IEEE Trans. Pattern. Analysis and Machine Intelligence*, 12, 7, 629-639, July 1990.
- [PER 91] PERROTT C. et HAMEY L., Objet Recognition, A Survey of the Literature, *Macquarie Computing Reports N^o. 91-0065C*, 1991.
- [PER 94] PERONA P., SHIOTA T. et MALIK J., Anisotropic Diffusion, *Kluwer Academic In Bart M. Ter Haar Romeny Editor*, 229-254, 1994.
- [PON 89] PONCE J. et KRIEGMAN D. J., On Recognizing and Positioning Curved 3D Objects From Image Contours, *In IEEE Workshop on Interpretation of 3D Scenes*, 61-67, Austin, Texas, November 1989.
- [PRI 99] PRIETO. F., Métrologie Assistée Par Ordinateur : Apport Des Capteurs 3D Sans Contact, PhD thesis, Institut National Des Sciences Appliquées Lyon, 1999.
- [RAD 99] RADOI E., Contribution À la Reconnaissance Des Objets 3D À Partir de Leur Signature Electromagnétique, *Thèse de Doctorat de l'Université de Bretagne Occidentale*, 1999.
- [RAO 97] RAO R., Dynamic Appearance Based Recognition, *In Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 540-546, San Juan, Puerto Rico, June 1997.
- [REY 77] REYNER S. W., An Analysis of a Good Algorithm for the Subtree Problem., *SIAM Journal of Computing*, 6, 4, 730-732, 1977.
- [RIL 64] RILEY J. et CALABI L., Certain Properties of Circles Inscribed in Simple Closed Curves, *Technical Report 59281, Park Math. Lab. Inc., One River Road, Carlisle, Massachusetts*, 1964.
- [RIP 96] RIPLEY B. D., Pattern Recognition and Neural Networks, *Cambridge, Cambridge University Press*, 1996.

- [RIV 02] RIVIERE D., MANGIN J., PAPADOPOULOS D., MARTINEZ J., FROUIN V. et REGIS J., Automatic Recognition of Cortical Sulci of the Human Brain Using a Congregation of Neural Networks, *Medical Image Analysis*, 6, 2, 77-92, Jun 2002.
- [SCH 89] SCHMITT M., Some Examples of Algorithms Analysis in Computational Geometry by Means of Mathematic Morphology Techniques, *In J. D. Boissonnat and J. P. Laumond, Editors, Lecture Notes in Computer Science, Geometry and Robotics*, 391, 225-246, Springer Verlag, Berlin 1989.
- [SCH 93] SCHMITT M. et MATTIOLI J., Morphologie Mathématique, *Masson*, 1993.
- [SCH 00] SCHÜPP S., Prétraitement et Segmentation D'images Par Mise En Oeuvre de Techniques Basées sur Les Équations Aux Dérivées Partielles : Application En Imagerie Microscopique Biomédicale, 2000, Thèse de doctorat à l'université de CAEN.
- [SHA 75] SHAMOS M., Geometric Complexity, *Proc. 7th ACM Symp. On Theory of Computing*, 224-233, 1975.
- [SHA 99] SHAHEEN M., Reconnaissance D'objets Polyédriques À Partir D'une Image Vidéo Pour la Téléopération, PhD thesis, Université Evry Val D'Essonne, 1999.
- [SHE 86] SHEN J. et CASTAN S., An Optimal Linear Operator for Edge Detection, *Proceedings Computer Vision and Patern Recognition, Miami*, 109-114, June 1986.
- [SHO 01] SHOKOUFANDEH A. et DICKINSON S., A Unified Framework for Indexing and Matching Hierarchical Shape Structures., *In Proceedings, 4th International Workshop on Visual Form, Capri, Italy, May 2001*.
- [SID 98] SIDDIQI K., SHOKOUFANDEH A., DICKINSON S. et ZUCKER S., Shock Graphs and Shape Matching, *Proceedings, IEEE International Conference on Computer Vision*, 222-229, Jan 1998.
- [SID 99] SIDDIQI K., SHOKOUFANDEH A., DICKINSON S. et ZUCKER S., Shock Graphs and Shape Matching, *International Journal of Computer Vision*, 30, 1-24, 1999.
- [SIN 97] SINGH M., CHATTERJEEAND A. et CHAUDHURY. S., Matching Structural Shape Descriptions Using Genetic Algorithms., *Pattern Recognition*, 30, 9, 1451-1462, 1997.
- [SMI 95] SMITH S. et BRADY J., A New Approach to Low Level Image Processing, *Report, Robotics Research Group Department of Engineering Science, Oxford University*, 1995.

- [SUG 98] SUGANTHAN P. et YAN H., Recognition of Handprinted Chinese Characters by Constrained Graph Matching, *Image and Vision Computing*, 16, 3, 191-201, Mar 1998.
- [SUN 03] SUNDAR H., SILVER D., GAGVANI N. et DICKINSON S., Skeleton Based Shape Matching and Retrieval, *Proceedings, Shape Modelling and Applications Conference, SMI 2003*, Seoul, Korea, May 2003.
- [THI 94] THIEL E., Les Distances de Chanfrein En Analyse D'images: Fondements et Applications., Settembre 1994., Thèse de Doctorat, Université Joseph Fourier - Grenoble I,.
- [TIK 77] TIKHONOV A. et ARSENIN V., Solution of Ill-Posed Problems, *Winston and Wiley*, 1977.
- [TRI 01] TRIBOULET J., SHAHEEN M., MALLEM M., FIORONI C. et PASSE-MARD J., Finding 3D Polyhedral Object Attitude Using a Virtual Model for Industrial Machining, *ETFA*, 57-65, October 2001.
- [TRU 97] TRUCCO E., UMASUTHAN M., WALLACE A. et ROBERTO V., Model Based Planning of Optimal Sensor Placements for Inspection., *IEEE Transactions on Robotics and Automation*, 13, 2, 182-194, April 1997.
- [TSA 81] TSAO Y. et FU K., Parallel Thinning Algorithm for 3-D Pictures, *Computer Graphics and Image Processing*, 17, 315-331, 1981.
- [TUR 91a] TURK M. et PENTLAND A., Eigenfaces for Recognition, *Journal of Cognitive Neuroscience*, 3, 1, 71-86, January 1991.
- [TUR 91b] TURK M. et PENTLAND A., Recognition Using Eigenfaces, *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 586-591, 1991.
- [VAP 71] VAPNIK V. et CHERVONENKIS A. Y., On the Uniform Convergence of Relative Frequencies of Events to their Probabilities, *Theory of Probability and its Applications*, 2, 264-280, 1971.
- [VER 98] VERGNE E., Contribution À la Localisation Absolue D'un Robot Mobile Pour L'exploration d'Environnements Inconnus, *Thèse Université d'Evry Val d'Essonne*, 1998.
- [WHE 95] WHEELER M. D. et IKEUCHI K., Sensor Modelling, Probabilistic Hypothesis Generation, and Robust Localization for Object Recognition, *IEEE Trans. On Pattern Analysis and Machine Intelligence*, 17, 3, 252-265, 1995.
- [WIL 97] WILSON R. C. et HANCOCK E. R., Structural Matching by Discrete Relaxation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19, 6, 634-698, 1997.
- [WIL 99] WILLIAMS M., WILSON R. C. et HANCOCK E. R., Deterministic Search for Relational Graph Matching, *Pattern Recognition*, 32, 7, 1255-1271, Jul 1999.

- [WOR 01] WORTHINGTON P. L. et HANCOCK. E. R., Object Recognition Using Shape-from-Shading, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23, 5, 535-542, 2001.
- [ZER 90] ZERUBIA J. et CHELLAPA R., Mean Field Approximation Using Compound Gauss-Markov Random Field for Edge Detection and Image Restoration, *ICASSP*, April 1990.