



HAL
open science

Reconnaissance d'objets polyédriques par indexation dans une base de modèles

Humberto Sossa

► **To cite this version:**

Humberto Sossa. Reconnaissance d'objets polyédriques par indexation dans une base de modèles. Modélisation et simulation. Institut National Polytechnique de Grenoble - INPG, 1992. Français. NNT: . tel-00341784

HAL Id: tel-00341784

<https://theses.hal.science/tel-00341784>

Submitted on 26 Nov 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THESE

présentée par

Humberto SOSSA

pour obtenir le titre de DOCTEUR

de l'INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

(Arrêté ministériel du 23 novembre 1988)

(Spécialité: Informatique)

**RECONNAISSANCE D'OBJETS
POLYEDRIQUES PAR INDEXATION DANS
UNE BASE DE MODELES**

Date de soutenance: 9 décembre 1992

Composition du jury:

Président: M. Christian LAUGIER

Rapporteurs: M. Henri MAITRE

M. Michel DHOME

Examineurs: M. Radu HORAUD

M. Adolfo GUZMAN

Thèse préparée au sein du laboratoire LIFIA - IMAG

Remerciements

Cette thèse est pour moi l'occasion de remercier tous ceux qui m'ont conseillé, aidé et consacré du temps pour amener à bon terme ce travail, et je tiens à exprimer mes remerciements et ma plus vive gratitude à :

Mon pays, le Mexique qui à travers le financement du CONACYT, a rendu possible mon séjour en France et la réalisation de cette thèse,

L'INPG et le laboratoire LIFIA de m'avoir accueilli durant ces trois années.

Le CINVESTAV pour son soutien,

Monsieur Christian Laugier, directeur de recherche de l'INRIA, pour avoir pris en charge la gestion de mon séjour au LIFIA et pour avoir accepté de présider mon jury,

Monsieur Radu Horaud, directeur de recherche au CNRS de m'avoir encadré et orienté sur le plan universitaire tout au long de cette recherche. Je lui suis particulièrement reconnaissant pour ses conseils, ses encouragements et sa confiance. Enfin, je suis doublement heureux de sa participation à ce jury,

Monsieur Henri Maitre, professeur de l'Ecole Nationale Supérieure des Télécommunications et Monsieur Michel Dhome, chargé de recherche au Laboratoire d'Electronique de l'Université Blaise Pascal qui m'ont fait l'honneur d'être rapporteurs de cette thèse,

Monsieur Adolfo Guzmán Arenas, président de Software-Pro International qui m'a fait l'honneur d'accepter d'être membre de mon jury.

J'exprime de chaleureux remerciements à toute l'équipe **Voir** pour avoir mis à ma disposition les matériels et logiciels indispensables à ce travail et pour les discussions fructueuses que nous avons eues ensemble.

Je tiens à citer la collaboration sympathique et efficace de Luce Morin et Long Quan pour tous les moments qu'ils ont consacrés à discuter ou relire ce mémoire.

Que mes frères et sœurs en Christ, ceux des villes de Mexico et Gières trouvent ici l'expression de ma profonde reconnaissance pour m'avoir soutenu sans cesse dans la prière.

Je remercie Monsieur Jacky Laureiro pour la fabrication des objets en bois utilisés dans les expériences de ce travail.

J'adresse un immense merci à mes parents qui ont su me comprendre, me faire grandir, me permettre d'atteindre mes buts en me donnant leur amour et en me fournissant l'aide dont j'ai pu avoir besoin.

Je tiens à joindre à mes remerciements, ma belle famille qui dans les moments difficiles ont toujours su me soutenir.

Je remercie plus particulièrement mon épouse Flor pour sa patience, sa compréhension et son soutien dans toutes circonstances par ses prières, ses conseils et ses encouragements.

Enfin, je rends grâces à Dieu par lequel toutes choses sont possibles : à Dieu le Père qui m'a donné la vie à travers son Fils Unique Jésus Christ, à Dieu le Fils, Jésus Christ, qui intercède toujours pour moi auprès du Père et à Dieu, l'Esprit Saint qui est toujours là pour me conduire en toutes circonstances.

Sommaire

1	Introduction	15
1.1	La vision par ordinateur	15
1.1.1	Le problème de la reconnaissance d'objets 3D	16
1.2	Le problème à résoudre	18
1.3	L'objectif	18
1.4	L'idée générale	19
1.5	Situation de départ	21
1.6	Notre approche de description des objets	21
1.7	Questions relatives à la recherche	23
1.8	La contribution de la recherche	23
1.9	Organisation du rapport	24
2	Résumé des travaux précédents	27
2.1	Résumé des méthodes de description de formes	28
2.1.1	Description des modèles	28
2.1.2	Méthodes utilisant des caractéristiques visuelles spécifiques	29

2.1.3	Méthodes utilisant une représentation 3D	29
2.1.4	Méthodes de représentation à partir d'une seule vue 2D	29
2.1.5	Méthodes de représentation à partir de plusieurs vues 2D	30
2.2	Tour d'horizon des primitives utilisées en vision par ordinateur	30
2.2.1	Primitives à points 2D/3D ou à contours 2D/3D spécifiques	31
2.2.2	Groupes de points 2D/3D ou de courbes 2D/3D	31
2.2.3	Primitives de surface	32
2.2.4	Primitives volumiques	33
2.2.5	Résumé sur les primitives utilisées en vision par ordinateur	35
2.3	Résumé des systèmes de reconnaissance utilisant un mécanisme d'indexation	36
2.3.1	Nevatia et Binford (1977)	36
2.3.2	Kalvin, Shonberg, Schwartz et Sharir (1986)	37
2.3.3	Lamdan et Wolfson (1988)	39
2.3.4	Ettinger (1988)	40
2.3.5	Stein et Medoni (1990)	42
2.3.6	Stein et Medoni (1991)	43
2.3.7	Résumé des systèmes de reconnaissance	44
3	La caractérisation des graphes	47
3.1	Rappels	48
3.1.1	Groupes	48
3.1.2	Théorie des matrices	51
3.1.3	Graphes et matrices associées	52
3.1.4	Fonctions matricielles généralisées	54
3.1.5	Polynômes généralisés	55
3.2	Le problème général de la caractérisation d'un graphe	55
3.2.1	Introduction	55
3.2.2	Enoncé du problème	56

3.2.3	Les différentes approches de solution	56
3.3	Le deuxième immanant	59
3.3.1	Définitions préliminaires	59
3.3.2	Résultats principaux	60
3.3.3	Caractérisation de la topologie d'un graphe	60
3.3.4	Un exemple	61
3.4	Caractérisation de l'apparence d'un graphe	63
3.5	Conclusion sur la méthode de caractérisation	71
4	La technique d'indexation choisie	73
4.1	Le problème	74
4.2	Une solution	75
4.3	Notre approche : hachage des graphes	76
4.4	La structure de la base de données choisie	77
4.5	Conclusions sur la méthode d'indexation choisie	79
5	Un système d'indexation de modèles	81
5.1	La modélisation d'un objet	81
5.1.1	De la vue caractéristique à la figure fondamentale	83
5.1.2	De la figure fondamentale à l'intersection de figures fondamentales	84
5.2	L'organisation de la base de modèles	85
5.3	Le mécanisme d'indexation	86
5.4	Résumé	89
6	Résultats d'expérimentation	91
6.1	Le processus hors ligne de construction de la base de modèles	92
6.1.1	Reduction du bruit	93
6.1.2	Décomposition du graphe en figures fondamentales	93
6.1.3	Evaluation des codes de hachage et stockage de l'information	93

6.1.4	Les modèles choisis	94
6.2	Le processus d'indexation	103
6.3	Un exemple détaillé	103
6.3.1	L'image d'entrée	104
6.3.2	Réduction du bruit	105
6.3.3	Division du graphe-image	107
6.3.4	Décomposition en figures fondamentales	109
6.3.5	Obtention des intersections de figures fondamentales	109
6.3.6	Evaluation des codes de hachage	110
6.3.7	Indexation dans la base de données	110
6.4	D'autres exemples	116
7	Conclusion	127
7.1	Résumé	127
7.2	Apports	128
7.3	Inconvénients et améliorations	129
7.4	D'autres améliorations	130

Liste des Figures

1.1	L'idée générale du système.	20
1.2	FEAGRO est un logiciel de vision bas niveau qui permet de traiter une image fournie par une carte de digitalisation. Voici un exemple d'une image traitée par ce système.	22
2.1	Les primitives classées par ordre de complexité.	31
3.1	Plusieurs paires de graphes non isomorphiques avec les mêmes ensembles de coefficients du polynôme caractéristique ont été découvertes pendant ces dernières années. Cela a amené les chercheurs à abandonner cette approche de caractérisation.	57
3.2	Le polynôme deuxième immanent favorise la caractérisation de graphes par rapport à sa contrepartie, le polynôme caractéristique. Pourtant, car ce polynôme est un compromis entre le polynôme permanent et le polynôme caractéristique il y a des exemples de graphes (avec une taille $n > 12$) avec les mêmes coefficients du polynôme d_2	59
3.3	(a) Un exemple de graphe à quatre nœuds et à trois arêtes. (b) sa matrice laplacienne associée et (c) l'ensemble des coefficients caractérisant sa topologie.	62
3.4	(a) Trois graphes ayant la même topologie mais non pas la même apparence. (b) Trois autres graphes ayant la même topologie et la même apparence.	63

3.5	(a) Un graphe à 11 nœuds et à 15 arêtes. (b) Deux de ses figures fondamentales. L'une d'ordre 1, l'autre d'ordre 2, toutes les deux obtenues à partir du nœud numéro 6.	65
3.6	Une figure fondamentale et ses trois BFF.	66
3.7	Les jonctions L, F et Y sont les trois types de jonctions considérés. La jonction de type T est considérée comme résultat du phénomène du chevauchement.	66
3.8	Les cotés "gauche" et "droite" d'une BFF sont déterminés par l'intermédiaire de son arête de direction.	66
3.9	Le code d'une BFF est obtenu en fonction des types de ses deux jonctions et du nombre d'arêtes à "gauche" et à "droite" de ces jonctions. (a) Tout d'abord, les BFF sont ordonnées dans l'ordre : LL, LF, LY, FL, FF, FY, YL, YF, YY et de façon ascendante en fonction du nombre d'arêtes à gauche et à droite des deux jonctions. (b) Ensuite, les BFF sont étiquetées suivant cet ordonnancement en éliminant en même temps les BFF résultats d'une opération de réflexion. Une étiquette suivie d'un "#" désigne une BFF éliminée.	68
3.10	Le code d'une BFF est un entier qui est une fonction des types de ses jonctions et du nombre d'arêtes à gauche et à droite de ces jonctions.	69
3.11	(a) Une figure fondamentale à 8 nœuds et à 9 arêtes, (b) ses trois BFF avec leurs codes, (c) sa matrice d'apparence locale correspondante et (d) l'ensemble des coefficients m_i du polynôme $d_2(M(G))$ caractérisant la FF.	70
4.1	La clé associée à chaque objet de la base de données est transformée en adresse d'un tableau de taille gérée par l'intermédiaire d'une fonction de hachage.	75
4.2	Le mécanisme d'indexation évalue le code de hachage d'un objet inconnu à fin de déterminer si cet objet se trouve dans le tableau de hachage ou pas.	76
4.3	La structure de la base de graphes.	78
5.1	Un exemple d'une vue caractéristique d'un objet simple en forme de "L" et quelques unes de ses figures fondamentales.	83
5.2	L'ensemble des IFF correspondant à l'ensemble des figures fondamentales de la vue caractéristique de l'objet en forme de "L".	85

5.3	La représentation d'un objet contient trois couches : une couche supérieure contenant une description à objet centré, une couche centrale où l'objet est décrit par une liste de vues caractéristiques et une couche inférieure où chaque vue caractéristique est décrite par une liste de figures fondamentales et une liste des intersections des figures fondamentales.	86
5.4	Un exemple d'une base de données simple. Elle est composée de trois objets, six VC, quatre FF et deux IFF. Seule les FF et les IFF sont stockés dans les tableaux de hachage.	87
5.5	Le mécanisme d'indexation prend en entrée un ensemble de figures fondamentales et leurs intersections. Il indexe ensuite la base de modèles et retourne la liste des modèles contenant les FF et IFF trouvées dans l'image.	88
6.1	Schéma du processus de construction de la base de modèles.	95
6.2	Les sept objets choisis pour former la base de modèles.	96
6.3	Pour chaque objet composant la base de modèles un nombre de vues caractéristiques sont choisies.	97
6.4	La vue choisie pour l'objet numéro 1.	97
6.5	La vue choisie pour l'objet numéro 2.	97
6.6	Les quatre vues choisies pour l'objet numéro 3.	98
6.7	Les deux vues choisies pour l'objet numéro 4.	98
6.8	Les huit vues choisies pour l'objet numéro 5.	99
6.9	Les deux vues choisies pour l'objet numéro 6.	100
6.10	Les deux vues choisies pour l'objet numéro 7.	100
6.11	Notre base expérimentale contenant les 20 vues appartenant aux 7 objets choisis.	101
6.12	Schéma du processus d'indexation.	102
6.13	(a) l'image de niveaux de gris. (b) l'image des contours et (c) l'image des lignes droites avec leurs jonctions.	104
6.14	Exemples des segments éliminés par le module de réduction du bruit.	104
6.15	(a) exemples des jonctions du type T éliminées. (b) exemples de segments colinéaires réduits à un seul segment.	106

- 6.16 (a) résultat du passage de l'image d'entrée par la première étape de réduction du bruit. (b) les T éliminées de l'image. (c) les métasegments trouvés. (d) en fin, l'image résultante du passage de l'entrée par la troisième étape de réduction du bruit. 107
- 6.17 (a) l'ensemble de segments considérés comme étant du bruit par notre système. (b) les deux graphes trouvés par notre système. Chaque graphe reçoit une étiquette. 108
- 6.18 Les deux nœuds indiqués par des cercles en blanc ne sont pas considérés par l'extracteur de FF pour générer les figures fondamentales respectives. 109
- 6.19 Les onze figures fondamentales appartenant au graphe "0". 111
- 6.20 Les seize figures fondamentales appartenant au graphe "1" 112
- 6.21 Notre base expérimentale contenant les 20 vues appartenant aux 7 objets choisis. Les vues dans la base de données qui matchent les objets inconnus sont marquées par l'étiquette correspondante. 115
- 6.22 (a) Une scène contenant un seul objet. (b) l'image des contours. (c) l'image des segments de droite. (d) premier passage de réduction du bruit. (e) élimination des T. (f) les métasegments trouvés. (g) réduction des graphes. (h) ensemble de segments considérés comment étant du bruit. (i) les bons graphes. 118
- 6.23 Notre base expérimentale contenant les 20 vues appartenant aux 7 objets choisis. Les vues dans la base de données qui matchent l'objet inconnu sont marquées par l'étiquette correspondante. 119
- 6.24 (a) Une scène contenant deux objets. (b) l'image des contours. (c) l'image des segments de droite. (d) premier passage de réduction du bruit. (e) élimination des T. (f) les métasegments trouvés. (g) réduction des graphes. (h) ensemble de segments considérés comment étant du bruit. (i) les bons graphes. 120
- 6.25 Notre base expérimentale contenant les 20 vues appartenant aux 7 objets choisis. Les vues dans la base de données qui matchent les deux objets inconnus sont marquées par l'étiquette correspondante. . 121
- 6.26 (a) Une scène contenant trois objets. Deux d'entre eux se cachent. (b) l'image des contours. (c) l'image des segments de droite. (d) premier passage de réduction du bruit. (e) élimination des T. (f) les métasegments trouvés. (g) réduction des graphes. (h) ensemble de segments considérés comment étant du bruit. (i) les bons graphes. . . 122

-
- 6.27 Notre base expérimentale contenant les 20 vues appartenant aux 7 objets choisis. Les vues dans la base de données qui matchent les trois objets inconnus sont marquées par l'étiquette correspondante. . 123
- 6.28 (a) Une scène contenant deux objets différents de ceux stockés dans la base d'objets. (b) l'image des contours. (c) l'image des segments de droite. (d) premier passage de réduction du bruit. (e) élimination des T. (f) les métasegments trouvés. (g) réduction des graphes. (h) ensemble de segments considérés comment étant du bruit. (i) les bons graphes. 124
- 6.29 Notre base expérimentale contenant les 20 vues appartenant aux 7 objets choisis. Les vues dans la base de données qui matchent le mieux les deux objets inconnus sont marquées par l'étiquette correspondante. 125
- 7.1 La caractérisation de propriétés comme le parallélisme, la colinéarité, et la symétrie permet de faire la différence entre ces deux objets. . . . 130

Liste des Tables

- 6.1 Voici la liste des vues contenues dans la base de données et leurs votes respectifs donnés par le mécanisme d'indexation en fonction des FF et des IFF du graphe "0". Les deux candidats choisis par le système qui ressemblent le plus à ce graphe sont indiqués par un "*" (5 ème colonne). 113
- 6.2 Voici la liste des vues contenues dans la base de données et leurs votes respectifs donnés par le mécanisme d'indexation en fonction des FF et des IFF du graphe "1". Les deux candidats choisis par le système qui ressemblent le plus à ce graphe sont indiqués par un "*" (5 ème colonne). 114

Chapitre 1

Introduction

1.1 La vision par ordinateur

L'homme est un être merveilleux qui pourrait être imaginé comme une machine douée d'impressionnantes possibilités la plupart d'entre elles encore inconnues. Depuis longtemps, il a cherché à comprendre son propre fonctionnement, notamment sa capacité à percevoir son environnement au moyen de ses cinq sens.

Parmi ceux-ci, la perception visuelle a attirée la curiosité des scientifiques depuis plusieurs siècles. Les contributions les plus importantes ont été faites d'abord par Newton (1704), qui a établi les bases pour les travaux modernes en vision des couleurs et ensuite par Helmholtz (1910), dont le traité en psychologie de la perception est encore d'actualité. D'autres apports plus récents ont été faits par Wertheimer (1912, 1923), Miles (1931), Wallach et O'Connell (1953) et Brindley (1970) (Marr [73]).

Grâce au développement de l'informatique, ces connaissances descriptives (neuropsychologie et psychologie de la perception) peuvent maintenant donner lieu à des tentatives pour doter des machines de capacités visuelles remplissant en quelque sorte la fonction de l'œil humain. Ceci permet à ces machines de comprendre leur environnement et de réaliser des tâches spécifiques en utilisant l'information perçue.

C'est ainsi que la Vision par Ordinateur (VO) est née.

La VO peut être définie comme suit : la vision par ordinateur est un processus de traitement de l'information, consistant à :

- **prendre** en entrée des données sensorielles provenant de capteurs visuels,
- **traiter** ensuite convenablement ces informations visuelles par un certain nombre de méthodes spécifiques et finalement,
- **produire** en sortie une *description symbolique* des données de l'entrée en termes d'objets et de relations entre ces objets.

La VO a été un champ de recherche très active et productive pendant les trois dernières décennies. Les premiers travaux importants ont été effectués par Roberts [90] qui en 1963 a développé un système qui prenait en entrée une image numérisée d'une scène contenant des objets polyédriques et produisait en sortie un dessin fil de fer "a line drawing" de la scène comme il apparaît depuis un point de vue spécifique. Ce travail traite des sujets de base tels que la recherche d'indices visuelles, la représentation d'objets et l'appariement. Depuis, des milliers des scientifiques se sont investis dans ce domaine de recherche.

Les domaines d'application de la VO sont très variés. Entre autres, nous pouvons citer quelques uns :

- la robotique,
- la reconnaissance de formes,
- l'analyse du mouvement,
- le diagnostique médical,
- la chimie et
- la perception télécommandée.

1.1.1 Le problème de la reconnaissance d'objets 3D

La reconnaissance d'objets 3D est un des problèmes principaux en VO. C'est une étape nécessaire pour beaucoup d'applications industrielles, telles que l'identification de pièces industrielles, l'automatisation des processus de manufacture. De plus elle offre la possibilité aux robots intelligents d'être équipés d'un système visuel à réalimentation.

La reconnaissance d'objets 3D en vision par ordinateur consiste à :

1. **identifier** un objet dans une scène à l'aide d'un modèle en établissant une correspondance entre l'objet et le modèle et
2. **obtenir** une information 3D utile telle que sa position et son orientation.

Dans le passé, des essais pour résoudre ce problème ont contribué au paradigme de calcul suivant. A partir d'une image d'entrée :

1. **obtenir** un ensemble d'indices visuels (primitives) cohérents, c'est-à-dire, des groupes de caractéristiques image (points, segments, surfaces, volumes...) provenant fortement d'un même objet.
2. **décrire** chaque objet à reconnaître en termes de ces indices (indices d'objet) et des relations entre ces indices.
3. **extraire** de l'information d'une image quelques indices d'image correspondant éventuellement aux indices d'objet.
4. **établir** une correspondance terme à terme (un appariement) entre un sous-ensemble d'indices d'image et un sous-ensemble d'indices d'objet.
5. **évaluer** la position et l'orientation de l'objet reconnu en fonction de l'appariement d'indices d'image avec ces indices d'objet.

Parmi ces cinq étapes, trois déterminent la compétence et la performance du système de reconnaissance : l'obtention des primitives, la description des objets en termes de primitives et le processus de la mise en correspondance.

Lorsque dans la base de données il n'y a qu'un seul modèle et lorsque le nombre des primitives décrivant ce modèle reste petit la question d'établir une correspondance (étape numéro 4 du paradigme) est, en général, résolue d'une manière satisfaisante lorsqu'on utilise des techniques de recherche exhaustives. D'autres implémentations du même paradigme considèrent les étapes 4 et 5 dans un processus unique qui prend la forme soit d'un graphe de recherche (Bolles et Cain [12]) ou d'un arbre de recherche (Ayache et Faugeras [4], Faugeras et Hebert [36], Bolles et Horaud [13], Horaud [54], Lowe [70], Grimson et Lozano-Pérez [44] et Pollard [88]). Cependant, si la complexité et le nombre des primitives composantes augmentent, ces implémentations éclatent car le processus de l'appariement est combinatoire en la complexité de calcul. Une solution pour affronter ce problème a été celle d'utiliser des contraintes (sous l'hypothèse d'objets rigides) de sorte que l'espace de recherche (la liste, le graphe ou l'arbre) est réduit autant que possible. Ainsi, n'importe quel processus (y compris celui de la recherche exhaustive) est réduit à visiter quelques centaines de nœuds au lieu de millions.

Néanmoins, si beaucoup d'objets au lieu d'un seul sont présents dans la base de modèles, les processus de recherche précédemment décrits éclatent à nouveau car la

complexité de calcul croît aussi avec le nombre d'objets (Grimson [43]). C'est pour cela que l'étape 4 du paradigme ci-dessus doit être décomposé davantage en :

- **indexation** de modèles et
- **appariement** ou vérification.

L'indexation de modèles peut être décrite brièvement comme suit : **étant donné un groupe d'indices d'image, extraire "rapidement" de la base de modèles les modèles contenant le groupe d'indices.**

1.2 Le problème à résoudre

Le problème qu'on se propose d'approcher dans cette recherche est celui de la **reconnaissance d'objets en présence d'une grande base de modèles. La question de l'indexation de modèles est alors l'obtention, à partir d'indices visuels d'une image, d'une liste restreinte à quelques modèles candidats à la reconnaissance.** On voudrait, répondre aux questions suivantes :

- Comment modéliser les objets 3D?
- Quelles primitives ou indices visuels utiliser pour décrire les objets et les modèles?
- Comment organiser les modèles obtenus dans la base? Autrement dit, étant donné un groupe d'indices visuels obtenus à partir d'une image, de quelle manière extraire **rapidement** de la liste de modèles ceux qui contiennent ce groupe d'indices?

1.3 L'objectif

L'objectif de cette recherche est de répondre aux questions posées ci-dessus au sujet de l'indexation dans une grande base de modèles par le développement d'un système qui décrit de façon automatique des objets 3D, utilise le même ensemble de primitives pour représenter les modèles et les objets de la scène et renvoyer une liste restreinte à quelques modèles candidats à la reconnaissance qui peuvent se trouver dans l'image d'entrée. Nous donnons, aussi, une série de solutions à d'autres problèmes liés rencontrés au cours de la recherche, notamment, celui de la réduction du bruit dans l'image et celui de la division d'un graphe. Nous présentons également des résultats satisfaisants pour une variété d'objets simples et moins simples, avec des scènes contenant un ou plusieurs objets se chevauchant éventuellement et même n'appartenant pas à la base de modèles.

1.4 L'idée générale

L'idée générale du système est la suivante (figure 1.1). Nous avons deux graphes. Le premier graphe est le résultat de la projection (de l'anglais *mapping*) du réseau des jonctions et des segments d'une image 2D sur le graphe. Le deuxième graphe est le résultat de la projection du réseau des jonctions et des segments d'une vue d'un objet 3D sur le graphe.

Le problème de l'isomorphisme de graphes étant NP-complet on le réduit au problème moins difficile qui consiste à comparer un sous-graphe de l'image avec un sous-graphe d'un modèle (figure 1.1 (b)).

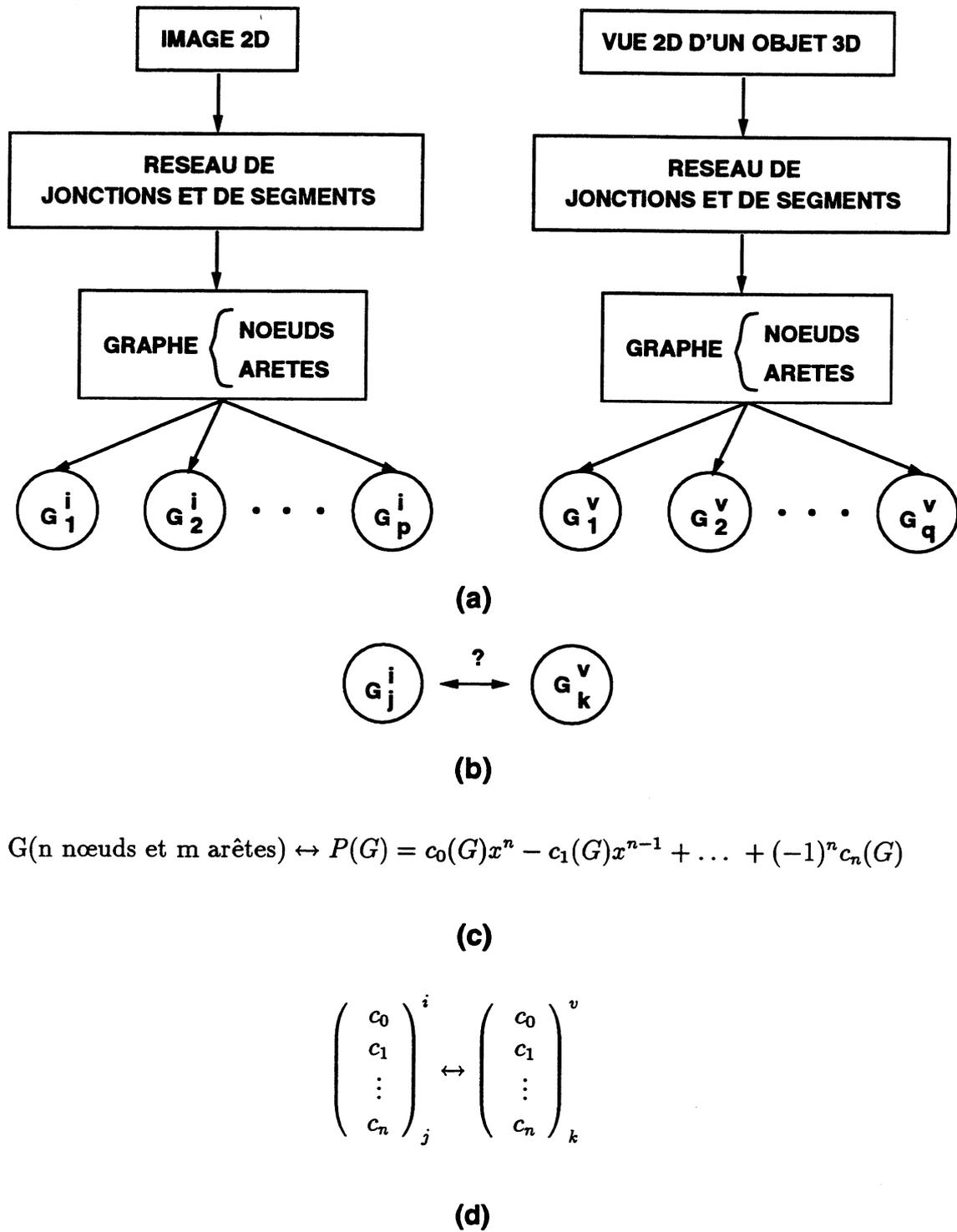
Pour un graphe G à n nœuds et à m arêtes nous proposons une approche pour caractériser le graphe G par l'ensemble des coefficients d'un polynôme (figure 1.1 (c)).

Par conséquent, le problème de comparer deux graphes est converti dans le problème de comparer deux ensembles de coefficients : les coefficients de deux polynômes (figure 1.1 (d)).

Nous essayons de répondre aux questions suivantes :

1. Y-a-t-il une correspondance unique entre le graphe et le polynôme?
2. Comment fait-on pour comparer rapidement les deux ensembles des coefficients? et
3. Comment fait-on pour organiser l'information (les données) dans la base de données?

Dans ce chapitre, nous commençons par décrire le type d'entrée qui sera utilisée par notre système. Nous présentons ensuite la description choisie. Puis, nous résumons la principale contribution de notre recherche. Finalement, nous exposons les grandes lignes de cette thèse.



$$G(n \text{ nœuds et } m \text{ arêtes}) \leftrightarrow P(G) = c_0(G)x^n - c_1(G)x^{n-1} + \dots + (-1)^n c_n(G)$$

Figure 1.1: L'idée générale du système.

1.5 Situation de départ

Voici l'environnement qui détermine de façon fondamentale les données de notre problème.

On dispose d'une caméra, d'une carte de digitalisation et d'un logiciel de vision bas niveau. La carte transforme le signal électronique de la caméra en image de niveau de gris.

A partir de cette image, le logiciel FEAGRO¹ permet de calculer au moyen d'un filtre de Deriche (Deriche [28]), le gradient de la fonction qui à chaque point associe son niveau de gris, de chercher les contours, c'est-à-dire les courbes de gradient maximal, puis de faire une approximation polygonale de ces contours. Enfin, le logiciel permet de calculer divers groupements perceptifs (jonctions, groupes de segments parallèles, parallélogrammes, symétries ...). Il comprend aussi d'autres fonctions comme la visualisation des images d'entrée et des divers résultats. La figure 1.2 montre un exemple d'image traitée par FEAGRO : de haut en bas et de droite à gauche, on a l'image d'origine, les contours extraits, l'approximation polygonale des contours et les intersections entre les segments de l'approximation polygonale.

Les données dont nous partons, que nous appellerons **images** sont le résultat obtenu par le logiciel de vision de bas niveau (figure 1.2 en bas à droite). Une telle image est constituée de segments de droites et de points appelés **jonctions**. Les jonctions sont à la fois les extrémités des segments, les points d'intersection de ces segments entre eux et le point où une arête est interrompue par une surface occultante.

Cette famille de segments de droites et de jonctions peut être considérée comme l'ensemble des nœuds et d'arêtes d'un graphe planaire. Ce choix a, bien entendu, ses inconvénients mais aussi ses avantages. D'une part, il est vrai que l'information métrique dans l'image est négligée mais, d'autre part l'information concernant la structure est présente et elle est aussi très riche pour décrire des objets. C'est le choix que nous allons adopter pour la suite.

1.6 Notre approche de description des objets

Un des sujets de recherche très répandu pour la modélisation d'objets 3D en vision par ordinateur est celui de l'utilisation de structures de graphes comme modèles de vision. Engelbrecht et Wahl [31], par exemple, représentent des objets polyédriques par des graphes attribuables dont les nœuds représentent des jonctions et dont les arêtes représentent des segments diédriques. Leur système reconnaît ensuite un objet par la mise en correspondance (un isomorphisme) entre le graphe représentant

¹FEAGRO (FEAture GROUping) est un logiciel de vision bas niveau développé au sien du LIFIA (Laboratoire d'Informatique Fondamentale et d'Intelligence Artificielle de l'INPG).

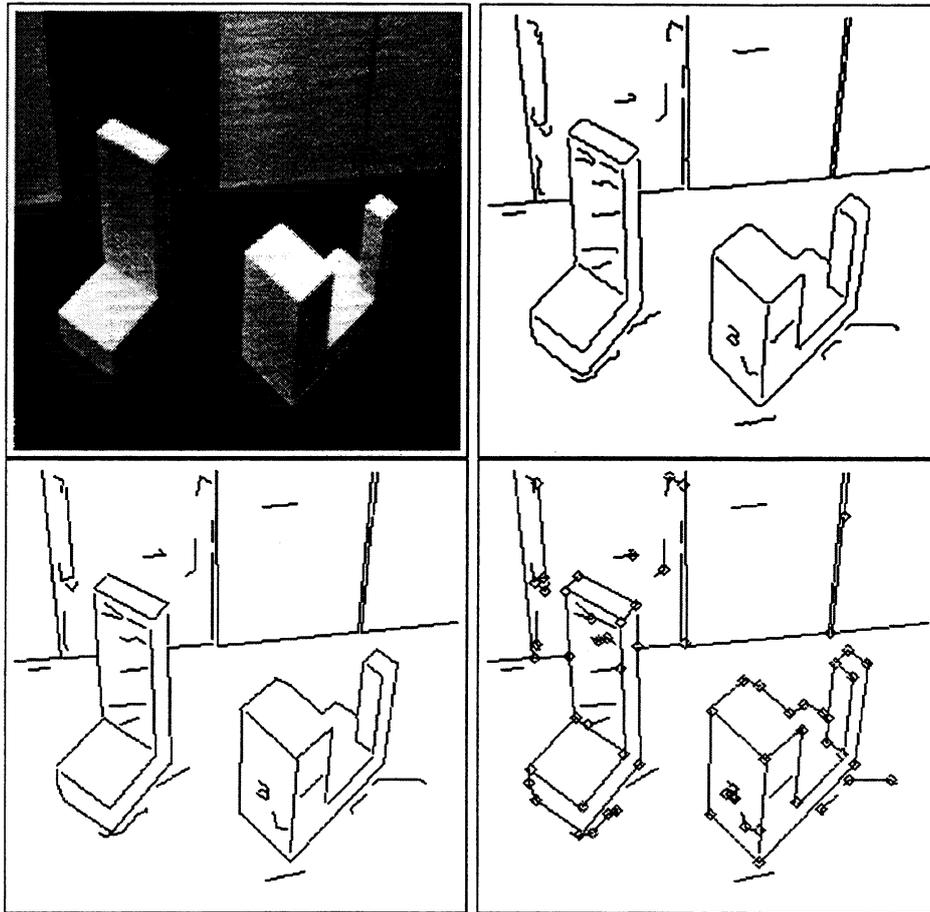


Figure 1.2: FEAGRO est un logiciel de vision bas niveau qui permet de traiter une image fournie par une carte de digitalisation. Voici un exemple d'une image traitée par ce système.

la scène et ceux représentant les modèles. DeFigueiredo [27] utilise aussi les graphes attribuables pour représenter des objets polyédriques. Dans ce cas, les nœuds du graphe représentent des surfaces et les arcs représentent la relation de proximité entre surfaces. Fan, Medioni et Nevatia [34] proposent une approche similaire. Cependant ils utilisent des portions de surfaces au lieu de surfaces complètes. Les relations choisies sont un facteur de chevauchement et la connexité entre surfaces. Lu et Wong [71] représentent des objets par des hyper-graphes attribuables. Les nœuds dans ces hyper-graphes représentent des surfaces alors que les arêtes représentent des limites entre surfaces. Enfin, les "hyper-arêtes" de ces hyper-graphes représentent des volumes (ensembles des surfaces et leurs limites).

L'approche de description que nous proposons pour résoudre le problème de l'indexation de modèles en présence d'une vaste base de modèles prend aussi la forme d'un graphe pour représenter les objets. Pourtant, à la différence de toutes ces approches qui utilisent des primitives spécifiques (jonctions, surfaces, portions de surfaces, volumes ...), notre méthode opère sur une représentation abstraite des

indices, plus précisément, des groupes d'indices. Donc les nœuds dans le graphe peuvent représenter n'importe quel type d'indices (jonctions ou surfaces ou portions de surfaces ou volumes ...) et les arêtes dans ce graphe représentent des relations entre des indices.

Ce graphe peut bien décrire un objet tout entier ou une partie de l'objet. La représentation est également valable pour décrire des données 2D/3D ou des objets 2D/3D. De plus, la reconnaissance 2D/2D, 2D/3D et 3D/3D est projetée (de l'anglais *mapped*) dans un même repère de travail.

1.7 Questions relatives à la recherche

Cette partie résume les questions clés que nous examinons dans ce travail :

1. Modélisation des objets et description de la scène :

- Quelles primitives ou indices visuels choisir?
- Quels critères doivent-ils respecter pour être choisis?
- Comment les extraire?
- Comment les évaluer?
- Quelle modèle choisir?
- Comment le modèle est-il affecté par le bruit, les occultations et les différents angles de vue?
- Quels sont ses limitations?

2. Indexation :

- Quelle méthode d'indexation choisir?
- Sur quels critères doit-on se baser pour choisir une telle ou telle méthode?
- Quelle est la performance en termes d'efficacité et de robustesse?
- Quels sont ses domaines d'application?

1.8 La contribution de la recherche

Le contribution de la recherche présentée dans cette thèse est celle d'une méthode pour décrire et indexer des objets 3D en présence d'une vaste base de modèles. Cette méthode est basée sur une approche de caractérisation provenant de la théorie des graphes. Elle utilise comme entrée l'information des contours de ces objets. Les principaux résultats sont les suivants :

1. un système qui reçoit en entrée une approximation polygonale d'une image de contours pour ensuite décrire et indexer les objets 3D (vues) qui se trouvent dans l'image. Le système opère à partir d'une technique de recherche appelée *hachage des graphes* (chapitre 4).
2. cette technique est possible grâce à une approche de caractérisation de graphes en termes d'un ensemble d'entiers utilisant le *polynôme deuxième immanent* associé à la matrice du graphe (chapitre 3).
3. dans la pratique les modèles et les objets-image sont représentés par des graphes.
4. le *hachage des graphes* est appliqué aux sous-graphes de ces graphes tant pour l'apprentissage comme pour la tâche de l'indexation d'objets.
5. le *hachage des graphes* est tolérant en présence d'une certaine quantité de bruit et au chevauchement.

1.9 Organisation du rapport

Le reste de ce mémoire est organisé comme suit. Dans le chapitre 2 nous donnons un résumé des travaux antérieurs concernant la représentation des objets 3D et de l'indexation en présence d'une vaste base de modèles.

Dans le chapitre 3 nous présentons de façon complète une méthode générale de la théorie de graphes pour caractériser un graphe binaire aussi bien qu'un graphe pondéré. Dans le premier cas la caractérisation est unique. Dans le deuxième cas la caractérisation n'est que locale.

Dans le chapitre 4 nous présentons notre technique d'indexation. Cette technique est possible grâce à la méthode de caractérisation du chapitre 3. Elle comporte :

1. le calcul de l'ensemble des coefficients du polynôme associé à un graphe,
2. le stockage de cette information dans la base de données,
3. la recherche de cette information dans la base de données.

Le chapitre 5 décrit un système d'indexation de modèles basé sur la technique d'indexation du chapitre 4. Un objet est décrit par un ensemble de vues caractéristiques. Chaque vue caractéristique (VC) est ensuite représentée par un graphe binaire. Toutefois la tâche de comparer directement une "VC de l'image" avec une VC d'un objet n'est pas facile car l'information est perturbée par le bruit et par le chevauchement. Pour surmonter ces problèmes, nous proposons de décomposer une VC en *figures fondamentales* (FF), chaque figure fondamentale est associée à un "petit" graphe.

Le chapitre 6 présente les résultats d'expérimentation obtenus avec des objets polyédriques.

Finalement, le chapitre 7 résume, les avantages et les inconvénients du système et suggère quelques directions de recherche en fonction des limitations actuelles.

Chapitre 2

Résumé des travaux précédents

La représentation de formes joue un rôle important dans le domaine de la reconnaissance d'objets. De nombreuses méthodes ont été développées pour résoudre le problème de la reconnaissance d'objets. La plupart sont fondées sur des descriptions utilisant des primitives géométriques ayant une complexité très variable :

- des points 2D ou 3D spécifiques,
- des contours 2D ou 3D isolés,
- des groupes de contours 2D,
- des surfaces 3D et
- des volumes.

selon que l'objet est 2D ou 3D. Dans ce travail, nous nous intéressons très spécialement aux méthodes de description des formes 3D.

Dans ce chapitre, nous exposons, tout d'abord, un résumé des différentes approches de description de formes (paragraphe 2.1). De même, nous présentons un tour d'horizon des différents types des primitives (indices visuels) utilisées pour obtenir ces différentes descriptions (paragraphe 2.2). Finalement, nous présentons

une série d'exemples des systèmes de reconnaissance d'objets 2D et 3D les plus représentatifs (paragraphe 2.3). Ceux-ci ont la caractéristique commune d'incorporer un mécanisme d'indexation pour réduire le temps de la recherche dans la base de modèles sur laquelle ils opèrent. Leurs avantages et leurs inconvénients sont aussi discutés.

Notre but n'est pas ici de présenter un travail de compilation complet mais d'exposer les approches les plus répandues. Nevatia [84], Ballard et Brown [6], Flynn et Jain [38], Bels et Jain [7] et Chin et Dyer [19], entre autres, fournissent d'excellents résumés concernant ces sujets.

2.1 Résumé des méthodes de description de formes

La forme des différents objets dans une scène peut être décrite de plusieurs manières selon qu'il s'agit d'objets 2D ou d'objets 3D. Dans ce travail, nous nous intéressons tout spécialement aux objets 3D. Un objet 3D peut donc être décrit soit en termes d'une ou de plusieurs de ses projections 2D soit en termes de son image de profondeur. Dans le premier cas, il est connu que les descriptions fournies ne comportent pas l'information relative à la forme 3D de l'objet. Dans le deuxième cas, les descriptions fournies sont plus riches mais les calculs qui découlent sont dans presque tous les cas beaucoup plus coûteux.

Dans cette partie nous décrivons les méthodes de modélisation d'objets 3D les plus répandues aujourd'hui. Celles-ci peuvent être classées en deux catégories possibles :

1. représentations où l'objet est centré par rapport à l'observateur, dites "*à objet centré*" et
2. représentations où l'observateur est centré par rapport à l'objet, dites "*à observateur centré*".

Le système de reconnaissance d'objets décrit dans ce mémoire utilise, comme nous allons le voir dans le chapitre 5, la versatilité d'une représentation qui n'est attachée à aucun de ces deux repères pour fournir aux étapes suivantes d'indexation et de vérification, un modèle relationnel de chaque objet. A notre connaissance une telle approche est unique.

2.1.1 Description des modèles

Les méthodes représentant les modèles d'objets 3D peuvent être divisées grosso modo en quatre catégories :

1. méthodes utilisant des indices visuels (caractéristiques visuelles) spécifiques,
2. méthodes utilisant une représentation 3D,
3. méthodes utilisant une représentation à partir d'une seule vue 2D et
4. méthodes utilisant une représentation à partir de plusieurs vues 2D.

2.1.2 Méthodes utilisant des caractéristiques visuelles spécifiques

A cette catégorie appartiennent toutes les approches n'utilisant qu'une partie des indices visuels (caractéristiques spécifiques) trouvées pour représenter les différents modèles (cf. Bolles et Cain [12]). Les modèles, dans ce cas, contiennent très peu de caractéristiques, par exemple, la distance entre deux points significatifs. Ces caractéristiques servent à décrire les différences parmi les modèles. Un objet dans la scène est identifié avec un modèle par l'étiquetage des caractéristiques significatives communes à l'objet et au modèle. Une représentation de ce type a le grand inconvénient qu'elle nécessite généralement une étude complète des modèles des objets avant de décider quelles indices visuels seront utilisés pour discriminer les modèles. La sélection de ces indices se fait par tâtonnements. Ainsi, si on veut ajouter des modèles à la base de données, des modifications seront nécessaires. En conséquence, la flexibilité du système est très restreinte.

2.1.3 Méthodes utilisant une représentation 3D

Cette catégorie comprend les approches utilisant des représentations avec un niveau d'organisation très élevé pour décrire des modèles 3D (cf. Brooks [16]). L'avantage principal de ce type de représentations est qu'elles fournissent des descriptions précises à objet centré. L'inconvénient principal est qu'il est nécessaire d'évaluer une translation du modèle 3D au moment de l'appariement avec une image 2D. Cette translation est toujours très coûteuse. En outre, l'obtention automatique d'une telle représentation est aussi très difficile à mettre en œuvre. En conséquence, dans presque tous les cas les modèles doivent être construits à la main.

2.1.4 Méthodes de représentation à partir d'une seule vue 2D

Dans ce cas, une seule vue (à partir d'un point de vue particulier) est utilisée pour représenter le modèle d'un objet donné (cf. Oshima et Shirai [86]). Les autres vues (à partir d'autres points de vue) sont ignorées. Ceci impose bien entendu des restrictions très fortes par rapport aux changements des points de vue.

2.1.5 Méthodes de représentation à partir de plusieurs vues 2D

Dans ce cas, un modèle est représenté par une collection de descriptions 2D obtenues à partir d'un ensemble de vues 2D fini (cf. Fan, Medioni et Nevatia [34]). Ces approches ont plusieurs avantages :

- Tout d'abord, elles sont plus faciles à obtenir que les représentations 3D.
- Deuxièmement, la même méthode peut être utilisée pour représenter les modèles et les objets dans la scène.
- Finalement, ces méthodes peuvent être conçues de sorte qu'aucune information a priori concernant les objets dans la scène soit nécessaire. Ceci n'est pas possible pour les méthodes utilisant des caractéristiques visuelles spécifiques à cause des modifications nécessaires lorsque des objets sont ajoutés à la base de modèles.

Le problème avec ces méthodes est que les vues contenues sont, en général, en grand nombre. Ceci affecte évidemment leur efficacité et en plus elles demandent de grands espaces mémoire. En conséquence, leur utilisation en vision par ordinateur est encore limitée, pour plus de détails, on peut se rapporter à (Faugeras, Mundi, Ahuja, Dyer, Pentland, Jain et Ikeuchi [35]).

Pour construire les modèles des objets de façon automatique et avoir des descriptions compatibles tant pour les modèles que pour les objets dans la scène, nous avons décidé d'utiliser une représentation à plusieurs vues. Dans la pratique le nombre de ces *vues caractéristiques* varie entre une et huit selon la complexité de l'objet. Ceci dans le but d'économiser de l'espace de mémoire et ainsi améliorer la vitesse des calculs.

2.2 Tour d'horizon des primitives utilisées en vision par ordinateur

Cette partie est destinée à présenter de façon réduite les différents types de primitives (indices visuels) souvent utilisées en vision par ordinateur pour :

- modéliser les objets dans une scène et/ou
- représenter les modèles de ces objets.

La classification adoptée ici est comparable à celle proposée par Dickinson, Pentland et Rosenfeld [30]. Seule la complexité des primitives est utilisée pour faire l'exposé. La figure 2.1 illustre les différents types des primitives classées en fonction de leur complexité.

Type de primitive	Complexite de la primitive
a points 2D/3D	
a contours 2D/3D	
Groupes de points 2D/3D	
Groupes de courbes 2D/3D	
de surface	
de volume	

Figure 2.1: Les primitives classées par ordre de complexité.

2.2.1 Primitives à points 2D/3D ou à contours 2D/3D spécifiques

La manière, peut-être la plus simple, pour modéliser des objets 3D consiste à utiliser des caractéristiques significatives : des jonctions 2D/3D, des courbes 2D/3D... Ces primitives sont, en général, faciles à extraire, faciles à utiliser et souvent elles demandent des temps de calcul courts. De ce fait, elles sont généralement choisies par la majorité des chercheurs pour la reconnaissance d'objets. Ceci est bien sûr valable lorsque le nombre des primitives composant les modèles reste petit. Si ce n'est pas le cas, le nombre d'appariements entre primitives objet et primitives modèle est grand. En conséquence son utilisation est limitée aux petites bases de modèles (Lamdan, Schwartz et Wolfson [67]). En outre, elles comprennent en général moins d'information qui est nécessaire pour décrire des objets 3D et elles sont également sensibles au bruit et aux occlusions. Finalement, leur simplicité suggère que l'interprétation fournie à partir de l'information d'entrée est plus ambiguë (quelques jonctions dans l'image peuvent correspondre à beaucoup de modèles dans la base). Cela peut demander une étape de vérification pour supprimer les ambiguïtés qui peut être très chère.

2.2.2 Groupes de points 2D/3D ou de courbes 2D/3D

La plupart des systèmes de reconnaissance d'objets existants essaient d'apparier des caractéristiques images (jonctions, contours...) avec des caractéristiques modèle comparables. Une approche qui est commune à plusieurs chercheurs consiste à for-

mer de petits groupes de caractéristiques image (points et/ou contours) pour ensuite les mettre en correspondance avec des groupes de caractéristiques modèle pour indexer, par exemple, une base de modèles. Ensuite, chaque hypothèse d'appariement est vérifiée ou rejetée par l'intermédiaire d'une information supplémentaire du modèle et de l'image. Pour reconnaître un objet, on doit trouver l'ensemble des bons appariements.

Un problème primordial de cette approche est que le nombre d'appariements est exponentiel en fonction du nombre de caractéristiques composant le groupe. Ceci suggère que, si on veut utiliser des primitives de ce type, on doit chercher une manière pour réduire la combinatoire.

Une façon d'accélérer le processus d'appariement consiste à utiliser les groupes de caractéristiques image comme des clés pour indexer un tableau spécial de caractéristiques modèle construit à l'avance et hors ligne. En ligne, ces clés fourniront aux étapes suivantes seulement les groupes de caractéristiques modèle ayant un appariement possible avec chaque groupe de caractéristiques image.

Lowe [70] utilise des groupes de segments pour indexer et reconnaître des objets 3D dans une base de données limitée à quelques modèles. Les segments sont regroupés en utilisant essentiellement trois types de relations : proximité, parallélisme et colinéarité

Lamdam, Schwartz et Wolfson [67, 66] utilisent des groupes à trois points (points d'intérêt) et de groupes à trois lignes pour reconnaître des objets plats à partir d'un point de vue arbitraire.

Dans [68], Lamdam et Wolfson généralisent leur approche pour la reconnaissance d'objets 3D à partir des images 2D de niveaux de gris.

Clemens et Jacobs [21, 20] utilisent des groupes à quatre et à cinq points pour indexer des modèles 3D à partir des images 2D.

2.2.3 Primitives de surface

Un troisième type de primitives utilisées pour représenter un objet 3D sont ses surfaces composantes. Celles-ci sont, en général, plus faciles à évaluer que les primitives de plus haut niveau, dites volumiques. Le problème est qu'elles sont très sensibles à l'occlusion lorsque leurs propriétés sont évaluées directement à partir de l'image. Une solution consiste à segmenter les surfaces visibles de l'objet en *portions* de surface (Fan [33]). L'utilisation de combinaisons de ces portions de surface réduit l'effet de l'occlusion. La segmentation des surfaces de l'objet se fait soit en portions régulières (triangles, splines 2D...) soit à des endroits spécifiques, notamment, au niveau des discontinuités de ces surfaces.

Dans le premier cas, le nombre de portions de surface trouvées est typiquement

très grand et, en plus, les points et les lignes où les portions se touchent n'ont aucune signification physique.

Dans le deuxième cas, le nombre de portions trouvées est moins grand, réduisant le nombre de calculs. En revanche, le prix de la segmentation est, dans la plupart des cas, très élevé car le problème de la segmentation de la surface d'un objet à des endroits spécifiques n'est pas un problème complètement résolu.

2.2.4 Primitives volumiques

Dans ce cas, il s'agit de représenter un objet, non plus par les surfaces qui le limitent, mais par le volume qu'il occupe. Il existe actuellement plusieurs façons de représenter un objet par des volumes. Nous exposons ici les plus utilisées.

Voxels

Une manière de représenter un objet consiste à le placer dans un système de coordonnées de référence et de diviser le volume occupé par l'objet, en petits éléments de volume (généralement appelés *voxels*). La forme normalement adoptée pour ces voxels est celle d'un cube ou d'un parallélépipède. La manipulation de ce type de primitives n'est pas rapide mais est en revanche très facile à obtenir. Si la taille du voxel utilisée est la même alors cette description peut occuper un grand espace de mémoire et en plus un grande quantité de voxels peuvent être nécessaires pour approcher un grand objet avec une forme compliquée.

Aggarwal et Kim [1] utilisent, par exemple, des voxels parallélépipédiques rectangulaires pour représenter des objets 3D à partir de trois vues orthogonales de l'objet.

Du fait qu'un grand nombre de voxels peuvent être nécessaires pour décrire un objet, leur application est plutôt limitée à la synthèse d'images.

Géométrie solide constructive

Le modélisation d'objets fondée sur la Géométrie Solide Constructive (de l'anglais, Constructive Solid Geometry - CGS), combine un ensemble fini de primitives de forme (généralement de formes simples telles que des cylindres, sphères...) et un ensemble aussi fini d'opérations binaires (d'union, d'intersection...) pour produire le solide voulu. Un modèle CGS est enregistré sous la forme d'un arbre. Des nœuds terminaux dans cet arbre représentent des primitives de base alors que des nœuds internes représentent des opérations binaires. Les arcs servent tout simplement à renforcer la séquence des opérations.

Cette représentation décrit bien l'enchaînement des opérations de conception et de fabrication d'un objet physique (soudure de pièces, enlèvement de matière ...) d'où son utilisation fréquente en CAO. Mais de par son aspect symbolique, elle ne peut être utilisée directement pour la visualisation, sauf si elle est convertie en un modèle plus approprié.

Ce type de représentation est plus utilisé en CAO qu'en vision par ordinateur. Il est adapté lorsqu'une information concernant la structure de l'objet est nécessaire.

Cylindres généralisés

Les cylindres généralisés ont été proposés par Binford [10] comme un outil pour décrire des objets 3D. Un cylindre généralisé (ou cône généralisé) nommé aussi "geon" est défini comme une forme plate arbitraire appelée *section transversale* ou "cross-section" se déplaçant le long d'une ligne courbe 3D arbitraire appelée *axe*. En général, la taille et la forme de la section transversale peuvent changer le long de l'axe. La règle décrivant ce changement est appelé *règle de déplacement*.

Agin [2] obtient des cylindres généralisés à partir d'images de profondeur par l'estimation, tout d'abord, de l'orientation et de la position de l'axe de déplacement. Ensuite, on fait passer des cercles le long des axes pour obtenir les cylindres. Les formes décrites sont donc restreintes à des sections transversales circulaires.

Nevatia et Binford [85] obtiennent des cylindres généralisés à partir des frontières des objets. Tout d'abord, une série de cylindres locaux le long d'axes droits sont évalués et ensuite ils sont élargis en courbant légèrement les axes. Parmi tous les cylindres engendrés seulement quelques uns sont sélectionnés. Ceux-ci sont utilisés pour décrire les différents objets en termes de leurs propriétés : longueur de l'axe, largeur moyenne de la coupe transversale, élongation, type (conique ou cylindrique) et leur connexité.

D'autres travaux utilisant des cylindres généralisés pour décrire des objets 3D ont été faits par Rao et Nevatia [89], Brooks [17] et Bergevin et Levine [9].

Les cylindres généralisés sont utiles pour des objets de forme allongée. Mais, ils ne sont pas appropriés pour décrire par exemple des objets polyédriques ou des cartes géographiques.

Superquadriques

Ces type de primitives est surtout utilisé pour représenter à partir de points 3D des volumes complexes et très divers de manière continue et compacte. Pentland [87], Bacjy et Solina [5] et Boulton et Gross [14] ont été parmi les premiers à les utiliser pour représenter des objets 3D. Une primitive de ce type peut être imaginée

comme un morceau d'argile qui peut être façonné jusqu'à ce que celle-ci adopte la forme de la partie étant modelée. La segmentation d'objets 3D complexes par des superquadriques peut être une tâche très difficile. En conséquence, son utilisation en vision par ordinateur est limitée plutôt aux problèmes d'approximation (de l'anglais, *fitting*) que au développement de systèmes de reconnaissance pour représenter des objets.

2.2.5 Résumé sur les primitives utilisées en vision par ordinateur

Les primitives spécifiques (points 2D/3D ou contours 2D/3D isolés) sont, généralement, faciles à extraire et à les utiliser. En outre, elles demandent des temps de calcul courts. Pourtant, chaque fois que l'on veut ajouter un modèle à la base de données, des modifications sont nécessaires. De plus, elles ne contiennent pas toute l'information nécessaire pour décrire complètement les objets 3D. En conséquence, la flexibilité du système les utilisant se voit dégradée.

Les primitives à groupes de points 2D/3D ou à groupes de courbes 2D/3D représentent de nos jours le choix de beaucoup de chercheurs parce qu'elles permettent de résoudre un certain nombre de problèmes en vision par ordinateur. Notamment, l'indexation en présence d'une grande base de modèles est possible pourvu que les groupes engendrés n'appartiennent qu'à une petite quantité des modèles de la base.

Les primitives surfaciques sont très sensibles à l'occlusion lorsque leurs propriétés sont évaluées directement à partir de l'image. L'utilisation de portions de surface régulières ou scindées à des endroits spécifiques réduit cet effet d'occlusion. Dans le premier cas, le nombre des portions peut être très élevé. Dans le deuxième cas, le prix pour avoir un petit nombre est élevé car le problème de la segmentation de la surface d'un objet à des endroits spécifiques n'est pas un problème bien résolu.

Les primitives volumiques permettent, dans l'ensemble, de représenter des objets 3D d'une manière très complète. Pourtant en général, elles sont plus difficiles à évaluer que les primitives surfaciques (sauf pour les cas de voxels) et les techniques pour les calculer ne sont pas encore complètement développées.

Dans le cadre de ce travail nous avons décidé d'utiliser des groupes de segments comme primitives pour décrire tant les modèles composant la base de données que les objets dans la scène traitée. Ceux-ci sont le résultat de l'approximation polygonale des contours de la scène.

2.3 Résumé des systèmes de reconnaissance utilisant un mécanisme d'indexation

Cette partie est destinée à décrire des systèmes de reconnaissance d'objets 2D et 3D typiques. Ceux-ci se caractérisent en ce qu'ils utilisent un mécanisme d'indexation pour accélérer le processus de la recherche en présence d'une vaste base de modèles. Chaque exemple est traité selon les points suivants :

- Comment les modèles sont représentés et comment la base de modèles est-elle organisée?
- Comment les objets de la scène sont-ils décrits?
- Comment les modèles sont-ils indexés dans la base?
- Quels sont les principaux avantages du système?
- Quels sont les principaux inconvénients du système?

2.3.1 Nevatia et Binford (1977)

Nevatia et Binford [85] présentent un système qui utilise des cylindres généralisés pour décrire et reconnaître des objets. Ce système est un des premiers à utiliser un mécanisme d'indexation pour réduire le temps de la recherche des modèles dans une base de données.

Représentation des modèles. Le modèle d'un objet quelconque est constitué d'un ensemble de vues, chacune d'elles étant représentée par un graphe relationnel. Un nœud de ce graphe représente une articulation (le point d'union entre des parties), et une arête du graphe représente une partie de l'objet caractérisée par un cylindre généralisé contenant l'information géométrique associée : longueur de l'axe, largeur moyenne de la coupe transversale, élongation et type (conique ou cylindrique).

Description de la scène. Les différentes parties des objets dans la scène à analyser sont aussi représentées par des cylindres généralisés. De même, un graphe relationnel est aussi obtenu pour chaque objet.

Les processus d'indexation et de mise en correspondance. Le processus de mise en correspondance s'est fait en deux étapes : une étape d'indexation pour réduire le nombre de candidats à la reconnaissance et une étape d'appariement (vérification) qui utilise les résultats de l'indexation. L'étape d'indexation se sert

des parties des objets (cylindres) de la scène pour indexer les modèles. L'approche consiste à générer un *code de description* pour chaque objet pour ensuite indexer dans la base les modèles ayant les mêmes codes de description. Le code de description est basé seulement sur les descriptions des parties d'objet dites discriminantes (pièces longues ou larges) de l'objet, chaque pièce engendrant un code. Les trois propriétés qui forment le code sont : la connexité de la pièce discriminante (relié d'un seul côté ou des deux), son type (long ou large) et si elle est conique ou non.

L'efficacité de la recherche est augmentée par l'ordonnement des modèles ayant les mêmes codes de description.

Dans la deuxième étape, l'objet est comparé à chaque modèle (pièce par pièce) suivant l'ordre de l'indexation en prenant en compte seulement les appariements entre pièces discriminantes similaires. Le processus d'appariement continue de cette manière jusqu'à avoir fini toutes les possibilités.

Avantages principaux. Les descriptions des modèles et des objets sont évaluées de façon automatique. La description n'a pas besoin d'une connaissance a priori des objets. En plus, elle est robuste au chevauchement partiel et permet le traitement d'objets articulés.

Inconvénients principaux. La petite quantité des descripteurs utilisés et le fait que ces descripteurs soient du type binaire limite sensiblement le nombre de combinaisons pour le codage des pièces composant les objets. En conséquence, beaucoup de pièces auront le même code réduisant l'efficacité de la recherche.

2.3.2 **Kalvin, Shonberg, Schwartz et Sharir (1986)**

Kalvin, Shonberg, Schwartz et Sharir [63] proposent pour la première fois une technique qui permet d'indexer de façon géométrique les modèles des objets dans une base de données. Cette technique appelée *hachage géométrique* utilisée en conjonction avec d'autres méthodes de reconnaissance permet d'identifier des objets à deux dimensions provenant d'une vaste base de données et avec des occlusions éventuelles. La performance de la méthode ne se voit pas vraiment dégradée lorsque le nombre de modèles augmente.

Représentation des modèles. L'obtention du modèle correspondant est obtenue par la décomposition du contour de l'objet 2D en sous-sections de contour appelées "footprints". Une footprint est une courbe obtenue à partir du contour de l'objet ayant les propriétés d'invariance à la translation, d'invariance à la rotation et une haute probabilité de correspondre qu'à une portion de contour d'un seul modèle.

Une fois que les footprints pour un contour donné sont engendrées, la base des modèles est organisée comme suit : un espace à cinq dimensions est divisé en hypercubes dont la taille est fixe. Les valeurs choisies pour les axes de cet espace sont, respectivement : les quatre premiers coefficients des sinus et cosinus de la décomposition de Fourier de chaque section du graphe G longueur d'arc versus l'angle de rotation. Ce graphe représente la fonction de courbure du contour, et l'angle total de rotation enfermé dans une fenêtre à partir d'un point donné. Chaque hypercube intersecté par une portion de footprint est enregistré dans un tableau de hachage. A chaque hypercube enregistré, on associe une liste de tous les modèles contenant cette portion de footprint.

Description de la scène. La même approche utilisée pour les modèles est utilisée aussi pour décrire les objets de la scène.

Les processus d'indexation et de mise en correspondance. De même qu'avec le système de Nevatia et Bindford l'identification d'objets se fait en deux étapes : l'indexation des modèles et la vérification. Pour chaque objet observé dans l'image, le mécanisme d'indexation utilise chacune des footprints associées à l'objet pour fournir une liste limitée des modèles dont les hypercubes associés contiennent des sections de cette footprint. Chacun des modèles formant cette petite liste est ensuite comparé avec la footprint pour sélectionner le candidat le plus proche. Finalement, la position et l'orientation de l'objet sont déterminés.

Avantages principaux. L'utilisation de cette technique de hachage par footprints permet au système d'une part, de rapidement sélectionner une liste de candidats parmi les nombreux présents dans la base de modèles et d'autre part d'utiliser cette liste des candidats pour choisir le meilleur par comparaison avec les footprints d'indexation.

Le système permet d'identifier des objets à deux dimensions en présence d'une vaste base de modèles (100 modèles).

La même description est utilisée pour les modèles et pour les objets. Elle est aussi évaluée de façon automatique. Elle est robuste aux occlusions et en plus elle n'a pas besoin d'une connaissance a priori des objets.

Inconvénients principaux. L'inconvénient principal de cette technique est le besoin de segmenter les contours de la scène observée en sous-parties telles que chacune provienne d'un objet différent. Ceci se fait en utilisant l'heuristique que différents objets produisent des angles concaves (aux points de coupure) lorsque qu'ils se superposent les uns sur les autres. Ceci peut introduire bien entendu des points de coupure supplémentaires pour des objets avec des concavités.

La non invariance aux changements d'échelle et sa limitation à la reconnaissance d'objets 2D sont d'autres inconvénients de cette méthode.

2.3.3 Lamdan et Wolfson (1988)

Lamdan et Wolfson [68] continuent avec la même pensée que Kalvin, Shonberg, Schwartz et Sharir, d'utiliser l'idée de hachage géométrique pour indexer une base de modèles et proposent une méthode générale de représentation et de reconnaissance d'objets. La méthode est valable pour la reconnaissance d'objets 2D et 3D. Un soin particulier est donné au problème de l'identification d'objets 3D dans des scènes où le phénomène d'occlusion est présent, en utilisant l'image 2D de niveaux de gris.

Représentation des modèles. Les objets sont représentés par des ensembles de caractéristiques géométriques (points ou lignes) invariants par rotation, translation et échelle. Les caractéristiques utilisées dans ce cas sont dits des points d'intérêt. La représentation d'un ensemble de points se fait par la définition d'un repère de coordonnées orthogonales basé sur une paire ordonnée de points (dans le cas 2D) de cet ensemble. Le reste des points est représenté par leurs coordonnées par rapport à ce repère. Cette procédure est répétée pour chaque paire de points. Ainsi par exemple, si un modèle d'un objet est défini par m points d'intérêt, pour chaque paire ordonnée de points modèle les coordonnées de tous les autres $m - 2$ points modèles sont exprimées dans le repère orthogonal défini par la paire de base. Chaque coordonnée (après un échantillonnage adéquat) est utilisée comme clé d'accès à un tableau de hachage, où la paire de base à partir de laquelle la coordonnée a été obtenue et le modèle sont enregistrés.

Description de la scène. La même approche est utilisée pour décrire les objets dans la scène.

Les processus d'indexation et de mise en correspondance. Le processus de reconnaissance extrait tout d'abord de l'image d'entrée un ensemble de points d'intérêt. A partir de cet ensemble le processus choisit une paire de points arbitraire et calcule les coordonnées des autres points dans la scène en prenant cette paire comme une base. Chaque coordonnée relative à cette paire est ensuite comparée aux clés d'accès des enregistrements dans le tableau de hachage. Lorsque la valeur de la coordonnée et de la clé d'accès sont les mêmes un vote est donné au modèle et à la paire de base correspondante. L'enregistrement (modèle, paire de base) avec le plus grand nombre de votes est choisi comme candidat pour l'étape de vérification.

Finalement, la vérification se fait en mettant en correspondance les segments du modèle avec les segments dans la scène.

Avantages principaux. L'avantage principal de cette approche est sa redondance. Ceci permet de reconnaître des objets dans une scène où le phénomène d'occlusion est présent. Des modèles ajoutés à la base de données peuvent être traités indépendamment sans recalculer le tableau de hachage.

Inconvénients principaux. Cette méthode est rapide lorsque qu'il n'y a qu'un seul modèle dans la base et lorsque dans l'image il y a peu de points n'appartenant pas à l'objet dans la scène. Mais, si la base commence à avoir plusieurs modèles ou si la scène contient beaucoup de points parasites ou tout simplement si un objet ne se trouve pas dans la base de modèles, alors le système devient inefficace sur le plan de la recherche. Ceci est dû, d'une part, aux primitives choisies (quelques points dans l'image peuvent se trouver dans beaucoup de modèles de la base) et d'autre part, à la formation de groupes parasites.

2.3.4 Ettinger (1988)

Ettinger [32] décrit un système de vision utilisant une représentation hiérarchique dans la structure (allant de l'objet tout entier aux sous-parties le composant) et hiérarchique dans l'échelle (des caractéristiques grossières aux caractéristiques fines) pour reconnaître des objets en présence d'une vaste base de modèles malgré le bruit et l'occlusion.

Représentation des modèles. La représentation utilisée, d'une part, pour organiser la base de modèles et d'autre part, pour reconnaître des objets dans une scène est obtenue par l'intermédiaire de deux mécanismes : l'extracteur de caractéristiques et l'extracteur de sous-parties. L'extracteur de caractéristiques prend en entrée le contour de l'objet à modéliser et trouve un ensemble de caractéristiques capturant l'information de courbure du contour (les changements significatives de courbure). L'extracteur de sous-parties utilise cette collection de caractéristiques pour décomposer l'objet en différentes sous-parties.

La représentation finale prend la forme d'un graphe dont les nœuds représentent les différentes sous-parties de l'objet et dont les arêtes spécifient comment les sous-parties sont reliées entre elles.

Un modèle est ajouté à la bibliothèque en utilisant ses sous-parties comme clés d'indexation. Pour cela, le système met en correspondance chacune des nouvelles sous-parties aux sous-parties déjà existantes dans la base. Deux sous-parties se ressemblant mutuellement (ayant les mêmes caractéristiques et dans le même ordre) sont regroupées en une seule clé. L'algorithme utilisé pour mettre en correspondance les sous-parties est le même utilisé pour le système de reconnaissance.

L'utilisation d'une telle représentation hiérarchique permet au système de di-

viser le grand problème de la reconnaissance d'un objet tout entier en petits sous-problèmes de la reconnaissance des sous-parties de l'objet.

Description de la scène. Seul l'extracteur de caractéristiques est utilisé pour décrire les objets dans la scène.

Les processus d'indexation et de mise en correspondance. Le système de reconnaissance prend en entrée, d'une part, la base de modèles et d'autre part, la description hiérarchique de la scène et fournit en sortie une liste des toutes les interprétations cohérentes de la scène. Pour cela, le système de reconnaissance combine des parties des différents schémas : hypothèse-prédiction-vérification avec la technique de recherche restreinte proposée par Grimson et Lozano-Pérez [44]. Les hypothèses et les prédictions se font au niveau des sous-parties alors que la vérification est effectuée en utilisant des contraintes géométriques parmi les caractéristiques des sous-parties pour réduire l'espace de recherche.

Une fois qu'une sous-partie a été identifiée et localisée dans la scène, cette connaissance est utilisée pour reconnaître les autres sous-parties. En utilisant la bibliothèque, tous les modèles contenant la sous-partie identifiée sont sélectionnés comme des candidats possibles. Le système prédit l'identité et la transformation du reste des sous-parties pour réduire encore la liste de candidats. Lorsque ces prédictions sont vérifiées, la liste de candidats est consolidée. Le système continue à prédire et à vérifier des composantes de la scène avec des composantes des modèles afin d'identifier chaque objet dans la scène avec un modèle de cette liste. Si un nombre suffisant des composantes d'un modèle ont été reconnues, la configuration des composantes résultante est considérée comme étant une des interprétations de la scène.

Avantages principaux. A la différence d'autres approches cette méthode utilise une représentation hiérarchique dans la structure et dans l'échelle. Ceci permet la reconnaissance d'objets d'une complexité variable, en présence du bruit, d'occlusion, des erreurs d'échantillonnage ainsi que la distorsion introduite par le capteur. De plus, l'évaluation de la représentation et de la création de la base de données se font de manière automatique.

Inconvénients principaux. Ce système a, à notre avis, deux inconvénients : la reconnaissance limitée à des objets à deux dimensions et la complexité du mécanisme de reconnaissance.

2.3.5 Stein et Medoni (1990)

Stein et Medoni [96] traitent le problème de la reconnaissance dans une scène contenant plusieurs objets plats s'occluant les uns les autres et d'un point de vue quelconque (perspective faible).

Représentation des modèles. La représentation du modèle ou de la scène est basée sur une approximation polygonale du contour de l'objet. Pour des raisons de robustesse, plusieurs approximations polygonales avec différentes tolérances concernant le nombre de lignes composantes sont utilisées. Pour chaque approximation polygonale un nombre de caractéristiques de base appelées *supersegments* est obtenu. Un supersegment est formé par les k segments à gauche et les k segments à droite d'un sommet quelconque de l'approximation polygonale. Il y a autant de supersegments que de sommets sur l'approximation polygonale. Un supersegment est caractérisé par sa cardinalité (nombre de segments), les angles entre segments consécutifs, la longueur d'arc (la somme des longueurs des segments), la localisation du sommet central, l'orientation (le vecteur entre les sommets d'avant et d'après du sommet du milieu) et le moment de deuxième ordre défini par la matrice de covariance des sommets du supersegment.

La base de données est organisée sous la forme d'un tableau de hachage dont les clés d'accès aux modèles sont les supersegments. Pour cela, chaque supersegment est représenté par un code. Ce code est formé en fonction des angles entre les segments consécutifs et le moment de deuxième ordre.

Description de la scène. Les contours de la scène sont décrits par la même méthode utilisée pour décrire les modèles.

Les processus d'indexation et de mise en correspondance. Le processus de la reconnaissance se fait essentiellement en deux étapes. Une étape d'indexation et une autre de vérification. L'étape d'indexation fournit une liste limitée à quelques candidats à la reconnaissance par l'intermédiaire des supersegments de chaque approximation polygonale des contours de la scène.

Pour la vérification tous les appariements possibles entre les supersegments de la scène et les supersegments de chaque modèle sont évalués pour générer des hypothèses d'appariement. Ces hypothèses avec les contraintes de distance, d'angle et de direction proposées par Grimson et Lozano-Pérez [44, 45] sont ensuite utilisées pour élaguer encore la liste de candidats jusqu'à trouver les modèles correspondants aux objets dans la scène.

Avantages principaux. Les modèles sont acquis de façon automatique. La redondance de la représentation utilisée permet de faire face aux problèmes de bruit et d'occlusion dans la scène.

2.3.6 Stein et Medioni (1991)

Stein et Medioni [97, 98] présentent leur approche *de hachage structural* pour la reconnaissance d'objets 3D en utilisant des données 3D à partir d'un point de vue arbitraire. L'approche est nouvelle en ce qu'ils utilisent deux types différents de primitives pour l'appariement : les lignes correspondant à des discontinuités de profondeur et d'orientation et des petites portions de surfaces "splashes". Les lignes servent à représenter des objets polyédriques alors que les splashes servent à représenter des objets non polyédriques.

Représentation des modèles. Suivant leur idée originale, d'utiliser de supersegments pour représenter les contours des objets 2D, Stein et Medioni généralisent leur méthode pour décrire leur premier type de primitives choisies : les contours 3D. A nouveau, la représentation pour décrire les contours 3D est fondée sur une approximation polygonale et plusieurs approximations polygonales avec différentes tolérances concernant le nombre de lignes composantes sont utilisées pour des raisons de robustesse. Les supersegments 3D (étant une extension de supersegments 2D) sont obtenus de la même façon. Les paramètres caractérisant un supersegment 3D sont sa cardinalité (nombre de segments), les angles de courbure entre segments consécutifs, et les angles de torsion entre bi-normales consécutives.

Le deuxième type de primitive "le splash" représente une portion de surface aux environs du centre de la portion. Un splash est construit comme suit. A une position donnée sur la surface de l'objet le vecteur normal est obtenu. En utilisant cette normale comme référence une tranche circulaire est évaluée. A partir d'un point arbitraire de cette surface, une série de points et leurs normales à la surface sont déterminés. Dans la pratique, ceci est fait en se déplaçant autour de la normale de référence d'un angle constant. Pour le premier point et la normale de référence un repère est obtenu. Ce point est ensuite représenté, à partir de ce repère, en coordonnées sphériques. Le reste des points est aussi représenté par rapport au premier point. Les deux applications résultantes avec l'angle de déplacement (pour chaque point) sont utilisées de façon combinée pour décrire une courbe dans l'espace 3D. Cette courbe résultante est donc représentée par la méthode décrite ci-dessus pour décrire des contours des objets 3D : la courbe est approchée par une approximation polygonale. Un supersegment 3D est évalué pour cette approximation polygonale. Le supersegment 3D résultant est caractérisé en fonction des mêmes paramètres déjà utilisés (cardinalité, les angles de courbure entre segments consécutifs . . .) plus d'autres comme par exemple le rayon du splash. De même qu'avec les contours 2D, plusieurs approximations polygonales avec différentes tolérances concernant le

nombre de lignes composantes sont utilisées pour des raisons de robustesse.

La base de données est aussi organisée sous la forme d'un tableau de hachage dont les clés d'accès aux modèles sont, bien entendu, les supersegments 3D et les splashes. Les supersegments 3D sont encodés soit en fonction des angles de courbure et des angles de torsion (dans le cas de supersegments 3D des objets polyédriques) soit en fonction des angles de courbure, des angles de torsion et les rayons des splashes (dans le cas des non polyédriques).

Description de la scène. Les contours dans la scène sont décrits aussi par des supersegments 3D et des portions de surface selon qu'il s'agit de contours polyédriques ou de contours non polyédriques.

Les processus d'indexation et de mise en correspondance. Le processus de la reconnaissance se fait essentiellement en deux étapes. Une étape d'indexation et une autre de vérification. L'étape d'indexation fournit une liste limitée à quelques candidats à la reconnaissance par l'intermédiaire des supersegments 3D et les splashes évalués à partir de la scène.

Pour la vérification tous les appariements possibles entre les supersegments de la scène et les supersegments de chaque modèle sont évalués pour générer des hypothèses d'appariement. Ces hypothèses avec les contraintes de distance, d'angle et de direction proposées par Grimson et Lozano-Pérez [44] et [45] sont ensuite utilisées pour élaguer encore la liste de candidats jusqu'à trouver les modèles correspondants aux objets dans la scène.

Avantages principaux. Les modèles sont acquis de façon automatique. L'utilisation de deux types de primitives permet au système de surmonter le problème de la reconnaissance dans les cas où soit l'information de contour n'est pas suffisante, soit l'information de surface n'est pas suffisante pour une classification correcte. La redondance de la représentation utilisée permet de faire face aux problèmes du bruit et de chevauchement dans la scène.

2.3.7 Résumé des systèmes de reconnaissance

D'autres travaux utilisant une étape d'indexation dans leurs systèmes de reconnaissance pour améliorer la performance dans la recherche ont été développés par Knoll et Jain [65], Hong et Wolfson [53], Breuel [15], Mehrotra et Grosky [74], Clemens et Jacobs [20], Swain et Ballard [99], Burns et Riesman [18], Jacobs [57], Rotwell [91, 93, 92], Swain et Ballard [99], Flynn et Jain [39] et Dickinson, Pentland et Rosenfeld [30, 29].

2.3. Résumé des systèmes de reconnaissance utilisant un mécanisme d'indexation**45**

En résumé, les systèmes décrits ou mentionnés ci-dessus utilisant un mécanisme d'indexation pour accélérer le processus de la recherche peuvent être classés comme suit :

- la plupart opèrent sur un champ limité aux objets 2D (Knoll et Jain [65], Kalvin [63], Hong et Wolfson [53], Ettinger [32], Mehrotra et Grosky [74], Stein et Medioni [96]) et Rotwell [91, 93, 92]. Ceci est dû, soit à la "facilité" du problème posé, soit parce que il s'agit d'un système en cours de développement.
- le reste des systèmes élargissant leur champ aux objets 3D :
 - sont limités à opérer en présence de petites bases de modèles (Nevatia et Binford [85], Lowe [70] et Lamdan et Wolfson [68]) à cause de la simplicité des primitives utilisées,
 - sont dépendants de l'efficacité de l'extraction des primitives, surtout s'il s'agit de primitives volumiques (Dickinson, Pentland et Rosenfeld [30, 29]),
 - sont capables d'opérer en présence de grandes bases de modèles (Kalvin [63], Swain et Ballard [99], Stein et Medioni [97], Stein et Medioni [98] et Burns et Riesman [18]).

Dans le cadre de ce travail, nous avons décidé de réaliser la mise en correspondance en graphes représentant les jonctions et les segments visibles des objets et d'utiliser la même description pour représenter tant les modèles des objets que les objets dans la scène.

Chapitre 3

La caractérisation des graphes

Nous considérons dans ce chapitre le problème suivant : étant donné un graphe G à n noeuds et à m arêtes, nous voulons envisager une méthode pour le caractériser, c'est-à-dire, trouver une méthode qui permette de distinguer les graphes. Les distinguer, en quoi, ou par rapport à quoi? Par rapport à leur topologie et à leur apparence.

- par **topologie** on entend la façon dont les arêtes du graphe sont reliées les unes par rapport aux autres,
- par **apparence** on entend la façon dont le graphe est dessiné dans le plan.

Le but de ce chapitre est de présenter une méthode générale provenant de la théorie de graphes pour caractériser ces deux propriétés d'un graphe.

Dans le premier cas, la caractérisation résultante est unique pour les graphes G avec une taille $n \leq 12$ (paragraphe 3.2.3). Dans le deuxième cas, la caractérisation de l'apparence résultante n'est unique que localement (pour les sommets et les arêtes les "plus proches" du "centre" du graphe). En tout cas, cette caractérisation couvre une grande quantité d'exemples.

Le chapitre est organisé comme suit : dans la partie 3.1, nous rappelons un certain nombre de notions et de définitions de l'algèbre et de la théorie des graphes, qui seront nécessaires pour la suite.

Dans la partie 3.2, nous exposons le problème en question : celui de la caractérisation d'un graphe. De même, nous présentons un compte rendu historique concernant les différentes approches qui ont essayé de donner une réponse à ce problème.

Dans la partie 3.3, nous exposons de façon complète la technique de caractérisation choisie. Dans la partie 3.3.3 nous appliquons cette technique pour caractériser la topologie du graphe. Dans la partie 3.4 nous expliquons comment la même technique est utilisée pour caractériser l'apparence du graphe.

Finalement, dans la partie 3.5, nous présentons quelques conclusions concernant la méthode de caractérisation proposée.

3.1 Rappels

Cette partie comporte quelques rappels concernant un certain nombre de notions et de définitions de l'algèbre et de la théorie des graphes nécessaires pour introduire ensuite les outils mathématiques dont nous avons besoin pour caractériser un graphe.

Pour quelques références sur ces sujets, voir Littlewood [69], James [58], Boerner [11], James et Kerber [59], Serre [95], Andrews [3], Cvetkovic [25, 24], Minc [80], Markus [72], Gantmacher [41], Turner [101], Merris [78, 79, 76], Harary [49], Berge [8], Tutte [102], Foulds [40] et Gross [47].

3.1.1 Groupes

Dans cette partie nous présentons les définitions de groupe, sous-groupe, permutation, groupe symétrique, classe, partition et caractère.

Un *groupe* G est un système composé d'un ensemble d'éléments a, b, c, \dots , muni d'une opération produit, telle que :

1. le carré d'un élément et le produit de deux éléments sont éléments de l'ensemble,
2. le produit est associatif. Par exemple, $a(bc) = (ab)c$,
3. l'ensemble contient l'élément d'identité I tel que $Ia = aI = a$ pour chaque élément a de l'ensemble et
4. chaque élément a de l'ensemble a un inverse a^{-1} appartenant à l'ensemble tel que $aa^{-1} = a^{-1}a = I$.

Pour un groupe fini G le nombre d'éléments g de G est appelé l'*ordre* du groupe.

Un *sous-groupe* H d'un groupe G est une partie de G qui est aussi un groupe pour la loi définie dans G .

Permutations

Soit Ω l'ensemble des entiers $\{1, 2, 3, \dots, n\}$. Une *permutation* de cet ensemble est une application

$$\sigma : \Omega \rightarrow \Omega$$

de Ω dans lui-même telle que, si $i, j \in \Omega$ et $i \neq j$, $\sigma(i) \neq \sigma(j)$. Si σ est une telle permutation, l'ensemble des entiers

$$\{\sigma(1), \dots, \sigma(n)\}$$

possède n éléments distincts et est donc à nouveau constitué des entiers $1, \dots, n$ rangés dans un ordre différent.

Une permutation σ des entiers $\{1, 2, 3, \dots, n\}$ est notée

$$\sigma = \begin{pmatrix} 1 & 2 & \dots & n \\ \sigma(1) & \sigma(2) & \dots & \sigma(n) \end{pmatrix} \quad \text{ou} \quad \sigma = (\sigma(1) \sigma(2), \dots, \sigma(n))$$

Une *transposition* est une permutation qui échange deux nombres et laisse les autres fixes. Ainsi

$$\sigma = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \end{pmatrix}$$

représente la permutation σ telle que $\sigma(1) = 2$, $\sigma(2) = 1$ et $\sigma(3) = 3$. Cette permutation est en fait une transposition.

Une *substitution* est une permutation qui échange tous les nombres de Ω . Ainsi

$$\sigma = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \end{pmatrix}$$

représente la permutation σ telle que $\sigma(1) = 3$, $\sigma(2) = 1$ et $\sigma(3) = 2$. Cette permutation est en fait une substitution.

Groupe symétrique

L'ensemble de toutes les permutations de Ω , c'est-à-dire, l'ensemble

$$S_{\Omega} = \{\sigma \mid \sigma : \Omega \rightarrow \Omega \text{ et } \sigma(\omega) \neq \omega \text{ pour } \omega \in \Omega\}$$

est un groupe, est appelé le *groupe symétrique* associé à Ω .

Si n est le nombre d'éléments de Ω alors le nombre d'éléments de G est $n!$ (l'ordre du groupe). Enfin, le groupe symétrique d'ordre $n!$ est noté S_n .

Ainsi par exemple,

$$S_3 = \left\{ \begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{pmatrix}, \begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \end{pmatrix}, \begin{pmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{pmatrix}, \right. \\ \left. \begin{pmatrix} 1 & 2 & 3 \\ 1 & 3 & 2 \end{pmatrix}, \begin{pmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \end{pmatrix}, \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{pmatrix} \right\}$$

Partitions

Si n est un entier, une séquence décroissante d'entiers positifs ou nuls

$$\alpha = \{\alpha_1, \alpha_2, \dots\}$$

est appelée une *partition* de n si elle satisfait

- (i) $\forall i \geq 1 \quad (\alpha_i \geq \alpha_{i+1})$
- (ii) $\sum_{i=1}^{\infty} \alpha_i = n.$

Les α_i sont appelées les *parties* de α . La partition α est écrite comme $(x_1^1, x_2^2, \dots, x_i^{y_i}, \dots)$ si l'entier x_i apparaît $y(i)$ fois pour tout i . Par exemple, on écrit la partition $(4, 4, 2, 2, 1)$ comme $(4^2, 2^2, 1)$.

Ainsi par exemple, pour $n = 3$, on a les trois partitions possibles : $1+1+1$, $1+2$ et 3 représentées en forme compacte par les symboles : (1^3) , $(1, 2)$ et (3) respectivement.

Classes d'éléments conjugués

Un élément x_1 d'un groupe G est dit *élément conjugué* de l'élément x_2 de G s'il existe un troisième élément $t \in G$ tel que $x_2 = tx_1t^{-1}$.

L'ensemble complet d'éléments x_1, x_2, \dots, x_r qui sont conjugués à un élément donné x_1 est appelé une *classe* du groupe et noté ρ . Le nombre r des éléments de ρ est *l'ordre de la classe*.

Il est connu (cf. Littlewood [69] et Thomas et Zund [100]) qu'il y a une relation étroite entre le nombre des partitions d'un entier n et le nombre de classes d'éléments conjugués d'un groupe S_n d'ordre $n!$. Cette relation s'exprime par le fait que S_n admet le même nombre de classes d'éléments conjugués que le nombre des partitions de n . Autrement dit, chaque partition de n définit une classe d'éléments conjugués.

Les trois classes d'éléments conjugués, du groupe S_3 d'ordre 6, définies par les trois partitions de $n = 3$ sont respectivement :

1. l'identité :

$$\begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{pmatrix}$$

correspondante à la classe (1^3) ,

2. les trois transpositions :

$$\begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \end{pmatrix} \quad \begin{pmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{pmatrix} \quad \begin{pmatrix} 1 & 2 & 3 \\ 1 & 3 & 2 \end{pmatrix}$$

correspondantes à la classe $(1, 2)$ et

3. les deux substitutions :

$$\begin{pmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \end{pmatrix} \quad \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{pmatrix}$$

correspondantes à la classe (3) .

De la même façon, il est facile de vérifier qu'il y a aussi cinq classes d'éléments conjugués du groupe symétrique d'ordre $4!$, d'ordres 1, 6, 8, 6 et 3, correspondant aux partitions (1^4) , $(1^2, 2)$, $(1, 3)$, (4) , et (2^2) .

Caractères

Une fonction scalaire χ d'un groupe G sur le corps des complexes \mathbf{C} : $\chi : G \rightarrow \mathbf{C}$ est appelée un *caractère* du groupe G . Dans le cas du groupe symétrique S_n à n éléments, on sait aussi qu'il y a une correspondance entre les partitions de n et les caractères du groupe. Dans le cas de S_n , χ est appelé caractère *irréductible*.

Le caractère correspondant à la partition α est noté par χ_α . Si la partition est du type $(k, 1^{n-k})$ alors χ_α est appelé caractère "single-hook". Pour plus de détails concernant la théorie de la représentation de groupes symétriques, voir James et Kerber [59].

3.1.2 Théorie des matrices

Dans cette partie nous donnons les notions d'ensemble de séquence strictement croissante d'indices entiers, de sous-matrice principale et de sous-matrice principale complémentaire.

Soit $\Gamma_{k,n}$ l'ensemble des n^k séquences $\alpha = (\alpha_1, \dots, \alpha_k)$ d'indices entiers, $1 \leq \alpha_i \leq n$, $i = 1, \dots, k$. Soit $Q_{k,n}$ l'ensemble de toutes les séquences strictement croissantes d'entiers telles que

$$Q_{k,n} = \{(\alpha_1, \dots, \alpha_k) \in \Gamma_{k,n} \mid 1 \leq \alpha_1 < \dots < \alpha_k \leq n\} \quad (3.1)$$

Ainsi, par exemple, $Q_{2,3}$ contient les séquences (1,2), (1,3) et (2,3). En général, $Q_{k,n}$ comporte C_n^k séquences.

Soit $M_{m,n}(\mathbf{C})$, le groupe des matrices complexes de taille $m \times n$, c'est-à-dire l'ensemble des matrices sur \mathbf{C} (le corps des complexes). Soit $M_n(\mathbf{C})$ l'ensemble de toutes les matrices carrées de taille $n \times n$. Si maintenant $B = (b_{ij}) \in M_{m,n}(\mathbf{C})$ et si $\alpha \in Q_{h,m}$ et $\beta \in Q_{k,n}$ alors $B[\alpha|\beta]$ désigne la sous-matrice principale de taille $h \times k$ de B qui contient les lignes de α et les colonnes de β . c'est-à-dire la matrice de taille $h \times k$ dont l'entrée (i,j) est l'entrée $(\alpha(i), \beta(j))$ de B . Si par exemple $B = (b_{ij}) \in M_{5,6}(\mathbf{C})$, $h = 2$, $k = 3$, $\alpha = (1, 3)$ et $\beta = (2, 3, 4)$, alors

$$B[\alpha|\beta] = \begin{bmatrix} b_{12} & b_{13} & b_{14} \\ b_{32} & b_{33} & b_{34} \end{bmatrix}$$

Si $\alpha = \beta$, on simplifie la notation à $B[\alpha]$.

De même, si $\alpha \in Q_{h,m}$ et $\beta \in Q_{k,n}$ alors $B(\alpha|\beta)$ désigne la sous-matrice principale de taille $(m-h) \times (n-k)$ de B complémentaire à $B[\alpha|\beta]$, c'est-à-dire, la matrice obtenue à partir de B en éliminant les lignes α et les colonnes β . Si, par exemple, $B = (b_{ij}) \in M_{5,6}(\mathbf{C})$, $h = 3$, $k = 4$, $\alpha = (2, 3, 5)$ et $\beta = (1, 2, 4, 6)$, alors $B(\alpha|\beta)$ désigne la matrice de taille 2×2

$$B(\alpha|\beta) = \begin{bmatrix} b_{13} & b_{15} \\ b_{43} & b_{45} \end{bmatrix}$$

A nouveau, si $\alpha = \beta$, on simplifie la notation à $B(\alpha)$.

3.1.3 Graphes et matrices associées

Nous considérons, dans un premier temps, des graphes finis, non orientés et simples, c'est-à-dire, des graphes ne contenant pas de boucles, et où deux sommets sont reliés par une arête au plus. Un graphe G est donné par le couple $G = (V, E)$, où V représente l'ensemble de sommets $\{v_1, v_2, \dots, v_n\}$ du graphe, et E représente l'ensemble des arêtes $\{e_1, e_2, \dots, e_m\}$ du graphe qui est aussi l'ensemble des paires des éléments de V .

La matrice d'adjacence d'un graphe

La matrice d'adjacence associée à G est la matrice symétrique $A(G) = (a_{ij})$ de taille $n \times n$, dont les entrées (a_{ij}) sont données par

$$a_{ij} = \begin{cases} 1 & \text{s'il y a une arête entre } v_i \text{ et } v_j \\ 0 & \text{sinon} \end{cases} \quad (3.2)$$

Comme G n'a pas de boucles, la diagonale principale de A ne contient que des 0.

Evidemment, $A(G)$ dépend non seulement de G mais aussi de l'étiquetage de ses sommets. Il est facile à voir, cependant, que les matrices d'adjacence obtenues à partir d'un même graphe avec différents étiquetages sont similaires par permutation.

Deux graphes $G_1 = (V_1, E_1)$ et $G_2 = (V_2, E_2)$ sont dits *isomorphes* s'il existe une bijection $\phi : V_1 \rightarrow V_2$ telle que $(v_1, v_2) \in E_1$ si et seulement si $(\phi(v_1), \phi(v_2)) \in E_2$.

Si A_1 et A_2 sont les matrices d'adjacence des graphes G_1 et G_2 , alors G_1 et G_2 sont isomorphes si et seulement si A_1 et A_2 sont similaires via une matrice de permutation P :

$$A_2 = P^T A_1 P \quad (3.3)$$

Ainsi, une condition nécessaire pour l'isomorphisme est que A_1 et A_2 soient similaires par permutation.

Deux graphes G_1 et G_2 sont *adjacence cœspectraux* si $\det(xI - A(G_1)) = \det(xI - A(G_2))$ (cf. Harary [49], Mowshowitz [83] et Merris, Rebman et Watkins [78]).

La matrice des degrés d'un graphe

Si d_i représente la valence ou degré (le nombre d'arêtes attachées) du sommet v_i avec $i = 1, \dots, n$ alors la matrice diagonale $D(G)$ associée au graphe G est une matrice dont l'entrée (i, j) est donnée par

$$d_{i,j} = \begin{cases} d_i & \text{si } i = j \\ 0 & \text{sinon} \end{cases} \quad (3.4)$$

La matrice laplacienne d'un graphe

La matrice laplacienne associée au graphe G est aussi une matrice de taille $n \times n$. Cette matrice est définie comme $L(G) = D(G) - A(G)$. Les entrées l_{ij} de $L(G)$, dans ce cas, sont données par

$$l_{ij} = \begin{cases} d_i & \text{si } i = j \\ -1 & \text{s'il y a une arête entre } v_i \text{ et } v_j \\ 0 & \text{sinon} \end{cases} \quad (3.5)$$

Cette matrice est aussi appelée *matrice d'admittance* (Cvetkovic, Doob et Sachs [25]) ou *matrice de Kirchhoff* (Constantine [22]). De même qu'avec la matrice d'adjacence, deux étiquetages différents de G donnent comme résultat des matrices laplaciennes similaires par permutation.

Ses origines remontent, apparemment, aux travaux de Kirchhoff [64] (d'où son nom) sur les réseaux électriques. Ses applications ne sont pas limitées qu'aux réseaux électriques, mais aussi à d'autres sujets de la théorie des graphes, notamment, la complexité d'un graphe.

Par rapport à la matrice d'adjacence, la matrice laplacienne a des propriétés qui font d'elle un choix préférable pour résoudre certains problèmes de la théorie des graphes, notamment celui de la caractérisation des graphes. Voici quelques unes de ces propriétés fondamentales :

1. $L(G)$ n'est pas seulement symétrique mais aussi semi-définie positive,
2. si le rang de $L(G)$ est $n - 1$, alors tous les cofacteurs de $L(G)$ sont non nuls et égaux au nombre des arbres de "spanning" en G . Un arbre de spanning est un sous-graphe contenant tous les nœuds de G . Le rang de $L(G)$ est $n - 1$ si et seulement si G est *connexe*. Un graphe est connexe si pour chaque nœud i de G $d_i \geq 1$.
3. $L(G)$ est singulière (la somme des éléments d'une ligne est nulle). Autrement dit, le vecteur $\mathbf{1}$, avec toutes les entrées égales à 1, est le noyau de $L(G)$,

Mohar [82, 81] et Constantine [23] présentent d'autres propriétés de la matrice laplacienne.

3.1.4 Fonctions matricielles généralisées

Soit $B = (b_{ij}) \in M_n(\mathbb{C})$. Soit S_n le groupe symétrique de degré n et H un sous-groupe de degré h de S_n . Soit χ un caractère du groupe H . Soit $\sigma \in H$ une permutation des éléments $1, 2, \dots, n$. La *fonction matricielle généralisée* correspondant au caractère χ et notée $d_\chi^H : M_n(\mathbb{C}) \rightarrow \mathbb{C}$ est définie par

$$d_\chi^H(B) = \sum_{\sigma \in H} \chi(\sigma) \prod_{t=1}^n b_{t\sigma(t)} \quad (3.6)$$

où la somme est exécutée sur toutes les permutations de H .

Immanants

Si χ est un caractère de S_n et si $H = S_n$ alors la fonction d_χ^H est appelée *immanent* ou *fonction de Schur* (Littlewood [69], Minc [80], Merris et Walkins [79], James et Liebeck [60], Johnson et Pierce [62] et Heyfron [52]).

Si, par exemple, $\chi = \varepsilon$ (le caractère d'alternance ou fonction signe), alors $d_\chi^H = \det(B)$ (le très connu déterminant).

Si $\chi \equiv 1$ (le *caractère principal*), alors $d_\chi^H = \text{per}(B)$ est la fonction *permanent*.

Le déterminant et le permanent d'une matrice sont donc des cas spéciaux d'immanants.

3.1.5 Polynômes généralisés

Soit $B(G) = (b_{ij}) \in M_n(\mathbf{C})$ la matrice représentant soit la matrice $A(G)$ soit la matrice $L(G)$ d'un graphe G à n nœuds et à m arêtes.

Soit χ un caractère du groupe H . Le *mineur principal généralisé* de taille $i \times i$ d'une matrice de taille $n \times n$ correspondante à χ est une fonction matricielle généralisée d_χ^H évaluée pour la matrice obtenue en éliminant $n - i$ lignes et leurs colonnes correspondantes.

Le *polynôme généralisé* correspondant au caractère χ du groupe symétrique H est défini par

$$p_\chi(x) = d_\chi(xI - B(G)) = b_0(G)x^n - b_1(G)x^{n-1} + \dots + (-1)^n b_n(G) \quad (3.7)$$

Les coefficients b_i de ce polynôme sont obtenus comme la somme de tous les mineurs principaux généralisés de taille $i \times i$ de $B(G)$ (cf. Turner [101] et Gantmacher [41]).

Alors si $H = S_n$ et si χ est la fonction signe, $p_\chi(x)$ est le polynôme caractéristique

$$\det(xI - B(G)) = x^n - d_1(G)x^{n-1} + d_2(G)x^{n-2} - \dots + (-1)^n d_n(G) \quad (3.8)$$

Si $H = S_n$ et si $\chi \equiv 1$, alors $p_\chi(x)$ est le *polynôme permanental*

$$\text{per}(xI - B(G)) = x^n - p_1(G)x^{n-1} + p_2(G)x^{n-2} - \dots + (-1)^n p_n(G) \quad (3.9)$$

3.2 Le problème général de la caractérisation d'un graphe

3.2.1 Introduction

Les problèmes concernant l'identification de graphes appartiennent aux problèmes les plus importants de la théorie des graphes. Parmi ces problèmes se trouvent :

1. l'isomorphisme de graphes,
2. la reconstruction d'un graphe et
3. l'ordonnancement et la classification de graphes.

Le problème de l'isomorphisme de graphes est celui qui consiste à trouver entre les ensembles des sommets de deux graphes une bijection qui minimise une distance, ou qui maximise le recouvrement entre les deux graphes.

Le problème de la reconstruction d'un graphe consiste à établir si chaque graphe est déterminé de façon unique par l'ensemble de ses *sous-graphes de sommet effacé*. Un sous-graphe de sommet effacé est le graphe obtenu par l'effacement d'un sommet quelconque v du graphe G et toutes les arêtes y attachées.

Finalement, le problème de l'ordonnement et la classification de graphes consiste à déterminer l'ordre lexicographique des graphes en utilisant des invariants de graphes. Parmi les invariants de graphes les plus utilisés pour classer des graphes, on trouve les valeurs propres (Harary [48] et Fisher [37]), les vecteurs propres (Cvetkovic [26]), les diviseurs élémentaires (Vince [104]), les moments spectraux (Cvetkovic, Doob, Gutman et Torgasev [24]) et les coefficients du polynôme caractéristique (Mowshowitz [83]), du polynôme permanental (Merris, Rebman et Watkins [78]) et du polynôme deuxième immanantal (Merris [76]).

Le problème de la caractérisation d'un graphe se trouve à la fois dans la catégorie des problèmes d'isomorphisme de graphes et de la classification de graphes.

3.2.2 Enoncé du problème

Etant donné un graphe $G = (V, E)$ fini, simple et non orienté à n nœuds et à m arêtes. On veut trouver un ensemble d'invariants isomorphiques pour le caractériser, c'est-à-dire, trouver un moyen pour lui donner une étiquette qui le distingue d'autres graphes non isomorphes.

3.2.3 Les différentes approches de solution

Plusieurs approches ont été proposées à travers l'histoire pour résoudre le problème de la caractérisation d'un graphe. Cette partie est destinée à présenter les approches les plus importantes.

Les approches spectrales

Plusieurs chercheurs ont déjà abordé ce problème qui consiste à savoir si un graphe G pouvait être caractérisé symboliquement en quelque sorte, c'est-à-dire trouver un moyen pour lui donner une étiquette qui le distingue d'autres graphes. Les premiers travaux reliés, Harary [48] et Fisher [37] utilisent les valeurs propres ("le" spectre) de la matrice d'adjacence $A(G)$ du graphe. Cette proposition a été rapidement abandonnée par la découverte de plusieurs exemples de graphes non isomorphes ayant le même spectre. La Figure 3.1 montre quelques exemples de ces graphes trouvés dans les travaux de Harary, King, Mowshowitz et Read [50] et [49].

Les approches utilisant des fonctions matricielles

D'autres auteurs ont proposé des approches "plus puissantes" pour résoudre le même problème. Turner [101], par exemple, est le premier à utiliser l'ensemble des fonctions matricielles généralisées. Dans un premier temps, il définit le polynôme car-

actéristique généralisé $p_x(x) = d_x(xI - A(G))$. Cette fonction est appliquée aux matrices d'adjacence des graphes. Avec ce polynôme, il arrive à distinguer un ensemble plus nombreux de graphes. Dans ce même travail, il présente aussi deux contre-exemples pour lesquels ces fonctions ne fonctionnent plus. La raison est due à l'interprétation des symboles 0 et 1 de la matrice $A(G)$, (cf. Johnson et Newman [61]).

Pour affronter cela ces derniers auteurs considèrent une matrice d'adjacence modifiée, équivalente à remplacer chaque 0 de $A(G)$ avec un 1 et chaque 1 avec un x indéterminé. En appelant la nouvelle matrice $A_x(G)$, ils trouvent que $\det(xI - A_x(G))$ distingue, par exemple, toutes les paires de la figure 3.1 sauf la paire (c).

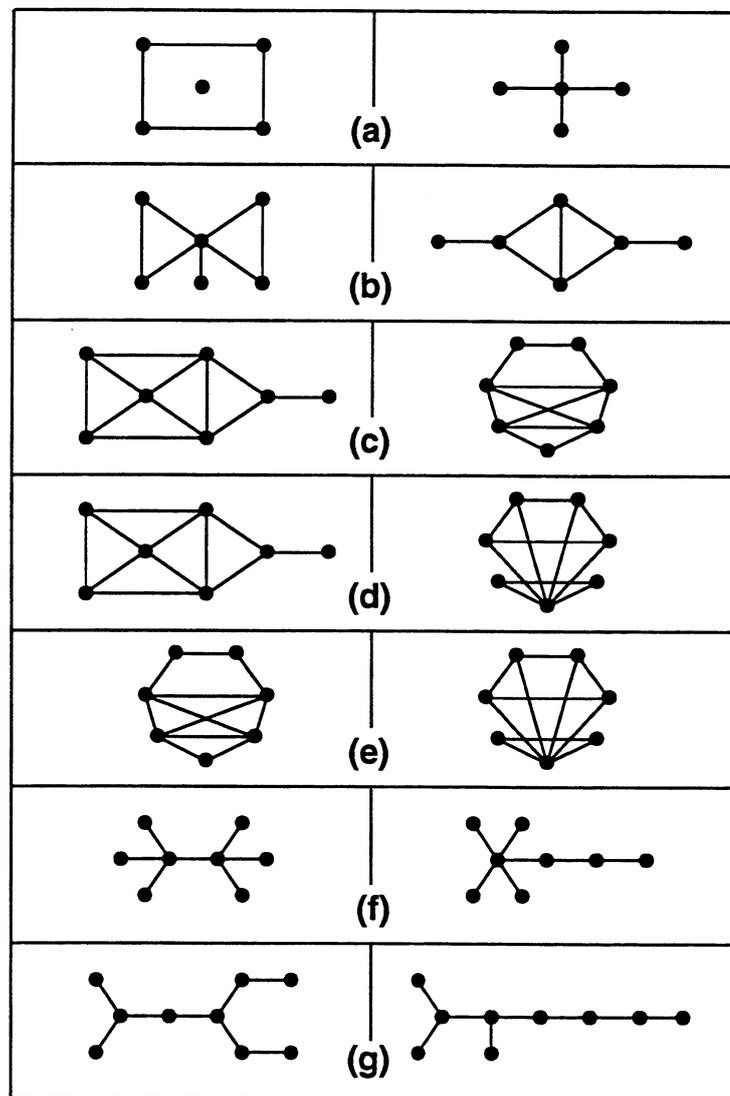


Figure 3.1: Plusieurs paires de graphes non isomorphiques avec les mêmes ensembles de coefficients du polynôme caractéristique ont été découvertes pendant ces dernières années. Cela a amené les chercheurs à abandonner cette approche de caractérisation.

L'article de Merris, Rebman et Watkins [78] initialise une sérieuse étude sur le *polynôme permanental* de la matrice d'adjacence laplacienne d'un graphe. Tout d'abord ils démontrent que le polynôme permanental donne de meilleurs résultats que le polynôme caractéristique pour distinguer des graphes lorsqu'on les applique à la matrice d'adjacence. Ensuite, ils montrent comment le polynôme permanental appliqué cette fois-ci à la matrice laplacienne permet de caractériser des graphes d'une taille quelconque. En effet, dans leur travail les auteurs déclarent n'avoir pas encore de connaissance d'une paire de graphes non isomorphes pour lesquels $per(xI - L(G_1)) = per(xI - L(G_2))!$

Bien que le polynôme permanental est préservé sous similarité par permutation, c'est-à-dire que pour deux étiquetages différents du même graphe, $per(xI - L(G_1)) = per(xI - L'(G_1))$ (les bonnes nouvelles), il n'est pas rentable à cause du manque d'un algorithme de calcul en temps polynômial (les mauvaises nouvelles), (cf. Valiant [103]), ce qui le rend inutilisable.

Le but du travail dû à Merris [76] est d'introduire précisément un compromis entre le $per[*]$ et le $det[*]$: le *deuxième immanent* d'une matrice. Ce polynôme a deux avantages. Tout d'abord, il est préservé aussi sous similarité par permutation (encore de bonnes nouvelles) et deuxièmement l'évaluation des ses coefficients est facile à réaliser car le problème de calcul est converti en calcul de déterminants, (Merris [76]) (les très bonnes nouvelles). Evidemment, ceci ne veut pas dire que avec ce polynôme on arrive à distinguer tous les graphes, car il existe des exemples qui démontrent le contraire (Cvetkovic, Doob et Sachs [25]). Pour des graphes de taille $n > 12$ il existe des graphes avec le même ensemble des coefficients du polynôme deuxième immanent. La figure 3.2 montre les deux premiers graphes de taille $n = 13$ avec les mêmes coefficients du polynôme deuxième immanent.

Il faut remarquer que le but de notre travail n'est pas de proposer une nouvelle méthode pour résoudre le problème de la caractérisation d'un graphe mais de choisir la plus adéquate à nos besoins pour étiqueter un petit graphe pour donner une réponse à la question de la reconnaissance d'objets tri-dimensionnels, en prenant en compte les propriétés:

- d'invariance,
- de forte discrimination (pas forcément parfaite) et
- de facilité et vitesse de calcul.

De cette discussion on peut conclure que le deuxième immanent est un très bon choix pour caractériser un graphe. Le reste de ce chapitre est consacré à présenter en détail le deuxième immanent et comment celui-ci peut être utilisé pour caractériser, dans un premier temps, la topologie d'un graphe (paragraphe 3.3.3), et dans un deuxième temps son apparence (paragraphe 3.4).

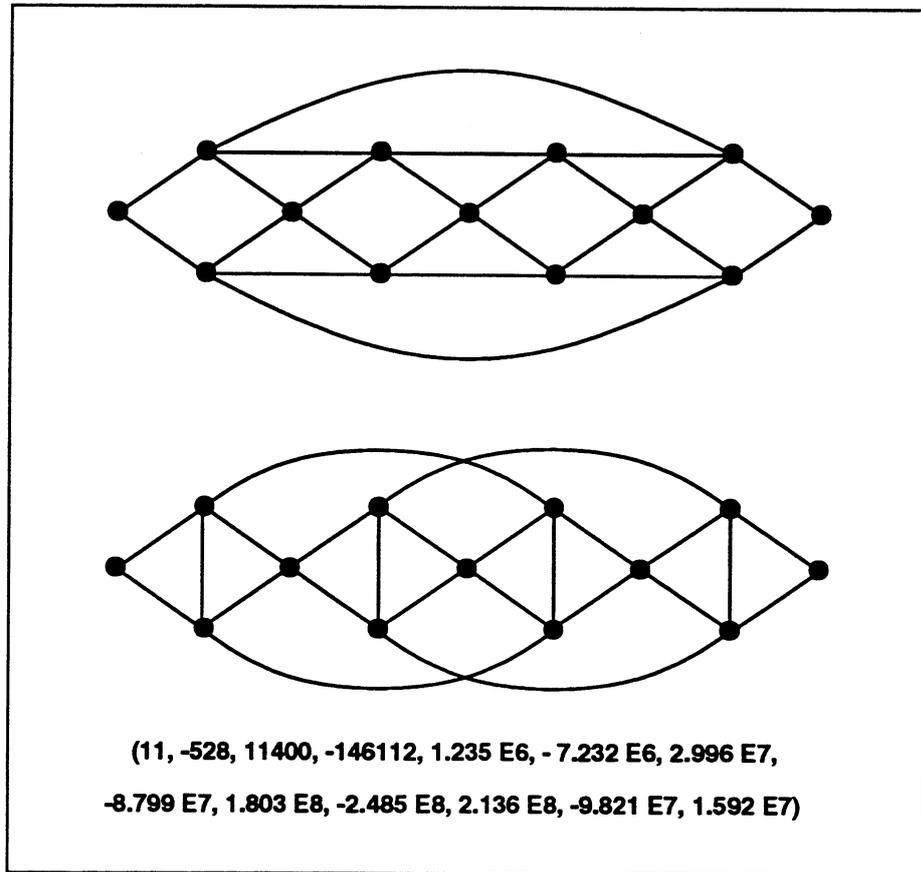


Figure 3.2: Le polynôme deuxième immanant favorise la caractérisation de graphes par rapport à sa contrepartie, le polynôme caractéristique. Pourtant, car ce polynôme est un compromis entre le polynôme permanental et le polynôme caractéristique il y a des exemples de graphes (avec une taille $n > 12$) avec les mêmes coefficients du polynôme d_2 .

3.3 Le deuxième immanant

Cette partie est consacrée à présenter un ensemble de résultats concernant le deuxième immanant : l'outil mathématique choisi pour caractériser un graphe. Pour arriver à ces résultats on donne d'abord une série de définitions de base nécessaires.

3.3.1 Définitions préliminaires

Le deuxième immanant d'une matrice $B = b_{ij}$ de taille $n \times n$ est défini par

$$d_2(B) = \sum_{\sigma \in S_n} \chi_2(\sigma) \prod_{t=1}^n b_{t\sigma(t)} \quad (3.10)$$

où χ_2 est un de caractères correspondant à la partition $(2, 1^{n-2})$ du groupe symétrique S_n des permutations (Merris [76]). Le polynôme d_2 de B est défini par

$$d_2(xI - B) = c_0(B) x^n - c_1(B) x^{n-1} + \dots + (-1)^n c_n(B) \quad (3.11)$$

Il est prouvé par Merris, Rebman et Watkins [78], car d_2 est un immanant donc d_2 est préservé sous similarité par permutation (pour deux matrices similaires du même graphe, les c_i du polynôme restent invariants).

3.3.2 Résultats principaux

Tout d'abord, considérons le polynôme d_2 d'une matrice générale $B = (b_{ij})$ de taille $n \times n$. Le coefficient $c_k(B)$ du polynôme $d_2(xI - B)$ peut être obtenu comme suit :

$$c_k(B) = \sum_{\alpha \in Q_{k,n}} d_2(B\{\alpha\}) \quad (3.12)$$

où $Q_{k,n}$ est l'ensemble de toutes les C_n^k séquences de taille k strictement croissantes d'indices formées à partir de l'ensemble $\{1, 2, \dots, n\}$, où α est un de ses éléments. $B\{\alpha\}$ est la matrice résultat de la somme directe de $B[\alpha]$ (la sous matrice principale de taille $k \times k$ correspondant à α , voir le paragraphe 3.1.2) et la matrice identité de taille $n - k$. Donc

$$B\{\alpha\} = B[\alpha] \oplus I_{n-k} = \begin{pmatrix} B[\alpha] & 0_k \\ 0_k & I_{n-k} \end{pmatrix} \quad (3.13)$$

où 0_k est la matrice nulle de taille k . Pour plus de détails voir Merris [75] et [76].

Il peut être démontré (Grone [46], Merris [76] et Merris et Watkins [79]) que pour $k \geq 2$:

$$d_2(B) = \sum_{i=1}^n b_{ii} \det B(i) - \det B \quad (3.14)$$

où $B(i)$ désigne la sous-matrice de B obtenue en effaçant la i -ème ligne et la i -ème colonne de la matrice B . Donc, pour $k \geq 2$

$$\begin{aligned} c_k(B) &= \sum_{\alpha \in Q_{k,n}} \left(\sum_{i=1}^n b_{ii} \det B\{\alpha\}(i) - \det B\{\alpha\} \right) \\ &= \sum_{\alpha \in Q_{k,n}} \left(\sum_{i=1}^n b_{ii} \det B\{\alpha\}(i) \right) - \sum_{\alpha \in Q_{k,n}} \det B\{\alpha\} \end{aligned} \quad (3.15)$$

3.3.3 Caractérisation de la topologie d'un graphe

Maintenant, pour le premier cas qui nous intéresse, c'est-à-dire caractériser la topologie d'un graphe $B = L(G)$, nous avons le résultat suivant, dû à Merris [76] :

Théorème 3.1 Soit G un graphe à n sommets et à m arêtes. On désigne par $L(G)$ la matrice laplacienne de G . On a alors :

$$d_2(xI - L(G)) = \sum_{k=0}^n (-1)^k c_k(G) x^{n-k} \quad (3.16)$$

Donc

$$\begin{aligned} c_0(G) &= n - 1, \\ c_1(G) &= 2m(n - 1), \\ &\vdots \\ c_k(G) &= \sum_{\alpha \in Q_{k,n}} \left(\sum_{i=1}^n b_{ii} \det B\{\alpha\}(i) - \det B\{\alpha\} \right) \end{aligned} \quad (3.17)$$

Pour la démonstration voir Merris [76]. Pour un graphe G à n nœuds et à m arêtes, la complexité de calcul des coefficients du polynôme d_2 est n^4 (Merris [76], Merris et Watkins [79] et Hartmann [51]). On rappelle que la complexité nécessaire pour le calcul du polynôme caractéristique est n^3 .

Il y a certaines raisons pour lesquelles on s'intéresse à la matrice laplacienne :

- Premièrement, on a vu que $L(G)$ n'est pas seulement symétrique mais aussi semi-définie positive de rang $\leq n-1$. (Le rang de $L(G)$ est $n-1$ si et seulement si G est connexe). Ceci est important car les immanants "s'adaptent" très bien aux matrices semi-définies positives (Merris [76]).
- Deuxièmement, $L(G)$ présente une grande importance dans l'étude de la complexité $\kappa(G)$ de G . Cette valeur $\kappa(G)$ est définie comme étant le nombre d'arbres de "spanning" de G . De plus, chaque cofacteur de $L(G)$ est égal à la complexité $\kappa(G)$ de G .
- Finalement, $L(G)$ peut mieux s'utiliser pour distinguer des graphes non isomorphes que la matrice d'adjacence. En effet, la plus grande complexité de $L(G)$ par rapport à celle de la matrice d'adjacence suggère l'existence de moins "d'accidents algébriques" (Merris [75, 76]).

3.3.4 Un exemple

Comme exemple d'application de ce résultat, trouvons l'ensemble des coefficients $c_k(G)$ qui caractérisent la topologie du graphe à quatre nœuds et à trois arêtes de la figure 3.3 (a).

Il est facile de vérifier que la matrice laplacienne correspond (pour l'étiquetage choisi) à celle montrée dans la figure 3.3 (b).

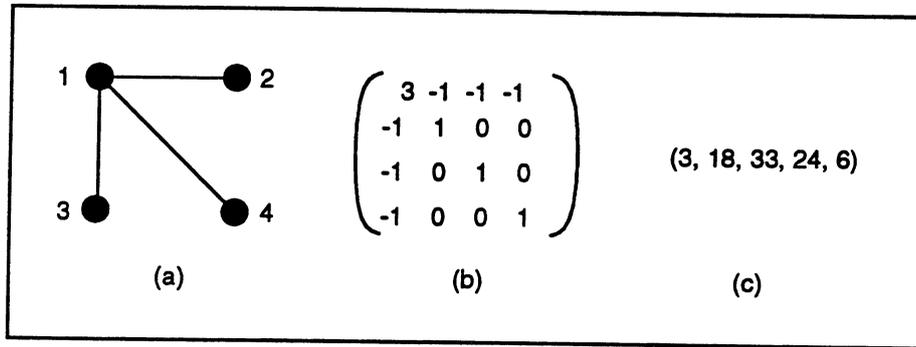


Figure 3.3: (a) Un exemple de graphe à quatre nœuds et à trois arêtes. (b) sa matrice laplacienne associée et (c) l'ensemble des coefficients caractérisant sa topologie.

En appliquant le Théorème 3.1 on peut voir que les valeurs pour les deux premiers coefficients sont les suivantes :

$$\begin{aligned} c_0(G) &= n - 1 = 3, \\ c_1(G) &= 2m(n - 1) = 18, \end{aligned}$$

Les valeurs des autres coefficients demandent un effort supplémentaire. Prenons par exemple le coefficient $c_2(G)$. Comme $k = 2$, donc,

$$Q_{2,4} = \{(1, 2), (1, 3), (1, 4), (2, 3), (2, 4), (3, 4)\}$$

Pour $\alpha = (1, 2) \in Q_{2,4}$ nous avons

$$L(G)[1, 2] = \begin{bmatrix} 3 & -1 \\ -1 & 1 \end{bmatrix}$$

De même,

$$L(G)\{1, 2\} = \begin{bmatrix} 3 & -1 \\ -1 & 1 \end{bmatrix} \oplus \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 3 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

et

$$\begin{aligned} c_2(G)_{\alpha=(1,2)} &= \sum_{i=1}^4 b_{ii} \cdot \det L(G)\{\alpha\}(i) - \det L(G)\{\alpha\} \\ &= (3) \begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{vmatrix} + (1) \begin{vmatrix} 3 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{vmatrix} \\ &+ (1) \begin{vmatrix} 3 & -1 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 1 \end{vmatrix} + (1) \begin{vmatrix} 3 & -1 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 1 \end{vmatrix} \end{aligned}$$

$$\begin{aligned}
 & - \begin{vmatrix} 3 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \\
 & = 3 + 3 + 2 + 2 - 2 = 8
 \end{aligned}$$

On peut facilement vérifier que $c_2(G)_{\alpha=(1,3)} = c_2(G)_{\alpha=(1,4)} = c_2(G)_{\alpha=(1,2)} = 8$ et que $c_2(G)_{\alpha=(2,3)} = c_2(G)_{\alpha=(2,4)} = c_2(G)_{\alpha=(3,4)} = 3$ donc $c_2(G) = 33$.

Si l'on procède de la même façon pour les autres coefficients, on obtient $c_3(G) = 24$ et $c_4(G) = 6$.

La figure 3.3 (c) montre cet ensemble de coefficients qui caractérisent la topologie du graphe G à quatre nœuds et à trois arêtes de la figure 3.3 (a).

3.4 Caractérisation de l'apparence d'un graphe

Un inconvénient de l'approche de caractérisation précédemment décrite est que seule la topologie du graphe est caractérisée. Ceci est normal puisque la matrice laplacienne sur laquelle le polynôme d_2 opère est par nature binaire : seule la connexité entre les nœuds du graphe est prise en compte et non pas son apparence (la façon dont le graphe est dessiné sur le plan).

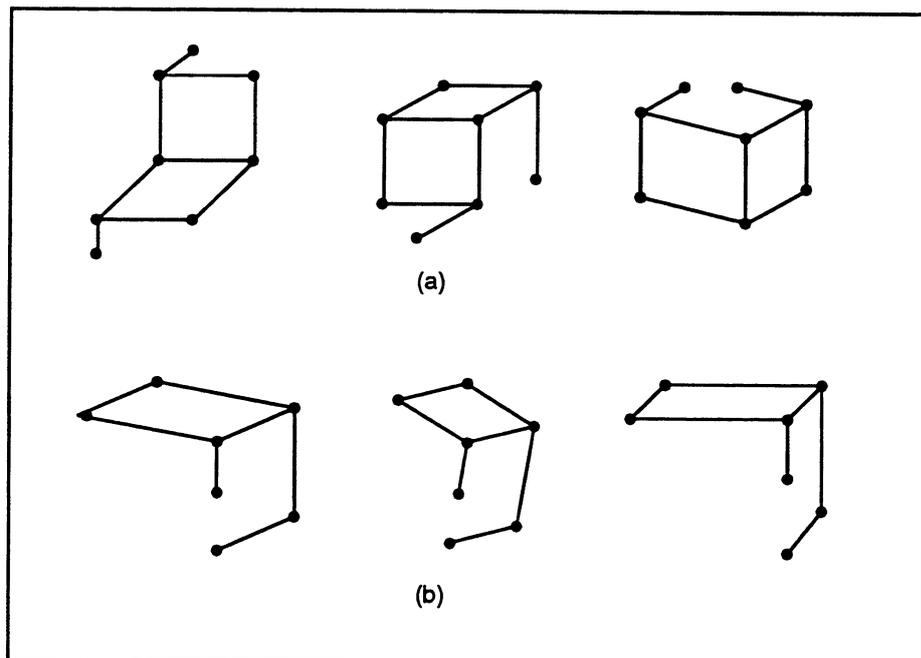


Figure 3.4: (a) Trois graphes ayant la même topologie mais non pas la même apparence. (b) Trois autres graphes ayant la même topologie et la même apparence.

Prenons comme exemple les graphes de la figure 3.4 (a). Dans cette figure on

peut voir que les trois graphes ont la même topologie, pourtant ils sont différents en forme ou en apparence. Evidemment, il s'agit du même graphe mais dessiné de trois manières complètement différentes.

Ce qu'on voudrait maintenant est de pouvoir se servir de la méthode déjà utilisée avec des matrices binaires pour opérer aussi sur des matrices pondérées différemment. Ceci devrait donc permettre de caractériser d'autres propriétés d'un graphe, par exemple son apparence, c'est-à-dire la forme dont le graphe est dessiné sur le plan. Ceci a comme but de distinguer le graphe avec d'autres graphes ayant la même topologie.

On voudrait donc, par exemple, faire la différence entre les trois graphes de la figure 3.4 (a) mais aussi pouvoir dire que les trois graphes de la figure 3.4 (b) "sont les mêmes", car ils ont "à peu près" la même forme.

Il est évident que ce problème est beaucoup plus difficile à résoudre que celui de la caractérisation de la topologie d'un graphe car non seulement les relations de connexité entre deux nœuds du graphe (l'existence d'une arête entre deux nœuds et le degré d'un nœud) doivent être prises en compte mais aussi comment les différentes arêtes sont placées les unes par rapport aux autres dans le graphe.

Une autre question consiste à savoir comment représenter cette information sous une forme matricielle en conservant les trois propriétés principales de la matrice laplacienne classique, à savoir :

- symétrique,
- semi-définie positive et
- singulière (la somme de chaque ligne est nulle).

afin de se servir du même polynôme d_2 , déjà utilisé pour calculer les coefficients à partir de la matrice laplacienne classique, pour évaluer maintenant les coefficients de la nouvelle matrice de structure.

Dans la partie 3.3.3 on a vu comment la méthode de calcul des coefficients du polynôme d_2 décrite dans la partie 3.3.2 nous permet de caractériser la topologie d'un graphe. Le reste de cette partie est consacré à décrire comment la même méthode de calcul des coefficients peut être aussi utilisée pour caractériser maintenant l'apparence d'un graphe. Ceci avec le but de pouvoir distinguer ce graphe avec d'autres graphes ayant éventuellement la même topologie mais non pas la même apparence.

Dans ce qui suit, nous allons considérer des graphes finis, simples, connexes et planaires, c'est-à-dire de graphes ne contenant pas de boucles, où deux nœuds sont reliés par une arête au plus, où aucun nœud est isolé et qui, en plus, peuvent être "enfoncés" dans le plan, c'est-à-dire lorsque le graphe est dessiné sur une surface S il n'y a pas deux arêtes qui s'intersectent.

Une *figure fondamentale* (FF) d'ordre k d'un graphe G est le sous-graphe formé par l'ensemble des arêtes qui sont à une distance k (en arêtes) d'un nœud i du graphe G . La figure 3.5 (b) montre les FF d'ordres $k = 1$ et $k = 2$ obtenues à partir du nœud numéro 6 du graphe G de la figure 3.5 (a).

Une figure fondamentale d'ordre $k = 1$ est appelée une *jonction*.

Dans ce travail on ne s'intéresse qu'aux figures fondamentales d'ordre $k \leq 2$.

Le nœud, à partir duquel la FF est générée est appelé le *nœud d'origine*, les nœuds à une distance d'une arête du nœud d'origine sont appelés les *nœuds primaires*, les nœuds à une distance de deux arêtes du nœud d'origine sont appelés les *nœuds secondaires*, et ainsi de suite.

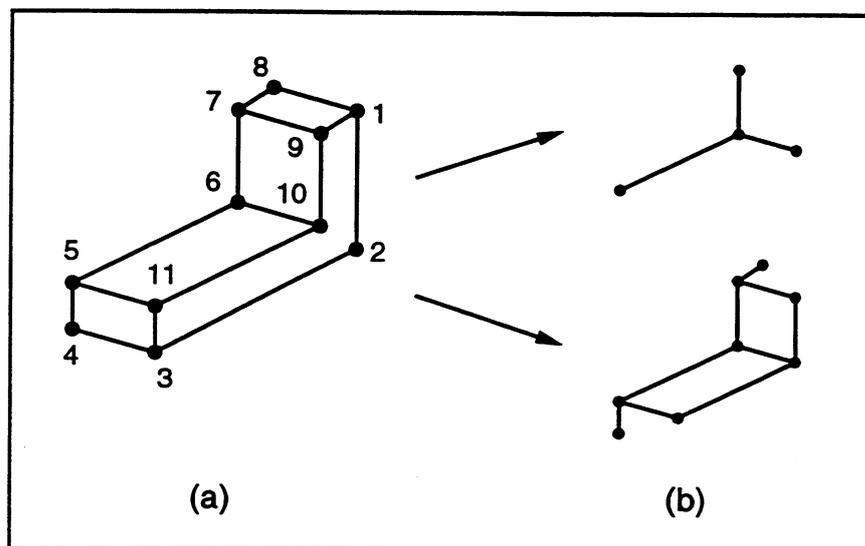


Figure 3.5: (a) Un graphe à 11 nœuds et à 15 arêtes. (b) Deux de ses figures fondamentales. L'une d'ordre 1, l'autre d'ordre 2, toutes les deux obtenues à partir du nœud numéro 6.

Une *branche d'une figure fondamentale* (BFF) est définie comme le sous-graphe formé par l'ensemble des arêtes qui sont à une distance $k = 1$ (en arêtes) du nœud d'origine et de n'importe quel nœud primaire de la FF. La figure 3.6 (b) montre les trois BFF de la FF de la figure 3.6 (a).

De la définition d'une BFF et de la figure 3.6 on peut voir qu'une BFF est tout simplement l'union de deux FF d'ordre $k = 1$ ayant une arête en commun.

Les deux jonctions formant une BFF sont désignées respectivement par j_o (pour le nœud d'origine) et j_p (pour le nœud primaire).

Nous allons considérer que ces jonctions peuvent appartenir à l'un de trois types : L, F et Y. Celles-ci sont montrées respectivement dans la figure 3.7 (a), dans la figure 3.7 (b) et dans la figure 3.7 (c). Les jonctions du type T (figure 3.7 (d)) con-

sidérées comme le résultat du phénomène du chevauchement sont traitées autrement (voir le chapitre 6).

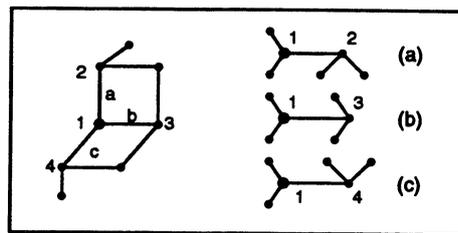


Figure 3.6: Une figure fondamentale et ses trois BFF.

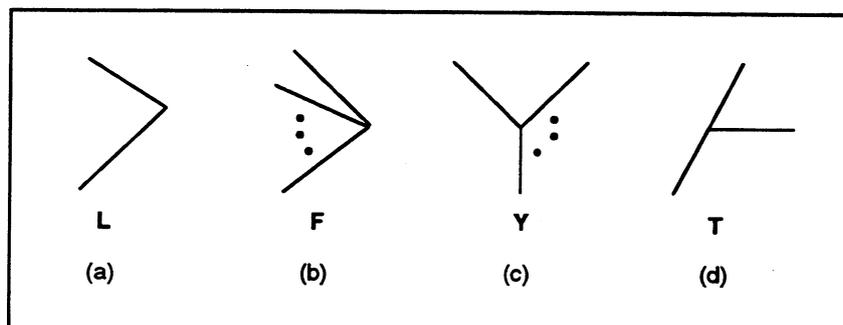


Figure 3.7: Les jonctions L, F et Y sont les trois types de jonctions considérés. La jonction de type T est considérée comme résultat du phénomène du chevauchement.

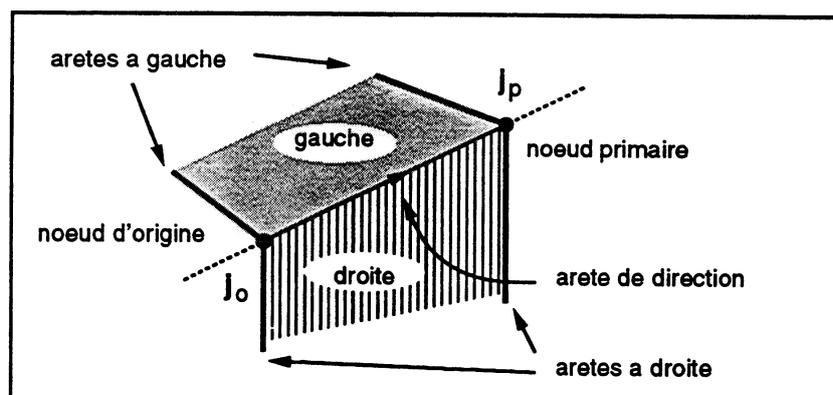


Figure 3.8: Les cotés “gauche” et “droite” d’une BFF sont déterminés par l’intermédiaire de son arête de direction.

Seules les BFF dont les deux jonctions sont de degrés $d_i \geq 2$ sont considérées.

L’arête reliant les deux jonctions d’une BFF prenant comme référence le noeud d’origine est appelée l’*arête de direction* de la BFF.

Les cotés *gauche* et *droit* d’une BFF sont déterminés respectivement par les régions à “gauche” à “droite” de l’arête de direction de la BFF. La figure 3.8 illustre ces derniers concepts.

Le nombre d'arêtes à gauche du nœud j_o , $N_g(j_o)$, est donné par le nombre d'arêtes situées à gauche de ce nœud et qui y sont connectées.

Le nombre d'arêtes à droite du nœud j_o , $N_d(j_o)$, est donné par le nombre d'arêtes situées à droite de ce nœud et qui y sont connectées.

$N_g(j_p)$ and $N_d(j_p)$ représentent respectivement le nombre d'arêtes à gauche et à droite du nœud primaire d'une BFF. Ainsi, pour l'exemple, de la figure 3.8 on a : $N_g(j_o) = 1$, $N_d(j_o) = 1$, $N_g(j_p) = 1$ and $N_d(j_p) = 1$.

Le code d'une BFF noté $C(BFF)$ est un entier qui est une fonction des types de ses deux jonctions (L, F, Y) et du nombre d'arêtes à "gauche" et à "droite" de ces jonctions. Celui-ci peut-être obtenu par l'approche suivante :

1. **choisir** le degré maximal d'une jonction (le nombre d'arêtes attachées à la jonction). Cette valeur va déterminer le nombre des BFF possibles.
2. **classer** les BFF, selon la taille maximale choisie, dans l'ordre : LL, LF, LY, FL, FF, FY, YL, YF, YY et de façon ascendante en fonction du nombre d'arêtes à gauche et à droite des ces deux jonctions.
3. **étiqueter** les BFF en commençant par donner un "1" à la première BFF, un "2" à la deuxième BFF et ainsi de suite, en éliminant en même temps les BFF résultant d'une opération de réflexion.
4. **fournir** la liste des BFF avec leur codes. Ceux-ci, sont égaux aux étiquettes données aux BFF en 3.

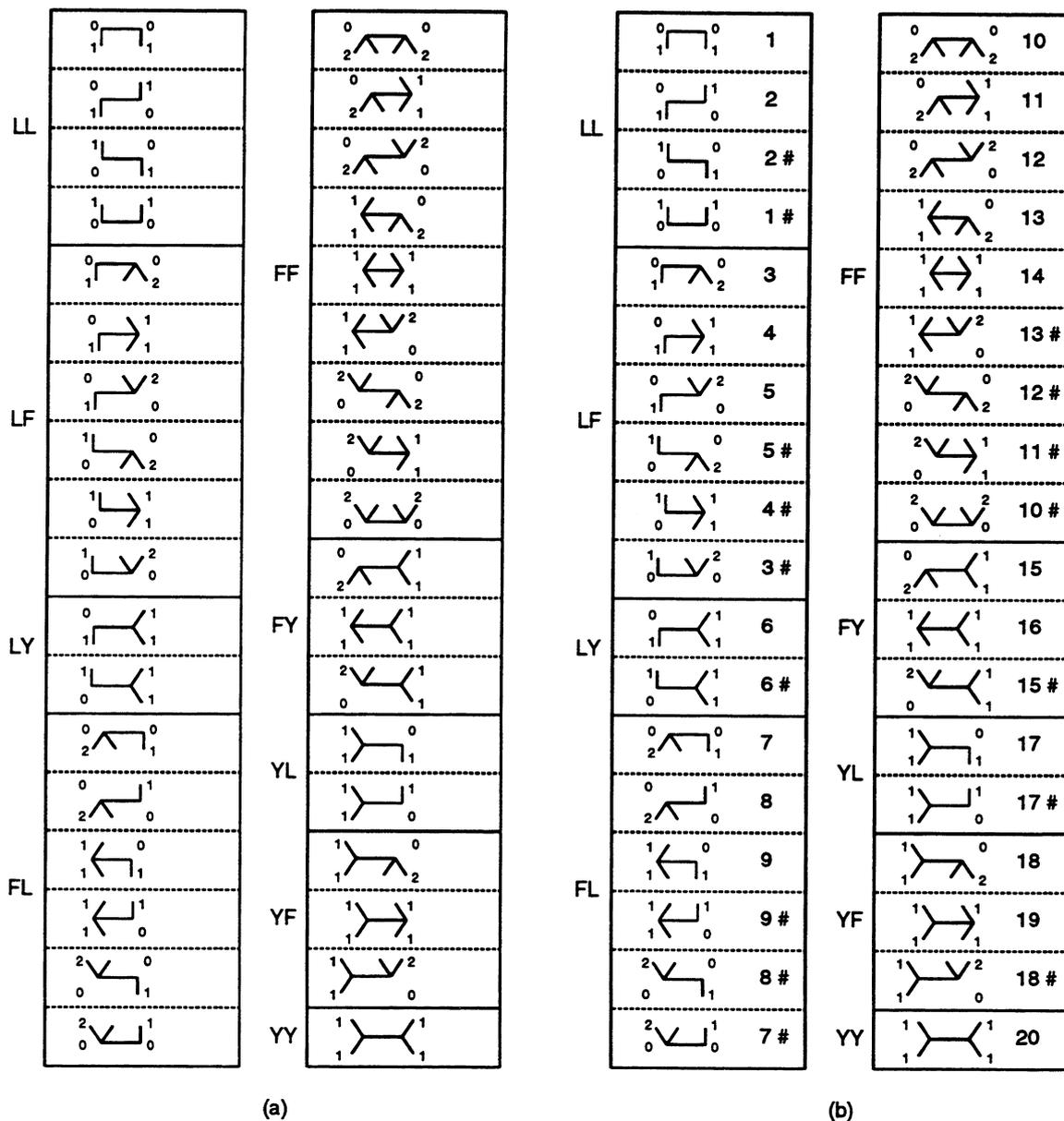
Pour mieux comprendre comment cette approche fonctionne, nous allons trouver l'ensemble de BFF et leurs codes correspondants lorsque la taille maximale pour une jonction est fixée, par exemple, à trois arêtes.

En tenant compte de cette contrainte, il n'est pas difficile de vérifier que les BFF sont classées selon la figure 3.9 (a). Une fois que les BFF ont été ordonnées, on les étiquette en éliminant, en même temps, les BFF qui sont le résultat d'une réflexion, (voir la figure 3.9 (b)). Le code d'une BFF est donc égal à son étiquette. Le tableau de la figure 3.10 montre les codes pour toutes les BFF possibles avec $d_i \leq 3$.

A ce moment, nous avons la possibilité de définir une nouvelle matrice pour représenter l'apparence locale du graphe décrivant une figure fondamentale quelconque.

La *matrice d'apparence locale* $M(G)$ d'un graphe G est une matrice de taille $q \times q$ dont l'entrée (m_{ij}) est donnée par :

$$m_{ij} = \begin{cases} D_i & \text{si } i = j \\ -C(BFF) & \text{s'il y a une arête entre les nœuds } i \text{ et } j \\ 0 & \text{sinon} \end{cases} \quad (3.18)$$



		types de jonctions																	
		L/L	L/F	L/Y	F/L	F/F	F/Y	Y/L	Y/F	Y/Y									
numero d'aretes a gauche et a droite		1																	
		2																	
			3																
				4															
					5														
						6													
							7												
								8											
									9										
										10									
											11								
												12							
													13						
														14					
															15				
																16			
																	17		
																		18	
																			19

Figure 3.10: Le code d'une BFF est un entier qui est une fonction des types de ses jonctions et du nombre d'arêtes à gauche et à droite de ces jonctions.

où $D_i = \sum_{j=0}^q C(BFF)_j \forall i \neq j$ et q désigne le nombre de nœuds primaires du graphe G plus un (le nœud d'origine). La figure 3.11 (c) montre la matrice $M(G)$ pour la FF de la figure 3.11 (a) pour l'étiquetage montré.

D'après l'équation 3.18 et l'exemple de la figure 3.11 (b) on peut voir que la matrice $M(G)$ a les mêmes propriétés que la matrice laplacienne classique. En conséquence, tous les résultats appliqués à cette dernière matrice sont aussi valables pour la matrice d'apparence.

Comme exemple d'application, essayons de trouver l'ensemble des coefficients c'_i du polynôme d_2 qui caractérisent l'apparence locale du graphe de la figure 3.11 (a). Il est facile de vérifier que cet ensemble de valeurs correspond à celui montré dans la figure 3.11 (d).

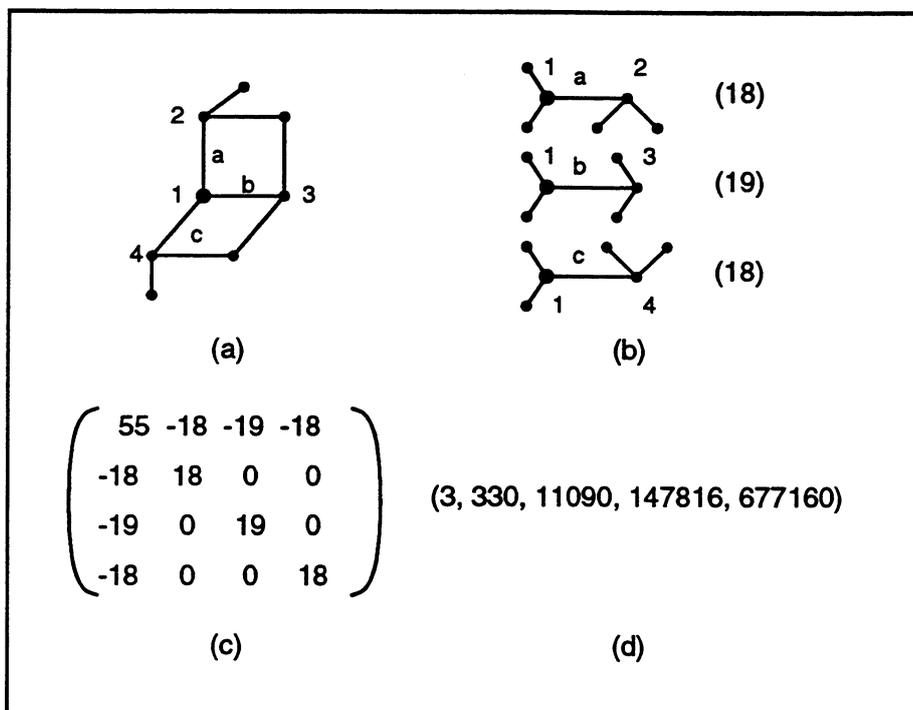


Figure 3.11: (a) Une figure fondamentale à 8 nœuds et à 9 arêtes, (b) ses trois BFF avec leurs codes, (c) sa matrice d'apparence locale correspondante et (d) l'ensemble des coefficients m_i du polynôme $d_2(M(G))$ caractérisant la FF.

3.5 Conclusion sur la méthode de caractérisation

Dans ce chapitre, nous avons :

1. décrit une méthode générale provenant de la théorie de graphes pour caractériser un graphe G à n nœuds et à m arêtes.
2. utilisé cette méthode pour caractériser deux propriétés du graphe : sa topologie et son apparence.

Par rapport à la topologie du graphe, nous avons vu :

- que celle-ci est caractérisée par un ensemble de $n + 1$ entiers : c_0, c_1, \dots, c_n ,
- que ces entiers décrivent de façon unique la topologie du graphe lorsque la taille du graphe $n \leq 12$,
- qu'ils sont les coefficients du polynôme d_2 et
- qu'ils peuvent être facilement évalués à partir de la matrice laplacienne du graphe en utilisant le Théorème 3.1.

Par rapport à l'apparence du graphe nous avons vu :

- que celle-ci est caractérisée localement (pour l'ensemble des arêtes les plus proches du "centre" du graphe) par un ensemble de $q + 1$ entiers : c'_0, c'_1, \dots, c'_q ,
- que ces entiers sont aussi les coefficients du polynôme d_2 , mais appliqué cette fois-ci à la matrice d'apparence $M(G)$ et
- que q représente, dans ce cas, le nombre de nœuds primaires du graphe G plus un (le nœud d'origine).

Finalement, nous avons vu que la complexité de calcul des coefficients du polynôme $d_2(B)$ pour un graphe G à n nœuds et à m arêtes est n^4 .

Chapitre 4

La technique d'indexation choisie

L'une des questions les plus importantes actuellement pour la communauté de la vision par ordinateur est d'organiser l'information de la base de données pour un système de reconnaissance d'objets, avec un accès efficace. Cette tâche est d'autant plus difficile que la taille de la base considérée commence à croître et que les objets quelle contient sont complexes.

Lorsque dans la base de données, il n'y a qu'un seul objet et lorsque le nombre des primitives décrivant cet objet reste petit, la mise en correspondance est, en général, résolue d'une manière satisfaisante par l'intermédiaire de techniques de recherche exhaustive.

D'autres implémentations plus performantes pour résoudre ce problème prendront la forme soit d'un graphe de recherche (Bolles et Cain [12]), soit d'un arbre de recherche (Ayache et Faugeras [4], Faugeras et Hebert [36], Bolles et Horaud [13], Horaud [54], Lowe [70], Grimson et Lozano-Pérez [44] et Pollard [88]).

Or, si la complexité et le nombre des primitives composantes augmentent alors ces implémentations éclatent car le processus de mise en correspondance est combinatoire dans la complexité de calcul. Une solution pour affronter ce problème a été d'utiliser un certain nombre de contraintes. En supposant les objets rigides l'espace de recherche (la liste, le graphe ou l'arbre) est réduit considérablement.

Ainsi, n'importe quel processus (même celui de recherche exhaustive) est contraint de visiter quelques centaines de nœuds au lieu de millions.

Néanmoins, si beaucoup d'objets et non un seul sont présents dans la base de modèles, même les processus de recherche mentionnés éclatent à nouveau car la complexité de calcul croît aussi avec le nombre d'objets (Grimson [43]). Une solution consiste à inclure un mécanisme d'indexation avant l'étape d'appariement pour améliorer la performance du système en limitant la liste des modèles, candidats à la reconnaissance, à quelques uns.

Dans ce chapitre nous décrivons une nouvelle méthode pour l'organisation de la base de données d'un système de reconnaissance d'objets 3D. Cette méthode qui est basée sur l'approche de caractérisation décrite dans le chapitre 3 permet au système, comme nous allons le voir dans le chapitre 6, de donner une solution au problème de reconnaissance par l'indexation en présence d'une vaste base de modèles.

La suite de ce chapitre est organisée comme suit. Dans la partie 4.1 nous présentons le problème de l'organisation de la base données pour un système de reconnaissance sous une nouvelle forme. On opère maintenant sur un ensemble de graphes que l'on stocke dans la base et que l'on essaie de retrouver rapidement.

Dans la partie 4.2 nous décrivons de façon brève la technique de hachage comme l'une des solutions possibles à ce problème alors que dans la partie 4.3 nous exposons l'approche proposée. Cette approche de construction de la base et d'indexation opère à l'aide de la méthode de caractérisation décrite dans le chapitre 3.

La partie 4.4 expose la structure de la base de données résultante.

Finalement, dans la partie 4.5 nous présentons quelques commentaires.

4.1 Le problème

Nous considérons maintenant le problème suivant. Étant donné une grande collection de graphes de taille n telle que $5 \leq n \leq 10$ qui sont stockés dans une base de données, nous voulons envisager une méthode pour savoir rapidement si un graphe donné est présent dans la base de données ou pas.

Donner une solution à ce problème exige la réponse aux questions suivantes :

1. Quelle information faut-il stocker sur le graphe? Quels attributs : sa taille, son nombre d'arêtes, les degrés d_i des ses nœuds ...?
2. Comment stocker cette information dans la base? Sous quelle forme : arbre de recherche, tableaux de hachage ...?
3. Enfin, comment y avoir accès rapidement?

4.2 Une solution

Une façon d'accomplir cette recherche consiste à utiliser la technique connue de *hachage* (Sedgewick [94]). Nous pouvons résumer cette opération comme suit (voir la figure 4.1). A chaque objet de la base de données est associé une clé numérique. Cette clé est transformée en adresse d'un tableau de taille gérée par l'intermédiaire d'une *fonction de hachage*. L'adresse d'un objet ainsi évaluée est aussi appelée le code de hachage de l'objet.

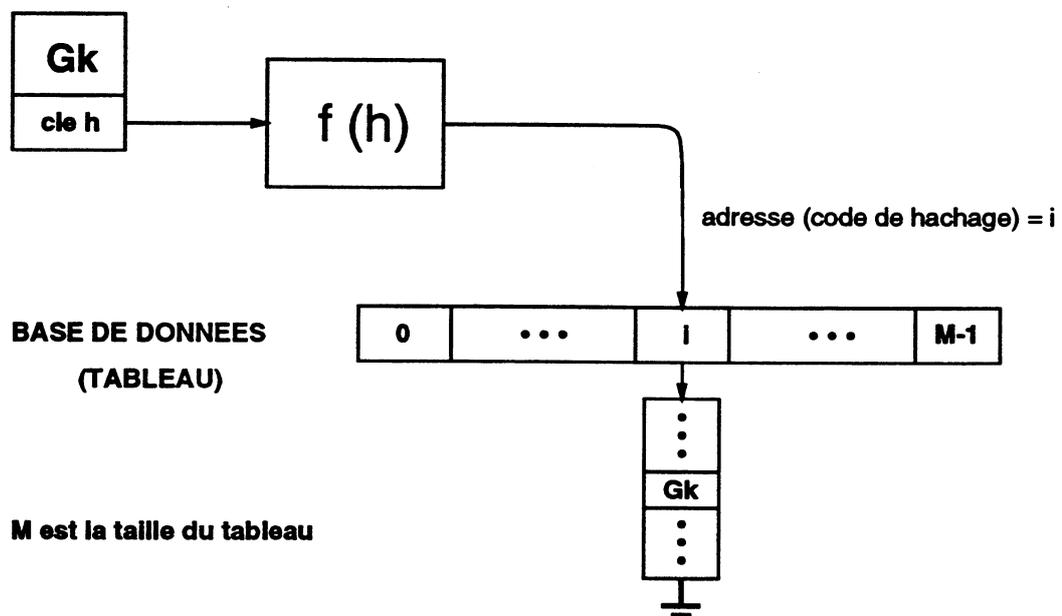


Figure 4.1: La clé associée à chaque objet de la base de données est transformée en adresse d'un tableau de taille gérée par l'intermédiaire d'une fonction de hachage.

Le processus de *construction de la base de données* consiste à évaluer précisément ce code de hachage pour chacun des objets à stocker dans la base d'objets. Plusieurs objets peuvent avoir la même adresse (le même code de hachage). Ainsi, une liste d'objets sera associée à chaque adresse. On dit, dans ce cas, que la base de données prend la forme d'un tableau de hachage où à chaque adresse est associée une liste d'objets.

Le processus d'*indexation* (figure 4.2) consiste à évaluer l'adresse, c'est-à-dire le code de hachage d'un objet inconnu afin de déterminer si cet objet se trouve dans le tableau de hachage ou pas.

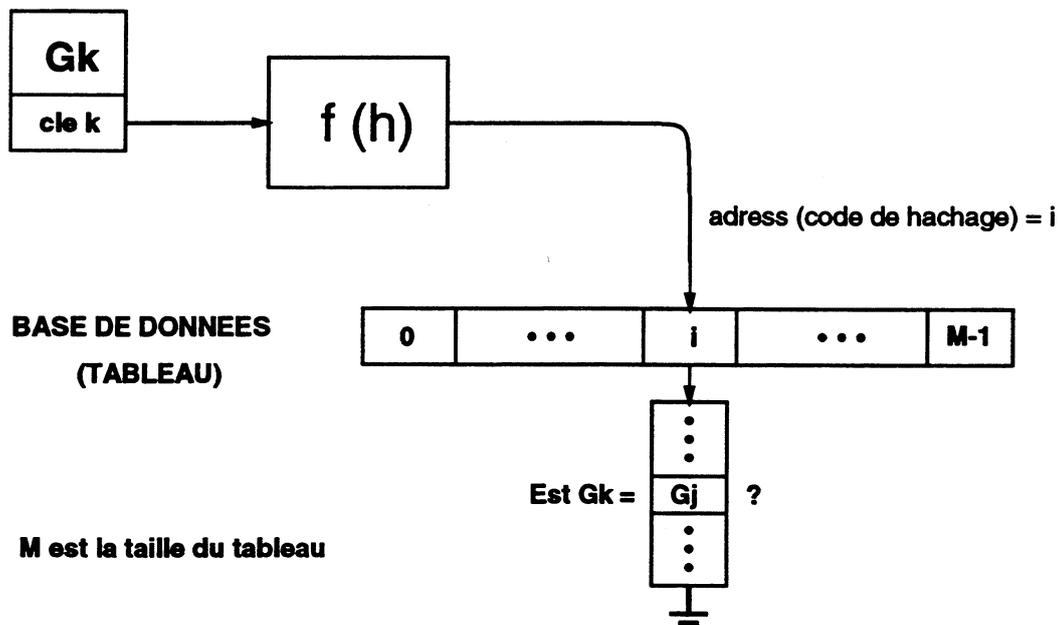


Figure 4.2: Le mécanisme d'indexation évalue le code de hachage d'un objet inconnu à fin de déterminer si cet objet se trouve dans le tableau de hachage ou pas.

4.3 Notre approche : hachage des graphes

Un inconvénient des techniques de hachage, mêmes des plus récentes, est précisément le problème que plusieurs objets peuvent partager le même code de hachage. Ainsi, si la taille de la liste des objets attachés à une telle adresse est très grande alors la vitesse du processus de recherche (d'un objet particulier) se verra sévèrement affectée détériorant ainsi évidemment les performances du système. Ce problème des techniques de hachage est connu dans la littérature avec le nom de *collision* Sedgewick [94] et Horowitz et Sahni [56] et représente aujourd'hui l'un des défis de plusieurs chercheurs en informatique.

Le but de cette partie est de présenter une méthode qui permet de faire du hachage en éliminant pratiquement les problèmes de collision par l'utilisation d'une fonction de hachage qui associe à chaque objet (à stocker ou à chercher dans la base d'objets) une clé unique.

Cette fonction de hachage est possible grâce à l'approche de caractérisation de graphes décrite dans le chapitre 3.

D'après ce chapitre, il y a deux propriétés d'un graphe G à n nœuds et m arêtes qui peuvent être caractérisées :

1. sa topologie, par un ensemble de $n + 1$ coefficients : c_0, \dots, c_n , obtenus à partir du polynôme d_2 appliqué à la matrice laplacienne $L(G)$ associée au graphe et

2. son apparence, par un ensemble de $q + 1$ coefficients : c'_0, \dots, c'_q , obtenus à partir du polynôme d_2 appliqué à la matrice d'apparence $M(G)$ associée au graphe.

Dans le premier cas, nous avons vu dans le chapitre 3 que la topologie du graphe est caractérisée de façon unique lorsque la taille n du graphe reste inférieure à 13 nœuds.

Dans le deuxième cas, nous avons constaté qu'au moins pour les exemples trouvés dans la pratique l'apparence locale du graphe est caractérisée de façon unique pour une bonne partie de l'ensemble de graphes spéciaux appelés figures fondamentales.

Dans ce que suit, nous allons voir comment la base d'objets, les graphes étant ces objets, est organisée en fonction de ces deux ensembles de coefficients.

4.4 La structure de la base de données choisie

Selon le chapitre 3, un graphe est caractérisé par les ensembles des coefficients $\{c_0, \dots, c_n\}$ et $\{c'_0, \dots, c'_q\}$. Notons que c_0 donne la taille (le nombre des nœuds) du graphe et, pour des graphes d'une taille fixe le reste des coefficients : c_1, \dots, c_n et c'_0, \dots, c'_q peut être utilisé comme la clé associée à chaque graphe. En raison de l'unicité de la clé associée à chaque graphe, la clé est tout simplement le code de hachage du graphe correspondant.

La figure 4.3 montre la structure de la base de données. Il y a autant de tableaux de hachage que de tailles des graphes.

Le *hachage des graphes* se sert de c_0 et, du reste des coefficients : c_1, \dots, c_n et c'_0, \dots, c'_q de façon séquentielle (figure 4.3). Un tableau "vertical" (en bas à gauche de la figure) permet la sélection de l'un des tableaux de hachage (en bas à droite de la figure) pointé par la valeur de c_0 . Ensuite une combinaison appropriée des coefficients : c_1, \dots, c_n et c'_0, \dots, c'_q (voir chapitre 6) permet de sélectionner l'une des adresses dans le tableau de hachage correspondant. Cette adresse représente l'endroit où le graphe se trouve.

Il y a deux avantages à ce choix :

1. la valeur du coefficient c_0 donne directement la taille du graphe. Ceci oblige le système à comparer des graphes dont la taille est la même et
2. l'unicité de la clé associée au graphe permet la sélection d'un seul graphe dans la base de données. Ceci élimine les problèmes de collision.

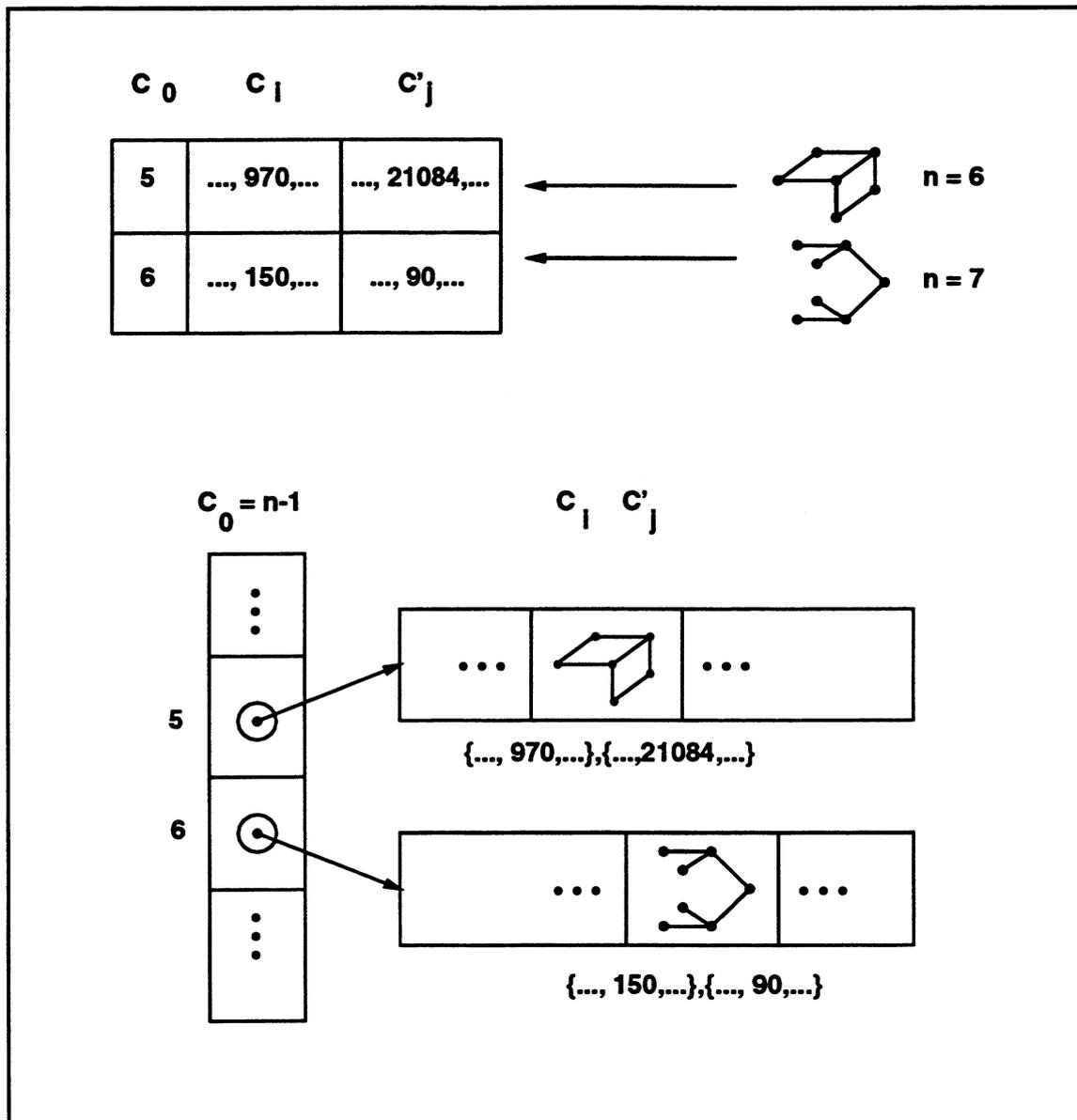


Figure 4.3: La structure de la base de graphes. Le haut de la figure montre deux graphes simples avec 6 et 7 nœuds et les valeurs de c_0, c_1, \dots, c_n et c'_0, \dots, c'_q pour ces graphes. Le bas de la figure montre l'emplacement de ces deux graphes dans les tableaux de hachage. Une double indexation est nécessaire pour accéder à chaque graphe dans cette base de données.

4.5 Conclusions sur la méthode d'indexation choisie

Pour conclure, dans ce chapitre nous avons présenté une technique d'indexation sous la forme de hachage des graphes.

Le hachage des graphes est possible grâce à l'approche de caractérisation d'un graphe décrite dans le chapitre 3.

La base de données obtenue par l'intermédiaire de cette méthode peut contenir tous les graphes dont la taille $5 \leq n \leq 10$. En outre, l'adresse de chaque tableau de hachage correspond à un seul graphe.

Chapitre 5

Un système d'indexation de modèles

Dans ce chapitre, nous discutons notre système d'indexation de modèles. Tout d'abord, dans la partie 5.1 nous discutons comment les différents objets qui formeront la base de données sont modélisés. Ensuite, dans la partie 5.2, nous expliquons comment la base de modèles résultante est organisée. Après, dans la partie 5.3, nous discutons du mécanisme d'indexation. Finalement, dans la partie 5.4, nous concluons.

5.1 La modélisation d'un objet

La modélisation d'objets a été largement étudiée dans le domaine de la vision par ordinateur. Dans le chapitre 2 nous offrons un état de l'art. D'après ce chapitre, nous avons vu qu'en général il y a deux classes de représentations possibles :

1. à objet centré et
2. à observateur centré.

Dans notre approche nous utilisons une représentation qui n'est attachée à aucun de ces deux repères. Un objet est, par essence, décrit par une liste de vues caractéristiques. Ces vues caractéristiques sont telles que la plupart des surfaces de l'objet soient contenues dans au moins l'une des vues caractéristiques. Dans la représentation que nous utilisons, la définition d'une vue caractéristique (VC) doit être comprise au sens large. Il s'agit d'un réseau de primitives objet (indices visuels) et des relations entre ces primitives. Ces primitives sont visibles simultanément d'un point de vue quelconque. Cette représentation n'utilise qu'une partie des aspects du graphe pour des raisons d'économie d'espace et de vitesse de calcul. Les primitives dans le réseau peuvent aussi bien être 2-D ou 3-D, à objet centré ou à observateur centré. La propriété importante ici n'est pas la dimension des primitives ou le repère auquel celles-ci sont attachées, mais plutôt les propriétés *intrinsèques* du réseau de primitives. Comme nous l'avons déjà mentionné, un tel réseau de primitives peut être représenté de façon conventionnelle par un graphe dont les nœuds représentent l'information concernant le degré des jonctions dans la VC et dont les arêtes représentent l'information relative à la connexité entre les jonctions dans l'image.

En général l'information saisie par un capteur encode rarement une VC complète associée à une vue d'un objet. L'information est corrompue par le bruit, par l'occlusion et par des positions de vue accidentelles dans l'image. En conséquence, il n'est pas envisageable d'encoder directement les vues caractéristiques dans les tableaux de hachage et d'essayer de les indexer. En revanche, nous avons ajouté une couche de plus à notre méthode de représentation. Chaque VC est décomposée en un ensemble de figures fondamentales (FF) se superposant éventuellement les unes aux autres. Chaque FF est tout simplement un sous-graphe du graphe encodant la VC. En résumé, une VC (un grand graphe) est faite de figures fondamentales (des petits graphes). Il y a plusieurs raisons en faveur de ce choix :

- on peut comparer seulement des graphes dont la taille est la même. Les graphes "bruts" extraits à partir de l'information de l'image ont rarement la même taille que les graphes associés aux vues caractéristiques des objets.
- l'approche de caractérisation fondée sur l'évaluation des coefficients du polynôme d_2 est efficace pour les graphes avec une taille $n \leq 12$ (voir le chapitre 3).
- un petit groupe de primitives image est plus facile à détecter qu'un groupe plus grand. En outre, à cause du problème d'occlusion, il est impossible de trouver dans une image un grand groupe de primitives qui appartiennent au même objet.
- l'ensemble de figures fondamentales croît de façon sous linéaire lorsque on ajoute des objets c'est-à-dire, des vues caractéristiques à la base de modèles. Autrement dit, l'ensemble des figures fondamentales est un "alphabet" permettant un grand "dictionnaire" de vues caractéristiques.

5.1.1 De la vue caractéristique à la figure fondamentale

La décomposition d'une vue caractéristique en figures fondamentales n'est pas une tâche facile. Elle est liée à la tâche qui consiste à décomposer un graphe en sous-graphes sous certaines contraintes. Jusqu'à présent nous n'avons pas de technique de décomposition d'usage général. Au lieu de cela nous utilisons une technique simple qui considère, pour chaque nœud du graphe, un sous-graphe "autour" de ce nœud central. Un tel sous-graphe n'est qu'une figure fondamentale, déjà définie dans le chapitre 3. Par exemple, la figure 5.1 montre une vue caractéristique d'un objet simple (a) et quelques unes de ses figures fondamentales (b,c,d,e,f).

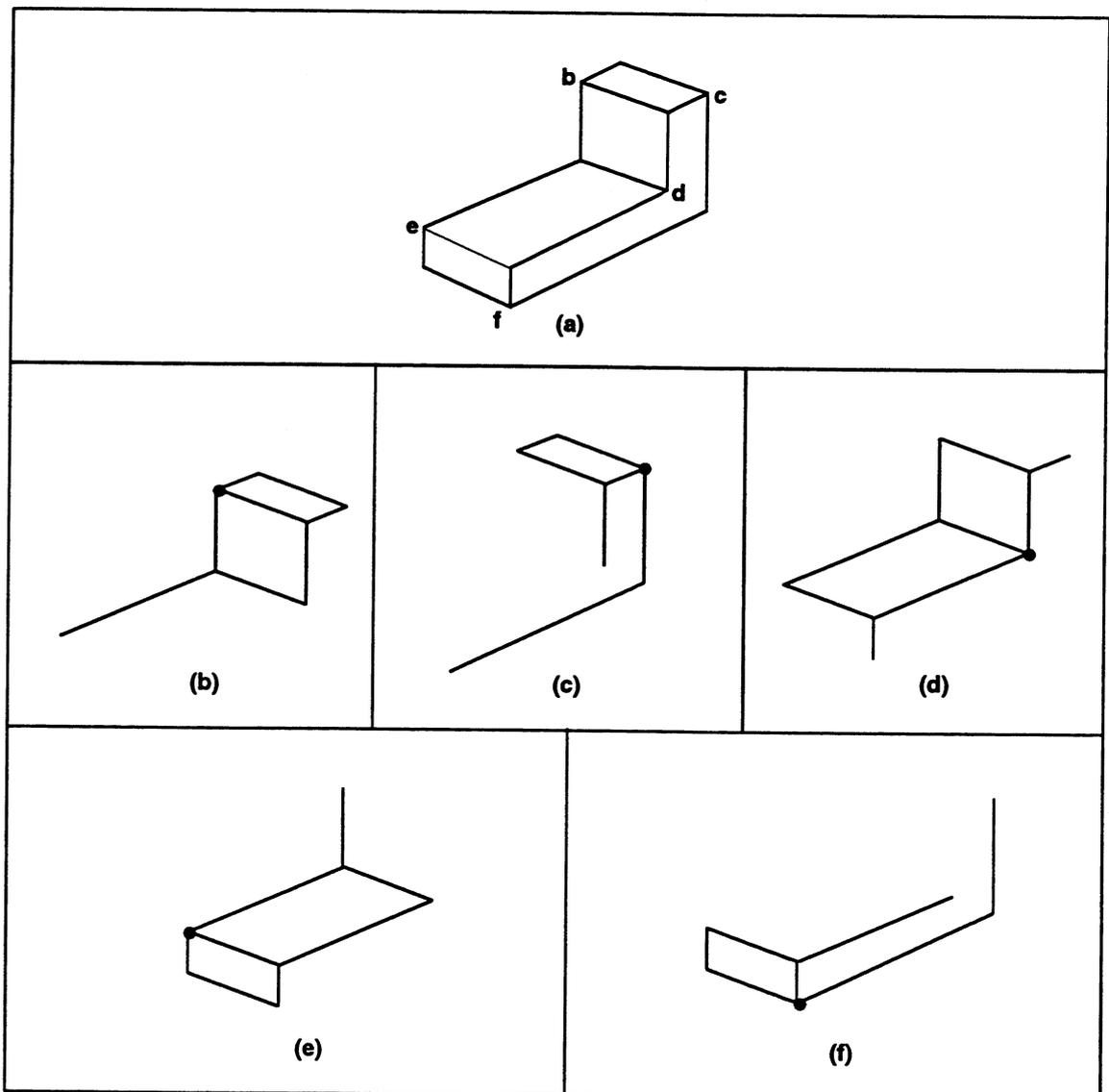


Figure 5.1: Un exemple d'une vue caractéristique d'un objet simple en forme de "L" et quelques unes de ses figures fondamentales.

Notons comment une même figure ((b) par exemple) peut apparaître plusieurs fois dans la même vue et comment les figures partagent les mêmes sommets et arêtes. Cette redondance est normale due à la nature de la méthode de génération utilisée. Ceci est pourtant important car dans certaines circonstances cela va nous permettre d'affronter les problèmes du bruit (l'absence de portions de segments ou même de segments complets, dédoublements de nœuds, apparition de segments fictifs provoqués par des ombres, la texture ...) et de l'occlusion.

Un compromis doit être fait pour la taille des figures fondamentales. Si la taille de la figure fondamentale est petite celle-ci peut appartenir à des milliers d'objets. En conséquence le processus d'indexation de modèles ne serait pas utile dans ce cas. En revanche, si la taille de la figure est très grande celle-ci est moins ambiguë et permet alors de pointer de façon unique vers une seule vue modèle. Le problème est que ces grandes figures sont beaucoup plus difficiles à détecter et à traiter.

Dans la pratique, la taille des figures fondamentales, c'est-à-dire leur nombre de nœuds, varie entre 5 et 10. Evidemment, ce choix est, d'une part cohérent avec la caractérisation des graphes par le polynôme d_2 , mais d'autre part, elle pose le problème d'ambiguïté car cette figure fondamentale peut appartenir à beaucoup de vues caractéristiques différentes.

5.1.2 De la figure fondamentale à l'intersection de figures fondamentales

Afin de surmonter ce problème, nous avons décidé d'introduire une caractérisation des relations possibles entre deux figures fondamentales. Pour chaque paire de figures fondamentales un nouveau arrangement est obtenu : l'intersection de figures fondamentales (IFF). Une intersection de figures fondamentales (IFF) est tout simplement l'ensemble des nœuds et des arêtes qui sont communes aux deux figures fondamentales. Comme chaque IFF est aussi un graphe, elle peut être aussi caractérisée par les coefficients du polynôme d_2 . La figure 5.2 montre les dix IFF obtenues à partir de l'ensemble des cinq figures fondamentales de la vue caractéristique de l'objet en forme de "L" de la figure 5.1. De même qu'avec les FF, notons comme la même IFF peut apparaître plusieurs fois dans la même vue caractéristique ((b,g) et (a,f)).

Afin qu'une intersection de figures fondamentales soit considérée comme acceptable il faut que :

- sa taille en nombre de nœuds soit supérieure ou égale à un seuil et
- elle ne contienne aucun nœud isolé, c'est-à-dire que le graphe associé soit connexe.

Si nous considérons, par exemple, que la taille minimale admissible pour une IFF est de 4 nœuds, seules les IFF (a,b,c,g et i) de la figure 5.2 sont considérées comme acceptables.

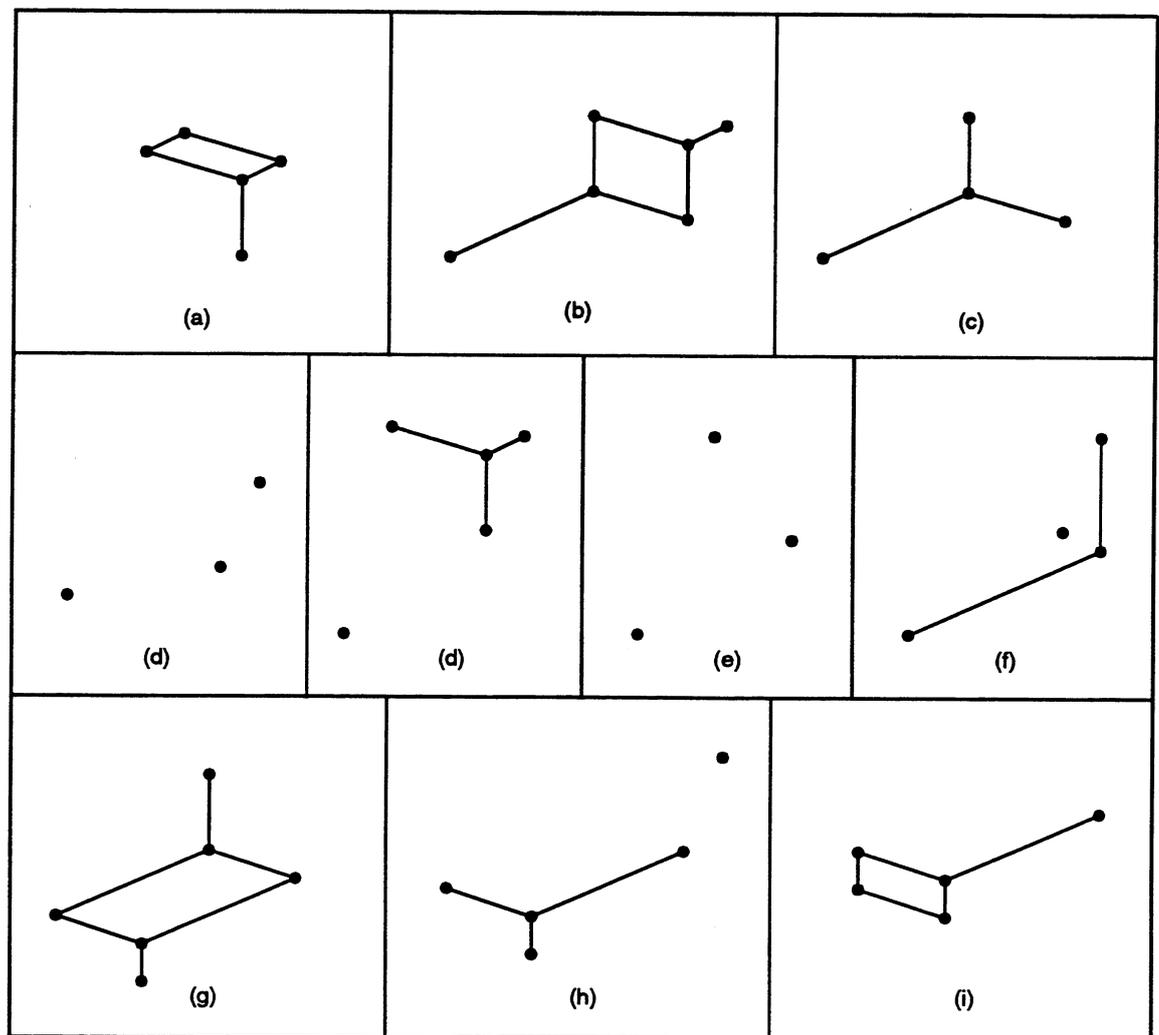


Figure 5.2: L'ensemble des IFF correspondant à l'ensemble des figures fondamentales de la vue caractéristique de l'objet en forme de "L".

En résumé, notre représentation des objets contient trois couches (voir la figure 5.3). La couche supérieure est une description à objet centré. Dans le niveau intermédiaire, l'objet est décrit par une liste de vues caractéristiques. Finalement, au niveau inférieure une vue caractéristique est décrite par une liste de figures fondamentales et une liste des intersections des figures fondamentales.

5.2 L'organisation de la base de modèles

L'organisation de la base de modèles suit la structure à trois couches décrite dans le paragraphe précédent, mais à l'inverse. La couche la plus basse contient une liste de figures fondamentales et une liste d'intersections de figures fondamentales. La couche

intermédiaire contient une liste de vues caractéristiques. Enfin, dans la couche supérieure, on a une liste d'objets. Seules les FF et les IFF sont organisées dans les tableaux de hachage suivant la technique d'indexation décrite dans le chapitre 4. La figure 5.4 montre un exemple d'une base de modèles simple composée de trois objets, six VC et quatre FF. Seul deux IFF sont montrées dans cette figure. Notez que les vues VC2 et VC3 contiennent toutes les deux les figures FF1 et FF2. Pourtant, leur intersection IFF1 en VC2 est différente de celle en VC3 (IFF2), de même qu'avec les mots "plaine" et "alpine" qui contiennent les mêmes lettres mais dans un ordre différent.

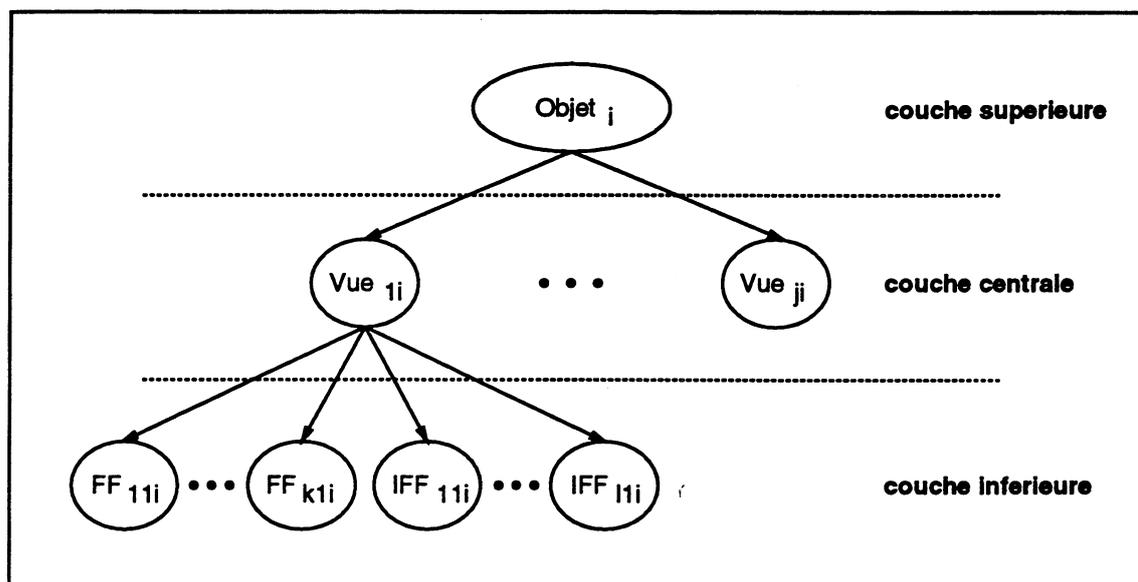


Figure 5.3: La représentation d'un objet contient trois couches : une couche supérieure contenant une description à objet centré, une couche centrale où l'objet est décrit par une liste de vues caractéristiques et une couche inférieure où chaque vue caractéristique est décrite par une liste de figures fondamentales et une liste des intersections des figures fondamentales.

5.3 Le mécanisme d'indexation

Le mécanisme d'indexation prend en entrée un ensemble de figures fondamentales et leurs intersections. On suppose que celles-ci ont été extraites d'une image. Le mécanisme consiste donc à déterminer quel modèle (ou modèles) contient ces figures afin de déterminer quel modèle (ou modèles) est probablement présent dans l'image.

Pour mieux fixer les idées, considérons l'exemple simple montré dans la figure 5.4. Supposons que FF1, FF2 et IFF2 ont été détectées dans l'image. Le processus d'indexation procède alors comme suit (voir la figure 5.5) :

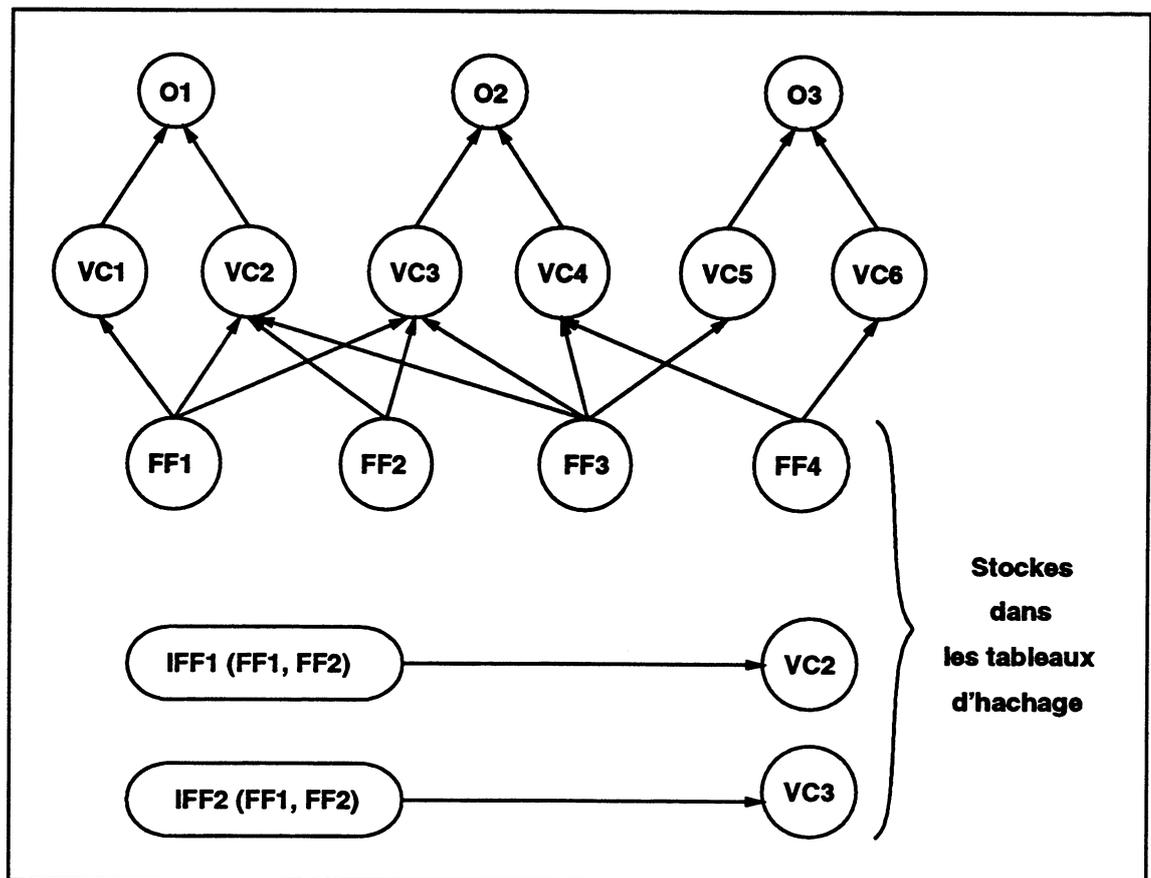


Figure 5.4: Un exemple d'une base de données simple. Elle est composée de trois objets, six VC, quatre FF et deux IFF. Seule les FF et les IFF sont stockés dans les tableaux de hachage.

1. le code de hachage de la figure fondamentale FF1 est évalué,
2. les tableaux de hachage sont parcourus en utilisant le code de hachage de FF1 afin de fournir la liste des vues caractéristiques contenant FF1. Dans ce cas, il s'agit des vues VC1, VC2 et VC3,
3. le code de hachage de la figure fondamentale FF2 est évalué,
4. en utilisant ce code de hachage, VC2 et VC3 sont sélectionnées,
5. l'intersection des deux listes (VC1, VC2, VC3) et (VC2, VC3) restreint la liste de vues caractéristiques à VC2 et VC3.
6. le code de hachage de IFF2 est évalué,
7. en utilisant ce code et le fait que IFF2 est le résultat de l'intersection de FF1 et FF2, VC3 est retournée,

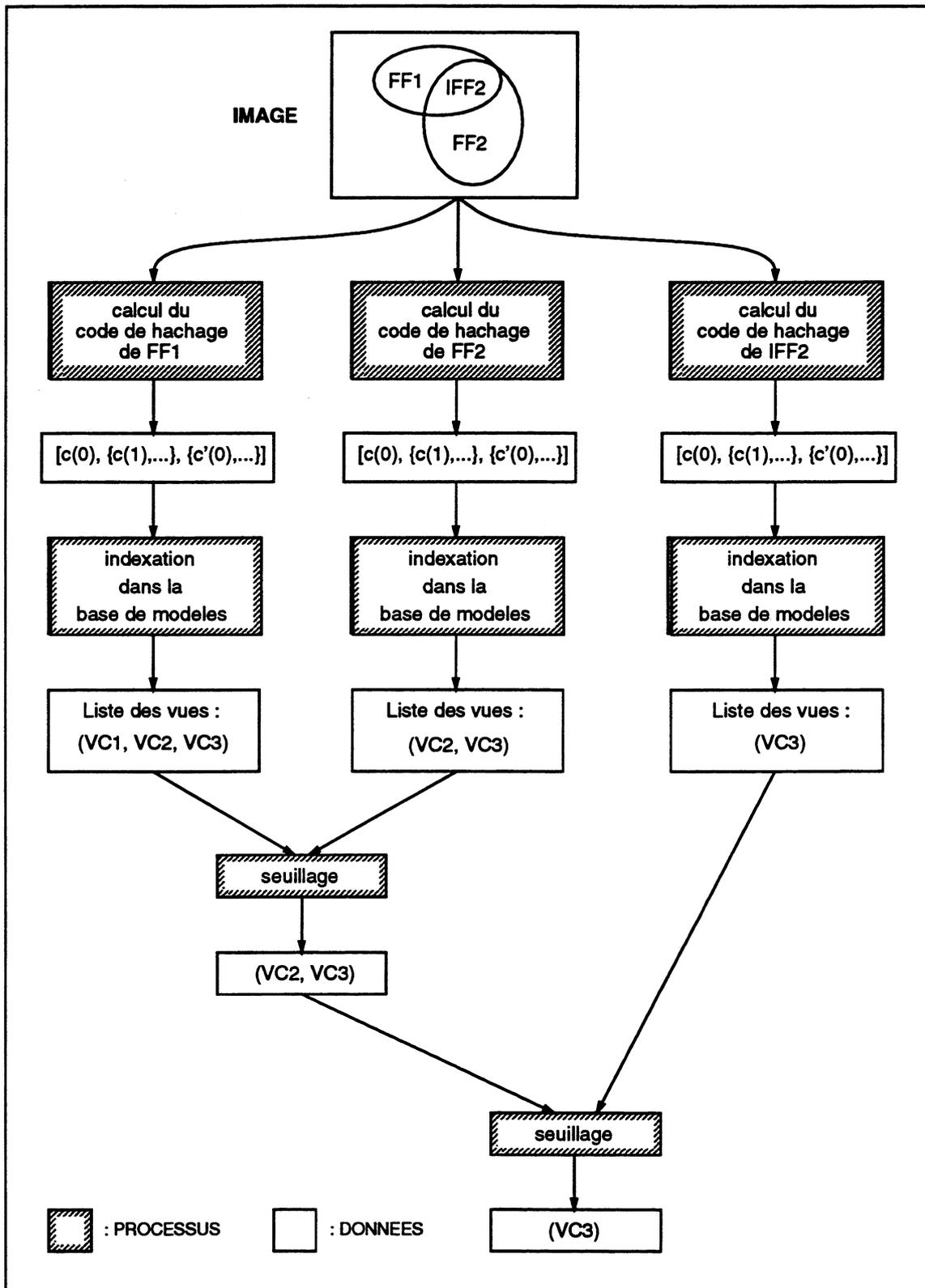


Figure 5.5: Le mécanisme d'indexation prend en entrée un ensemble de figures fondamentales et leurs intersections. Il indexe ensuite la base de modèles et retourne la liste des modèles contenant les FF et IFF trouvées dans l'image.

8. finalement, l'intersection des deux listes (VC2, VC3) et (VC3) lève l'ambiguïté entre VC2 et VC3 en sélectionnant VC3 comme le meilleur candidat.

5.4 Résumé

En résumé, pour notre système d'indexation de modèles :

1. la représentation proposée contient trois couches : une couche supérieure contenant une description à objet centré. Une couche centrale contenant une description de l'objet par une liste de vues caractéristiques et une couche inférieure où chaque vue caractéristique est décrite par une liste de figures fondamentales et une liste des intersections de figures fondamentales.
2. la base de modèles suit cette structure à trois couches, mais à l'inverse. En bas, on a une liste de figures fondamentales et une liste d'intersections de figures fondamentales. Au milieu, on a une liste de vues caractéristiques. Enfin, en haut, on a une liste d'objets. Seules les FF et les IFF sont organisées dans les tableaux de hachage.
3. le mécanisme d'indexation prend en entrée un ensemble de figures fondamentales et leurs intersections et essaye de déterminer quel modèle (ou modèles) contiennent ces figures afin de déterminer quel modèle (ou modèles) sont probablement présents dans l'image.

Chapitre 6

Résultats d'expérimentation

Afin de tester le système d'indexation de modèles qui a été développé dans notre laboratoire, nous avons décidé d'utiliser une bibliothèque composée principalement d'objets polyédriques. Ce domaine a été choisi pour des raisons de facilité. Parmi les objets choisis quelques uns sont simples, d'autres sont assez compliqués. La plu part d'entre eux contiennent beaucoup de similitudes. Ceci rend le problème plus intéressant à résoudre. L'information à stocker dans la base de données et l'information à interpréter sont extraites à partir d'images 2D. Les tests ont été réalisées pour montrer les aspects suivants du système :

1. Construction automatique d'une base de modèles par :

- la prise d'un nombre d'images à partir de plusieurs "points de vue caractéristiques" pour chaque objet à stocker dans la base,
- le traitement de chaque image de sorte que des lignes droites et des jonctions soient obtenues,
- le mappage du réseau des segments de droite et des jonctions dans un graphe, où un nœud représente une jonction et une arête représente une ligne,
- la décomposition de ce graphe en sous graphes (les figures fondamentales),
- l'obtention des intersections entre paires de figures fondamentales,

- l'évaluation des codes de hachage des figures et des intersections et
- le stockage de chaque vue caractéristique, ses figures fondamentales et ses intersections associées dans la base de données.

2. Recherche et identification des objets dans une scène par :

- l'extraction du graphe à partir de l'image d'entrée.
- la décomposition de ce graphe en sous graphes (les figures fondamentales),
- l'obtention des intersections entre paires de figures fondamentales.
- l'évaluation des codes de hachage des figures et des intersections.
- l'indexation d'un ou de plusieurs modèles dans la base.
- la sélection des modèles. Les modèles qui ont reçu les plus de votes correspondent aux objets identifiés.

Les images peuvent être d'une complexité variable. Celles-ci peuvent contenir les propriétés suivantes :

- un ou plusieurs objets se cachant éventuellement les uns avec les autres. Certains de ces objets peuvent être des objets non contenus dans la base de modèles.
- un certain degré de bruit si cela ne cause pas trop de problèmes dans la recherche. Ce bruit peut être dû au capteur, à l'éclairage, aux ombres, à la texture ...
- variations dans les figures fondamentales et en conséquence dans leurs intersections telles que rotations, translation, ou échelle.
- variations dans les paramètres globaux de la scène tels que la rotation, la translation, l'échelle ou la symétrie.

3. Recherche efficace en fonction du nombre d'objets de la base.

6.1 Le processus hors ligne de construction de la base de modèles

Le processus hors ligne (off line) de construction de la base de modèles procède comme suit : pour chaque objet à stocker dans la base de modèles, (figure 6.1, étape 1) un certain nombre d'images sont prises à partir de plusieurs points de vue "caractéristiques" en accord avec la complexité de l'objet. Chaque image représente alors une vue caractéristique potentielle associée à un objet. Celles-ci sont obtenues par l'intermédiaire du système d'acquisition et de traitement d'images FEAGRO. Chaque image est traitée de façon à ce que des segments de droites et des jonctions soient extraits. Une jonction est l'intersection de deux ou plusieurs droites. Le réseau des droites et des jonctions est représenté sous la forme d'un graphe, où chaque nœud encode une jonction et chaque arête encode un segment de droite [55]. Cette information constitue l'entrée de notre système.

6.1.1 Réduction du bruit

Afin d'avoir les modèles de nos objets les plus précis possibles, notre système filtre l'image d'entrée pour réduire au mieux le bruit qui y est contenu (voir la figure 6.1, étape 2). Ce module sera expliqué dans la partie 6.3.2.

6.1.2 Décomposition du graphe en figures fondamentales

Le graphe-image résultant est ensuite décomposé en un ensemble de sous-graphes (figure 6.1, étape 3). Ces sous-graphes représentent alors les différentes figures fondamentales et leurs intersections entre figures fondamentales composant le graphe-image.

6.1.3 Évaluation des codes de hachage et stockage de l'information

Pour chaque figure et chaque intersection les codes de hachage respectifs sont évalués (figure 6.1, étape 4). Ceux-ci sont obtenus par la méthode de caractérisation des graphes proposée dans le chapitre 3. Par l'intermédiaire de ces codes de hachage, les figures fondamentales et les intersections avec leurs vues associées sont stockées dans la base de données (figure 6.1, étape 5). Ceci se fait par l'utilisation de la technique d'indexation proposée dans le chapitre 4.

Dans la pratique :

1. le code de hachage d'une figure fondamentale est composé par : les coefficients c_0, c_{n-1}, c'_{q-1} et le type du nœud générateur.
2. le code de hachage d'une intersection est composé par les coefficients c_0 et c_{n-1} du graphe associé à l'intersection ainsi que les codes des deux figures qui forment l'intersection.

Le type du nœud générateur vaut :

- 1, si le sommet est du type L,
- 2, si le sommet est du type F,
- 3, si le sommet est du type Y.

Il y a plusieurs avantages à ce choix :

- la valeur du coefficient c_0 donne directement la taille du graphe. Ceci oblige le système à comparer des graphes dont la taille est la même,

- le calcul de c_0 ne demande que l'évaluation d'une simple soustraction : $c_0 = n - 1$,
- le coefficient c_{n-1} discrimine très bien les graphes sauf dans le cas des arbres (Merris [77]). Dans la pratique, nous avons constaté (pour le type de figures fondamentales choisies) que cet inconvénient est surmontable si on utilise le type du nœud générateur comme partie du code de hachage du graphe. Ce choix est basé évidemment sur l'expérience.

Il y a deux raisons pour lesquelles on n'utilise pas les ensembles complets des coefficients $\{c_0, \dots, c_n\}$ et $\{c'_0, \dots, c'_q\}$:

1. tout d'abord, parce qu'au fur et à mesure que la taille et la complexité du graphe augmentent, le calcul de ces coefficients devient plus coûteux.
2. deuxièmement, la gestion d'ensembles de coefficients de taille variable ($5 \leq n \leq 10$) est mal aisée.

6.1.4 Les modèles choisis

Dans le cadre de ce travail, nous avons sélectionné sept objets pour construire les modèles correspondants. La figure 6.2 montre ces sept objets. Le nombre de vues caractéristiques choisies pour chaque objet est montré dans le tableau de la figure 6.3. Les vues caractéristiques pour chacun des sept objets choisis sont montrées dans les figures 6.4, 6.5, 6.6, 6.7, 6.8, 6.9 et 6.10. La taille de chaque image montrée dans ces figures est de 256×256 pixels.

Afin que l'information contenue dans ces images soit utilisable par notre système d'indexation, il faut que celle-ci passe par une étape de compression. Cette compression de l'information prend la forme d'une approximation polygonale des contours obtenus à partir de l'information des niveaux de gris des objets dans l'image. Ceci est possible grâce au système d'acquisition et de traitement d'images FEAGRO. Le réseau des jonctions et des segments de droite contenus dans la nouvelle image constitue l'entrée de notre système.

Après avoir été traitées par le processus de construction de la base de données dans ses différentes étapes : réduction du bruit, décomposition en FF et en IFF, calcul de codes de hachage ... les vingt vues sont stockées dans notre base de données expérimentale. Elles constituent donc les modèles de notre système. La figure 6.11 montre toutes les vues contenues dans notre base de données. A partir de scènes contenant un ou plusieurs des sept objets choisis, notre système essaye de les identifier en utilisant leurs modèles.

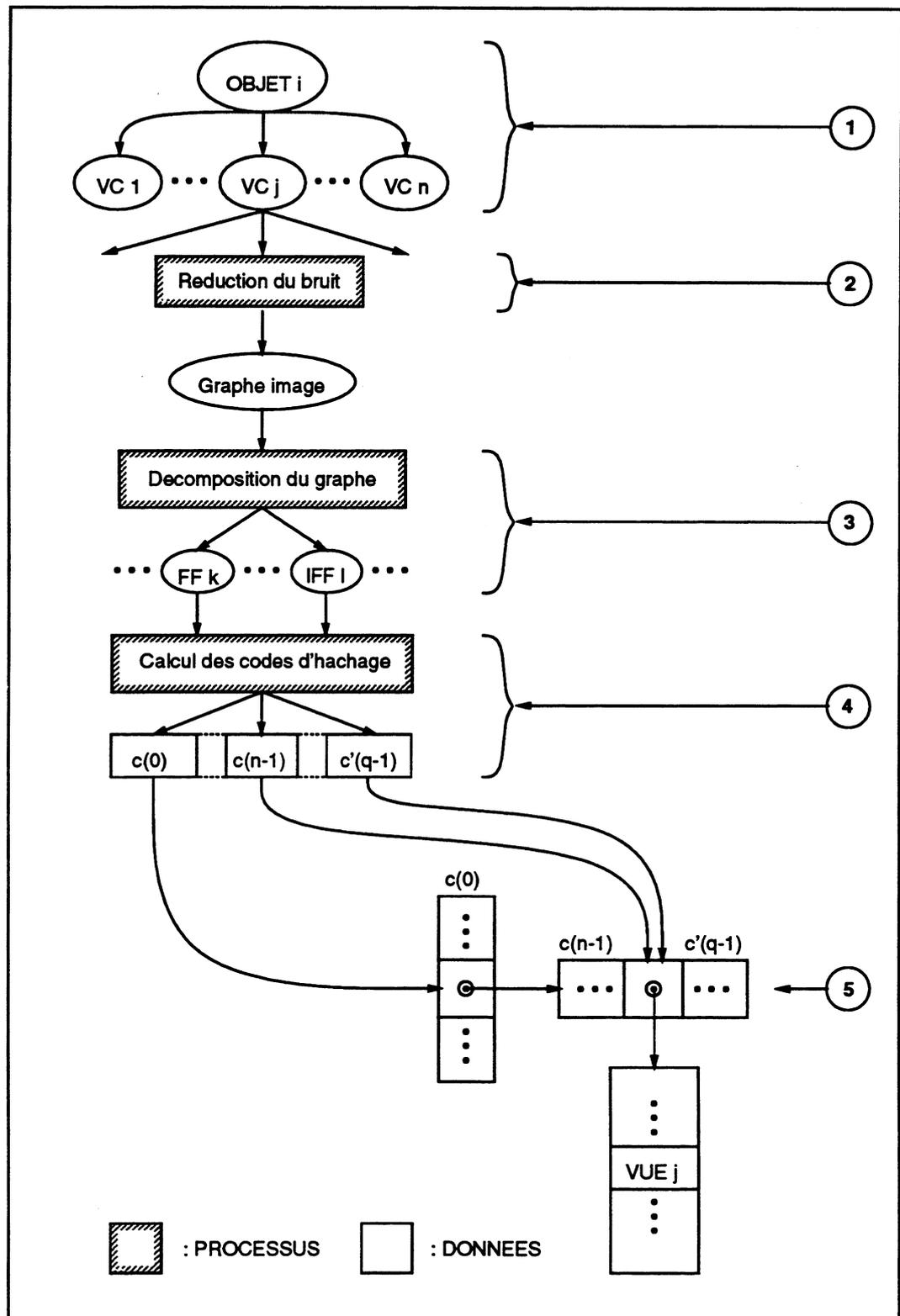


Figure 6.1: Schéma du processus de construction de la base de modèles.

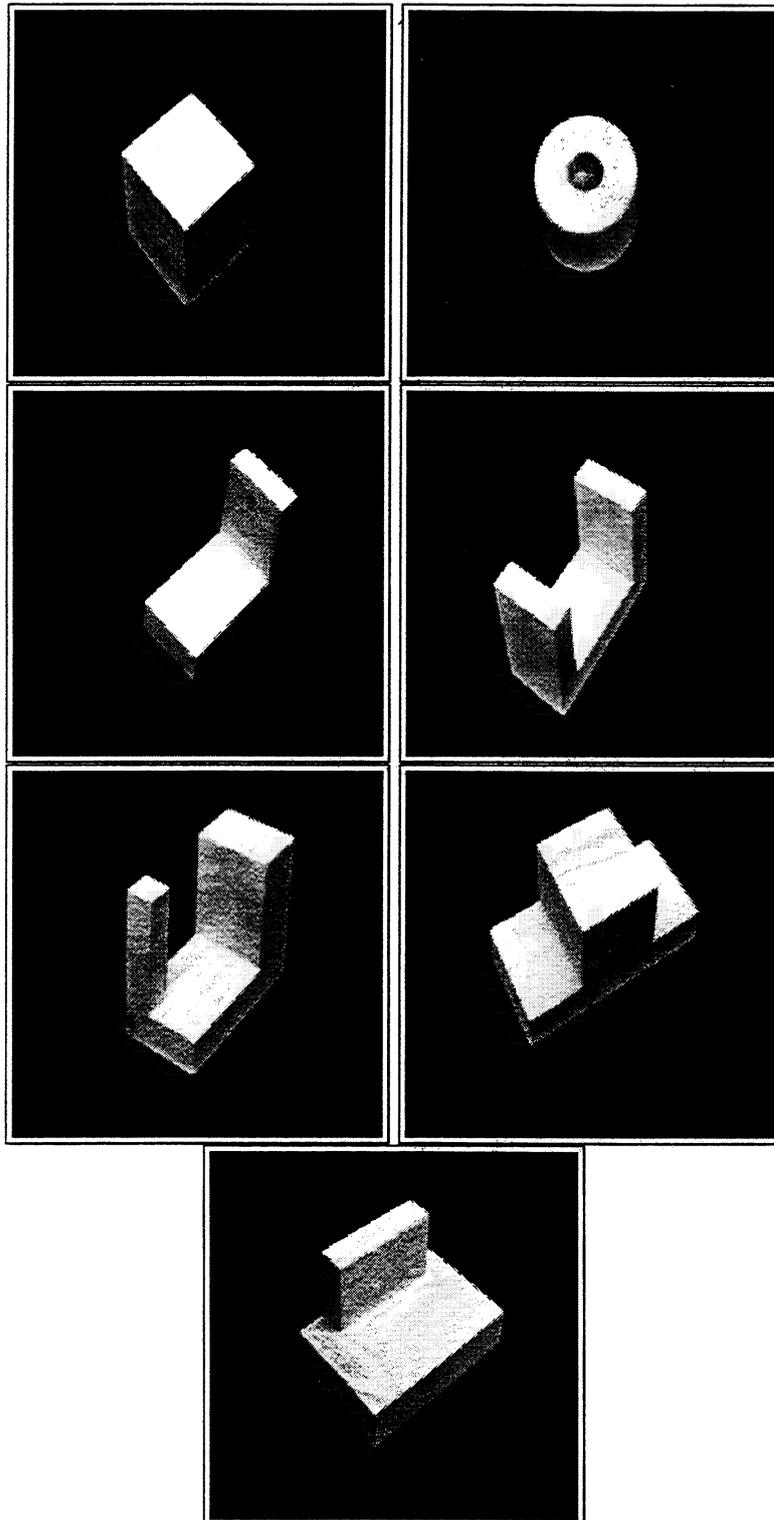


Figure 6.2: Les sept objets choisis pour former la base de modèles.

Objet No.	Nombre de vues caracteristiques
1	1
2	1
3	4
4	2
5	8
6	2
7	2
	total = 20

Figure 6.3: Pour chaque objet composant la base de modèles un nombre de vues caractéristiques sont choisies.

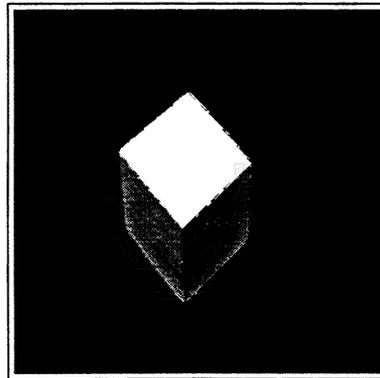


Figure 6.4: La vue choisie pour l'objet numéro 1.

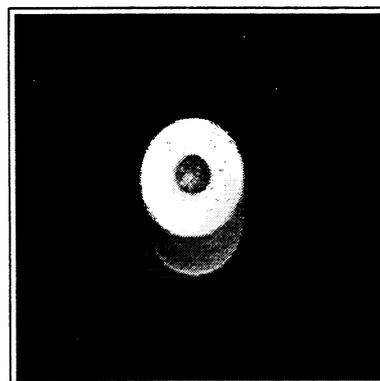


Figure 6.5: La vue choisie pour l'objet numéro 2.

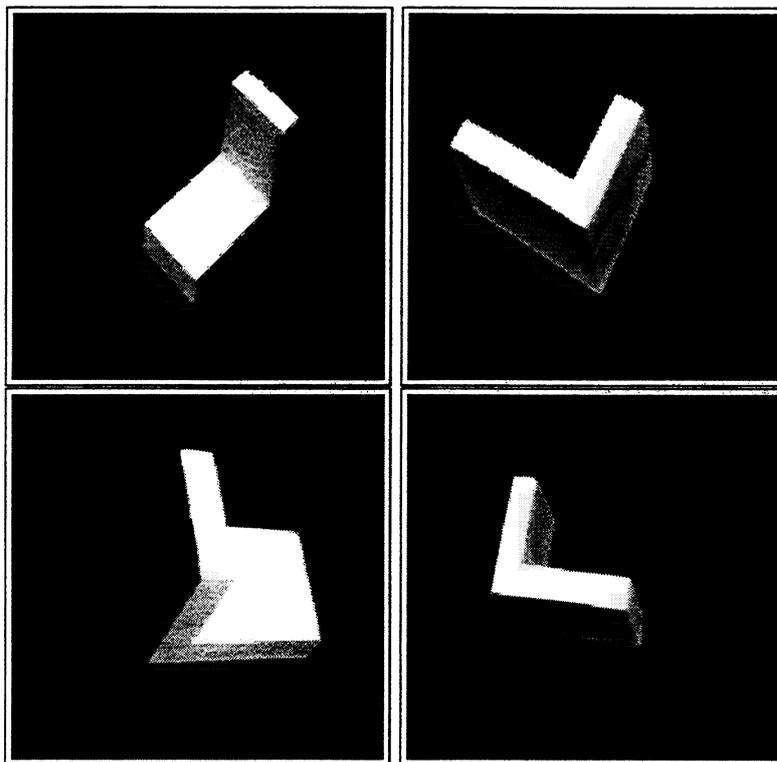


Figure 6.6: Les quatre vues choisies pour l'objet numéro 3.

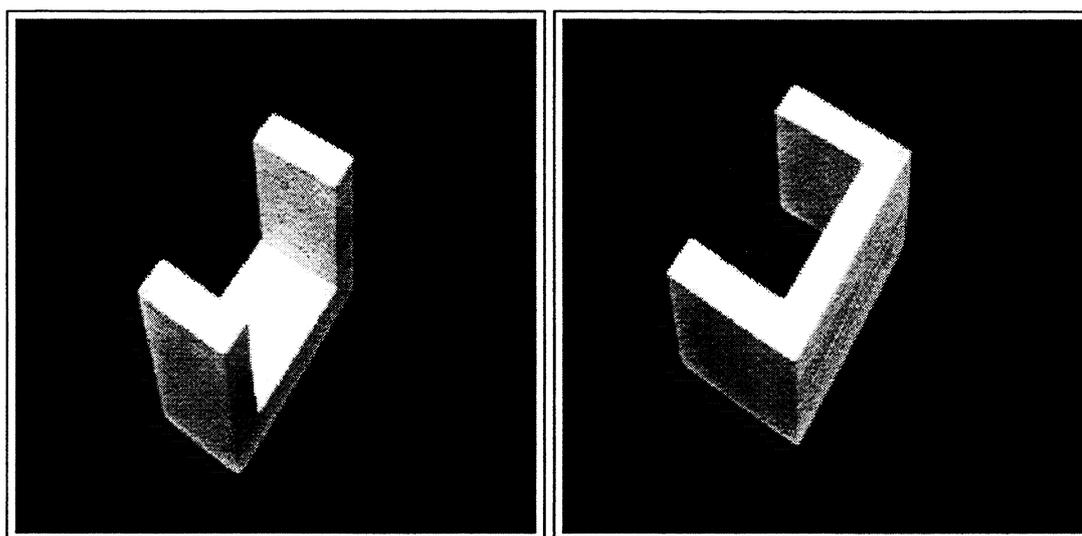


Figure 6.7: Les deux vues choisies pour l'objet numéro 4.

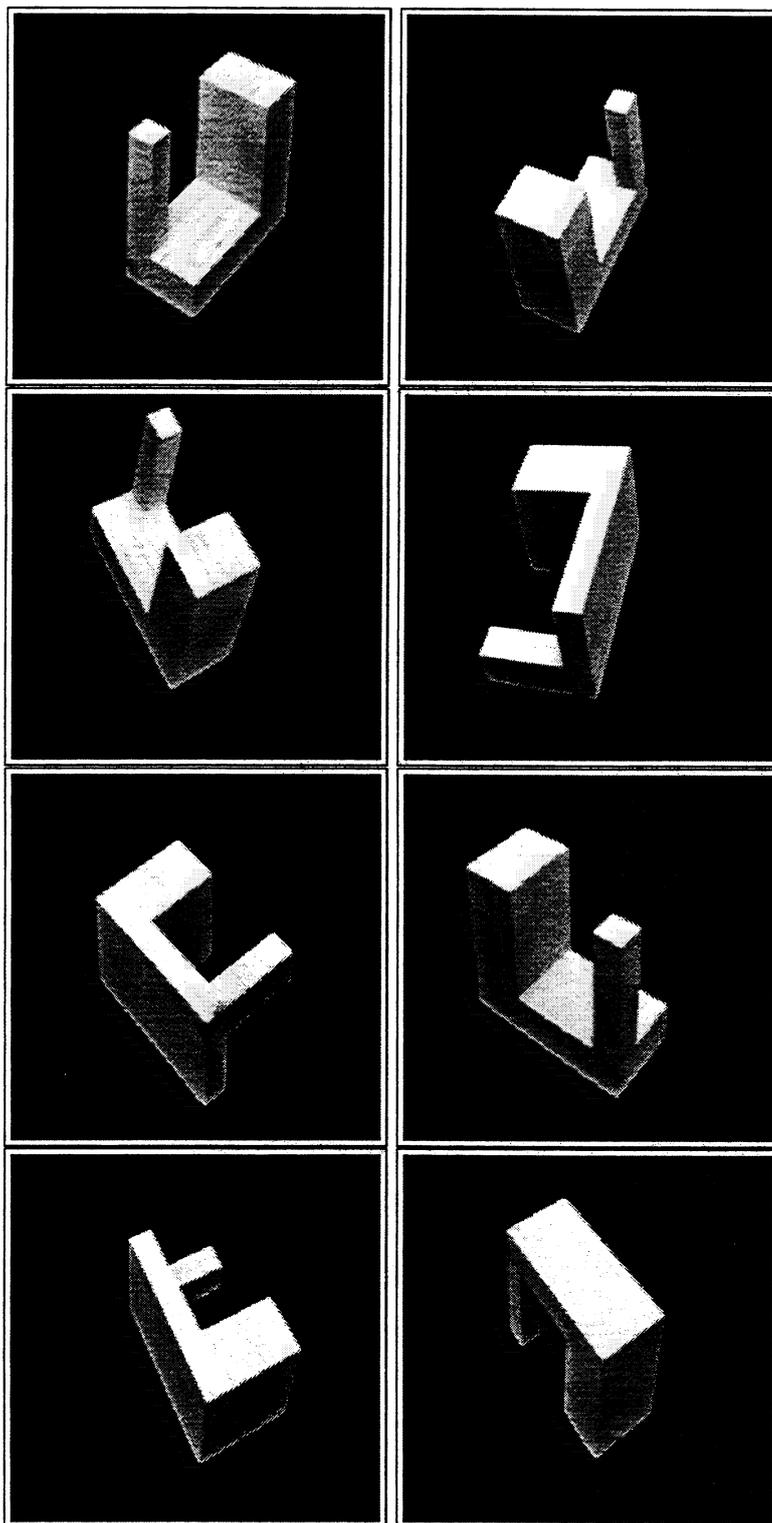


Figure 6.8: Les huit vues choisies pour l'objet numéro 5.

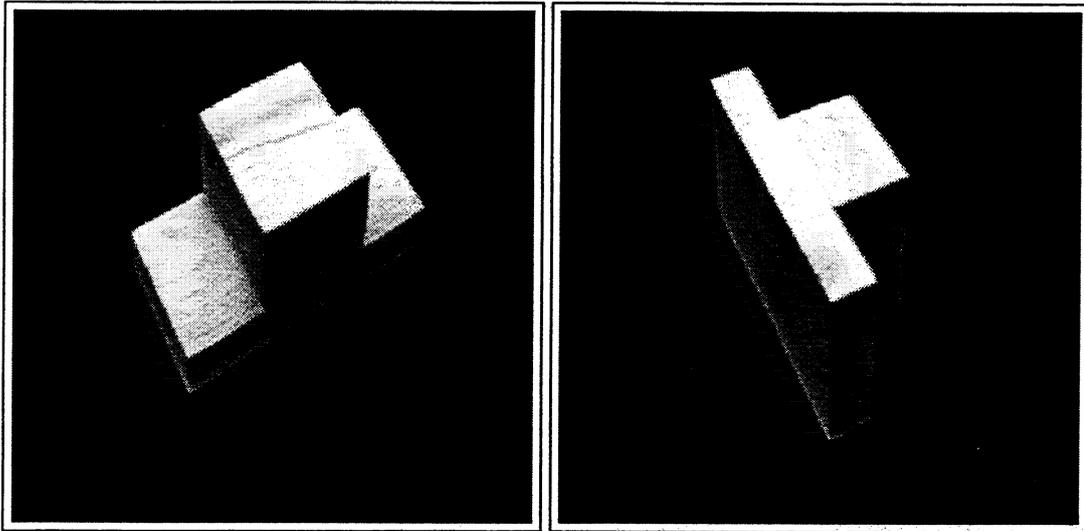


Figure 6.9: Les deux vues choisies pour l'objet numéro 6.

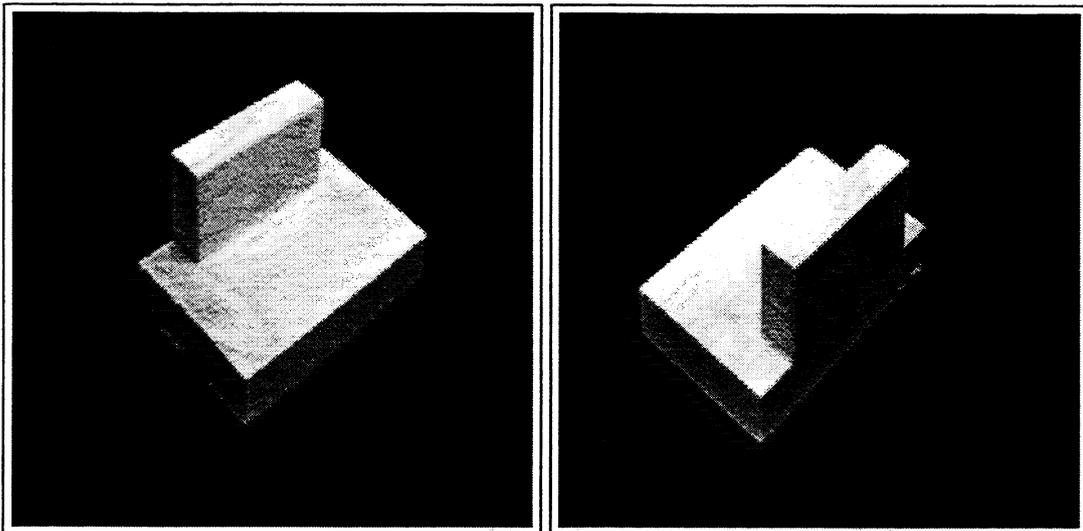


Figure 6.10: Les deux vues choisies pour l'objet numéro 7.

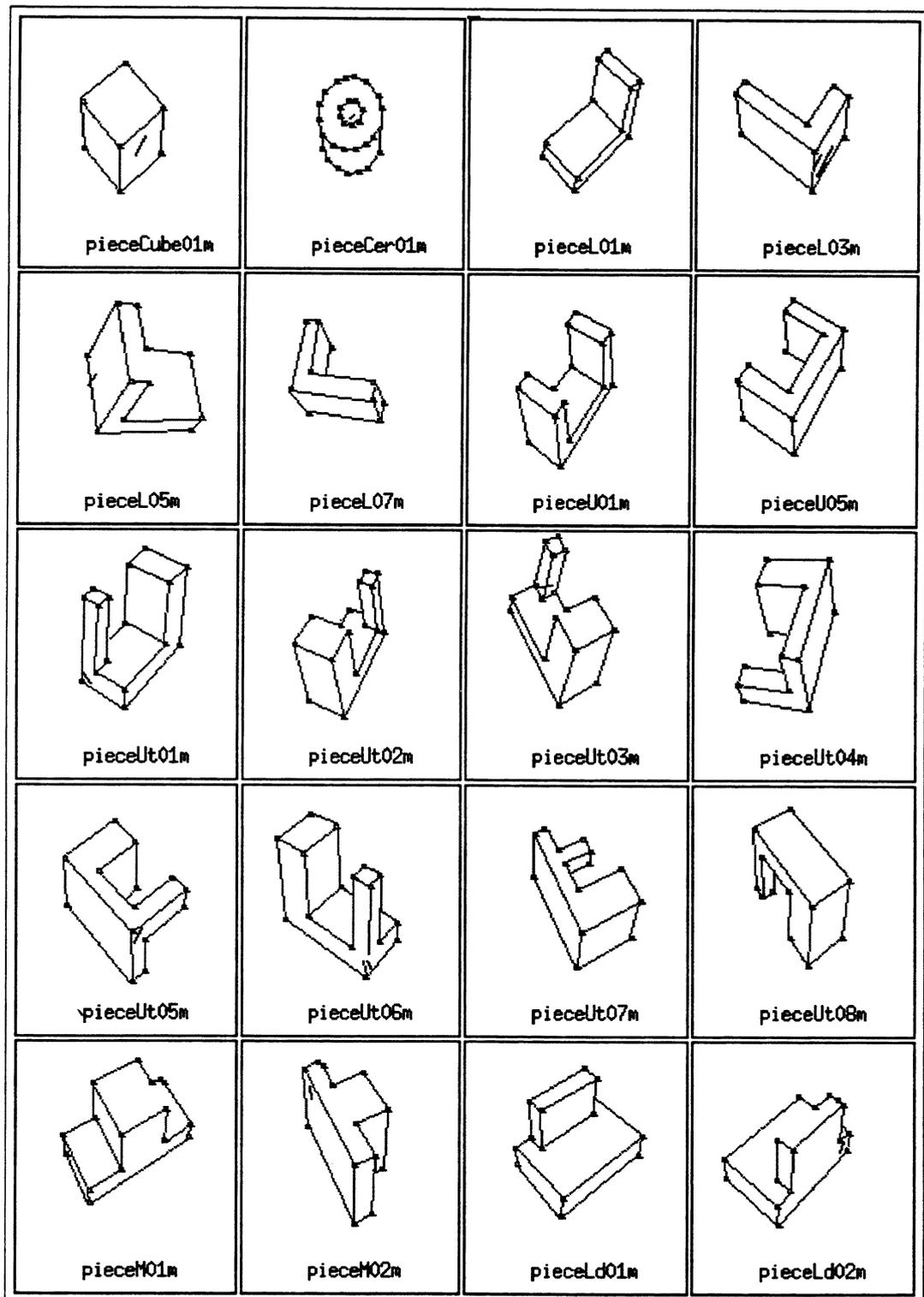


Figure 6.11: Notre base expérimentale contenant les 20 vues appartenant aux 7 objets choisis.

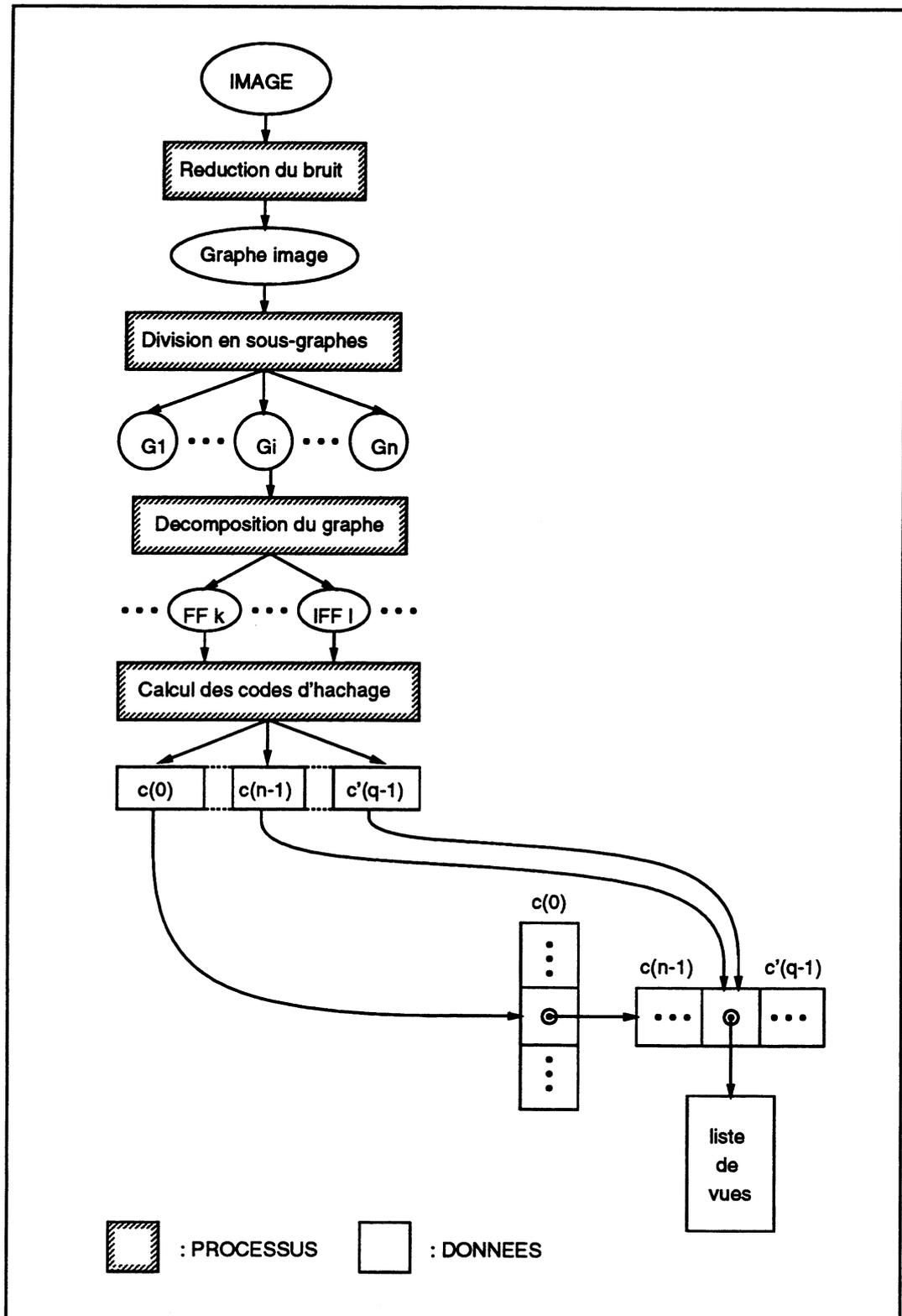


Figure 6.12: Schéma du processus d'indexation.

6.2 Le processus d'indexation

Le processus d'indexation analyse une image et essaye de trouver quels objets y sont présents. L'implémentation actuelle permet l'analyse des images contenant un ou plusieurs objets à la fois.

Le même processus décrit dans le paragraphe précédent avec quelques modifications :

1. **prend en entrée une image.** Celle-ci est fournie par FEAGRO. Elle peut contenir plusieurs objets,
2. **essaye de réduire au mieux le bruit contenu dans l'image,**
3. **divise le graphe-image en composantes connexes de façon que chaque composante provienne au mieux possible d'un objet identique.** Pour chacun de ces graphes le même processus d'indexation :
 - **extraît** respectivement les figures fondamentales et les intersections.
 - **calcule** le code de hachage pour chaque figure et chaque intersection, Celui-ci est obtenu de la même façon que pour le processus de construction de la base de données.
 - **fournit** la liste de modèles respectifs en indexant dans la base de modèles. Ceci se fait par l'intermédiaire de la technique d'indexation du chapitre 4.

Les modèles qui ont reçu le plus de votes sont sélectionnés comme les candidats les plus proches de l'objet en question.

La même procédure est répétée pour le reste des composantes connexes (objets) dans l'image.

La figure 6.12 résume de manière schématique le fonctionnement du processus d'indexation.

6.3 Un exemple détaillé

Afin de mieux comprendre le fonctionnement du mécanisme d'indexation, nous expliquons en détail chacune de ses étapes. Pour cela nous prenons l'une de nos scènes. Celle-ci est montrée dans la figure 6.13 (a).

6.3.1 L'image d'entrée

L'entrée qui sera traitée par le mécanisme d'indexation consiste de l'image des segments de droite avec leurs jonctions montrée dans la figure 6.13 (c). Celle-ci est obtenue à partir de l'image de contour (figure 6.13 (b)) par l'intermédiaire du même système d'acquisition et de traitement d'images déjà utilisé pour construire la base de modèles. L'image de contour a été extraite à partir de l'image de niveaux de gris (figure 6.13 (a)).

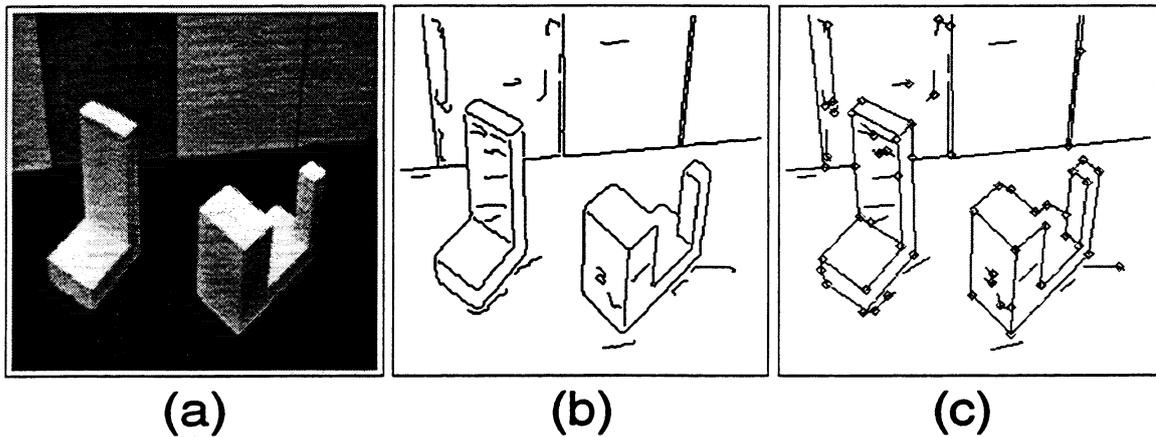


Figure 6.13: (a) l'image de niveaux de gris. (b) l'image des contours et (c) l'image des lignes droites avec leurs jonctions.

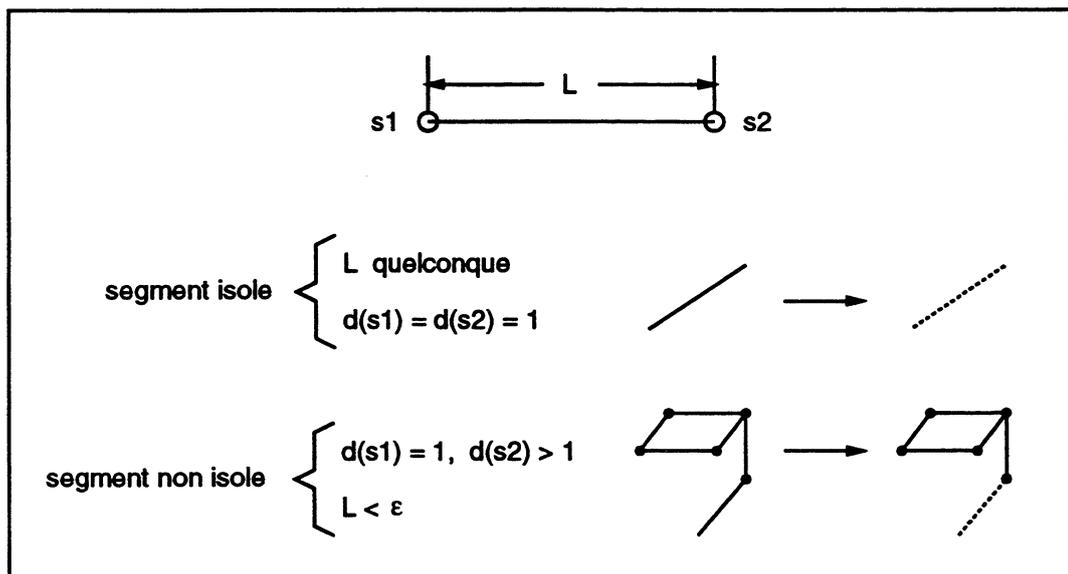


Figure 6.14: Exemples des segments éliminés par le module de réduction du bruit.

6.3.2 Réduction du bruit

Afin de réduire le temps de calcul des étapes suivantes et d'améliorer la performance du mécanisme d'indexation nous avons décidé d'intégrer à notre système un module de réduction du bruit. Ce module prend en entrée l'image des jonctions et des lignes droites fournie par FEAGRO et essaie de réduire le bruit qui y est contenu. Il comporte trois étapes :

1. **filtrage des segments parasites.** Cette étape consiste à éliminer de l'image tous les segments dont les caractéristiques sont montrées dans la figure 6.14.
2. **réduction des T et obtention de métasegments.** Cette étape consiste, d'une part, à éliminer de l'image les jonctions du type T et d'autre part, à regrouper les segments colinéaires dans un seul segment (un métasegment). La figure 6.15 montre des exemples de ces deux cas.
3. **filtrage de petits graphes.** Cette étape consiste à éliminer de l'image tous les graphes pour lesquels :

$$(n < 11 \text{ et } \frac{f_1}{n} < 2) \text{ ou } \frac{f_2}{f_1} \leq 7$$

ou

$$f_1 = \sum_{i=1}^n d(i),$$

$$f_2 = \sum_{j=1}^m t(j),$$

m est le nombre d'arêtes du graphe, $d(i)$ est le degré du sommet i et $t(j)$ est la taille de l'arête j du graphe.

Le facteur (f_1/n) nous permet de mesurer la complexité du graphe. Plus grande est cette valeur, plus complexe est le graphe. Si $f_1 = 0$ alors le graphe est totalement non connexe. Si $f_1 = 3$ alors le graphe est *complet*. Un graphe est dit complet si pour toute paire de sommets, il existe une arête qui relie les deux sommets (Gondran [42]). Dans ce cas, la valeur 2 limite la complexité des graphes à celle d'un graphe avec au plus un *cycle*. Un cycle est une chaîne dont les extrémités coïncident (Gondran [42]).

Le facteur f_2 limite la taille moyenne des arêtes du graphe à 7 pixels.

Autrement dit, si la taille n du graphe trouvé dans l'image est inférieure à 11 nœuds et celui-ci a au plus un cycle ou si la taille moyenne de ses arêtes est inférieure ou égale à 7 pixels alors on l'élimine.

Les figures 6.16 (a), 6.16 (b), 6.16 (c) et 6.16 (d) montrent respectivement :

- l'image résultante obtenue par le fait de faire passer l'image de la figure 6.13 (c) par l'étape de filtrage des segments parasites,
- les endroits où une jonction du type T a été trouvée et éliminée. Celles-ci sont indiquées par des croix,
- les métasegments trouvés dans l'image. Ceux-ci sont indiqués par deux carrés aux bout des métasegments. Notons l'absence des croix (là où il y avait une jonction du type T) indiquant la disparition du sommet correspondant.
- l'image résultante lorsque on fait passer l'image de la figure 6.16 (c) par l'étape d'élimination des petits graphes.

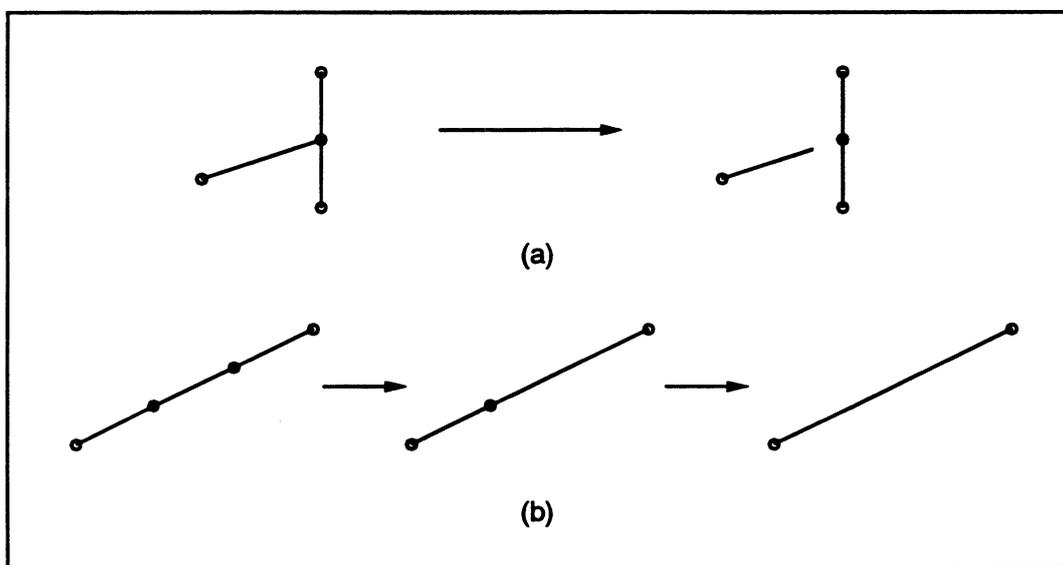


Figure 6.15: (a) exemples des jonctions du type T éliminées. (b) exemples de segments colinéaires réduits à un seul segment.

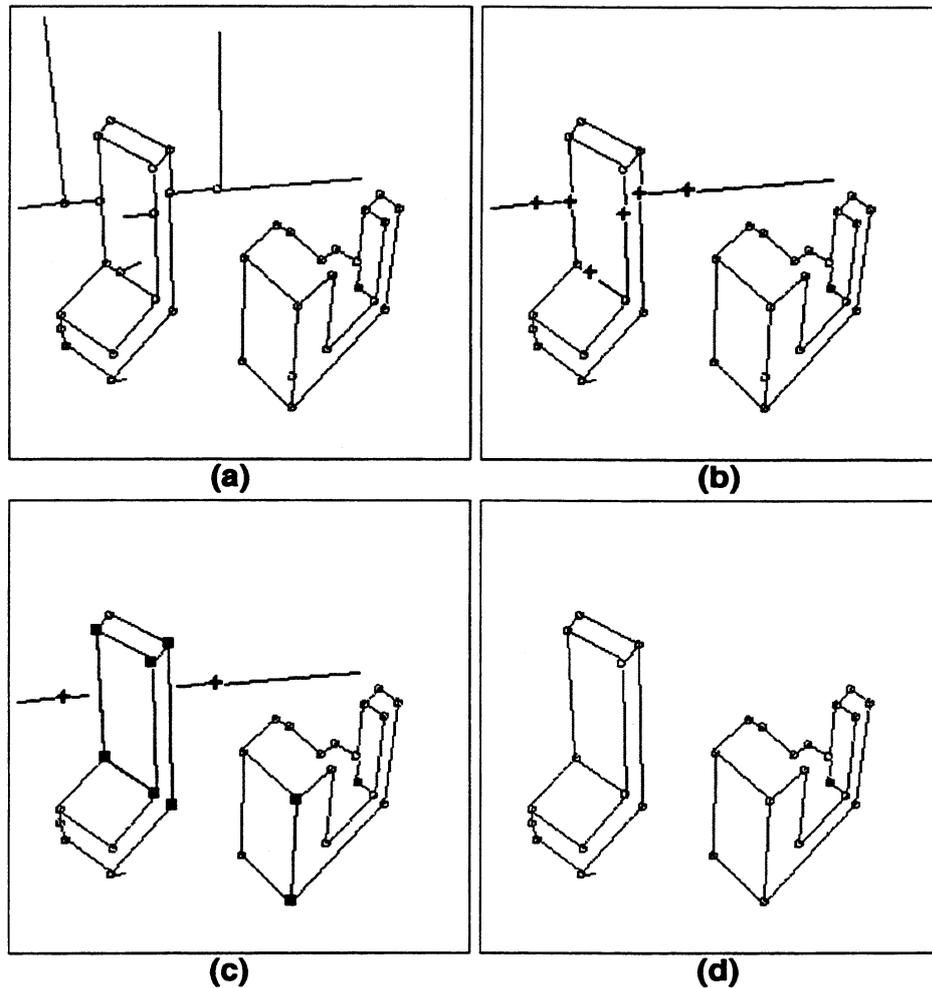


Figure 6.16: (a) résultat du passage de l'image d'entrée par la première étape de réduction du bruit. (b) les T éliminées de l'image. (c) les métasegments trouvés. (d) en fin, l'image résultante du passage de l'entrée par la troisième étape de réduction du bruit.

6.3.3 Division du graphe-image

Lorsque dans la scène n'il y a qu'un seul objet, le mécanisme d'indexation peut indexer dans la base de modèles et rendre une liste limitée à quelques modèles candidats représentant peut-être l'objet dans la scène.

Si, au contraire, dans la scène il y a plusieurs objets et si en plus certains d'entre eux s'occulent alors le mécanisme d'indexation peut encore rendre une liste limitée à quelque candidats mais, dans ce cas, il n'est pas capable de dire quel modèle représente quel objet dans la scène.

Afin de pouvoir appliquer la méthode d'indexation séparément pour chacun des objets contenus dans l'image, notre système incorpore une étape supplémentaire.

Cette étape se sert des résultats fournis par le module de réduction du bruit afin de diviser le graphe représentant l'image d'entrée filtrée en composantes connexes tels que chacun de ces sous-graphes provienne d'un seul objet. La division du graphe-image se fait en trois étapes :

1. tout d'abord, les jonctions du type T, dans l'image, sont éliminées au mieux. Cette élimination a été déjà fournie par le module de réduction du bruit. Voir la figure 6.16 (b).
2. ensuite, les segments colinéaires sont regroupés en métasegments. Ces regroupements ont été également fournis par le module de réduction du bruit. Voir la figure 6.16 (c).
3. finalement, une procédure d'étiquetage parcourt chaque graphe dans l'image restante en étiquetant les nœuds et les arêtes appartenant au même graphe. Cette procédure est répétée pour chaque graphe dans l'image. L'image 6.17 (b) montre les deux graphes trouvés dans l'image. Bien que les petits graphes ont reçu aussi une étiquette pendant l'étiquetage, ils n'apparaissent point dans l'image, tout simplement parce qu'ils avaient été déjà considérés comme du bruit par le module de réduction du bruit.

La figure 6.17 (a) englobe l'ensemble des segments considérés comme étant du bruit par notre système, au moment de faire passer l'image d'entrée de la figure 6.13 (c) par les différentes étapes ci-dessus expliquées.

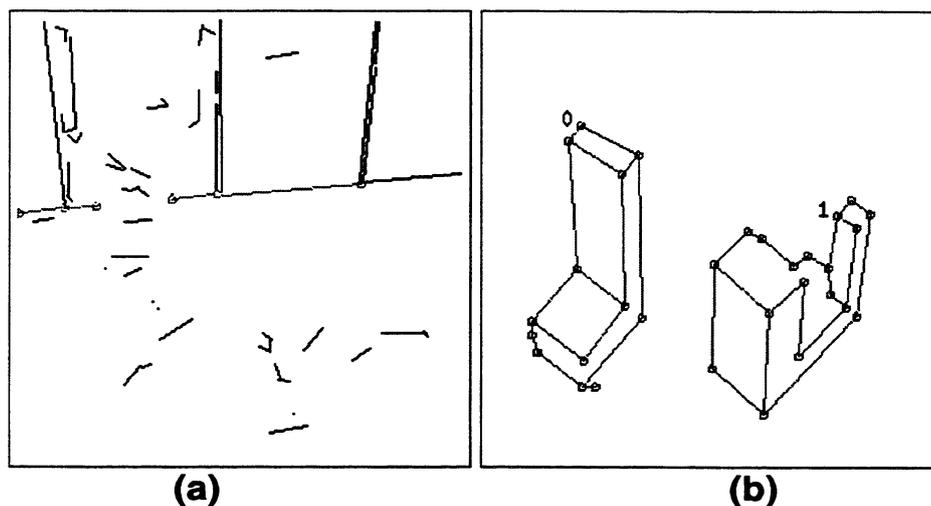


Figure 6.17: (a) l'ensemble de segments considérés comme étant du bruit par notre système. (b) les deux graphes trouvés par notre système. Chaque graphe reçoit une étiquette.

6.3.4 Décomposition en figures fondamentales

L'étape suivante consiste à prendre chacun des graphes trouvés dans l'image pour ensuite le décomposer en ses figures fondamentales respectives. Celles-ci sont obtenues par l'intermédiaire de l'extracteur de figures fondamentales à partir de chaque nœud de chaque graphe. Les cas suivants sont exclus :

1. le degré du nœud générateur, $d(i) = 1$,
2. le nœud a un voisin qui tombe dans le cas antérieur.

Les nœuds numéro 54 et 28 du graphe de la figure 6.18 sont des exemples. En conséquence, ils ne sont pas pris en compte par l'extracteur de FF pour générer les figures fondamentales respectives.

Pour l'exemple que nous sommes en train de détailler, la figure 6.17 (b) montre les deux graphes considérés par l'extracteur de FF. La taille du premier graphe est de 13 nœuds alors que celle du deuxième est de 18 nœuds.

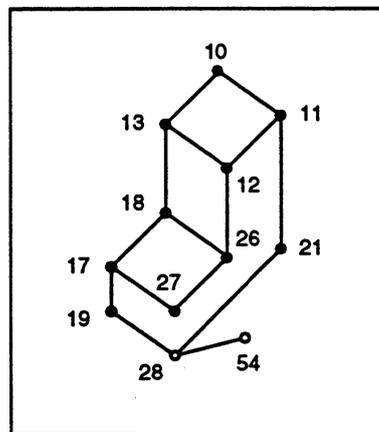


Figure 6.18: Les deux nœuds indiqués par des cercles en blanc ne sont pas considérés par l'extracteur de FF pour générer les figures fondamentales respectives.

Les tableaux des figures 6.19 et 6.20 montrent respectivement les 11 et les 16 FF trouvées pour chacun des deux graphes de la figure 6.17 (b). Pour chaque FF, les données suivantes sont montrées : sa taille n (en nombre de nœuds), son nombre d'arêtes (m), sa séquence des degrés (la liste de degrés $d(i)$ de chaque nœud en ordre ascendant), le type du nœud générateur et les trois coefficients : c_0 , c_{n-1} et c'_{q-1} . Également, on a ajouté une neuvième colonne pour les remarques.

6.3.5 Obtention des intersections de figures fondamentales

L'extracteur des IFF prend l'ensemble des figures fondamentales d'un graphe et évalue l'ensemble des IFF correspondant. Son obtention suit le processus de géné-

ration décrit dans le chapitre 5, paragraphe 5.1.2.

6.3.6 Evaluation des codes de hachage

Une fois que les FF et les IFF ont été obtenues pour chaque graphe valable dans l'image, le mécanisme d'indexation procède à évaluer les codes de hachage respectifs.

Le code de hachage pour un FF est composé par les coefficients c_0, c_{n-1}, c'_{q-1} et le type du nœud générateur.

Le code de hachage pour un IFF est composé par les coefficients c_0 et c_{n-1} du graphe associé à l'IFF ainsi que les codes des deux FF qui forment l'IFF.

Les codes de hachage des FF des graphes de notre exemple correspondent donc aux valeurs des coefficients ainsi qu'au type du nœud générateur montré respectivement dans les tableaux des figures 6.19 et 6.20.

Pour des raisons de simplicité, seuls les codes de hachage des FF sont montrés.

6.3.7 Indexation dans la base de données

Par l'intermédiaire des codes des FF et des IFF trouvées dans l'image, le mécanisme d'indexation indexe la base de modèles et fournit les listes des vues (une liste pour chaque graphe) les plus proches de chaque graphe. L'obtention de ces deux listes suit la technique d'indexation du chapitre 4.

Pour notre exemple, les tableaux 6.1 et 6.2 montrent les votes donnés à chaque vue caractéristique par le mécanisme d'indexation en fonction des FF (2^{ème} colonne), des IFF (3^{ème} colonne). La quatrième colonne montre les totaux. Enfin, la cinquième colonne montre la liste des vues dont les votes dépassent un seuil donné. Les vues sont indiquées par une "*".

La figure 6.21 illustre autrement ces résultats. Dans ce cas, les candidats choisis par le système comme étant les plus proches aux deux objets présents dans la figure 6.17 (b) sont étiquetés par l'étiquette du graphe correspondant.

FF	n	m	sequences des d(i)	type	c(0)	c(n-1)	c'(q-1)	remarques
	6	6	1 1 2 2 3 3	1	5	304	90	
	7	7	1 1 2 2 2 3 3	2	6	524	21084	
	7	6	1 1 1 1 2 3 3	1	6	150	122	elle n'existe pas dans la base de donnees
	6	5	1 1 1 2 2 3	1	5	98	38	
	6	5	1 1 1 2 2 3	1	5	98	38	
	7	7	1 1 2 2 2 3 3	3	6	524	24256	
	8	9	1 1 2 2 3 3 3 3	3	7	2436	147816	
	7	8	1 2 2 2 3 3 3	2	6	1618	44812	
	8	9	1 1 2 2 3 3 3 3	3	7	2436	156028	
	7	8	1 2 2 2 3 3 3	2	6	1618	44812	
	6	6	1 1 2 2 3 3	1	5	304	122	

Figure 6.19: Les onze figures fondamentales appartenant au graphe "0".

Nom de la vue	vote des FF	vote des IFF	total	liste des vues
pieceCube01m	1	0	1	
pieceCer01m	0	0	0	
pieceL01m	6	7	13	*
pieceL03m	2	2	4	
pieceL05m	2	1	3	
pieceL07m	2	1	3	
pieceU01m	4	6	10	
pieceU03m	2	1	3	
pieceUt01m	5	6	11	
pieceUt02m	5	4	9	
pieceUt03m	4	2	6	
pieceUt04m	2	1	3	
pieceUt05m	2	1	3	
pieceUt06m	4	6	10	
pieceUt07m	2	1	3	
pieceUt08m	2	1	3	
pieceM01m	6	10	16	*
pieceM03m	3	1	4	
pieceLd01m	3	2	5	
pieceLd02m	3	1	4	

Table 6.1: Voici la liste des vues contenues dans la base de données et leurs votes respectifs donnés par le mécanisme d'indexation en fonction des FF et des IFF du graphe "0". Les deux candidats choisis par le système qui ressemblent le plus à ce graphe sont indiqués par un "*" (5 ème colonne).

Nom de la vue	vote des FF	vote des IFF	total	liste des vues
pieceCube01m	1	0	1	
pieceCer01m	1	0	1	
pieceL01m	3	2	5	
pieceL03m	5	2	7	
pieceL05m	7	8	15	*
pieceL07m	2	1	3	
pieceU01m	6	10	16	*
pieceU03m	5	2	7	
pieceUt01m	4	2	6	
pieceUt02m	8	11	19	*
pieceUt03m	6	9	15	*
pieceUt04m	3	3	6	
pieceUt05m	6	3	9	
pieceUt06m	2	1	3	
pieceUt07m	6	2	8	
pieceUt08m	5	2	7	
pieceM01m	7	3	10	
pieceM03m	6	5	11	
pieceLd01m	1	0	1	
pieceLd02m	7	5	12	

Table 6.2: Voici la liste des vues contenues dans la base de données et leurs votes respectifs donnés par le mécanisme d'indexation en fonction des FF et des IFF du graphe "1". Les deux candidats choisis par le système qui ressemblent le plus à ce graphe sont indiqués par un "*" (5 ème colonne).

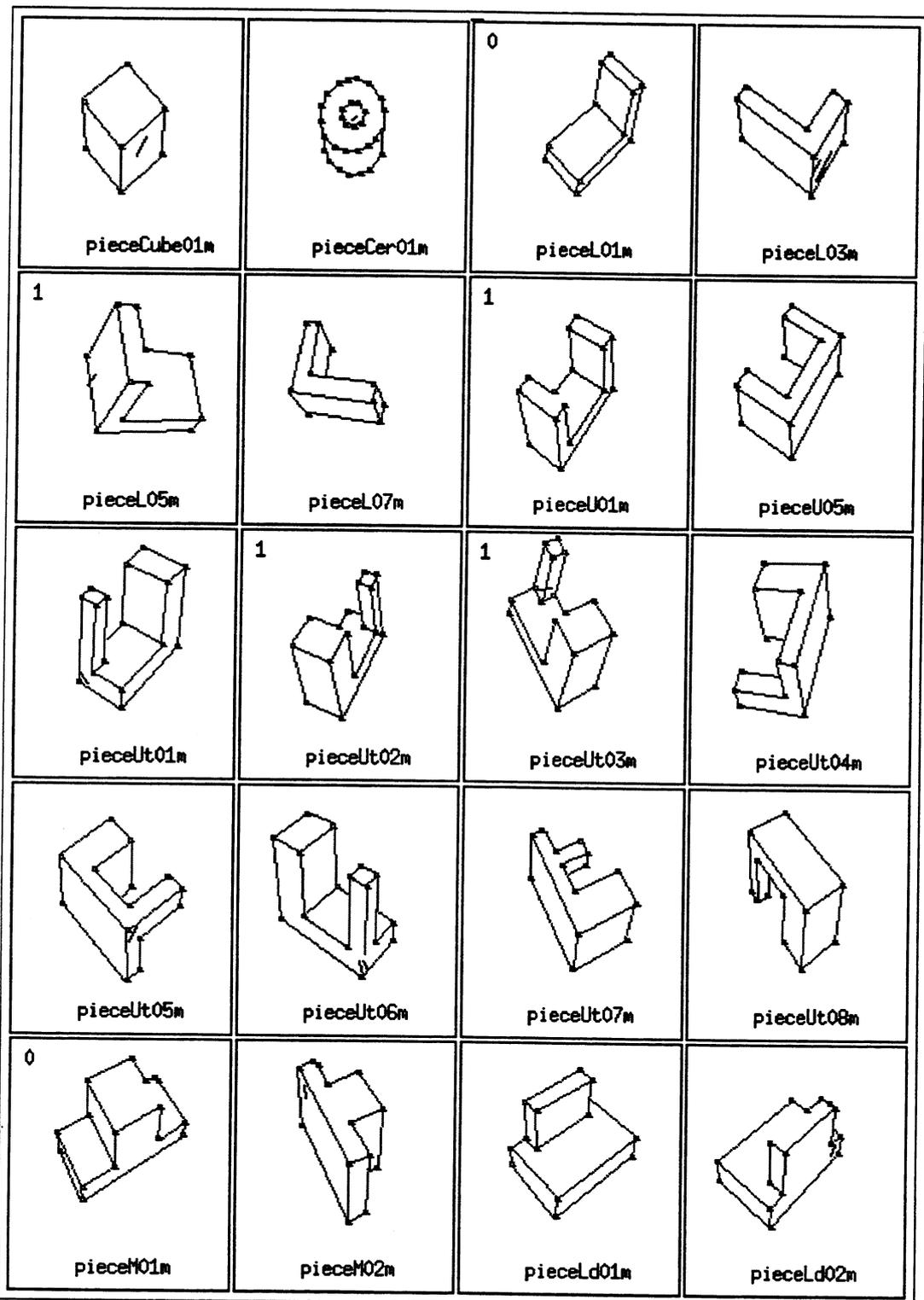


Figure 6.21: Notre base expérimentale contenant les 20 vues appartenant aux 7 objets choisis. Les vues dans la base de données qui matchent les objets inconnus sont marquées par l'étiquette correspondante.

6.4 D'autres exemples

Notre système a été confronté à d'autres exemples pour tester sa performance devant d'autres situations possibles. Ces situations sont les suivantes :

- **un seul objet dans la scène.** La scène utilisée pour cette expérience est montrée dans la figure 6.22 (a). La figure 6.22 (b) montre l'image de contours rendue par FEAGRO alors que l'image de la figure 6.22 (c) montre l'approximation polygonale correspondante. Cette dernière information est l'entrée de notre système. Les figures 6.22 (d), 6.22 (e), 6.22 (f) et 6.22 (g) montrent les différentes étapes lorsque l'image de la figure 6.22 (c) est passée par le module de réduction du bruit. Enfin, les figures 6.22 (h) et 6.22 (i) montrent respectivement :

1. l'ensemble des segments considérés comme étant du bruit par notre système et
2. le graphe de l'objet correspondant.

La figure 6.23 montre les vues choisies par le mécanisme d'indexation d'après les FF et les IFF du graphe de la 6.22 (i).

- **deux ou plusieurs objets isolés.** La scène choisie pour cette expérience est celle de la figure 6.24 (a). La figure 6.24 (b) montre l'image de contours rendue par FEAGRO alors que l'image de la figure 6.24 (c) montre l'approximation polygonale correspondante. Les figures 6.24 (d), 6.24 (e), 6.24 (f) et 6.24 (g) montrent les différentes étapes de réduction du bruit. La figure 6.25 (h) montre l'ensemble des segments considérés comme étant du bruit par notre système alors que la figure 6.24 (i) montre les deux graphes utilisés par le système pour indexer dans la base de modèles.

La figure 6.25 montre les vues choisies par le mécanisme d'indexation pour chacun des deux graphes de la figure 6.25 (i).

- **des objets s'occultant les uns les autres.** Notre scène de teste est celle de la figure 6.26 (a). La figure 6.26 (b) montre l'image de contours alors que l'image de la figure 6.26 (c) montre l'approximation polygonale correspondante. Les figures 6.26 (d)-(g) montrent les différentes étapes de réduction du bruit. La figure 6.26 (i) montre les trois graphes utilisés par le système pour indexer dans la base de modèles.

La figure 6.27 montre les vues choisies par le mécanisme d'indexation pour chacun de trois graphes de la figure 6.26 (i).

- **des objets différents de ceux qui sont dans la base d'objets.** La scène pour cette expérience est celle de la figure 6.28 (a). Les figures 6.28 (b) et 6.28 (c) montrent respectivement les images de contours et de segments de

droite correspondantes. La figure 6.28 (i) montre les graphes utilisés par le système pour indexer dans la base de modèles.

La figure 6.29 montre les vues choisies par le mécanisme d'indexation pour le graphe de la figure 6.28 (i).

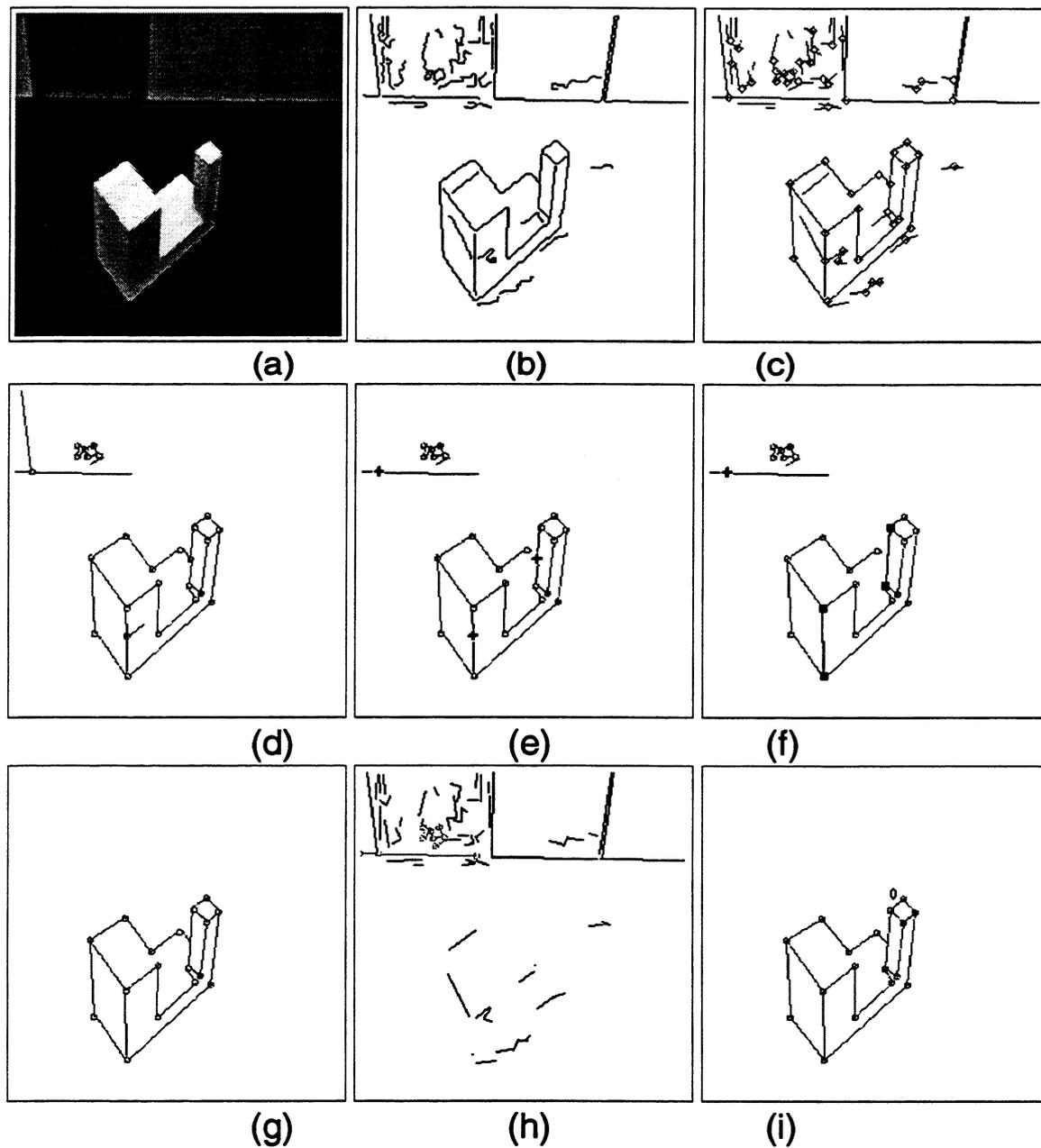


Figure 6.22: (a) Une scène contenant un seul objet. (b) l'image des contours. (c) l'image des segments de droite. (d) premier passage de réduction du bruit. (e) élimination des T. (f) les métasegments trouvés. (g) réduction des graphes. (h) ensemble de segments considérés comment étant du bruit. (i) les bons graphes.

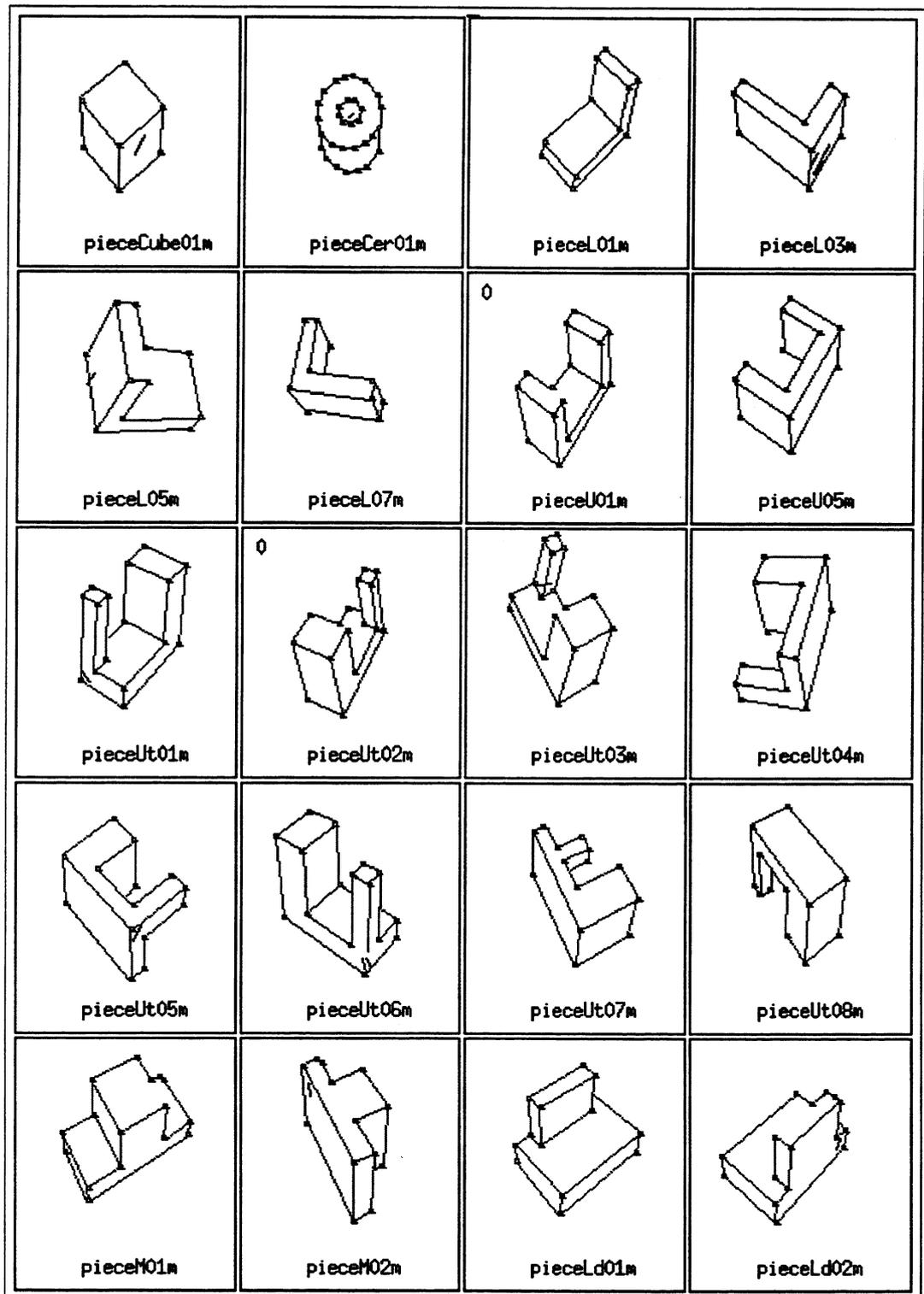


Figure 6.23: Notre base expérimentale contenant les 20 vues appartenant aux 7 objets choisis. Les vues dans la base de données qui matchent l'objet inconnu sont marquées par l'étiquette correspondante.

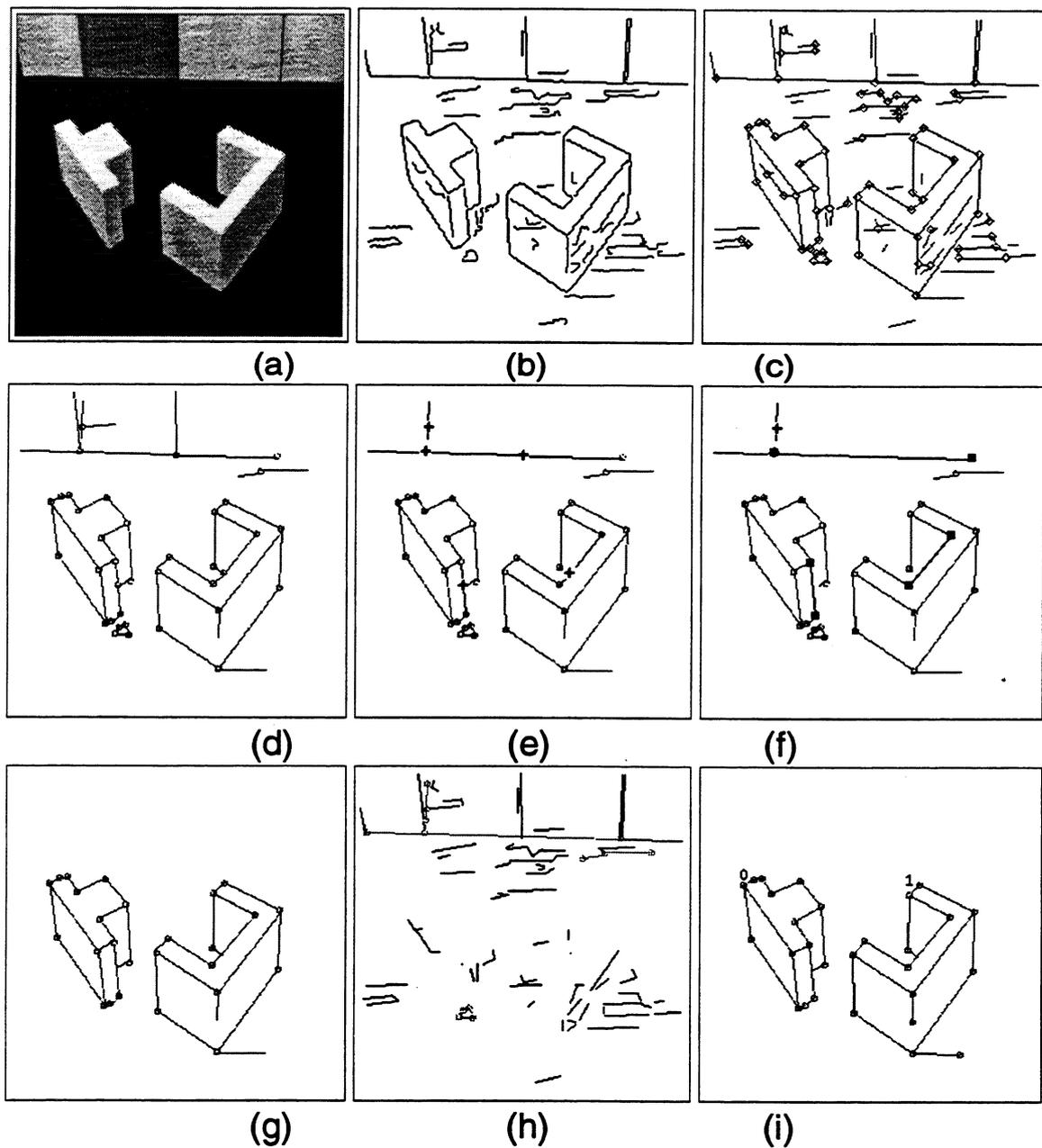


Figure 6.24: (a) Une scène contenant deux objets. (b) l'image des contours. (c) l'image des segments de droite. (d) premier passage de réduction du bruit. (e) élimination des T. (f) les métasegments trouvés. (g) réduction des graphes. (h) ensemble de segments considérés comme étant du bruit. (i) les bons graphes.

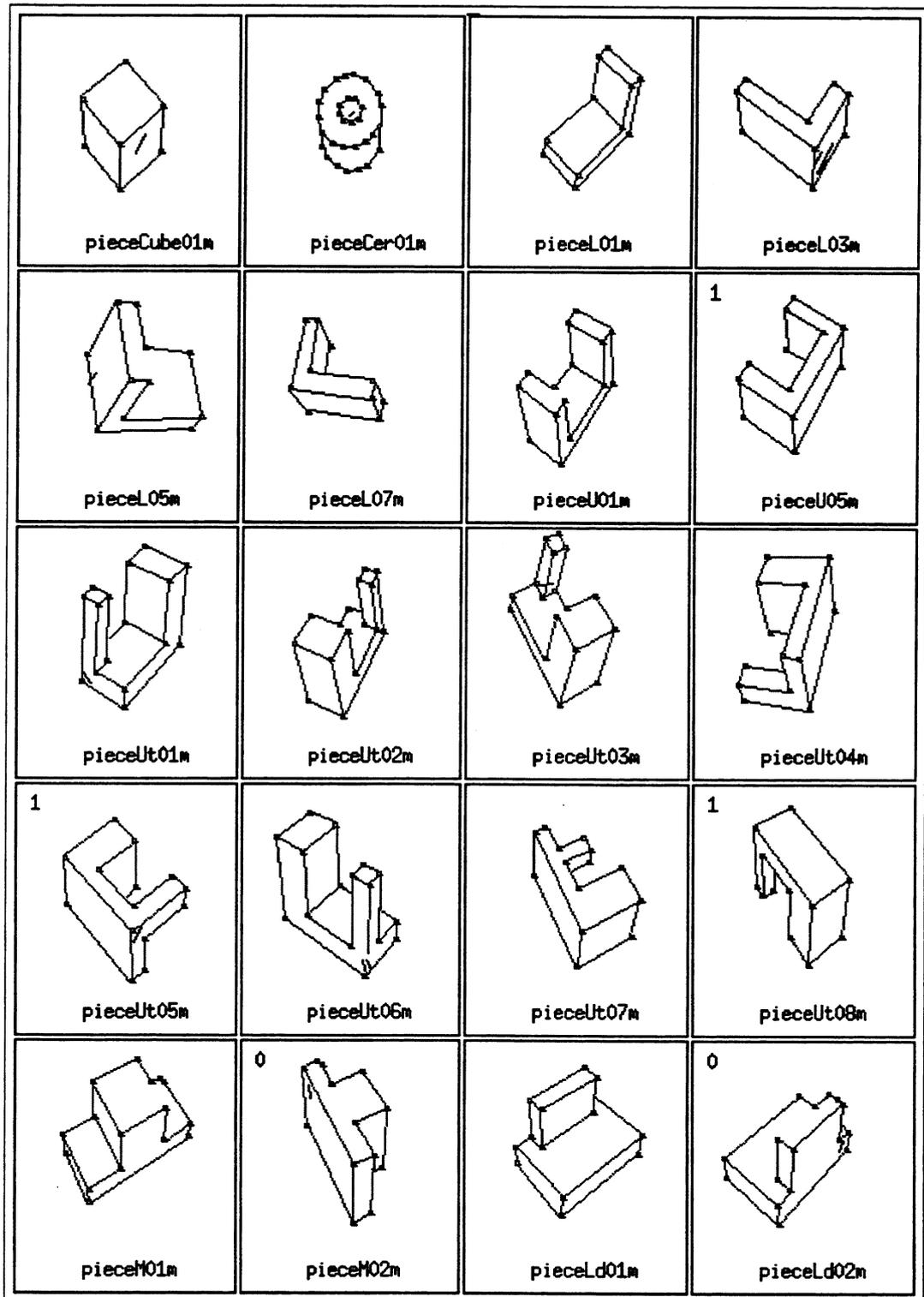


Figure 6.25: Notre base expérimentale contenant les 20 vues appartenant aux 7 objets choisis. Les vues dans la base de données qui matchent les deux objets inconnus sont marquées par l'étiquette correspondante.

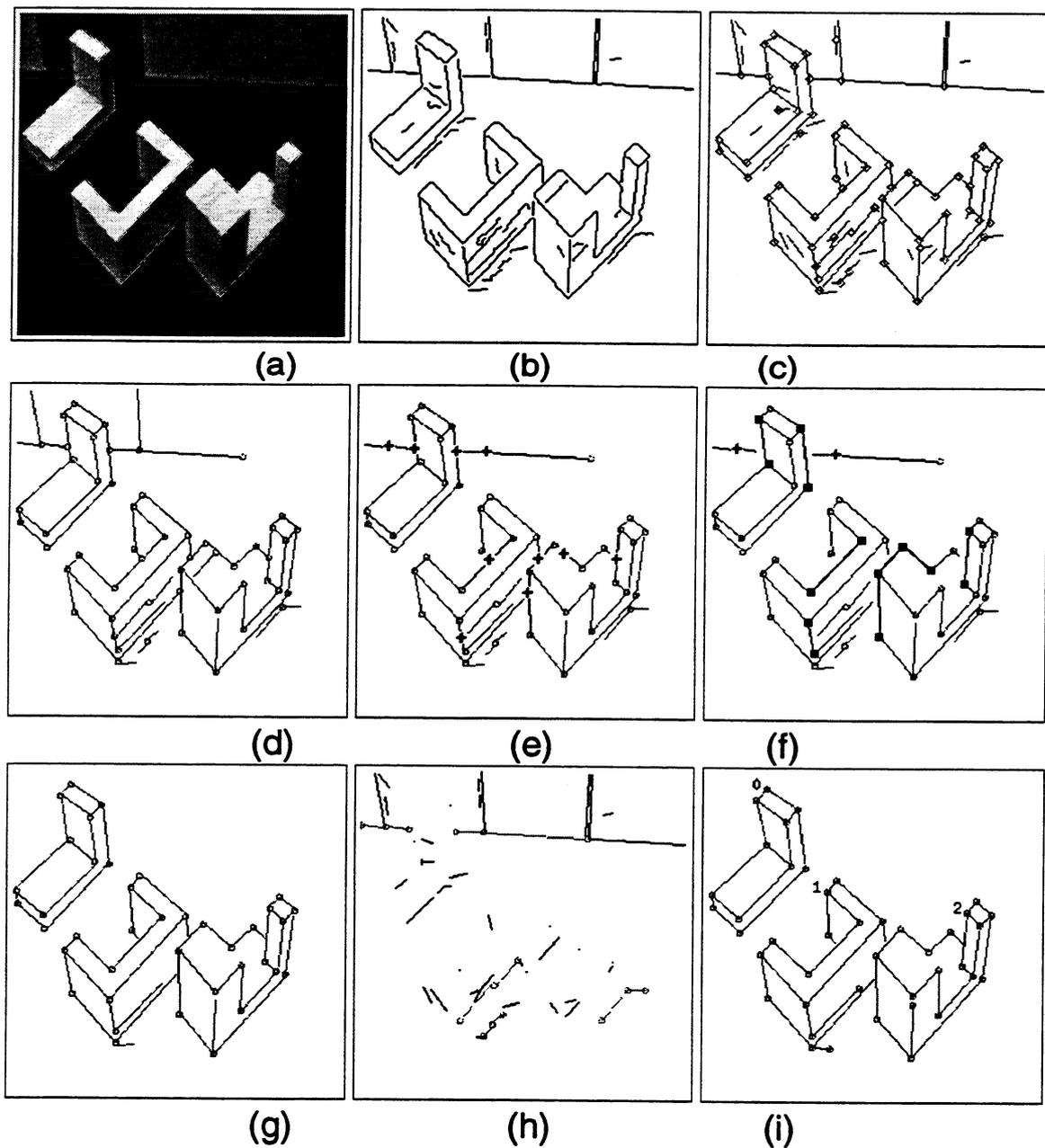


Figure 6.26: (a) Une scène contenant trois objets. Deux d'entre eux se cachent. (b) l'image des contours. (c) l'image des segments de droite. (d) premier passage de réduction du bruit. (e) élimination des T. (f) les métasegments trouvés. (g) réduction des graphes. (h) ensemble de segments considérés comme étant du bruit. (i) les bons graphes.

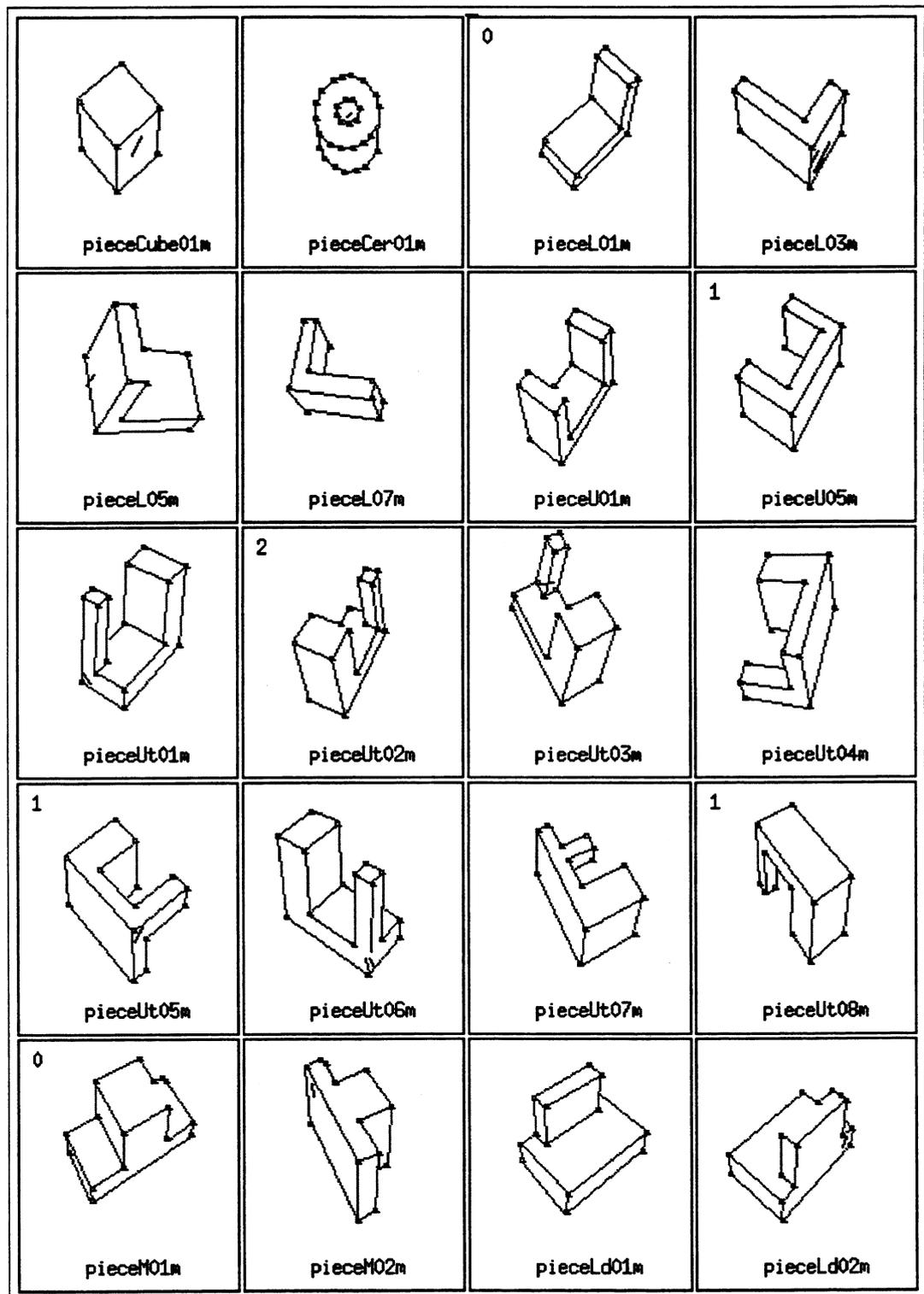


Figure 6.27: Notre base expérimentale contenant les 20 vues appartenant aux 7 objets choisis. Les vues dans la base de données qui matchent les trois objets inconnus sont marquées par l'étiquette correspondante.

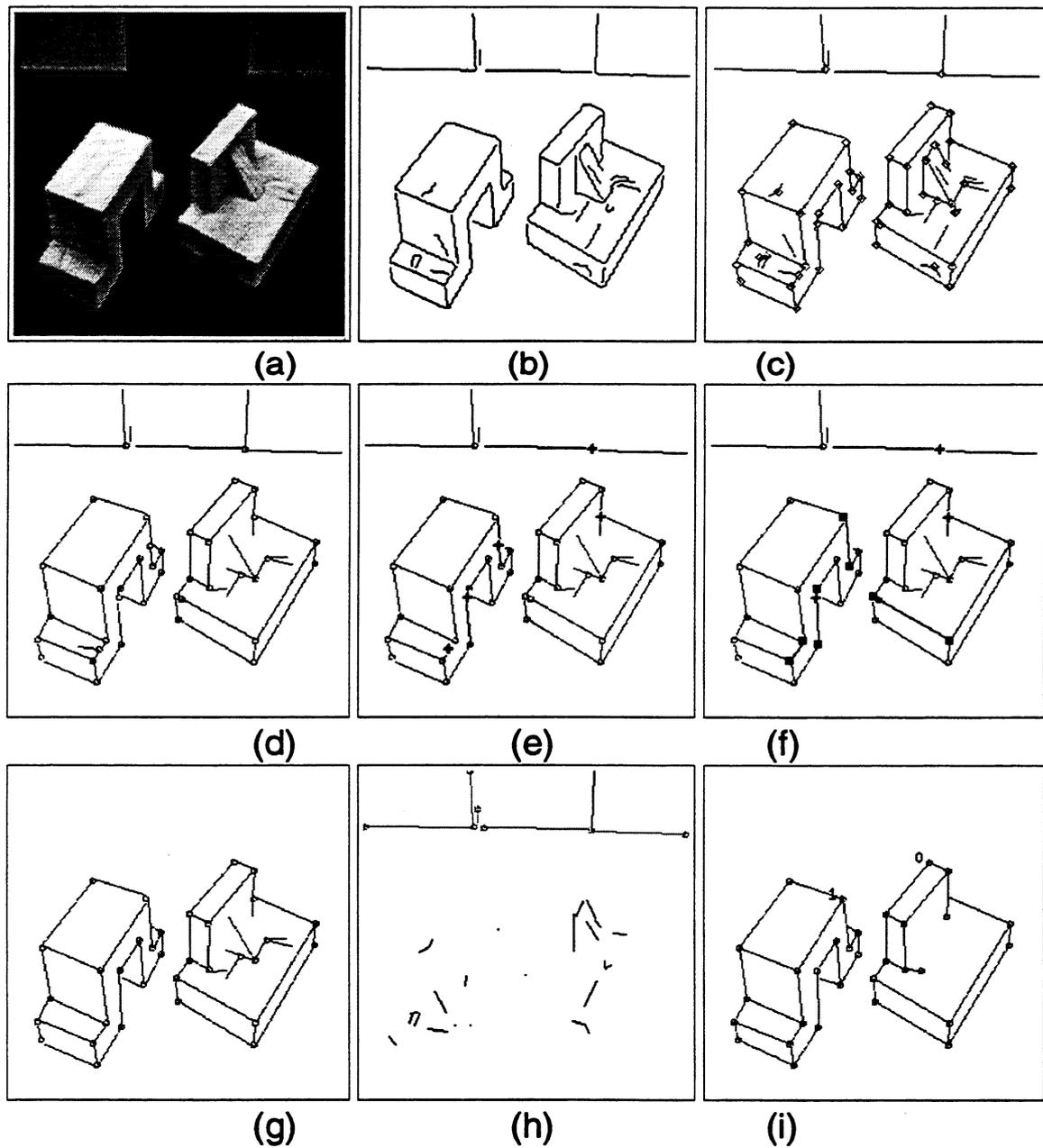


Figure 6.28: (a) Une scène contenant deux objets différents de ceux stockés dans la base d'objets. (b) l'image des contours. (c) l'image des segments de droite. (d) premier passage de réduction du bruit. (e) élimination des T. (f) les métasegments trouvés. (g) réduction des graphes. (h) ensemble de segments considérés comme étant du bruit. (i) les bons graphes.

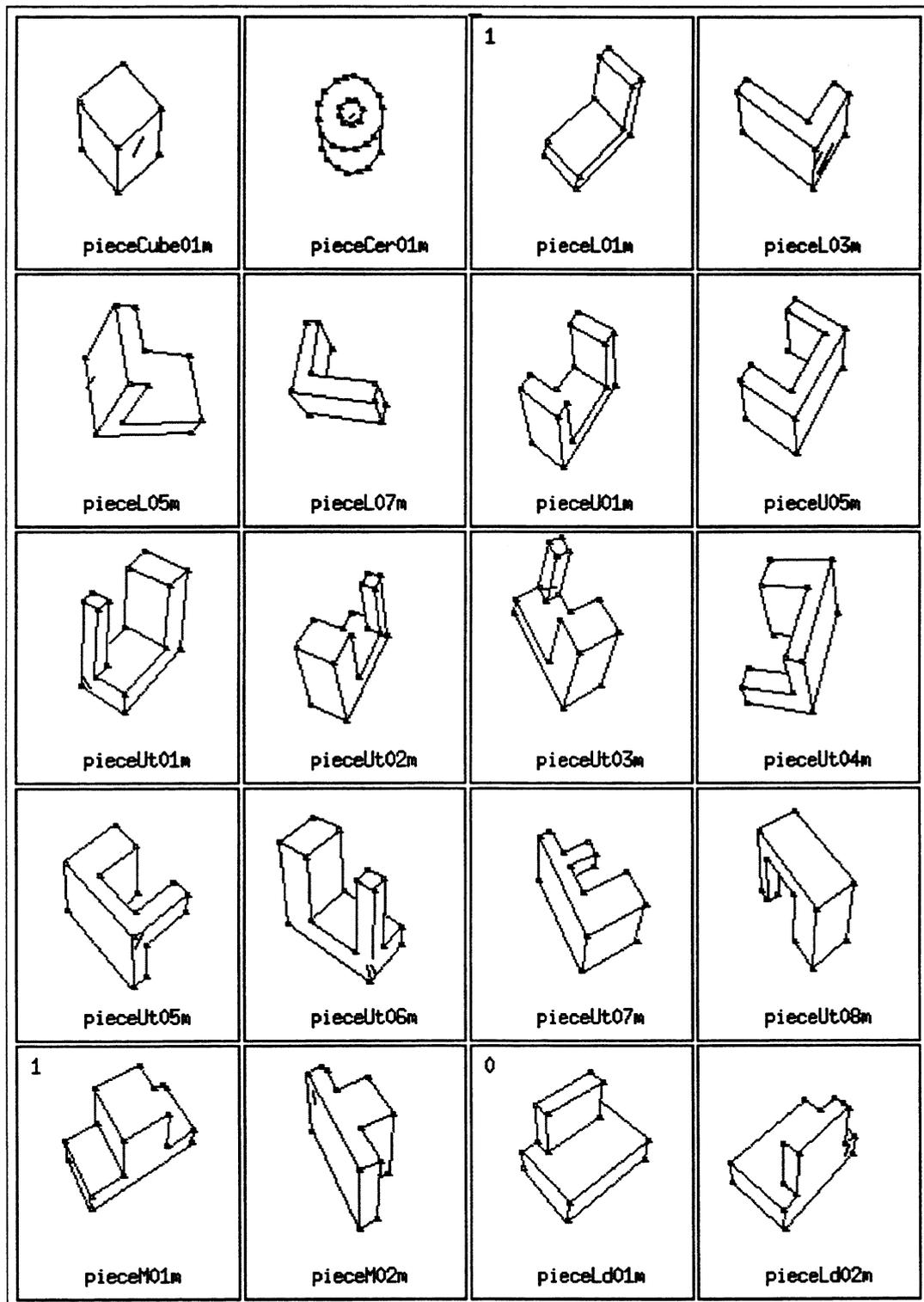


Figure 6.29: Notre base expérimentale contenant les 20 vues appartenant aux 7 objets choisis. Les vues dans la base de données qui matchent le mieux les deux objets inconnus sont marquées par l'étiquette correspondante.

Chapitre 7

Conclusion

7.1 Résumé

Dans ce travail nous avons abordé le problème de la reconnaissance d'objets en présence d'une vaste base de modèles, en développant un système qui :

- **décrit** de façon automatique des objets 3D,
- **utilise** le même ensemble de primitives pour représenter les modèles et les objets de la scène et
- **fournit** une liste restreinte à quelques modèles candidats à la reconnaissance.

Les objets sont décrits automatiquement par l'intermédiaire d'une représentation à trois couches :

1. une couche supérieure contenant une description à objet centré,
2. une couche centrale contenant une description de l'objet par une liste de vues caractéristiques et
3. une couche inférieure où chaque vue caractéristique est décrite par une liste de figures fondamentales et une liste des intersections de figures fondamentales.

La base de modèles suit cette structure d'organisation à trois couches, mais à l'inverse. En bas, on a une liste de figures fondamentales (FF) et une liste d'intersections de figures fondamentales (IFF). Au milieu, on a une liste de vues caractéristiques. Enfin, en haut, on a une liste d'objets. Seules les FF et les IFF sont organisées dans des tableaux de hachage.

Une propriété importante de la représentation résultante est que celle-ci est évaluée sans aucune connaissance a priori concernant les modèles existants. Ceci est important lorsque l'environnement de travail est inconnu ou lorsque le nombre d'objets est grand.

Au moment de la recherche, le mécanisme d'indexation prend en entrée un ensemble de figures fondamentales et leurs intersections et essaye de déterminer quel modèle (ou modèles) contient ces figures afin de déterminer quel modèle (ou modèles) est probablement présent dans l'image.

Les processus de construction de la base de données et de recherche utilisent la même technique d'indexation.

Cette technique d'indexation est possible grâce à une approche de caractérisation provenant de la théorie des graphes en termes d'un ensemble d'entiers. Ces entiers sont facilement évalués par l'aide du polynôme deuxième immanent appliqué à la matrice associée au graphe.

De plus, la complexité de calcul des coefficients du polynôme $d_2(G)$ pour un graphe G est n^4 .

7.2 Apports

L'apport principal de cette recherche est celui d'une méthode générale provenant de la théorie des graphes pour caractériser deux propriétés d'un graphe G à n nœuds et à m arêtes : sa topologie et son apparence.

Caractérisation de la topologie du graphe :

- celle-ci est caractérisée par un ensemble de $n + 1$ entiers : c_0, c_1, \dots, c_n ,
- ces entiers décrivent de façon unique la topologie du graphe lorsque la taille du graphe $n \leq 12$,
- ils sont les coefficients du polynôme d_2 et
- ils peuvent être facilement évalués à partir de la matrice laplacienne du graphe.

Caractérisation de l'apparence du graphe :

- celle-ci est caractérisée localement (pour l'ensemble des arêtes les plus proches du "centre" du graphe) par un ensemble de $q + 1$ entiers : c'_0, c'_1, \dots, c'_q ,
- ces entiers sont aussi les coefficients du polynôme d_2 , mais appliqué cette fois-ci à la matrice d'apparence $M(G)$.

7.3 Inconvénients et améliorations

La performance et l'efficacité de notre système, du point de vue de la recherche, dépendent fondamentalement des étapes suivantes :

- **diviser** le graphe-image en sous-graphes tels que chacun d'eux provient d'un même objet et
- **trouver** les bonnes figures fondamentales et leurs intersections pour chacun de ces sous-graphes.

Ces deux étapes du système dépendent en plus de deux facteurs primordiaux : la complexité de l'image et le bruit qui y est contenu. Dans le cas idéal d'une image simple où il n'y a pas de bruit, diviser le graphe-image en sous-graphes et trouver les figures fondamentales respectives pour chacun de ces sous-graphes est un problème facile à résoudre.

Dans la pratique, on peut toujours s'arranger pour avoir des scènes simples mais il est évident qu'il est pratiquement impossible d'éviter la présence d'une certaine quantité de bruit dans une image, même si l'image est très simple. Par rapport à cette dernière situation il existe encore deux cas :

1. le bruit contenu dans l'image n'affecte pas tellement la structure du graphe-image. En conséquence, notre système peut encore donner de bons résultats.
2. la structure du graphe-image se voit sévèrement affectée par le bruit. En conséquence, la performance de notre système se verra aussi affectée.

Dans le premier cas, on a constaté qu'en général le système donne de bons résultats sauf peut-être dans certains cas où il faut passer l'image par le module de réduction du bruit afin que le système ait un bon comportement. Ceci dépend de deux facteurs :

1. de la partie de la structure du graphe affectée par le bruit et
2. du rapport (information / bruit).

Dans le deuxième cas, où la structure du graphe est sévèrement affectée par le bruit, il faut résoudre au moins l'un des problèmes suivants :

1. **améliorer** le module de réduction du bruit,
2. **améliorer** le module de division du graphe-image en sous-graphes,
3. **proposer** une autre technique de regroupement afin d'obtenir d'autres groupements (d'autres figures fondamentales différentes à celles proposées dans ce travail) moins sensibles au bruit.

7.4 D'autres améliorations

Une fois que le système d'indexation a fourni une liste restreinte à quelques modèles candidats à la reconnaissance, il nous faut encore déterminer :

- l'**identité** de chaque objet image avec un modèle de la liste restreinte, c'est-à-dire établir une correspondance terme à terme objet-modèle et
- la **position** de chaque objet dans l'image par rapport à un repère.

Une solution possible pour résoudre ces deux questions consiste à utiliser par exemple un ensemble d'invariants géométriques attachés à chaque figure fondamentale. Cette information peut être utilisée d'avantage pour restreindre encore les listes de modèles pendant le processus d'indexation.

Une autre amélioration possible consiste à caractériser d'autres propriétés du graphe, comme le parallélisme, la colinéarité, et la symétrie de ses arêtes. Ceci devra permettre à notre système. par exemple, de pouvoir faire la différence entre les objets de la figure 7.1

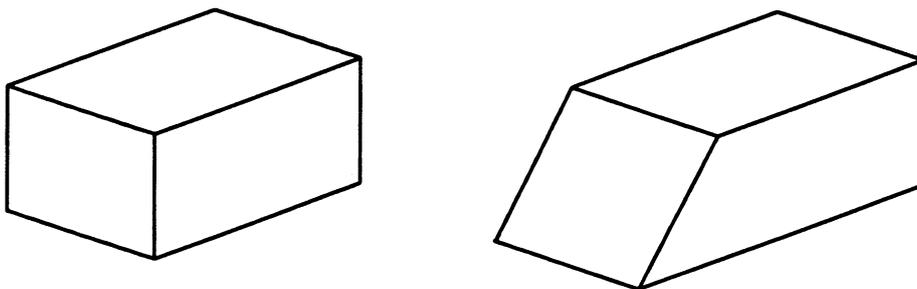


Figure 7.1: La caractérisation de propriétés comme le parallélisme, la colinéarité, et la symétrie permet de faire la différence entre ces deux objets.

Finalement il faut essayer de généraliser la méthode aux objets non polyédriques sachant que leur diversité est beaucoup plus grande que celle des objets polyédriques.

Bibliographie

- [1] J. Aggarwal and Y. Kim. Rectangular parallelepiped coding: A volumetric representation of three-dimensional objects. *International Journal of Robotics and Automation*, 2(3):127–134, 1986.
- [2] G. J. Agin. *Representation and Description of Curved Objects*. PhD thesis, Stanford University, 1972.
- [3] G. E. Andrews. *The Theory of Partitions*. Addison-Wesley, Publishing Company, Inc., Massachusetts, 1976.
- [4] N. Ayache and O. D. Faugeras. HYPER: A new approach for the recognition and positioning of two-dimensional objects. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, PAMI-8(1):44–54, January 1986.
- [5] R. Bacjy and F. Solina. Three dimensional object representation revisited. In *First International Conference on Computer Vision*, pages 231–240, London, June 1987.
- [6] D. H. Ballard and C. M. Brown. *Computer Vision*. Prentice Hall Inc., 1982.
- [7] P. J. Bels and R. C. Jain. Three-dimensional object recognition. *ACM Computing Surveys*, 17:75–145, 1985.
- [8] C. Berge. *Graphes*. Gauthier-Villars, Paris, 1983.
- [9] R. Bergevin and M. Levine. Recognition of 3-d objects in 2-d line-drawings: An approach based on geons. Technical Report CIM-88-24, McGill Research Centre for Intelligent Machines, 1988.
- [10] T.O. Binford. Visual perception by computer. In *Proc. IEEE Conference on Systems and Control*, Miami, USA, December 1971.

- [11] H. Boerner. *Representations of Groups*. American Elsevier, New York, 1970.
- [12] R. C. Bolles and R. A. Cain. Recognizing and locating partially visible objects, the Local-Feature-Focus method. *International Journal of Robotics Research*, 1(3):57–82, 1982.
- [13] R. C. Bolles and R. Horaud. 3DPO: A Three-Dimensional Part Orientation System. *International Journal of Robotics Research*, 5(3):3–26, Fall 1986.
- [14] T. Boult and A. Gross. On the recovery of superellipsoids. In *DARPA Image Understanding Workshop*, pages 1052–1063, Cambridge, Massachusetts, April 1988.
- [15] T. M. Breuel. Adaptive model base indexing. In *Proc. DARPA IU Workshop*, pages 805–814, 1989.
- [16] R. A. Brooks. Symbolic reasoning among 3D models and 2D images. *Artificial Intelligence*, 17(1-3):285–348, August 1981.
- [17] R. A. Brooks. Model-based three-dimensional interpretations of two-dimensional images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(2):140–150, 1983.
- [18] J. B. Burns and E. M. Riseman. Matching complex images to multiple 3d objects using view descriptions networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 328–334, Champaign, Illinois, June 1992.
- [19] R. T. Chin and C. R. Dyer. Model-based recognition in robot vision. *ACM Computing Surveys*, 18:67–108, 1986.
- [20] D. Clemens and D. Jacobs. Model group indexing for recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4–9, 1991.
- [21] D. Clemens and D. Jacobs. Space and time bounds on indexing 3-d models from 2-d images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(10):1007–1017, 1991.
- [22] G. M. Constantine. Schur convex functions on the spectra of graphs. *Discrete Math.*, 45:181–188, 1983.
- [23] G. M. Constantine. Graph complexity and the laplacian matrix in blocked experiments. *Linear and Multilinear Algebra*, 28:49–56, 1990.
- [24] D. M. Cvetkovic, M. Doob, I. Gutman, and A. Torgasev. *Recent Results in the Theory of Graph Spectra*. North-Holland, Amsterdam, 1988.
- [25] D. M. Cvetkovic, M. Doob, and H. Sachs. *Spectra of Graphs*. Academic Press, New York, 1980.

- [26] D. M. Cvetkovit. Some graph invariants in the eigenvectors of the adjacency matrix. In *Proceedings of the Eighth Yugoslav Seminar on Graph Theory*, pages 31–42, 1987.
- [27] R. deFigueiredo. A framework for automation of 3d machine vision. *TI Tech. J.*, pages 62–72, 1987.
- [28] R. Deriche. Using Canny's criteria to derive a recursively implemented optimal edge detector. *International Journal of Computer Vision*, 1(2):167–187, 1987.
- [29] S. Dickinson, Pentland A., and A. Rosenfeld. 3-d shape recovery using distributed aspect matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):174–198, 1992.
- [30] S. Dickinson, Pentland A., and A. Rosenfeld. From volumes to views: An approach to 3-d object recognition. *CVGIP: Image Understanding*, 55(2):130–154, 1992.
- [31] J. R. Engelbrecht and F. M. Wahl. Polyhedral object recognition using hough-space features. *Pattern Recognition*, 21(6):155–168, 1988.
- [32] G. J. Ettinger. Large Hierarchical Object Recognition Using Libraries of Parameterized Model Sub-Parts. In *Proc. Computer Vision and Pattern Recognition*, pages 32–41, Ann Arbor, Michigan, USA, June 5-9 1988.
- [33] T. J. Fan. *Describing and Recognizing 3-D Objects Using Surface Properties*. PhD thesis, Institute for Robotics and Intelligent Systems. University of Southern California, 1988.
- [34] T. J. Fan, G. Medioni, and R. Nevatia. Recognizing 3-d objects using surface descriptions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(11):1140–1157, 1989.
- [35] O. D. Faugeras, J. Mundy, N. Ahuja, Ch. Dyer, A. Pentland, R. Jain, and K. Ikeuchi. Why aspect graphs are not (yet) practical for computer vision. *CVGIP: Image Understanding*, 55(2):212–218, 1992.
- [36] O.D. Faugeras and M. Hebert. The representation, recognition, and locating of 3-d objects. *International Journal of Robotics Research*, 5(3):27–52, Fall 1986.
- [37] M. Fisher. On hearing the shape of a drum. *J. Combinatorial Theory*, 1(1):105–125, 1966.
- [38] P. J. Flynn and A. K. Jain. CAD-Based Computer Vision: From CAD Models to Relational Graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-13(2):114–132, February 1991.

- [39] P. J. Flynn and A. K. Jain. 3d object recognition using invariant feature indexing of interpretation tables. *CVGIP: Image Understanding*, 55(2):119–129, March 1992.
- [40] L. R. Foulds. *Graph Theory Applications*. Springer, New York, 1992.
- [41] F. R. Gantmacher. *Théorie des matrices, Tome 1*. Dunod, Paris, 1966.
- [42] M. Gondran and M. Minoux. *Graphes et algorithmes*. Editions Eyrolles, 61, Bd. Saint-Germain Paris 5, 1985.
- [43] W.E. Grimson. The effect of indexing on the complexity of object recognition. In *Third International Conference on Computer Vision*, pages 644–651, Osaka, Japan, December 1990.
- [44] W.E.L. Grimson and T. Lozano-Perez. Model-based Recognition and Localization from Sparse Range or Tactile Data. *International Journal of Robotics Research*, 3(3):3–35, Fall 1984.
- [45] W.E.L. Grimson and T. Lozano-Perez. Localizing overlapping parts by searching the interpretation tree. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(4):469–482, 1987.
- [46] B. Grone. An inequality for the second immanant. *Linear and Multilinear Algebra*, 18:147–152, 1985.
- [47] J. L. Gross and Th. W. Tucker. *Topological Graph Theory*. Wiley-Interscience, New York, Chichester, Brisbane, Toronto, Singapore, 1987.
- [48] F. Harary. The determinat of the adjacency matrix of a graph. *SIAM Rev.*, 4:202–210, 1962.
- [49] F. Harary. *Graph Theory*. Addison-Wesley, Reading, MA, 1969.
- [50] F. Harary, C. King, A. Mowshowitz, and R. Read. Cospectral graphs and digraphs. *Bull. London Math. Soc.*, 3:321–328, 1971.
- [51] W. Hartmann. On the complexity of immanants. *Linear and Multilinear Algebra*, 18:127–140, 1985.
- [52] P. Heyfron. Some inequalities concerning immanants. *Math. Proc. Camb. Phil. Soc.*, 109:15–30, 1991.
- [53] J. Hong and H. Wolfson. An improved model-based matching method using footprints. In *International Conference Pattern Recognition*, pages 72–78, Rome, Italy, 1988.
- [54] R. Horaud. New Methods for Matching 3-D Objects with Single Perspective Views. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, PAMI-9(3):401–412, May 1987.

- [55] R. Horaud, F. Veillon, and Th. Skordas. Finding Geometric and Relational Structures in an Image. In O. Faugeras, editor, *Computer Vision – ECCV 90, Proceedings First European Conference on Computer Vision, Antibes, France*, pages 374–384. Springer Verlag, April 1990.
- [56] E. Horowitz and S. Sahni. *Fundamentals of Data Structures*. Computer Science Press, Inc., Maryland, 1982.
- [57] D. W. Jacobs. Space efficient 3d model indexing. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 439–444, Champaign, Illinois, June 1992.
- [58] G. James. *The Representation Theory of the Symmetric Groups*. Springer-Verlag, New York, 1978.
- [59] G. James and A. Kerber. *The Representation Theory of the Symmetric Groups*. Addison-Wesley, Publishing Company, Inc., Massachusetts, 1981.
- [60] G. D. James and M. W. Liebeck. Permanents and immanants of hermitian matrices. *Proc. London Math. Soc.*, 3(55):243–265, 1987.
- [61] C. R. Johnson and N. Newman. A note on cospectral graphs. *J. Comb. Theory Ser. B*, 28:96–103, 1980.
- [62] Ch. R. Johnson and S. Pierce. Inequalities for single hook immanants. *Linear Algebra and its applications*, 102:55–79, 1988.
- [63] A. Kalvin, E. Schomberg, J. T. Schwartz, and M. Sharir. Two-dimensional model-based, boundary matching using footprints. *The International Journal of Robotics Research*, 5(4):38–54, 1986.
- [64] G. Kirchhoff. Über die auflösung der gleichungen, aus welche man bei der untersuchung der linearen verteilung galvanischer strome gefurht wird. *Ann. Phys. Chem.*, 72:497–508, 1847.
- [65] T. F. Knoll and R. C. Jain. Recognizing partially visible objects using feature indexed hypotheses. *IEEE Journal of Robotics and Automation*, 2(1):3–13, March 1986.
- [66] Y. Lamdan, J. Schwartz, and H. Wolfson. Object recognition by affine invariant matching. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 335–344, 1988.
- [67] Y. Lamdan, J. Schwartz, and H. Wolfson. On recognition of 3-d objects from 2-d images. In *IEEE International Conference on Robotics and Automation*, pages 1407–1413, 1988.
- [68] Y. Lamdan and H. Wolfson. Geometric hashing: A general and efficient model-based recognition scheme. In *Second International Conference on Computer Vision*, pages 238–249, Tampa, Florida, USA, December 1988.

- [69] D. Littlewood. *The Theory of Group Characteres*. University Press, London, 1958.
- [70] D. Lowe. Three-dimensional Object Recognition from Single Two-dimensional Images. *Artificial Intelligence*, 31:355–395, 1987.
- [71] S. Lu and A. K. C. Wong. Analysis of 3-d scene with partially occluded objects for robot vision. In *International Conference Pattern Recognition*, pages 99–103, Rome, Italy, 1988.
- [72] M. Markus and H. Minc. *A Survey of Matrix Theory and Matrix Inequalities*. Allyn and Bacon, INC., Boston, 1964.
- [73] D. Marr. *Vision*. W. H. Freeman, San Francisco, 1982.
- [74] R. Mehrotra and W. Grosky. Shape matching utilizing indexed hypotheses generation and testing. *IEEE Journal of Robotics and Automation*, 5(1):70–77, February 1989.
- [75] R. Merris. The laplacian permanental polynomial for trees. *Czech. J. Math.*, 107:397–403, 1982.
- [76] R. Merris. The second immanantal polynomial and the centroid of a graph. *SIAM Journal Alg. Disc. Meth.*, 7:484–503, 1986.
- [77] R. Merris. Almost all trees are co-immanantal. *Linear Algebra ant its Applications*, 150:61–66, 1991.
- [78] R. Merris, K. R. Rebman, and W. Watkings. Permanental polynomials of graphs. *Linear Algebra Applications*, 38:273–288, 1981.
- [79] R. Merris and W. Watkings. Inequalities and identites for generalized matrix functions. *Linear Algebra Applications*, 64:223–242, 1985.
- [80] H. Minc. *Permanents, Encyclopedia of Mathematics and its Applications, vol. 6*. Addison-Wesley Publising Company, MS, 1978.
- [81] B. Mohar. Laplacian matrices of graphs. *Preprint Ser. Dept. Math. Univ. E. K. Ljubljana*, 26:385–392, 1988.
- [82] B. Mohar. The laplacian spectrun of graphs. *Preprint Ser. Dept. Math. Univ. E. K. Ljubljana*, 26:353–384, 1988.
- [83] A. Mowshowitz. The characteristic polynomial of a graphe. *Journal of Combinatorial Theory*, 12(B):177–193, 1972.
- [84] R. Nevatia. *Machine Perception*. Prentix-Hall, Inc., Englewood Cliffs, New Jersey, 1982.
- [85] R. Nevatia and T. Binford. Description and recognition of complex-curved objects. *Artifitial Intelligence*, 8:77–98, 1977.

- [86] M. Oshima and Y. Shirai. Object recognition using three-dimensional information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 3(4):353–361, 1983.
- [87] A. Pentland. Perceptual organization and the representation of natural form. *Artificial Intelligence*, 28:293–331, 1986.
- [88] S. B. Pollard, T. P. Pridmore, J. Porril, J. E. W. Mayhew, and J. P. Frisby. Geometrical Modeling from Multiple Stereo Views. *Int. J. of Robotics Research*, 8(4):3–32, August 1989.
- [89] K. Rao and R. Nevatia. Generalized cone descriptions from shape descriptions from 3-d data. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 256–263, 1986.
- [90] L.G. Roberts. Machine perception of three-dimensional objects. Technical Report 315, MIT, Lincoln Laboratory, May 1963.
- [91] C. Rothwell, A. Zisserman, D. Forsyth, and J. Mundy. Using projective invariants for constant time library indexing in model based vision. In *British Machine Vision Conference*, pages 62–70, London, September 1991.
- [92] C. Rothwell, A. Zisserman, D. Forsyth, and J. Mundy. Canonical frames for planar object recognition. In *Second European Conference on Computer Vision*, pages 757–772, Santa Margherita Liguere, Italy, May 1992.
- [93] C. Rothwell, A. Zisserman, D. Forsyth, and J. Mundy. Efficient model library access by projectively invariants indexing functions. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 109–114, Champaign, Illinois, June 1992.
- [94] R. Sedgewick. *Algorithms*. Addison-Wesley Publishing Company, Inc., Massachusetts, 1988.
- [95] J. Serre. *Représentations Linéaires des Groupes Finis*. Hermann, Paris, 1978.
- [96] F. Stein and G. Medioni. Efficient two dimensional object recognition. In *Proceedings of International Conference on Pattern Recognition*, pages 13–17, Atlantic City, New Jersey, June 1990.
- [97] F. Stein and G. Medioni. Structural hashing: Efficient three dimensional object recognition. In *Proceedings Computer Vision and Pattern Recognition*, pages 244–250, Hawaii, USA, June 1991.
- [98] F Stein and G. Medioni. Structural hashing: Efficient 3-d object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):125–145, February 1992.

- [99] M. J. Swain and D. H. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11–32, November 1991.
- [100] J. D. Thomas and J. D. Zund. A note on the theory of immanants. *Rendiconti del Circolo Matematico di Palermo*, XXVIII:143–150, 1979.
- [101] J. Turner. Generalized matrix functions and the graph isomorphism problem. *SIAM J. Appl. Math.*, 16:520–526, 1968.
- [102] W. T. Tutte. *Graph Theory*. Addison-Wesley Publishing Company, Inc., Massachusetts, 1984.
- [103] L. G. Valiant. The complexity of computing the permanent. *Theoretical Computer Science*, 8:189–201, 1979.
- [104] A. Vince. Elementary divisors of graphs and matroids. *Europ. J. Combinatorics*, 12:445–453, 1991.

Résumé

Cette thèse s'intéresse au problème de la reconnaissance d'objets en présence d'une vaste base de modèles. L'indexation de modèles peut être décrite comme suit: étant donné un groupe d'indices image, extraire rapidement de la base de modèles les modèles contenant le groupe d'indices.

La combinatoire du paradigme classique d'appariement indices-modèles étant prohibitive dans ce cas, nous proposons une méthode d'indexation par une technique de hachage de graphes.

Nous décrivons d'abord les principes: codage d'image et des modèles sous une forme de graphes, gestion d'une base de modèles par tables hash-codées, comparaison de graphes sur la base de leur caractérisation polymômiale, accès à la base de modèles.

Ensuite, nous décrivons le système mis en œuvre selon ces principes, et nous présentons quelques résultats d'expérimentation.

Mots-clés : - vision par ordinateur - reconnaissance d'objets - indexation - théorie des graphes - caractérisation de graphes - polynôme deuxième immanent - hachage.