Institut National
Polytechnique de Lorraine

CRIN-CNRS
Inria Lorraine

Ecole doctorale IAE+M
Informatique

Département de Formation Doctorale

Reconnaissance par indexation en vision par ordinateur

THÈSE

présentée et soutenue le **Vendredi 13 novembre 1992**

pour l'obtention du

Doctorat de l'Institut National Polytechnique de Lorraine

(Spécialité Informatique)

par

Stéphane Paris

Composition du jury :

Président : Roger MOHR

Rapporteurs : James L. CROWLEY
Michel HABIB

Examinateurs : Jean-Paul HATON
Gérald MASINI
Amedeo NAPOLI

INPL
INSTITUT NATIONAL POLYTECHNIQUE DE LORRAINE

AUTORISATION DE SOUTENANCE DE THESE DU DOCTORAT DE L'INSTITUT NATIONAL POLYTECHNIQUE DE LORRAINE

o0o

VU LES RAPPORTS ETABLIS PAR :
Monsieur CROWLEY James, Professeur, LIFIA/INPG Grenoble,
Monsieur HABIB Michel, Professeur, C.R.I.M. Montpellier.

Le Président de l'Institut National Polytechnique de Lorraine, autorise :

Monsieur PARIS Stéphane

à soutenir devant l'INSTITUT NATIONAL POLYTECHNIQUE DE LORRAINE, une thèse intitulée :
"Reconnaissance par indexation en vision par ordinateur"

en vue de l'obtention du titre de :

DOCTEUR DE L'INSTITUT NATIONAL POLYTECHNIQUE DE LORRAINE

Spécialité : **"INFORMATIQUE"**

Fait à Vandoeuvre le, **29 Octobre 1992**
Le Président de l'I.N.P.L.,
M. LUCIUS

2, avenue de la Forêt de Haye - B.P. 3 - 54501 VANDŒUVRE CEDEX

Téléphone : 83. 59. 59. 59 - Télex : 961 715 F - Télécopie : 83. 59. 59. 55

Remerciements

Ce mémoire est pour moi l'occasion d'exprimer des remerciements à tous ceux qui m'ont conseillé et aidé à réaliser ce travail dans de bonnes conditions, et je suis heureux aujourd'hui de remercier plus particulièrement :

Monsieur Roger Mohr, Professeur à l'institut Polytechnique de Grenoble, qui me fait l'honneur de présider ce jury,

Monsieur James L. Crowley, Professeur à l"institut Polytechnique de Grenoble, et Monsieur Michel Habib, Professeur au Centre de Recherche en Informatique de Montpellier, pour avoir accepté d'être les rapporteurs de cette thèse,

Messieurs Jean-Paul Haton, Professeur de l'Université de Nancy I, et Amedeo Napoli, chargé de recherche CR1 CNRS au Centre de Recherche en Informatique de Nancy, pour avoir accepté d'examiner mon mémoire,

Monsieur Gérald Masini, chargé de recherche CR1 CNRS au Centre de Recherche en Informatique de Nancy et directeur de mes travaux, pour le constant intérêt qu'il a porté à ce travail.

J'exprime des remerciements chaleureux à Mademoiselle Houda Chabbi, étudiante en thèse dans l'équipe, et à Mademoiselle Marie-Odile Berger, Chargée de Recherche CR2 INRIA au Centre de Recherche en Information de Nancy, pour les nombreuses et fructueuses discussions que nous avons eues ensemble et sans lesquelles ce travail ne serait pas tout à fait le même.

Et merci de tout cœur à tous mes amis, Sylvie Karst, Annie L'Homée, Véronique Bouras, Régine Karst, Anja Habacha, Georges Paris, Julian Anigbogu, Eric Domenjou, Yves Laprie, Pierre, Marquis et Philippe Hoggan, pour les soutiens moral et matériel qu'ils m'ont apportés.

Table des matières

Introduction

La vision par ordinateur présente une complexité qui a été sous estimée à l'origine. Si des problèmes sont rapidement apparus, aujourd'hui ils ne sont pas tous résolus. C'est ainsi que la création d'un système de vision autonome, c'est-à-dire capable de répondre aux besoins de déplacements et de manipulations d'un robot, s'avère limitée à des environnements définis par avance. Cependant, l'indépendance du robot demande une "vision" pouvant s'adapter à différents types d'environnements et donc très complexe à réaliser. Le plus simple pour comprendre l'ensemble des difficultés rencontrées dans le domaine de la vision par ordinateur est d'observer l'évolution des recherches dans le temps et la façon dont les chercheurs ont adapté leur travail au fur et à mesure de leurs découvertes.

Brève chronologie

L'histoire de la vision artificielle a commencé dans les années soixante par des recherches sur la compréhension de scènes composées d'objets polyédriques à partir de données parfaites[1] [Waltz, 1975] [Minsky, 1975] [Mackworth, 1977a] : l'étude du monde des blocs. Dans ce cadre d'étude, les données à interpréter sont acquises à partir de dessins où, contrairement aux images réelles, il n'y a pas de pertes, de déformations et de parasitages de l'information. Cependant, les chercheurs se sont très vite rendus compte que les données réelles étaient loin d'être parfaites [Barrow and Tennenbaum, 1981]. En effet, elles sont souvent fortement déformées, parasitées ou partiellement occultées ; en un mot, bruitées.

Partant de cette constatation, diverses études sur le comportement humain en vision[2] [Thorpe, 1988] [Thorpe, 1991] [Droulez, 1991] [Fréganc and Schalchli, 1991] [Zeki, 1991] mettent en évidence la diversité des données utilisées par l'être humain pour la compréhension de scènes : la couleur, la texture et la forme des objets. Ces études ont également permis d'observer la façon dont la vision humaine utilise l'information sous ses formes *bidimensionnelles* (la projection rétinienne) et *tridimensionnelles* (reconstruction 3D par analyse de séquences d'images rétiniennes).
La nouvelle génération de systèmes se focalise sur la reconnaissance d'objets à partir

1. L.G. Roberts, *Machine Perception of Three Dimensional Solids*, Optical and Electro-Optical Information Processing, MIT Press, 1965
2. Wertheimer *Principles of Perceptuals Organization*, 1923

d'images acquises par caméras. Ainsi, contrairement à ceux du monde des blocs, ces systèmes doivent faire appel à des outils mathématiques qui sont hélas très sensibles à la dégradation des données.

Observons plus attentivement le bruit dans les images et sa concéquences au sein des systèmes de vision de la nouvelle génération.

Les images acquises par caméras sont partiellement détériorées par les différents traitements postérieurs à leur acquisition. Elles sont tout d'abord enregistrées sous un format lisible pour un système informatique, c'est-à-dire sous un format binaire. Les images sont donc échantillonnées puis stockées dans une matrice dont chaque élément est appelé *pixel*[3]. Chaque pixel est localisé par ses coordonnées dans l'image et indique la valeur de l'intensité lumineuse qui lui est associée. Cette opération, appelée *discrétisation*, fournit une représentation de l'image plus ou moins fidèle suivant la résolution choisie pour l'échantillonnage qui correspond à la taille de la matrice: le nombre de lignes et de colonnes. De ces images digitalisées, on peut extraire les contours (contours d'objets, d'ombres, etc.) qui sont les chaînes de pixels de fort *gradient*: les pixels dont la valeur de la différence d'intensité avec leurs voisins est marquée. On peut également extraire les régions, qui représentent les zones de l'image dont le gradient est nul (ou quasiment). Les contours peuvent être polygonalisés sous forme de segments 2D comme ceux de la scène illustrée par la figure 0.1.a. Cette scène est composée d'une

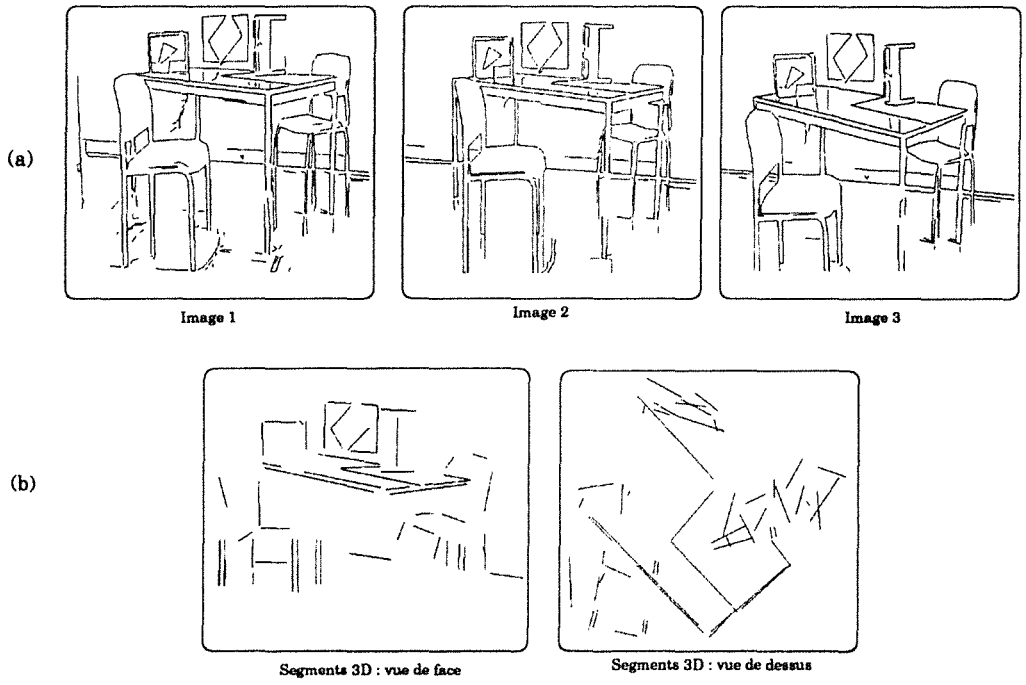


Figure 0.1. (a) *Les segments* 2D *des 3 images.* (b) *Reconstitution en segments* 3D *vue de face et de dessus, après intégration des informations* 2D *fournies par les 3 images en A.*

table à quatre pieds, de deux chaises, à quatre pieds également, l'une se situant devant

3. "picture element"

et l'autre derrière la table, et de parallélépipèdes en carton posés sur la table. On peut facilement constater, d'une part, des pertes et des déformations au niveau des segments composant l'information visuelle à interpréter et, d'autre part, la présence de segments parasites. Ce triplet d'image est utilisé pour retrouver l'information 3D. En connaissant les coordonnées du centre focal et du plan image de chacune des trois caméras, il est possible d'apparier les pixels qui proviennent d'un même point physique 3D puis de reconstruire ce point physique en reprojetant les pixels correspondant. La figure 0.1.b présente les résultats de la reconstruction 3D de la scène à partir des segments 2D de la figure 0.1.a. Cette reconstruction cumule les déformations et les pertes de chaque image et donc perturbe encore plus fortement l'information visuelle utile à la reconnaissance. Ainsi à partir des résultats de la reconstruction 3D présentés dans la figure 0.1.b en vues de face et de dessus, il est difficile, même pour l'œil humain, de retrouver la table, les chaises et les cartons.

Ces bruits et ces occultations dans les images entraînent l'utilisation de nouvelles techniques, aptes à prendre en compte ces phénomènes, pour l'identification et la localisation des objets. L'identification revient à étiqueter des zones images en termes d'"objets", ce qui demande une base de modèles des objets à reconnaître, tandis que la localisation revient à déterminer le déplacement permettant de positionner le modèle sur l'image de l'objet, ce qui dépend du type de représentation utilisée pour les images et les modèles. Nous pouvons donc, en une première approche, classer les systèmes selon le type d'images (informations perçues) et le type de modèles (connaissances déjà intégrées) qu'ils utilisent. Ainsi, nous avons les systèmes où :

- les images et les modèles sont bidimensionnels (2D) ; ce sont des images de caméras et les modèles représentent les projections des objets sur un plan.
- les images et les modèles sont tridimensionnels (3D) ; un capteur, composé d'un émetteur et d'un récepteur, délivre directement l'information de profondeur.
- les images sont 2D et les modèles sont 3D ; l'objectif est d'identifier les objets d'une scène observée en 2D.
- les images et les modèles sont 2,5D [Marr, 1982] ; l'information 2D délivrée par un système de caméras est utilisée pour retrouver l'information 3D de la scène.

De plus, la vision artificielle a des contraintes spécifiques suivant les domaines d'applications tel que l'imagerie biomédicale, la navigation robotique, la surveillance ou encore la manipulation robotique. Par exemple, la navigation ou la surveillance a besoin de connaître la géométrie 3D de la scène observée (environnement extérieur du robot) tandis que la manipulation a besoin de connaître la position des objets qui composent cette scène. En effet dans le premier cas, une modélisation des objets susceptibles d'être perçus n'est pas nécessaire alors que dans le deuxième cas elle est obligatoire.
Dans ce mémoire, nous distinguerons les systèmes qui ont pour tâche la manipulation robotique en les regroupant sous la terminologie de systèmes de *compréhension de scènes*.

Le contexte de l'étude

Dans le cadre de la compréhension de scènes, présentons tout d'abord le schéma global du système MultiAGents pour l'Interprétation et la Compréhension de scènes (MAGIC) développé dans notre laboratoire. Ce système analyse des informations de type 2,5D. Il est composé de cinq parties coopérant comme l'illustre la figure 0.2. Il

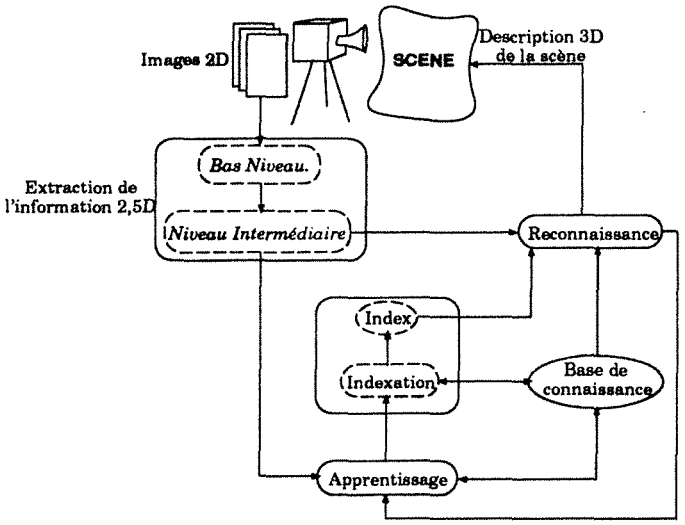


Figure 0.2. *Le schéma du système de vision* MAGIC.

est prévu qu'une architecture multi-agents soit utilisée pour implanter les différents modules du système.

- *L'extraction de l'information* 2,5D correspond à l'acquisition de l'information 2D et à la reconstruction de l'information 3D : le *bas niveau* extrait les informations 2D ; le *niveau intermédiaire* fournit une description 3D construite à partir de l'information 2D. Cette description 3D présente sous la forme de *primitives-image* les parties d'objets non décomposables (les faces par exemples) et sous la forme d'*indices-image* les relations entre les primitives-image (la connexité de deux faces ou l'occultation d'une face par une autre).

- *La base de connaissance* est l'ensemble des modèles d'objets. Un modèle est la description 2D et/ou 3D d'un objet ou d'une classe d'objets. Par analogie aux informations extraites des images, la description se présente en termes de *primitives-modèle* et d'*indices-modèle*.
  A ces modèles peut être ajoutée une description des scènes et des classes de scènes : une scène est décrite par les positions relatives des objets qui la composent.

- *L'index* est une représentation des modèles d'objets de la base de connaissance suivant une description adaptée à l'information extraite des images. Un processus d'*indexation* construit une classification de l'ensemble des modèles des objets de la base à partir de la représentation utilisée pour décrire les images des objets en

termes de primitives-image et d'indices-image. Lors de l'observation d'un nouvel objet, le processus d'indexation réorganise la classification existante en tenant compte des nouvelles informations.
Le processus de reconnaissance se sert de l'index pour sélectionner rapidement les modèles qui peuvent être associés aux données fournies par la partie extraction du système.

- *La reconnaissance* est un processus d'appariement entre l'information 2D et/ou 3D de l'image de la scène et les modèles des objets. Il s'agit d'*identifier* et de *localiser* les objets de la scène observée.

- *L'apprentissage* répond à des besoins d'autonomie et d'évolutivité. Le système doit être capable de constituer lui-même ses propres connaissances. Bien que ce module ne soit pas en cours d'étude dans notre laboratoire, il est possible d'en dresser les principes généraux.
Lors de l'observation d'un nouvel objet, les informations 2D et 3D qui le décrivent, sont stockées pour former un nouveau modèle. De même, si la scène observée n'est pas modélisée dans la base, sa composition en termes d'objets est enregistrée.
Ainsi, le module d'apprentissage doit donc indiquer au module de pilotage du robot les déplacements à effectuer pour observer les différents points de vues nécessaires pour extraire l'information 2D et 3D utiles à la modélisation de l'objet observé.

L'étude

L'étude que nous proposons est destinée à la reconnaissance des objets d'une scène observée avec des caméras à partir d'une base de modèles d'objets. Pour mener à bien ce travail, il a été nécessaire de le scinder en deux. Tout d'abord, il est utile de définir ce qu'est la reconnaissance d'objets et de cerner le plus concrètement possible les différents problèmes rencontrés. Ce travail est présenté sous la forme d'une étude bibliographique (chapitres 1 et 2). Ensuite, un des problèmes posés lors de cette étude bibliographique est étudié et une solution est proposée et discutée. C'est la deuxième partie du travail, qui porte sur l'indexation et qui est présentée dans les chapitres 3 et 4.

Les problèmes rencontrés en reconnaissance d'objets

La reconnaissance d'objets doit permettre de décrire une scène en termes des objets qui la composent, par appariement entre les primitives-image fournies par le module d'extraction et les primitives-modèle fournies par la base de connaissance. Ce processus est de complexité exponentielle puisque tous les appariements entre les primitives-image et les primitives-modèle sont à tester. Ce problème est fort complexe et il n'existe actuellement ni de solution générale ni de principe à suivre. Cependant, la complexité de l'algorithme peut être réduite en décrivant les images et les modèles par les **mêmes**

primitives et les **mêmes** indices. Le terme *description intermédiaire* est couramment utilisé pour nommer ce principe qui est de plus en plus considéré comme solution possible dans la pratique. Cette homogénéisation de l'information peut être faite :

- par construction de primitives 3D à partir de primitives image 2D [Marr, 1982] [Faugeras, 1988]. Le module d'*extraction* est structuré en un *bas niveau* et en un *niveau intermédiaire*. Le bas niveau extrait les primitives 2D, le niveau intermédiaire construit les primitives-image et les indices-image 3D à partir des primitives-image 2D.

- par inférence de configurations 3D à partir d'indices 2D [Lowe, 1985a, chapitre4]. Dans ce cas, un ensemble d'heuristiques de configurations 2D est proposée comme typique afin d'établir des hypothèses de configurations 3D.

- par indexation de la base de connaissance en une classification des modèles d'objets. Cette classification peut être hiérarchique comme les graphes de décision [Brooks, 1981] [Burns and Kitchen, 1987] [Burns and Riseman, 1992] ou plane telle que les tables de hachage [Stein and Médioni, 1992] [Sossa and Horaud, 1992]. Les clefs de hachage sont prédéfinies et représentent des groupes de primitives-image supposés pertinents pour l'identification des objets de l'image à analyser. Les graphes de décision expriment des hiérarchies suivant l'information structurelle des modèles des objets pour servir de guide au processus de reconnaissance lors du regroupement des primitives-image en groupes-image définissant les objets perçus.

Ce mémoire s'intéresse particulièrement à cette dernière catégorie et les chapitres 3 et 4 présentent les processus d'indexation et de reconnaissance que nous avons développés.

La technique d'indexation structurelle étudiée

L'*index* que nous avons développé [Paris and Masini, 1991] [Paris, 1991] est une classification structurelle hiérarchique de l'ensemble des modèles d'objets. Il est construit par un processus d'*indexation* qui réorganise la hiérarchie pour chaque nouveau modèle enregistré dans la base de connaissance. L'index créé est un graphe de décision formé de deux types de nœuds : les nœuds *entrée* représentent les types d'indices connus par le système, comme les relations de connexité entre deux faces ou d'occultation d'une face par une autre, alors que les nœuds *structurels* représentent une structure géométrique présente dans plusieurs modèles.
Afin de mettre à jour de manière efficace la base de connaissance lors de la modélisation, et donc lors de l'indexation d'un nouvel objet, nous avons opté pour un processus d'indexation incrémentiel.
L'adjonction incrémentielle d'un nouvel élément dans une classification hiérarchique est un problème ouvert car, actuellement, il n'existe pas de solution qui, à partir d'un nouvel élément à enregistrer, puisse indiquer sans erreur quelles sont les classes à fusionner, quelles sont les classes à décomposer et dans quelle classe le nouvel élément

doit être stocké. Malgré tous ces problèmes, l'utilisation d'un processus incrémentiel est primordial pour un fonctionnement efficace.

Dans le cadre de la reconnaissance en vision par ordinateur, nous proposons une solution simple et efficace qui permet d'ajouter un modèle d'objet à l'index courant, en utilisant l'information déjà contenue dans ce dernier. L'idée est de ne pas fusionner ni de décomposer l'existant mais seulement d'ajouter les nœuds structurels propres au nouveau modèle. Ceci est possible car la structure des objets (l'information utilisée pour indéxer les modèles) peut se représenter par des éléments de bases (les primitives-modèle et indices-modèle) qui se composent pour former des éléments plus évolués.

Nous avons également étudié le processus de reconnaissance [Paris and Boufama, 1992] afin de valider l'efficacité de la méthode d'indexation par des jeux d'essais. Son algorithme a la particularité d'être similaire à celui du processus d'indexation.

Le plan du mémoire

Le principe de regroupement perceptuel, qui est à l'origine de l'index et des processus d'indexation et de reconnaissance, a été étudié dès 1923 par Wertheimer dans le cadre de la psychologie humaine. Depuis, ces études ont été reprises par des chercheurs en vision par ordinateur [Lowe, 1985a] [Fukushima, 1988].

Chapitre 1 : Les systèmes de reconnaissance d'objets

Pour présenter de manière précise le regroupement perceptuel parmi les techniques de reconnaissance, le chapitre 1 est une étude bibliographique des systèmes de reconnaissance. Nous les scindons en deux grandes catégories : les processus qui identifient toutes les occurrences d'un modèle donné dans une image et les processus qui identifient tous les objets présents dans l'image. Pour la première catégorie, la sélection du modèle à identifier dans l'image est faite auparavant par l'utilisateur ou est imposée par la tâche du robot. En revanche, pour la deuxième catégorie, la sélection des modèles susceptibles d'être présents dans l'image est faite par le processus de reconnaissance. Parmi les processus de cette seconde catégorie, nous distinguons trois classes de techniques de sélection :

- les techniques de sélection guidées par les modèles,
- les techniques de sélection guidées par l'image,
- les techniques de modélisation des objets tels qu'ils sont perçus dans les images.

La sélection guidée par les modèles est typpique du système ACRONYM [Brooks, 1981]. Le but est de construire une hiérarchie des modèles d'objets qui sert ensuite à guider le regroupement perceptuel dans les images. Mais cette hiérarchisation est établie suivant des modèles décrits par des primitives correspondant à certains environnements bien définis et est donc difficile à généraliser.
La sélection guidée par les images est symbolisée par le système RAF de Grimson/Lozano-Pérez [Grimson and Lozano-Pérez, 1987]. L'objectif est d'élaguer les hypothèses

d'appariement entre les primitives-image et les primitives-modèle en se servant des contraintes fournies par les indices-image et indices-modèle. Ceci revient à construire un arbre de recherche *en profondeur d'abord* en sélectionnant une primitive-image et en vérifiant toutes les primitives-modèle qui correspondent. Cette méthode est souvent utilisée car elle est mieux adaptée aux descriptions image. Nous décrivons, entre autres, dans ce chapitre HYPER [Ayache and Faugeras, 1986], SCERPO [Lowe, 1985b], BONSAI [Flynn and Jain, 1991] et "STICKS, BLOBS AND PLATES" [Mulgaonkar *et al.*, 1984].
Les systèmes de la troisième catégorie représentent les objets par des modèles 2D s'approchant le plus possible de la représentation image. Ce principe de modélisation est utilisé pour restreindre le coût de l'algorithme de reconnaissance mais, en contrepartie, la taille d'une telle modélisation croît exponentiellement. En effet, un objet, même simple, peut comporter quelques milliers de points de vues 2D [Ahuja *et al.*, 1992] et le problème de la complexité exponentielle n'est donc pas supprimé.

En synthèse nous concluons que seuls les deux premiers principes semblent actuellement raisonnables.

Chapitre 2 : Les techniques d'indexation

En considérant donc les deux premiers principes étudiés dans le chapitre précédent, le chapitre 2 présente un ensemble de techniques qui peuvent être utilisées pour indexer les modèles d'objets sous la forme de classifications hiérarchiques :

- Les techniques d'*analyse de données* sont des procédés de classification d'objets caractérisés par l'utilisation de vecteurs d'attributs purement numériques.
  Ces techniques se regroupent en deux familles distinctes. D'une part, les classifications hiérarchiques ascendantes (CHA) regroupent au fur et à mesure les objets les plus proches suivant une mesure de similarité. Le processus itère en considérant initialement chaque objet comme une classe. A partir de ces classes initiales, un arbre est construit où chaque niveau correspond à la fusion des classes les plus proches suivant un critère préétabli, jusqu'à n'obtenir qu'une seule classe pour l'ensemble des objets. Ainsi chaque niveau de l'arbre exprime une classification possible de l'ensemble des objets.
  D'autre part, les techniques de classification par partitions (CP), dont l'une des plus connues est fondée sur les *nuées dynamiques*, regroupant initialiement tous les objets dans une classe unique. L'objectif est inverse au précédent : partant de la classe représentant l'ensemble des objets, le processus cherche à former k classes suivant des critères préétablis de dissimilarité inter-classes et de similarité intra-classes. Le nombre k de classes à créer est fixé a priori. Les techniques de classification par partitions peuvent être utilisées pour créer aussi bien des classifications planes que des classifications hiérarchiques. Dans ce dernier cas, le processus décrit précédemment est réappliqué pour chacune des k classes jusqu'à obtenir un partitionnement où chaque classe ne représente qu'un seul objet.

  Pour les deux types de classifications, le principal problème est de pouvoir estimer le nombre de classes. De plus, lorsqu'un nouvel objet doit être pris en compte, il

n'existe pas de technique efficace qui réorganise la classification courante de telle façon que le résultat soit comparable à celui obtenu en repartitionnant l'ensemble [Vogel and Wong, 1979].

- Les *classifications conceptuelles* construisent des classifications d'ensembles d'objets décrits par des vecteurs d'attributs hétérogènes. Ces attributs sont :
  - *quantitatifs* : une mesure absolue, proportionnelle ou par intervalles,
  - *qualitatifs* : une mesure nominale ou ordinale,
  - *hiérarchique* : par classification de formes 2D par exemple (Figure 0.3).

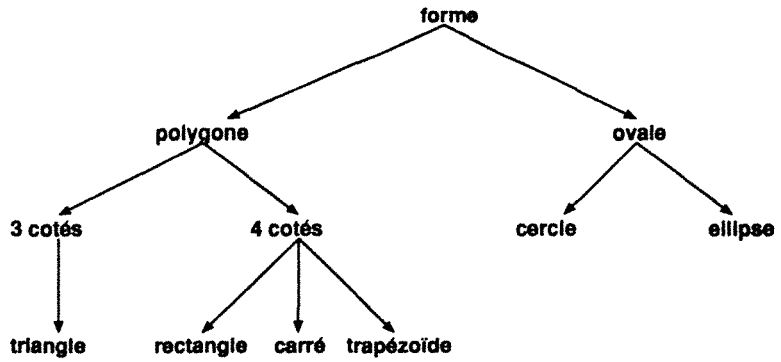


Figure 0.3. *Une mesure de similarité hiérarchique des formes.*

Le fonctionnement est globalement similaire à celui des techniques de classification par partitions, en particulier les *nuées dynamiques*. Partant d'un ensemble d'objets initialement considéré comme formant une seule classe et connaissant le nombre k de classes à créer, le processus forme les noyaux des classes recherchées et cherche itérativement à les stabiliser.

Les mêmes problèmes sont rencontrés : l'estimation du nombre de classes et le fonctionnement incrémentiel.

- Les *tables de hachage* décrivent les objets par une *clef* ou un jeu de clefs, suivant que l'on désire une représentation des objets plane ou hiérarchique. L'utilisation des tables de hachage est comparable à celle de l'index d'un livre. Les mots "clefs" du livre permettent de saisir les idées principales du livre.
  Par rapport aux précédentes techniques, cette méthode a l'avantage d'effectuer la réorganisation de la classification courante de façon incrémentielle en un temps constant [Clemens and Jacobs, 1991].

  Cependant, dans le cadre de la vision par ordinateur, les outils de traitement d'images ne sont pas suffisamment stables pour extraire fiablement les clefs (contours, régions, etc.).

- Les *graphes de décision*, que nous allons étudier, utilisent une représentation structurelle des objets. La structure d'un objet est moins sensible au bruit dans

les images que ne le sont des caractéristiques comme le type de surface des faces 3D, la texture de la surface ou encore la couleur d'une surface. Les modèles des objets sont alors des graphes relationnels, où les nœuds représentent les primitives et les arcs les relations entre ces primitives (les indices-modèle).
Un graphe de décision forme une classification hiérarchique structurelle des objets. Parmi les techniques que nous allons présenter, aucune n'est incrémentielle et il n'en existe pas à notre connaissance.

Nous allons consacrer à chacun de ces principes d'indexation une étude qui fera ressortir les inconvénients et les avantages de chacune dans le cadre de la reconnaissance en vision par ordinateur.

Chapitre 3 : L'indexation

Ce chapitre commence par une présentation succincte du système MAGIC pour situer l'index et le processus d'indexation.
Le fonctionnement du processus d'indexation consiste à construire des structures qui puissent être retrouvées dans les objets et qui forment des classes structurelles d'objets. Le processus d'indexation recherche les parties structurelles communes à plusieurs objets. Les objets étant décrits par des graphes relationnels, un processus de recherche d'isomorphismes de sous-graphes est utilisé pour retrouver ces parties communes. Nous présentons donc le processus de recherche d'isomorphisme de sous-graphe avant de décrire le processus d'indexation proprement dit. Le fonctionnement de l'indexation est ensuite expliqué en détails. L'algorithme est présenté, sa terminaison est prouvée et la complexité du graphe d'indexation est donnée dans le pire cas, dans le meilleur cas et en moyenne. Le chapitre se termine par une explication des expérimentations d'indexation.

Chapitre 4 : La reconnaissance

Le processus de reconnaissance utilise l'index présenté dans le chapitre précédent. Nous commençons par une présentation générale accompagnée d'un exemple pédagogique. Le principe est similaire à celui de l'indexation mais, au lieu d'enregistrer les classes structurelles, le processus de reconnaissance les utilise. Nous détaillons l'algorithme, dérivé de celui de l'indexation, puis nous testons son efficacité sur un ensemble de scènes.

Conclusion et perspectives

Nous concluons en analysant les différentes améliorations souhaitables pour que le processus de reconnaissance soit efficace : restreindre la taille de l'index, paralléliser les processus d'indexation et de reconnaissance, prendre en compte les erreurs lors de l'extraction des primitives et des indices-image. Nous présentons pour finir les futures directions de recherche que nous désirons prendre.

1

Les systèmes de reconnaissance d'objets

La compréhension automatique de scènes est un processus d'identification et de localisation des objets qui composent la scène observée. Nous regroupons *l'identification* et la *localisation* des objets sous le terme *reconnaissance* et les informations fournies au processus de reconnaissance sont, d'une part, la description par des primitives 2D/3D (points, courbes, surfaces) de la scène examinée et, d'autre part, les modèles des objets connus a priori.

Nous allons passer en revue un certain nombre de systèmes que nous jugeons représentatifs et, afin de les évaluer, nous définissons un certain nombre de critères [Zhao, 1991] [Suetens *et al.*, 1992] :

1. *La robutesse* : comment se dégradent les performances d'un système en présence de bruits ou d'occultations ?

   Le bruit dans les images est synonyme d'erreurs dans les mesures donc, a fortiori, dans les caractéristiques des *primitives* (points, courbes, surfaces, volumes) et des *indices* (relations entre les primitives). Ainsi ces erreurs, tout comme l'occultation, peuvent empêcher la détection de primitives ou d'indices utiles à la description de la scène. Aussi, l'un des objectifs d'un système de vision est d'obtenir une description des images qui soit la plus représentative possible des objets présents dans la scène observée.

   Certains systèmes ont des tâches, telle que la saisi d'objets posés en tas [Ayache and Faugeras, 1986] [Ikeuchi, 1987], qui les autorisent à ne pas tenir compte de ce critère, en particulier des phénomènes d'occultation. C'est une limitation très forte pour la reconnaissance car la généralisation de ces systèmes à d'autres tâches s'avère difficile.

2. *La généricité* : il s'agit, en fait, de savoir si les modèles d'objets proposés décrivent la géométrie, la couleurs, la texture, la fonction, etc. de chaque objet ou si, en plus de ces informations propres à chaque objet, ces modèles sont regroupés dans des classifications planes ou hiérarchiques suivant ces mêmes caractéristiques [Brooks, 1981] [Thirion, 1989] : géométrie, couleur, texture, fonction, etc.

Par exemple décrire un objet suivant sa fonction revient en fait à définir les liens existant entre cette fonction et les attributs physiques des primitives le décrivant [Stark and Bowyer, 1991] [Stark and Bowyer, 1992]. Par exemple, la classe des chaises représente un ensemble de modèles de chaises de géométries diverses, mais, de fonction identique (en général y poser son postérieur mais ce n'est nullement limitatif). Si une telle modélisation est réalisable, il est possible de concevoir un système de vision qui, lors de la compréhension d'une scène, est capable d'identifier une chaise par sa fonction sans en connaître la forme exacte.

3. *L'accessibilité aux modèles* : la construction de primitives image et d'indices image décrivant au mieux les objets des images n'est pas suffisant pour limiter la combinatoire lors de l'étape de sélection des modèles devant correspondre aux objets image. Cette combinatoire va en grandissant avec la taille de la base de connaissance contenant les modèles. Pour répondre à ce problème, des techniques d'*indexation* sont proposées pour partionner l'ensemble des modèles en classes de modèles identifiables dans l'image. C'est le thème de ce mémoire et à travers l'étude bibliographique proposée dans ce chapitre nous allons indiquer pourquoi ce critère est primordiale pour définir un système de vision. Déjà, intuivement, nous pouvons dire que le problème majeur d'un système de vision est de pouvoir mettre, le plus facilement possible, deux représentations légèrement distinctes des objets qui ne sont pas tout à fait identiques : d'une part, le système possède une représentation image qui comporte des bruits et des occultations, d'autre part, le modèle d'un objet est une description numériquement plus stable car non entâchée de bruit et ne présentant pas d'occultation par d'autres objets.

4. *Le matériel* : de la qualité du matériel dépend le temps mis pour la compréhension de scènes : plus les opérations sont activées en parallèle plus le système est efficace. Ce critère ne sera pas considéré par la suite car parmi les études présentées seuls les réseaux neuronaux proposent une structure adaptée au parallélisme [Fukushima, 1988] [Feldman, 1985].

En plus de ces quatre critères permettant de comparer les systèmes de reconnaissance entre eux, nous allons distinguer deux catégories de systèmes présentées en deux sections distincts :

- la section 1.1 décrit les techniques de *recherche des occurrences d'un modèle* dans une image en supposant que ce modèle a été sélectionné auparavant soit par l'utilisateur soit par la tâche qui est imposée au robot.

- la section 1.2 s'intéresse aux techniques de *perception des objets* d'une scène : c'est-à-dire aux techniques d'identification et de localisation des objets qui composent cette scène. Ici, la sélection des modèles qui décrivent la scène observée n'est pas effectuée en amont de la phase de reconnaissance mais fait partie de celle-ci.

Parmi les techniques de recherche des occurrences d'un modèle, la sous-section 1.1.5 présente la *décomposition d'une scène en objets image* qui symbolise la charnière

avec les techniques de perception des objets. Cette technique découle des premières études sur les systèmes de type perception des objets datant du début des années 1960. Cependant, leur objectif n'est pas de reconnaître les objets de la scène mais de fournir la meilleure représentation d'une image en terme d'objet image : c'est-à-dire regrouper l'information image, qui est souvent présentéee par des primitives et des relations entre ces primitives, de façon à ne décrire que les objets présents dans l'image. Cette technique a permis de mettre en évidence deux paradigmes importants :

- *une description intermédiaire* : la construction de primitives image *complexes* ou de groupes de primitives image pour représenter les différents objets qui composent la scène observée.
- *la composition* : la modélisation des objets par *composition* de modèles plus simples.

Après avoir présenté les techniques de recherche d'occurrences d'un modèle dans une image en section 1.1 et celles de perception des objets en 1.2, la section 1.3 énumère les différents avantages et inconvénients des deux catégories et explique l'intérêt de l'indexation pour la reconnaissance, ce qui introduit le chapitre suivant sur l'étude bibliographique des différentes méthodes d'indexation possibles.

1.1 La recherche des occurrences d'un modèle

Parmi l'ensemble des techniques de recherche d'occurrences, nous décrivons successivement le *template matching* tel qu'il est défini dans [Duda and Hart, 1973, pages 276–290] , la transformée de Hough, les réseaux de neurones et la décomposition en objets. Mis à part les réseaux de neurones, toutes ces techniques ne tiennent pas compte du critère d'accessibilité puisqu'elles considèrent le modèle à reconnaître comme étant sélectionné auparavant. Toutes ces techniques présentent principalement des modélisations 2D des objets sauf la technique de décomposition en objets image qui n'utilise pas de modèles des objets.

La technique du *template matching* reconnaît des formes 2D simples décrites par des matrices de pixels. Cette modélisation des formes est à la fois simple et rigide car l'appariement est quasi immédiat entre le modèle qui est une matrice de pixels et une zone de l'image qui est également une matrice de pixels. Les deux représentations sont identiques mais le moindre bruit ou la moindre occultation partielle de l'objet dans l'image entraîne des déformations difficiles à tenir compte lors de l'appariement. Ainsi les images ne doivent comporter qu'un taux très faible de bruit et d'occultation.

La transformée de Hough propose d'identifier et de localiser des formes paramétriques. Cette paramétrisation permet de reconnaître simultanément plusieurs classes de formes identiques telles que celle des cercles ou des ellipses. Mais, il semble que ce genre de modélisation soit difficile à rendre indépendant de l'utilisateur.

Les modèles élastiques sont des modèles munis de contraintes qui permettent de tolérer de fortes déformations dues aux bruits ou de ne pas connaître avec précision les dimensions de l'objet à reconnaître.

Les réseaux de neurones sont des systèmes apprenant seuls la forme à observer. Les modèles ne sont pas paramétriques, mais, la configuration multicouche permet de tenir compte de certaines déformations dues aux bruits et d'observer une homothétie des modèles.

La décomposition en objets propose de décrire l'image de la scène observée par des groupes de primitives image représentant au mieux les objets présents. Elle est l'étape intermédiaire entre les techniques de recherche des occurrences d'un modèle et celles de perception des objets.

1.1.1 La technique du *template matching*

Les techniques de template matching ont été conçues pour répondre à la question suivante [Duda and Hart, 1973, pages 276–290] : existe-t-il dans l'image de la scène observée, l'occurence d'une forme a priori sélectionnée ?

Les modèles des formes sont des *matrices caractéristiques*. Par exemple, le modèle donné d'un triangle est la matrice caractéristique illustrée en figure 1.1.

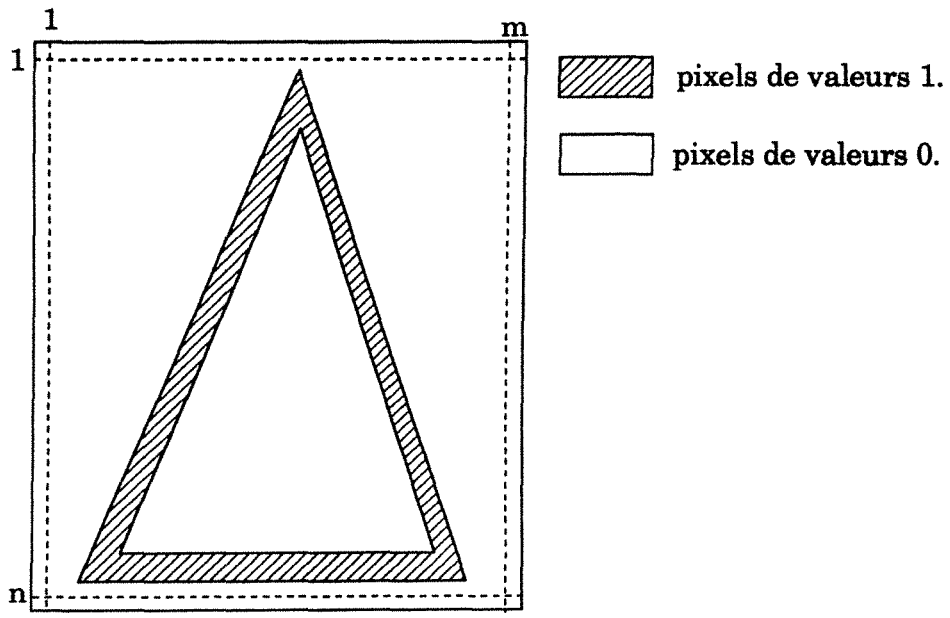


Figure 1.1. *La matrice caractéristique* $(n \times m)$ *d'un triangle : la zone inscrite 1 représente la forme [Duda and Hart, 1973].*

L'objectif est donc de détecter la présence ou l'absence d'une matrice caractéristique donnée dans une image. Ce principe est utilisé aussi bien avec des contours qu'avec des régions. Il existe différents critères d'appariement entre une matrice caractéristique et une zone image : les valeurs absolues des différences entre pixels, les carrés des différences entre pixels ou les mesures probabilistes qui définissent en chaque pixel de l'image la probalité de représenter un pixel de la matrice caractéristique du modèle à apparier. L'appariement se fait par la minimisation de la distance entre la matrice et le représentant potentiel de l'image observée.

Evaluation

Le principe est peu robuste puisque la modélisation des objets par des matrices caractéristiques ne tolère, lors de la phase de reconnaissance, que peu de bruit dans l'image ainsi qu'un faible taux d'occultation.

La généricité est nulle car la modélisation des objets se fait au niveau pixel. L'objectif est de rechercher dans une scène une forme donnée, suffisamment simple pour être modélisée par une matrice de pixels.

Compte tenue de ces limitations, il semble difficile de généraliser ce principe à d'autres tâches. De plus la modélisation par des matrices caractéristiques ne permet pas de reconnaître des formes génériques un cercle de rayon variable. Les techniques de template matching permettent de reconnaître des formes qui peuvent être décrites par des matrices de pixels. Mais, nombre de formes complexes ou plus génériques ne peuvent être simplement représentées par des matrices de pixels.

1.1.2 La transformée de Hough

Parmi les techniques qui prennent en compte des modèles plus évolués, la transformée de Hough recherche une forme définie par une équation paramétrique. Elle est donc utilisée quand la forme n'est pas localisée dans l'image, mais, qu'elle peut-être décrite par un vecteur de paramètres. Cette technique est relativement peu sensible aux différentes perturbations apportées par les distorsions dues aux bruits dans l'image et par les ruptures de la forme dans l'image dues aux occultations [Ballard and Brown, 1982, pages 123–131].

Duda et Hart [Duda and Hart, 1973, page 273] ont été les premiers à utiliser cette technique pour détecter des droites. Prenons, par exemple, l'équation d'une droite: $y = Ax + B$, représentant au sens des moindres carrés un ensemble de n points (Figure 1.2.a). Alors les paramètres A et B de cette droite peuvent être calculés à partir des

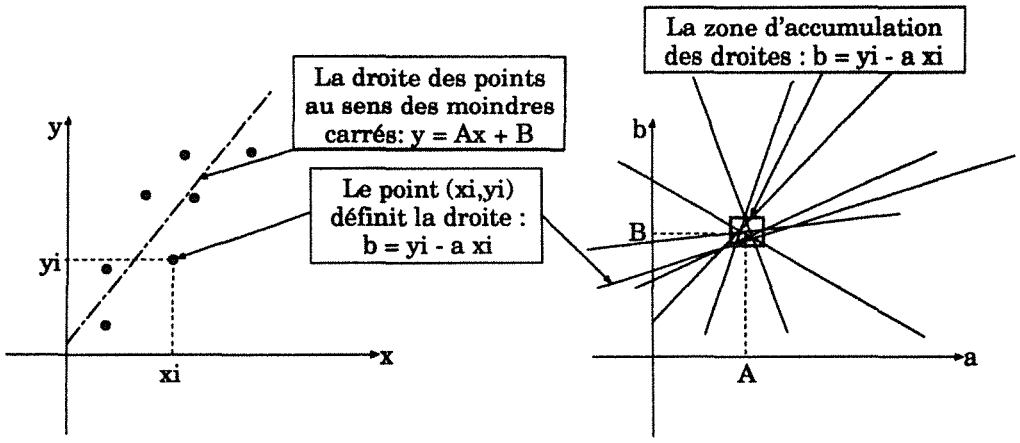
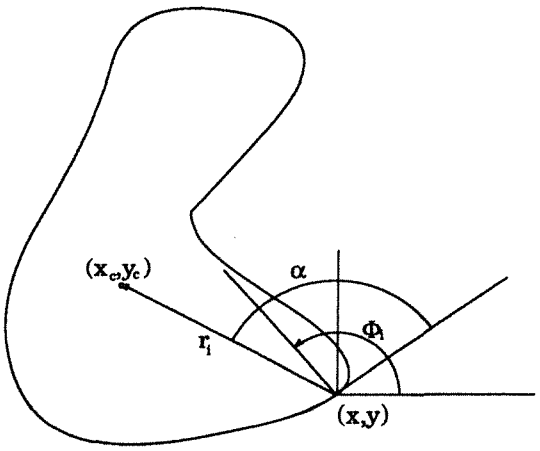


Figure 1.2. *(a) Les n points forment une droite; (b) chaque point définit une droite dans l'espace de dimension a et b.*

coordonnées des points de l'image: les coordonnées (x_i, y_i) de chaque point définissent

une droite dans l'espace des paramètres (a, b) : $b = y_i - ax_i$. L'ensemble des n droites $b = y_i - ax_i$, obtenues pour tous les points (x_i, y_i), définit une zone d'accumulation illustrée par la figure 1.2.b qui représente les valeurs A et B des paramètres de la droite, au sens des moindres carrés, de l'ensemble des n points.

Cette technique a été généralisée à des formes plus complexes mais toujours paramétriques telles que celle présentée en figure 1.3. On parle alors de *transformée de*



Inclinaison par rapport à l'horizontale	ensemble des couples $k = (r_i\ \alpha_i)$	
Φ_1	k_1^1	$k_1^{n_1}$
Φ_2	k_2^1	$k_2^{n_2}$
Φ_m	k_m^1	$k_m^{n_m}$

Figure 1.3. *Une forme identifiable par la technique de la Transformée de Hough Généralisée.*

Hough généralisée. La forme est initialement décrite par une table indéxée par le gradient angulaire Φ_i de la normale au contour. Cette table indique, pour chaque Φ_i, l'ensemble des couples (r_i, α_i) qui définissent le même point central (x_c, y_c). Ensuite, lors de la localisation de cette forme dans une image donnée, cette table est utilisée comme suit :

- pour chaque point (x_i, y_i) de l'image observée, on calcule le gradient angulaire Φ_i de la normale au contour.
- à l'aide de la table de description de la forme, ce gradient Φ_i permet de calculer l'ensemble des centres (x_c, y_c) possibles :

$$\begin{pmatrix} x_c \\ y_c \end{pmatrix} = \begin{pmatrix} x_i \\ y_i \end{pmatrix} + r\Phi_i \begin{pmatrix} \cos(\alpha_i^j(\Phi_i)) \\ \sin(\alpha_i^j(\Phi_i)) \end{pmatrix}$$

enfin, tous ces calculs de centres devant converger vers un point d'accumulation si la forme existe dans l'image observée, la transformé de hough recherche les points d'accumulation dans l'espace (x_c, y_c).

Evaluation

La robustesse est pris en compte grâce à la recherche des points d'accumulations dans l'espace des paramètres permettant de tolérer le bruit dans les images : une zone d'accumulation définit une zone d'erreurs autorisée, et le fait d'avoir une représentation paramétique permet de localiser une forme en en n'observant qu'une partie.

La généricité est limitée car, bien que les modèles représentent des classes de formes définies par des équations paramétriques telles que la classe des cercles ou des ellipses, cette modélisation ne permet pas de regrouper les modèles d'objets. Pour ce type de généricité des caractéristiques telles que la fonction des objets ou la structure géométrique des objets sont utilisées.

La transformée de Hough et sa généralisée identifient et localisent des formes dont les modèles géométriques sont purement paramétriques.

1.1.3 Les modèles élastiques

Les modèles élastiques, autrement appelés *snakes*, sont des descriptions de contours déformables suivants certaines contraintes [Suetens *et al.*, 1992]. Par exemple, Fua [Suetens *et al.*, 1992] opère la reconnaissance de buildings dans des images aériennes en utilisant un modèle élastique dont les contraintes expriment que la forme des buildings doit être rectangulaire avec une région interne d'intensité uniforme. Ici, le modèle n'a pas de dimensions exactes mais des contraintes sur sa forme et sa région interne. La reconnaissance d'un building se fait en initialisant le modèle élastique à l'intérieur du rectangle supposé être l'image du building. Puis l'élastique s'expanse jusqu'au contour du rectangle image en verifiant que la région contenue est bien uniforme en intensité. En fait, un modèle élastique représente les caractéristiques de régularisation et photométrique du modèle à optimiser. Nous retrouvons là la notion de suivi de l'information au niveau pixel utilisée par les techniques de *template matching* et par la transformée de Hough mais avec des contraintes d'appariements globales qui permettent de prendre en compte le bruit.

Evaluation

Le critère de robustesse est pris en compte car l'utilisation de contraintes globales d'appariement permet d'envisager des bruits tels que la perte de certains pixels dans l'image de la forme ou la présence de pixels parasites. Mais cette souplesse ne va pas sans d'autres contraintes.

- Préalablement, les modèles doivent être grossièrement localisé sur l'image. Cette localisation grossière présuppose elle-même une identification des zones image susceptibles de contenir des occurrences du modèle recherché.
- Les modèles élastiques sont obligés d'utiliser des algorithmes de minimisation de fonctionnelles qui sont assez coûteux en temps de calcul.

La généricité est faible puisque les modèles élastiques sont utilisés pour reconnaître des classes d'objets paramétrés. Par exemple, la classe des buildings telle que l'a définie Fua [Suetens *et al.*, 1992].

1.1.4 Les réseaux neuronaux

Contrairement aux techniques proposées jusqu'ici, les réseaux de neuronnes prennent en compte le problème de sélection des modèles à apparier. Nous allons décrire l'un des réseaux les plus connus, développé par Fukushima [Fukushima, 1988]. Le réseau qu'il nous propose a subi trois évolutions majeures. Dans la première version ce réseau est un réseau multicouche tel que la première couche représente les pixels de l'image digitalisée et les couches suivantes regroupent au fur et à mesure ces pixels jusqu'à décrire un modèle au niveau de la dernière couche. Ces regroupements sont définis au sein du réseau par un *apprentissage non supervisé*[1] des formes à identifier. Ainsi, chaque forme est décrite au sein du réseau comme un arbre dont la racine (représentant la forme apprise) est en dernière couche et dont les feuilles (représentant les pixels) sont en première couche. Ce réseau est capable d'identifier une image ne contenant qu'une seule forme qui ne soit pas déformée par rapport à son modèle. En deuxième version, Fukushima divise les couches du réseau en deux de façon à pouvoir tenir compte des déformations des formes à identifier. Enfin, la dernière version adapte le réseau de façon à identifier des images contenant plusieurs motifs.

Apprentissage non supervisé

Parmi les réseaux de neurones présentant des techniques de reconnaissance de motifs par *apprentissage non supervisé*, Fukushima [Fukushima, 1988] présente une application à la vision artificielle où ses résultats sont une référence. Le premier système qu'il a développé est le COGNITRON. C'est un réseau *multicouche*, c'est-à-dire un réseau constitué de plusieurs couches successives. Une *couche* est un ensemble de *cellules* et le lien d'une couche à l'autre se fait par un ensemble de *synapses* qui chacun connecte un groupe de cellules $(C_i, \ldots, C_j)$, de la couche n, à une cellule C_k de la couche $n+1$.

Les cellules du COGNITRON sont de deux sortes : les cellules *excitatrices* et les cellules *inhibitrices*. Les cellules excitatrices $(C_i, \ldots, C_j)$ de la cellule C_k sont connectées à une cellule inhibitrice C_h de la couche n, comme le montre la figure 1.4 : les synapses $(a_i, \ldots, a_j)$ sont les pondérations de l'influence des cellules excitatrices $(C_i, \ldots, C_j)$ sur la cellule C_k ; Les synapses $(b_i, \ldots, b_j)$ sont les pondérations de l'influence des cellules excitatrices $(C_i, \ldots, C_j)$ sur la cellule inhibitrice C_h. Le synapse h est la pondération de l'influence de la cellule inhibitrice C_h sur la cellule C_k. L'activation de la cellule C_k est assurée par la fonction :

$$w = \begin{cases} \frac{1+e}{1+h} - 1 & si \quad \frac{1+e}{1+h} - 1 \geq 0 \\ 0 & si \quad \frac{1+e}{1+h} - 1 < 0 \end{cases}$$

1. c'est-à-dire autonome vis à vis de l'utilisateur

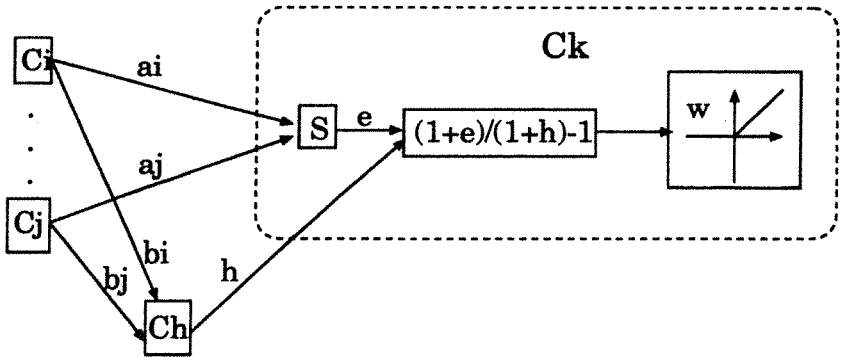


Figure 1.4. *Le fonctionnement de la cellule C_k.*

où e est la somme des influences des cellules $(C_i, \ldots, C_j)$ et h est l'influence de la cellule inhibitrice C_h. La cellule inhibitrice C_h sert à pondérer l'influence du groupe $(C_i, \ldots, C_j)$ sur la cellule C_k par rapport à l'influence du groupe sur d'autres cellules de la couche $n + 1$ (Figure 1.5).

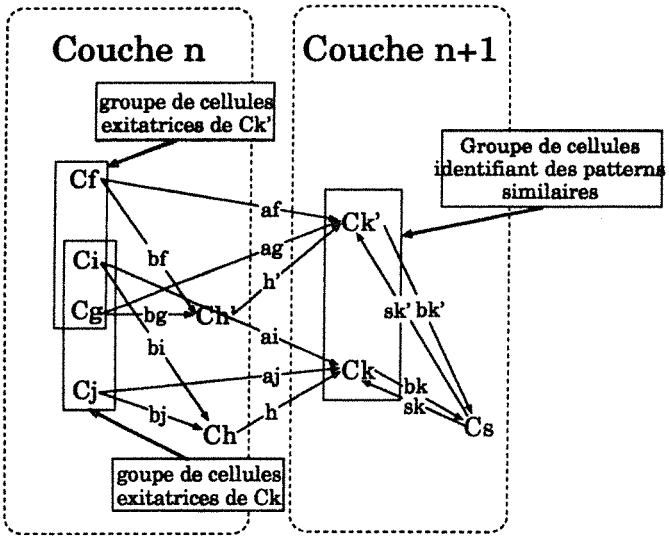


Figure 1.5. Les liens exitateurs et hinibiteurs entre deux couches du COGNITRON.

De plus, comme les cellules $C_{k'}$ voisines de C_k, au niveau de la couche $n + 1$, sont susceptibles d'être activées par des motifs similaires, une cellule inhibitrice C_s est associée à la cellule C_k et à ses voisines $C_{k'}$ pour permettre de sélectionner la plus représentative (figure 1.5).

La faiblesse du COGNITRON est le manque de robustesse au bruit (déformation du motif à analyser par rapport au modèle ayant été appris par le réseau) et ses variations (changemement d'échelle, de forme, etc.).

Prendre en compte le bruit

Fukushima présente alors le NEOCOGNITRON qui reprend le principe du COGNITRON tout en l'améliorant au vu de ses faiblesses. Le NEOCOGNITRON divise les couches en deux. Pour chaque couche, la sous-couche des *cellules simples* acquiert l'information des cellules complexes de la couche précédente tandis que la sous-couche des *cellules complexes* généralisent le motif observée par les cellules simples de la même couche. Les cellules complexes permettent de reconnaître des formes avec des variations pour la localisation et de tenir compte des déformations. La figure 1.6 illustre le fonctionnement du NEOCOGNITRON. La première couche a une seule sous-couche

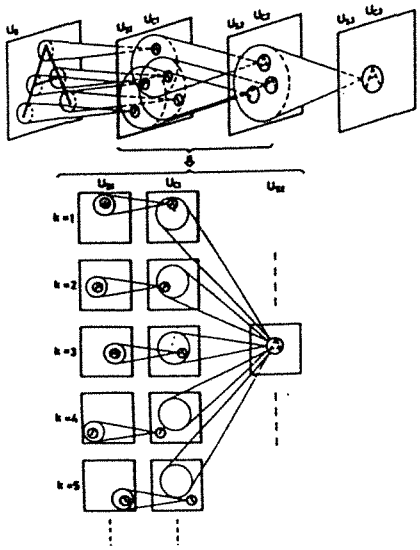


Figure 1.6. *Le schéma de fonctionnement du* NEOCOGNITRON *[Fukushima, 1988].*

réceptrice U_0. La couche 1 est composé des sous-couches simple U_{s1} et complexe U_{c1}. Les cellules de U_{s1} qui représentent la même forme dans différentes positions, sont rassemblées en une seule cellule de U_{c1}. Ainsi plusieurs modèles légèrement déformés les uns par rapport aux autres sont enregistrés dans le réseau lors de l'apprentissage.

Construire des centres d'intérêts

Puis Fukushima s'est intéressé à la reconnaissance d'images contenant plusieurs formes. Le système se focalise sur une forme, l'identifie, passe à la suivante et ainsi de suite. Le NEOCOGNITRON est transformé de manière à permettre la focalisation sur une forme et la reconnaissance de celle-ci. A l'ensemble des couches précédemment expliquées est ajouté des couches dont la direction des synapses est inverse. Ce deuxième réseau, illustré par la figure 1.7, permet à partir d'une hypothèse de forme émise en dernière couche de présenter en première couche l'image supposée de cette forme, ceci afin de la comparer avec celle en cours d'analyse. Cette segmentation virtuelle est le processus inverse de la reconnaissance de la forme analysée. Ainsi cette prédiction permet de gérer les seuils de détection quand une partie de la forme n'est pas correctement perçue lors de la reconnaissance. Elle permet, également, la focalisation lors

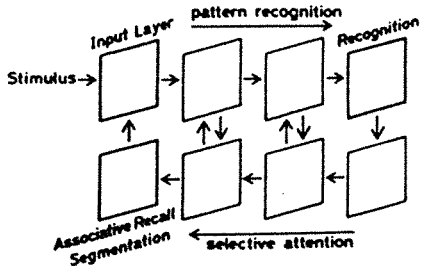


Figure 1.7. *Le système par focalisation [Fukushima, 1988].*

de la reconnaissance en se concentrant sur l'information qui est suppose représenter la forme en cours d'analyse.

Evaluation

La généricité est prise en compte par la structure multicouche du réseau. Les réseaux de neurones s'intéressent à des modèles dont la généricité revient à former des classes de motifs partageant une caractéristique image commune. L'ensemble des arbres créés par l'apprentissage des formes apprises forme une forêt où certains noeuds sont partagés par plusieurs arbres et donc représentent une information commune à plusieurs formes.

Mais la robustesse est faible car l'occultation n'est pas considérée puisqu'il s'agit de reconnaître des caractères dans une image. Outre cela, Il est à noter que les données image sont les pixels et ceci implique une taille de réseau tout à fait importante.

Par contre, les réseaux définissent le besoin d'un processus apte à former des groupes perceptuels image susceptibles de provenir d'un même objet. Ils présentent un principe attractif: *la construction d'une description intermédiaire guidée par les modèles.*

1.1.5 La décomposition en objets image

L'ensemble des techniques de recherche des occurrences d'un modèle que nous venons de voir proposent principalement des appariements de formes 2D avec des images 2D. Mais nous nous intéressons également aux techniques d'appariement entre des modèles 3D d'objets et des images 2D. La *décomposition en objets* est la description d'images 2D en termes d'objets 3D. Dans ce cadre, les occultations sont fréquentes et donc la séparation des primitives image suivant les objets auxquels elles appartiennent, est difficile. Cette partie est inspirée de l'article de Mackworth [Mackworth, 1977a].

Le contexte historique

Les études dans ce domaine ont été introduites par les travaux de Roberts (Roberts 1965[2]) [Waltz, 1975] [Minsky, 1975] [Mackworth, 1977a]. Il présente un système de type *perception des objets*. Le principe général est la reconnaissance d'objets 3D polyédriques dans des images 2D à partir d'un ensemble de trois modèles: un cube, un coin et un prisme hexagonal (Figure 1.8).

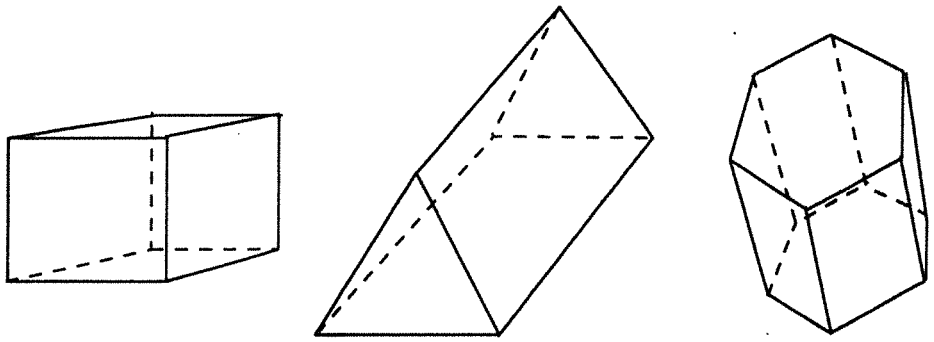


Figure 1.8. *Les modèles du système de Roberts [Mackworth, 1977a].*

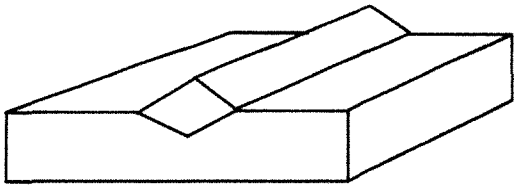
Les objets des scènes observées sont décrits par composition de ces trois modèles. Falk propose une extension du principe à une base de neuf modèles ainsi que la gestion du bruit dans les images (Falk 1972)[3].

La représentation des objets par composition de modèles de base et la recherche dans les images de groupes de primitives évoluées provenant d'un même objet sont les deux critères utilisés par Roberts et Falk. Nous les retrouverons dans l'ensemble des systèmes de type *perception des objets* [Brooks, 1981] [Lowe, 1985b]. Mais, les modèles utilisés ne sont pas toujours les meilleurs éléments pour une décomposition simple et claire. Par exemple, comment décomposer l'objet de la figure 1.9? Faut-il utiliser le prisme triangulaire ou le parallélogramme comme modèle de base?

Falk définit lui-même les limites de son système INTERPRET: l'identification demande que l'information 3D soit fournie et, en même temps, l'information 3D est extraite par identification des objets. Notons que le principe de *regroupement perceptuel* présenté par Lowe [Lowe, 1985a, chapitre 4], présenté parmi les systèmes de perception des objets en Section 1.2.2, provient également de l'étude de ce problème. Ayant défini le principal problème à l'évolution de tels systèmes: le regroupement des informations image en objets image, les chercheurs ont proposé différentes solutions.

2. L.G. Roberts, *Machine Perception of Three Dimensional Solids.*, Optical and Electro-Optical Information Processing, MIT Press, 1965

3. G. Falk, *Interpretation of Imperfect Line Data as a Three-dimentional Scene.*, Artificial Intelligence, Volume 3, Numéro 2, pages 101–144, 1972



Comment décomposer cette objet ?

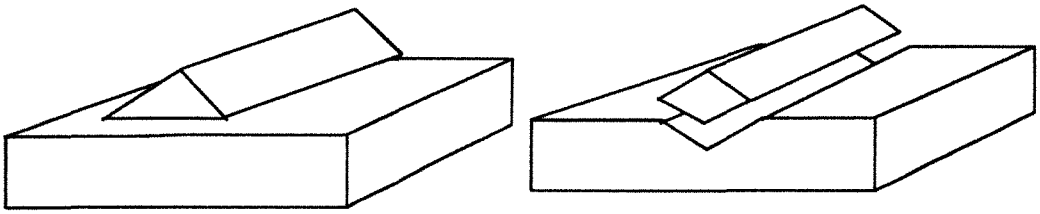


Figure 1.9. *Comment connaître a priori les modèles de base qui seront utiles ? [Mackworth, 1977a].*

Regrouper les régions provenant d'un même objet

Peu après les travaux de Roberts en 1965, Guzman (1968)[4] propose, pour retrouver les objets des images, la construction d'un catalogue, qui est illustré en figure 1.10, des

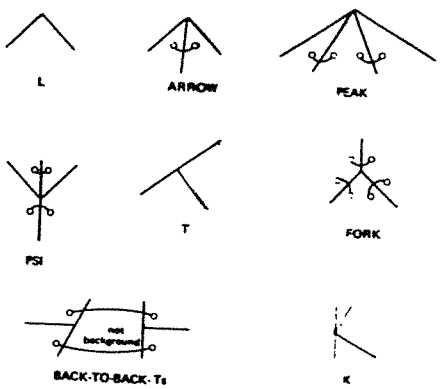


Figure 1.10. *Le catalogue des jonctions utilisé par Guzman [Mackworth, 1977a].*

différentes jonctions de segments qu'une image peut présenter où les arcs qui relient deux régions indiquent que celles-ci proviennent d'un même objet. Le système SEE recherche une description de l'image se rapprochant le plus possible des objets à identifier en vérifiant la cohérence des étiquetages des régions faits à partir du catalogue.

4. A. Guzman, *Decomposition of a Visual Sceneinto Three-dimentional Bodies.* AFIPS Proc. Fall Joint Comp. Conf., Volume 33, pages 291–304, 1968.

Sa faiblesse provient de ce que l'étude de la cohérence est locale aux jonctions alors que l'interprétation est globale. Par exemple, la figure 1.11 peut-être interprétée

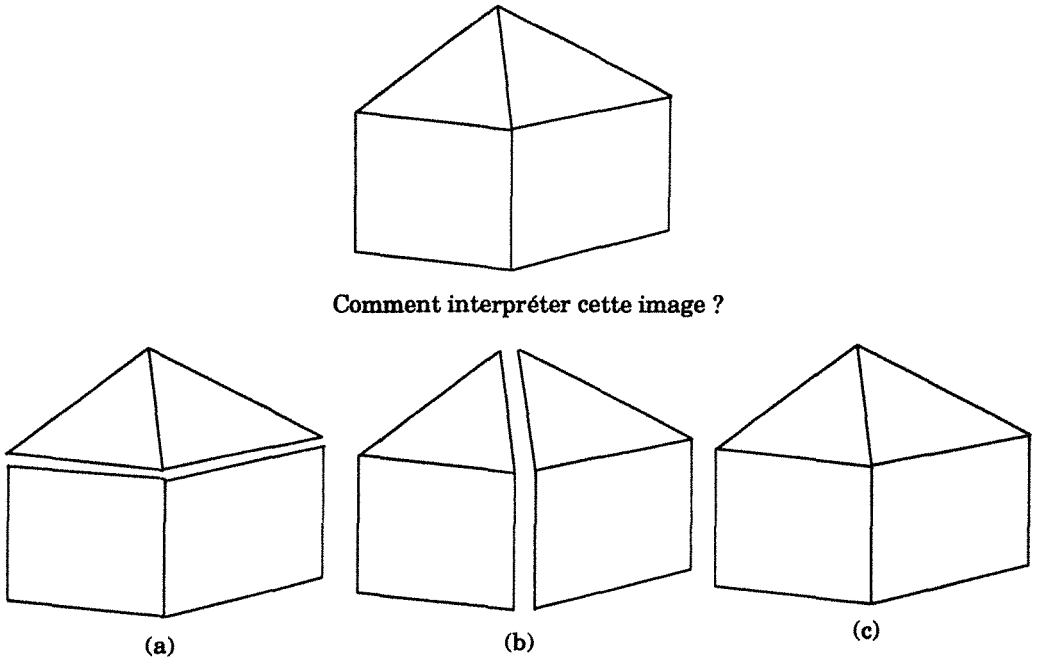


Figure 1.11. *Un exemple de limite du système* SEE *[Mackworth, 1977a].*

de plusieurs manières : **(a)** une maison qui forme un tout (le même objet), **(b)** une pyramide reposant sur une brique ou encore **(c)** deux pans d'objets différents accolés l'un à l'autre. SEE ne peut proposer, par le principe même de sa méthode de vérification de cohérence, que la première solution. Malgré cette faiblesse, Falk l'utilise afin de gérer le bruit dans les images car les bords des régions indiquent que deux segments appartiennent à un même contour, même si ceux-ci ne sont pas connexes.

Interpréter les segments image

Clowes[5] et Huffman[6] forment, à leur tour, un catalogue des jonctions fondé sur la signification des arêtes plutôt que sur celle des régions pour déterminer les différents objets de l'image (Figure 1.12) :

- *une arête est étiquetée* — si les deux régions qui la bordent forment une connexion concave et proviennent d'un même objet.
- Si la connexion entre les deux régions est convexe l'étiquette de *l'arête est étiquetée* +.

5. M. B. Clowes, *On Seeing Things.* Artificial Intelligence, Volume 2, Number 1, pages 79–112, 1971
6. D. A. Huffman, *Impossible Objects as nonsense sentences.* Machine Intelligence, Volume 6, B. Meltzer and D. Michie Editors, Edinburgh University Press, Edinburgh, pages 295–323, 1971

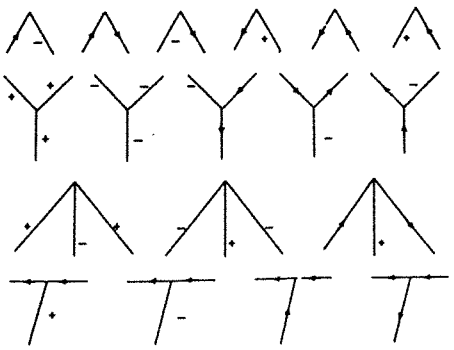


Figure 1.12. *Le catalogue utilisé par Clowes et Huffman [Mackworth, 1977a].*

- Si ce n'est ni une connexion concave ni une connexion convexe, les deux régions ne proviennent pas du même objet et l'une d'elles occulte l'autre. Dans ce cas, itl'étiquette → est affectée à l'arête.

Par exemple si l'on considère les deux premières jonctions en Y à gauche, la première représente un sommet convexe, comme celui d'un cube par exemple, alors que la deuxième représente un sommet concave, comme celui d'un coin d'une pièce par exemple.

La cohérence de la recherche des objets dans une image est vérifiée par similitude de l'étiquetage des deux extrémités de chaque segment de l'image. Clowes propose un processus d'étiquetage par une recherche en largeur d'abord tandis que Huffman propose en profondeur d'abord.

Les faiblesses de ce principe sont le coût algorithmique dû aux techniques de recherche en profondeur ou en largeur d'abord et la signification de l'étiquette de concavité. Cette dernière représente des relations ambiguë entre les faces : les faces peuvent tout aussi bien provenir du même objet que de plusieurs objets différents. Dans la figure (1.13), les deux étiquettes − sont fausses car elles lient le cube avec le fond de

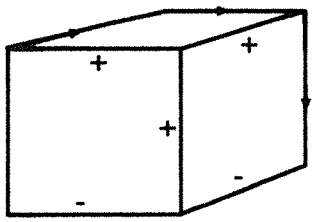


Figure 1.13. *Un exemple de mauvaise décomposition [Mackworth, 1977a] : le cube n'est pas séparé du fond de l'image.*

l'image. En fait, la relation de concavité ne fait pas la distinction entre le cas où un objet repose sur un autre et celui où un objet possède une arrête de concavité.

Comment éliminer ces faiblesses ?

Waltz [Waltz, 1975] propose alors un système qui améliore celui de Clowes/Huffman. Il ajoute à l'ensemble des étiquettes de Clowes/Huffman les *brisures* et les ombrages. Une brisure représente le lien entre deux faces coplanaires. L'ensemble des etiquettes est illustré par le réseau de la figure 1.14. Le catalogue passe de dix huit à cinquante

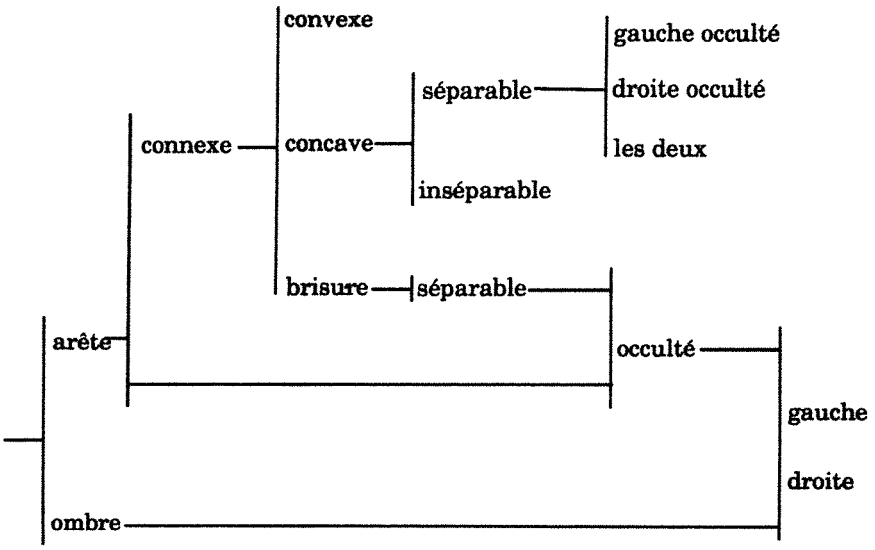


Figure 1.14. *L'étiquetage que Waltz propose [Mackworth, 1977b].*

trois jonctions possibles. De plus, Waltz propose un filtrage des étiquetages incohérents de manière à accélérer la phase d'interprétation. Waltz démontre la qualité et l'efficacité de son système par des essais tout à fait spectaculaires (Figure 1.15).

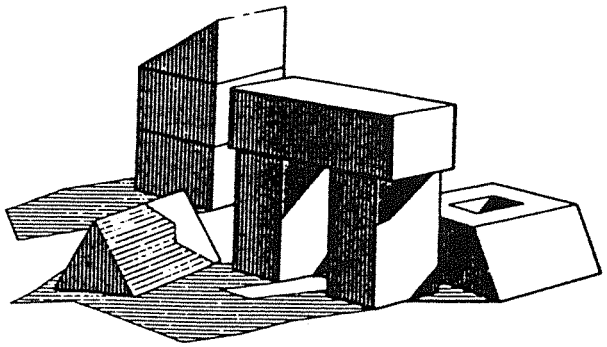


Figure 1.15. *Une image que Waltz sait reconnaître [Waltz, 1975].*

Une formalisation complète du procédé de décomposition en objets

Enfin, le système POLY de Mackworth [Mackworth, 1973] formalise complètement le procédé en modélisant les objets dans l'*espace dual* associé.
Dans un espace dual, les points deviennent les plans et les plans les points : l'équation

$a_x x + a_y y + a_z z + 1 = 0$, où les points de coordonnées $(xyz)^t$, sont les variables et les plans de coordonnées $(a_x a_y a_z)^t$ sont les constantes, est transformée en une équation $xa_x + ya_y + za_z + 1 = 0$ où les plans sont les variables et les points les constantes. De manière plus illustrée, le plan P de coordonnées $(a_x a_y a_z)^t$ a pour représentant dans l'espace dual associé le *dual* I de coordonnées $(a_x, a_y, a_z)^t$. Utilisant cette représentation, Mackworth définit un plan projectif G en $z = 1$ (Figure 1.16).

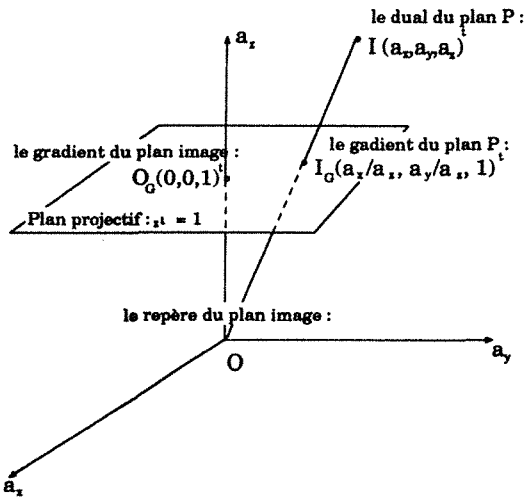


Figure 1.16. *L'espace dual et le plan projectif défini dans celui-ci [Mackworth, 1973].*

Pour chaque dual I d'un plan P est définit un gradient I_G de coordonnées $(\frac{a_x}{a_z}, \frac{a_y}{a_z}, 1)$ de telle manière que l'inclinaison du plan P par rapport au plan image est indiquée par le vecteur $\overrightarrow{O_G I_G}$ où le point O_G est le gradient du plan image. Si de plus cette représentation est faite dans le même repère que celui associé à l'image et que le plan image est parallèle au plan $Oa_x a_y$ en $z = 1$ alors un segment de l'image bordé par deux régions provenant d'un même objet, est de direction perpendiculaire à celle définie par les gradients des deux régions. Cette propriété sert à Mackworth pour regrouper les régions qui proviennent d'un même objet. Prenons par exemple, l'image de la pyramide de la figure 1.17.a. La figure 1.17.b illustre les relations entre les régions, comme par exemple la connexité convexe entre les régions B et C, et sont obtenues en observant le plan projectif dual associé à la pyramide tel qu'il soit confondu avec le plan image de la pyramide.
Les régions A, B et C ont respectivement pour gradients G_A, G_B et G_C tandis que les arêtes $1 \cdots 5$ sont respectivement représentée dans le plan projectif dual par les arêtes $1' \cdots 5'$. Par exemple, la droite $1'$ représente la connexité entre les régions A et B et est perpendiculaire au segment image 1. Il est à noter que la représentation duale introduit, pour cet exemple, un gradient fictif G_D qui est à associé en fait à la région A et donc l'arête $6'$ entre G_A et G_D est également fictive.
Dans cet exemple, le système POLY indique que les segments $1'$, $3'$ et $4'$ sont compatibles entre eux, c'est-à-dire qu'il est possible de positionner les gradients G_A, G_B et G_C tel que $1'$ soit perpendiculaire à 1, $3'$ à 3 et $4'$ à 4. Ainsi les segments 1 et 4 sont identifiés comme des connexions concaves de deux régions et le segment 3 comme une

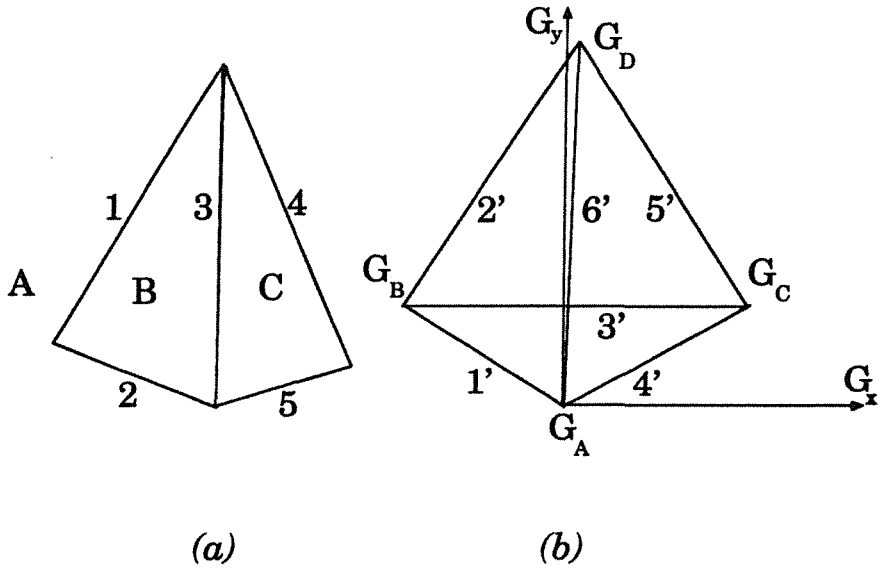


Figure 1.17. *(a) une image d'une pyramide. (b) la représentation dans le projeté de l'espace dual G des connexions entre deux régions provenant d'un même objet.*

connexion convexe. Puis, le système en déduit que les segments 2 et 5 expriment des occultations.

Evaluation

Le critère de robutesse semble être oublié puisque l'on suppose avoir des segmentations des images en contours et en régions qui soient peu bruitées.

De plus, ni le critère d'accessibilité ni le critère de généricité ne sont envisagés par ces techniques puisque l'on suppose possible d'utiliser un nombre restreint de modèles de base pour décrire des objets plus complexes.

Toutefois n'oublions pas les deux principes qui en découlent et que nous retrouverons tout au long de ce chapitre: le regroupement des primitives en une description intermédiaire et la composition de modèles simples pour représenter les objets complexes, même si cette dernière, telle qu'elle a été présentée, semble parfois inadaptée.

1.1.6 Synthèse

Les techniques de *template matching* nécessitent des images avec un faible taux de bruits et peu d'occultations. Les formes identifiables sont des plus simples puisqu'elles sont représentées par des matrices de pixels. L'appariement consiste en un étiquetage des pixels image par des pixels modèles dont la cohérence est contrôlée par une minimisation d'une distance entre la forme et l'image. La robustesse et la généricité de ces techniques sont donc faibles.

La transformée de Hough est une technique d'appariement de formes paramétrées. Ces formes peuvent être identifiées malgré certaines occultations ou certains bruits dans l'image. La Transformée de Hough tient donc compte du critère de robustesse. Ces modèles, de généricité faible, sont plus évolués que ceux des techniques de *template matching*.

Les modèles élastiques sont tout à fait adaptés à certaines scènes [Suetens *et al.*, 1992] où le système doit reconnaitre une forme donnée dont certains de ses paramètres sont lâches. La robustesse des modèles élastiques est due à leur capacité d'éviter les déformations dues aux bruits, mais, les occultations perturbent fortement leur comportement. La généricité des modèles élastiques est liée au fait qu'il peuvent définir des formes de façon plus ou moins précise. Par exemple la classe des buildings utilisée par le système développé ar Fua [Suetens *et al.*, 1992].

Les réseaux de neurones sont fondés sur des techniques proches de la transformée de Hough : les pixels procèdent par des votes pour se regrouper en primitives de plus en plus évoluées jusqu'aux modèles. Ils présentent une structure qui a l'avantage d'être parallélisable et ils cherchent à prendre en compte le critère de robustesse puisque les motifs à identifier sont modélisés avec certaines déformations possibles. La généricité est assurée par la structuration de chaque couche en deux sous-couches permettant de généraliser les motifs observés. Mais si le système observe un motif non modélisé, il indiquera alors le motif qui lui est le plus proche au lieu d'indiquer le nœud d'une des couches intermédiaires qui le décrit le mieux. Contrairement aux techniques précédentes qui omettent de répondre au problème de sélection des modèles susceptibles d'être présents dans l'image, l'accessibilité aux modèles est également due à la structure multicouche. La classification hiérarchique que forme le réseau est proche d'un graphe de décision dont les noeuds seraient les classes de modèles.

La décomposition en objets étudie comment représenter une image quand on désire y identifier les objets présents. Pour ce faire, deux paradigmes sont avancés :

- *La description intermédiaire* : c'est la formation des groupes de primitives image susceptibles de provenir d'un même objet.
- *La composition* : les objets complexes sont décrits par des compostions d'objets plus simples.

Les études sur la décomposition en objets cherchent à définir la meilleure représentation d'une image en termes d'objets image de façon à faciliter sa reconnaissance. Ainsi seul le besoin d'une description intermédiaire est réellement considéré à défaut de ceux de robustesse et de généricité.

Pour la reconnaissance d'une scène fort encombrée d'objets multiples et variés, un système de vision doit présenter une description identique ou quasi-identique aux modèles des objets et aux objets image de la scène de manière à faciliter les appariements. Pour la plupart, les techniques de recherche des occurrences d'un modèle ne répondent pas à ce critère d'accessibilté. Pourtant ce critère est nullement négligeable et une autre classe de systèmes se proposent d'y répondre : les techniques de perception des objets.

1.2 La perception des objets

Les systèmes décrits par la suite possèdent des modèles d'objets plus ou moins hétérogènes : texture, couleur, forme 2D/3D. Nous désignons par *primitives* les éléments de base utilisés pour constituer les modèles et pour former la description de la scène observée. Le principe général de la reconnaissance correspond à un processus en deux phases : *Prédiction* et *Vérification*[7] La prédiction forme les hypothèses d'appariement entre les groupes de primitives image et les modèles. La vérification observe que les appariements conjecturés par la phase de prédiction sont des interprétations correctes de l'image.

Nous distinguons par la suite trois grandes familles de systèmes dont les représentants les plus connus sont ACRONYM, RAF[8] et le système d'Ikeuchi.
ACRONYM propose une modélisation générique. L'ensemble des modèles des objets est partitionné hiérarchiquement pour former le *graphe de restriction*. Ce dernier permet de construire le *graphe de prédiction* qui correspond aux connaissances que peut fournir le graphe de restriction sur la position de la caméra et des objets dans la scène observée. Le processus de reconnaissance est un cycle où le graphe de restriction guide la recherche de groupes de primitives à apparier dans le graphe image et ces groupes image servent à spécifier les classes d'objets. ACRONYM symbolise les systèmes qui infèrent les hypothèses d'appariement à partir des modèles des objets et des classes de modèles d'objets.
RAF n'utilise pas de modèles génériques : les modèles décrivent la géométrie de chaque objet. Le processus de reconnaissance opère en trois phases : l'organisation des primitives image, la sélection des modèles à apparier, la vérification des hypothèses d'appariement. L'organisation est assuré par la transformée de Hough où les paramètres sont définis par les modèles des objets. La sélection des modèles correspondant aux objets image, est faite séquentiellement. Cette étape n'est en fait pas réellement étudiée. La phase de vérification d'un appariement entre un modèle d'objet et un objet image correspond à un arbre de recherche contraint par les appariements entre les primitives modèle et image. Contrairement à ACRONYM, les systèmes tels que RAF préférent construire les hypothèses à partir des primitives image.
De manière à assurer au mieux la formation des hypothèses d'appariement, c'est-à-dire les deux phases d'organisation et de sélection présentées par Grimson, Ikeuchi propose de décrire les modèles d'objets tels qu'ils sont perçus. Il représente chaque objet par son *graphe d'aspects* qui est l'ensemble des configurations 2D observables de l'objet. Le processus de reconnaissance est guidé par l'information fournie par les graphes d'aspects.

7. Cette définition n'est pas à confondre avec les systèmes *PVH* (Prédiction-Vérification d'Hypothèses) qui forment une classe dans la famille des *PV*.
8. Recognition and Attitude Finder.

1.2.1 Prédictions dirigées par les modèles

La théorie développée par Brooks dans le système ACRONYM [Brooks, 1981] propose que la reconnaissance soit faite par prédiction à partir des modèles (Figure 1.18).

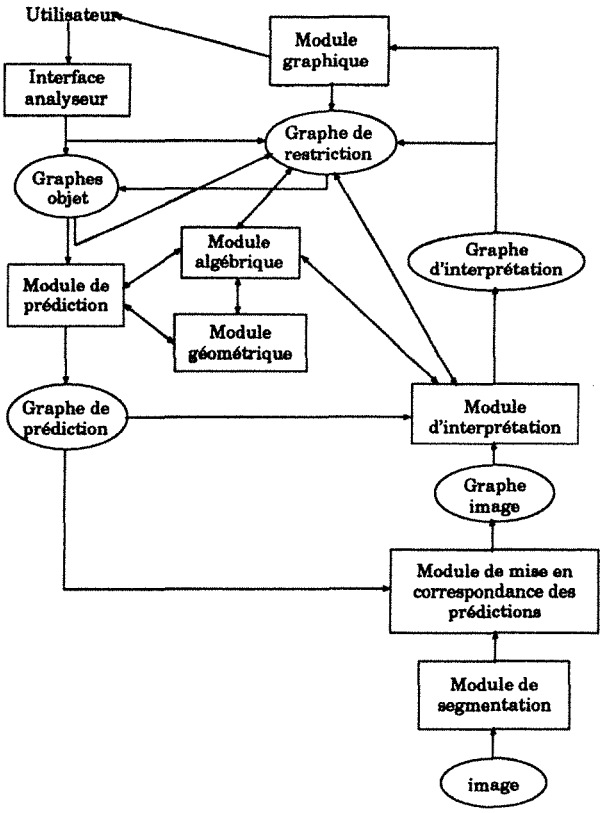


Figure 1.18. *La structure du système* ACRONYM *[Brooks, 1981].*

Les objets sont modélisés par des *graphes objet*. Les noeuds de ces graphes sont des *cônes généralisés*. Les arcs sont les relations de positions (translation et rotation) et d'inclusions (partie-de) entre les noeuds. Un cône généralisé est composé de l'axe principal de la primitive considérée, de la courbe de référence qui définit la forme de la primitive en vue perpendiculaire par rapport à l'axe et d'une règle de déplacement qui indique la déformation de la courbe de référence lors de son déplacement le long de l'axe. Les cônes généralisés sont munis d'attributs de dimensions. Par exemple, un cylindre est représenté par son axe principal, un disque comme surface de référence et une règle de déplacement égale à l'identité (Figure 1.19).
Les images sont également représentées par des graphes. Les nœuds de ces graphes sont des ellipses et des parallélogrammes représentant les projections image des cônes généralisés et les arcs les relations entre les primtives image (ellipses et parallélogrammes). L'ensemble des *graphes objet* est hiérarchiquement partitionné pour donner le *graphe de restriction*. Un noeud de restriction est composé de l'ensemble des contraintes définies sur les graphes objet de la classe représentée par ce noeud. Il présente donc

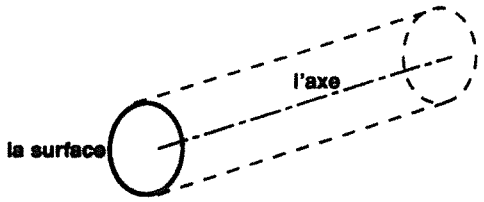


Figure 1.19. *Le cône généralisé d'un cylindre.*

des plages de tolérance sur les dimensions des cônes généralisés des graphes objet et aussi sur les contraintes qui lient ces cônes généralisés. Les arcs de restriction sont orientés des classes les plus génériques (les moins contraignantes numériquement) vers les classes les plus spécifiques (ne représentant qu'un seul graphe objet). Les graphes objet et le graphe de restriction sont construites par le concepteur.

Prenons, par exemple, la reconnaissance de vues aériennes d'aéroport. Les graphes objet sont les différents types d'avion susceptibles d'être présents dans les images observées et le graphe de restriction qui en découle décrit les différentes classes d'avions telles que les avions à réaction et les avions à hélices. Chaque graphe objet comprend un cylindre généralisé pour représenter le fuselage de l'avion et des parallélépipèdes pour les ailes. Il faut donc mettre en correspondance les formes 2D des images observées avec les formes 3D des graphes objet.

Pour cela, les graphes objet et le graphe de restriction sont utilisés pour construire le graphe de prédiction de l'image. Ce graphe permet de prédire les types de primitives image pertinentes ainsi que leurs relations. Dans notre exemple, les images aériennes d'aéroports, le graphe image décrit l'image, illustré en figure 1.20, à l'aide de parallé-

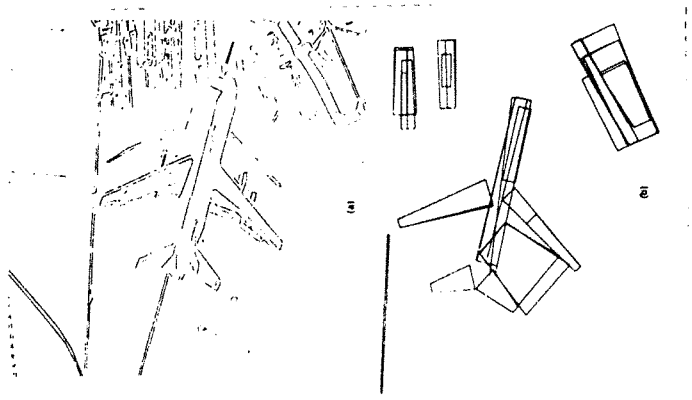


Figure 1.20. *Une image aérienne d'aéroport : les fusellages et les ailes sont décrits dans le graphe image par des parallélogrammes.*

logrammes représentant la projection 2D des fuselages et des ailes.

Ainsi, initialement, le graphe de prédiction guide les regroupements des parallé-

logrammes et des ellipses de l'image en groupes représentant les objets image. Ces groupes, une fois formés, contraignent les classes de modèles d'objet du graphe de restriction : l'appariement entre un objet image et une classe de modèles permet de restreindre les plages de tolérance associées à cette classe. Les classes de modèles deviennent donc plus spécifiques et le graphe de restriction apporte, à son tour, de nouvelles contraintes sur les regroupements des primitives image. Les restrictions sur les plages de tolérance sont traitées par un système algébrique de manipulation de contraintes (CMS).

Evaluation du système ACRONYM

Le système de manipulation de contraintes est coûteux. La complexité du processus de reconnaissance s'en ressent, mais, le modèle est générique. Il offre la possibilité d'identifier la classe d'un objet sans en connaître la forme exacte grâce à la hiérarchie décrite par le graphe de restriction. Par contre, la composition qui permet de décrire les objets en sous-parties, est adaptée aux objets manufacturés, mais, ne l'est peut être pas aux objets de la nature. Et comme la décomposition d'un objet en modèles simples n'est pas unique, il n'est pas certain qu'un objet sera perçu suivant la même composition que la modélisation l'a prédite.

Le critère de robustesse est prise en compte par l'utilisation de *groupes perceptuels* pour décrire l'image : les ellipses, les parallélogrammes et les groupes d'ellipses et/ou de parallélogrammes. Ces groupes permettent de parer aux inconvénients dus aux bruits et aux occultations. Comme nous allons le voir, cette idée de regroupement a été également développée par nombre de chercheurs dont, entre autres, Lowe [Lowe, 1985a, chapitre 4].

Le système a été testé sur des images aériennes d'aéroports qui sont propices à fournir des parallélogrammes (les carlingues des avions) mais il semblerait que ces primitives soient plus difficiles à utiliser dans le cadre de vision de type robotique.

Le système TRIDENT

Le système TRIDENT [Zaroli, 1983] [Masini and Zaroli, 1984] [Thirion, 1989] présente une simulitude avec ACRONYM. L'appariement ne se fait plus au niveau 2D, mais les images sont supposées 3D. Les tests faits jusqu'ici n'ont portés que sur des données synthétiques.

TRIDENT utilise une technique de relaxation discrète au lieu du système de manipulation de contraintes d'ACRONYM [Mohr and Henderson, 1986]. Nous retrouvons les techniques de filtrage utilisées par Waltz [Waltz, 1975] et Mackworth [Mackworth, 1977b]. Ces techniques sont plus efficaces mais moins précises numériquement que les systèmes algébriques.
Par ailleurs, TRIDENT forme les prédictions différemment d'ACRONYM car c'est un système mixte qui autorise la formation des hypothèses d'appariement aussi bien à partir des modèles qu'à partir des primitives image. La première itération du processus de reconnaissance étant forcément guidée par les primitives image, il est donc

supposé que l'image observée contiendra au moins un groupe de primitives image qui permettra d'identifier avec une forte certitude un modèle d'objet. Si ce n'est pas le cas, le système obtiendra un nombre d'hypothèses très important et donc la reconnaissance sera hautement combinatoire. Aussi, pour élaguer au plus tôt les fausses hypothèses, TRIDENT utilise une modélisation plus générique que celle d'ACRONYM. La hiérarchie est fondée sur la représentation des objets par composition de volumes simples (parallélépipèdes, sphères, etc.) et/ou de surfaces planes munis de contraintes de composition portant sur divers attributs (géométrie, couleur, etc.). De plus, les compositions des scènes et une classification de celles-ci sont décrites dans cette hiérarchie. Cette généricité supplémentaire permet de prédire des objets simplement en sachant quel type de scène est observé. Par exemple, si la scène est supposée être une salle de bain, le système peut prédire la présence de l'objet *baignoire*. Mais ces modélisations sont fort restrictives et à manipuler avec précaution car parfois les salles de bains servent de chambre noire pour le dévelopement de photographies...

Evaluation des systèmes dirigés par les modèles

La prédiction à partir des modèles est possible s'ils sont décrit dans une hiérarchie guidant la formation des hypothèses d'appariements. Cette généricité, telle qu'elle est présentée par ACRONYM ou TRIDENT, ne peut pas être construite automatiquement sans l'aide de l'utilisateur. De plus, les prédictions sont faites suivant des groupes perceptuels établis à l'avance: ellipses, parallélogramme, faces planes. Il serait plus intéressant de créer automatiquement les groupes image afin qu'ils soient adaptés aux diverses scènes observées.

Afin de répondre à ce problème, certains systèmes, dont l'un des représentants est RAF, propose de former les hypothèses d'appariement à partir des primitives et des indices image.

1.2.2 Organiser, sélectionner et vérifier

Le système RAF [Grimson and Lozano-Pérez, 1987] opère en une phase de prédiction distincte de la phase de vérification. La prédiction est la formation des hypothèses d'appariement entre un groupe de primitives image et un modèle. La vérification teste qu'il existe, pour chaque hypothèse, une transformation qui amène le modèle à *recouvrir* le groupe de primitives image.
La formation des hypothèses revient à construire un *arbre d'interprétation*. Cet arbre est le développement du processus de reconnaissance qui opère les appariements par une recherche en profondeur d'abord de la cohérence des contraintes locales d'appariement entre une primitive image et une primitive modèle. La taille de l'arbre est n^s si n est le nombre de primitives modèle et s le nombre de primitives image. Le processus de reconnaissance est donc combinatoire. Aussi, Grimson et Lozano-Pérez ont-ils introduit des heuristiques d'élagage. Un seuil de terminaison, également appelé mesure de *bonne* interprétation, permet de déterminer si une hypothèse est suffisamment *correcte* pour ne pas la tester jusqu'aux feuilles; c'est un seuil sur le rapport entre

le nombre de primitives appariées et le nombre total de primitives image. La mesure de *bonne* interprétation est délicate car si elle permet de sélectionner une hypothèse correcte elle ne permet nullement de certifier que c'est la meilleure. Pour répondre à ce problème, la mesure de *bonne* interprétation rend une liste de *bonnes* hypothèses parmi lesquelles doit se trouver la meilleure. A chaque nouvelle *bonne* hypothèse détectée, le seuil de *bonne* interprétation est remis à jour. De plus, toujour pour limiter la combinatoire lors de la création de l'arbre d'interprétation, des contraintes vérifient, pour un ensemble de primitives image et modèle appariées, l'adéquation entre les positions relatives définies sur les paires de primitives image et celles définies sur les paires modèle. Ces contraintes permettent de tester la cohérence d'arcs ou de chemins car le développement d'un arbre d'interprétation s'apparente au problème de la recherche de la cohérence dans un étiquetage [Mackworth, 1977b] [Grimson, 1990]. Une autre série de mécanismes est ajoutée pour limiter la combinatoire, en opérant des prétraitements de regroupements perceptuels avant la construction de l'arbre d'interprétation. L'utilisation de primitives évoluées telles que les segments ou les faces est envisagée dans le système. La transformée de Hough est aussi utilisée avec les primitives et les contraintes d'un modèle comme paramètres pour former des groupes image.

Evaluation du système RAF

Grimson a étudié la complexité du système RAF [Grimson, 1988] [Grimson, 1991] et a formalisé le fonctionnement idéal d'un processus de reconnaissance guidé par les primitives image de la manière suivante [Grimson, 1990] :

1. *Organisation :*
   Cette phase construit des groupes de primitives image. Chaque groupe est formé de primitives provenant d'un même objet.

2. *Sélection :*
   Il s'agit de sélectionner les modèles d'objets, contenus dans une base, susceptibles d'être appariés aux groupes image formés précédemment.

3. *Vérification :*
   Les hypothèses d'appariements entre les groupes image et les modèles doivent être vérifiées. La vérification de chaque appariement se fait en dévelopant un arbre de recherche pour trouver la correspondance optimale puis en calculant la transformation qui amène le modèle dans une position proche de celle du groupe image.

Le calcul des complexités a permis de déduire certains points importants sur la nature des hypothèses :

- *Si le groupe image décrit un seul objet et que le modèle associé représente bien cet objet* :
  Le coût de la mise en correspondance des données du modèle avec les données image, est en $O(m^2 + ams)$ où m est le nombre de données image, a le nombre de données modèle et s un coefficient associé au seuil de terminaison.

- *Si le groupe image recouvre des parties d'objets différents et la scène observée est fort encombrée* :
  Le coût est exponentiel. Donc à éviter à tout prix.

- *Si le groupe image recouvre des parties d'objets différents , mais, la scène est peu encombrée* :
  Le coût est dans ce cas cubique. C'est plus intéressant, mais, notons tout de même que le processus n'est pas censé gérer le taux d'encombrement de la scène.

Ces résultats mettent en avant l'utilité de définir correctement le groupe image et le modèle sélectionné. Grimson propose d'utiliser des techniques qui doivent assurer la formation de groupes image tels que chacun décrive un unique objet. Pour cela, le système RAF utilise la transformée de Hough.

La complexité du processus de reconnaissance et le critère de robustesse ont été très sérieusement étudiés. Pour limiter la taille de l'arbre d'interprétation certaines heuristiques sont utilisées qui sont hélas des paramètres suplémentaires à estimer pour l'utilisateur.

Les modèles ne sont pas génériques, mais, la formation des hypothèses se faisant à partir des primitives image, la généricité n'est pas utile. Par ailleurs,le problème de la sélection n'est pas non plus considéré. Les modèles d'objets sont simplement stockés dans une base de connaissance. Il faut créer un arbre d'interprétation pour chacun des modèles de la base et si la base est importante, la complexité du système s'en ressent.

Nous allons maintenant présenter des systèmes de la même famille où chacun propose une solution à un des problèmes cité ci-dessus. Le système HYPER [Ayache and Faugeras, 1986] propose une procédure de vérification des plus efficaces. Le système SCERPO propose de s'intéresser à la phase d'organisation en reprenant les concepts décrits par l'école Wertheimer, Clowe, Huffman, *et al.*: le *regroupement perceptuel*. Les systèmes BONSAI et "Sticks, Blobs and Plates" s'orientent vers une étude de la phase de sélection des modèles.

Le système HYPER

Pour la phase de vérification, HYPER [Ayache and Faugeras, 1986] présente une technique qui semble tout à fait efficace. La transformation qui amène le modèle dans la position du groupe image est calculée par une technique itérative des moindres carrés pondérés par les erreurs de mesure des primitives image: le filtre de Kalman. Par contre, les hypothèses sont construites à partir d'heuristiques.
Les plus grands segments image sont sélectionnés pour identifier le modèle et pour estimer la première transformation T_0 qui amène le modèle dans une position proche du groupe image. A partir de la prédiction de la transformation, le système prédit la position d'une primitive image non encore appariée. Puis la transformation est à nouveau estimée en verifiant que l'erreur de localisation reste en deçà d'un seuil préfixé. Le processus est itéré pour l'ensemble des primitives modèles.

Les étapes d'organisation et de sélection ne sont pas du tout étudiées dans ce système. Par contre, il présente un processus de vérification tout à fait intéressant au

vu des résultats.

Le système SCERPO

Le système SCERPO [Lowe, 1986] propose un processus de reconnaissance qui met en correspondance des modèles 3D avec des scènes décrites par des images 2D. Le principe général est le *regroupement perceptuel* des primitives image ; nous retrouvons les principes avancés par Clowes [Clowes, 1971], Huffman [Huffman, 1971]. Ces regroupements, construits à partir de relations entre les primitives 2D image, servent à former les hypothèses d'appariement avec les modèles de la base : un catalogue de relations 2D permet d'inférer des attitudes 3D ainsi que la position de la caméra dans la scène. La figure 1.21 décrit les groupes perceptuels 2D recherchés et les inférences 3D qui en dé-

Relations 2D	Inférence 3D	Exemples
colinéarité	colinéarité 3D	
points curvilignes ou arcs de cercles	points curvilignes 3D ou arcs de cercles 3D	
connexité de plusieurs segments de courbes ou de droites	sommets 3D d'un objet	
jonction "T"	L'arête continue occulte l'autre en 3D	
deux segments continues se croisent	ceux ne sont pas des contours occultants	
un groupe de segments parallèles	segments parallèles en 3D	
plusieurs segments convergent vers un même point	segments parallèles (points de fuite)	
points colinéaires équidistants ou parallèles équidistantes	points colinéaires équidistants ou lignes coplanaires	
patterns identifiables dans un ensemble de points ou de segments	les mêmes patterns sont identifiables en 3D	
segments virtuels parallèles entre les tangentes aux discontinuités de deux coubes	deux arêtes : une arête d'objet et une arête d'ombre	

Figure 1.21. *Les relations* 2D *et leurs inférences* 3D.

coulent. Le système SCERPO vérifie la présence d'un objet particulier dans une image à partir d'un certain nombre de groupes perceptuels image. Par exemple, l'agrafeuse dans la figure 1.22 est décrit par un ensemble de parallélogrammes permettant d'estimer la position 3D de l'objet. La version du système présenté dans [Lowe, 1986] utilise uniquement les segments de droites comme primitives image. Le système recherche pour chaque groupe perceptuel extrait de l'image les appariements initiaux avec les modèles. Puis, des hypothèses plus sûres sont créées en regrouant les appariements concernant les groupes perceptuels image. Chaque hypothèse est ensuite confirmée si le projeté du modèle sur l'image est proche du groupe à apparier. L'estimation de la transformation est assurée en utilisant une méthode itérative des moindres carrés.

Ce système démontre l'intérêt d'opérer le mieux possible la phase d'organisation perceptuel

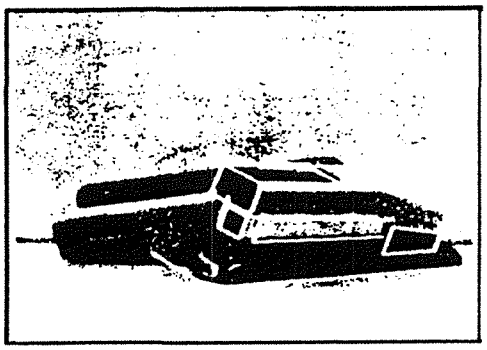


Figure 1.22. *Une image analysée par le système* SCERPO.

Le système BONSAI

Flynn [Flynn and Jain, 1991] utilise également les arbres d'interprétation pour assurer la reconnaissance d'objets perçus par un capteur laser 3D. A l'encontre de Grimson et Lozano-Pérez, il étudie comment limiter le nombre d'arbres d'interprétation à générer. Il propose alors d'utiliser une procédure itérative qui, se basant sur les caractéristiques intrinsèques des primitives image, sélectionne les modèles susceptibles de correspondre. A chaque itération, les primitives sélectionnées sont celles correspondant à un nombre de modèles le plus faible possible. Cette technique de sélection des modèles n'est pas assurée d'identifier tous les objets image par des modèles: nous n'avons aucune preuve que tous les objets présents dans l'image soient détectés. Il se peut qu'un objet, présent dans l'image, n'ait pas de primitive dont les carcatéristiques soient suffisamment pertinentes pour permettre l'identification de l'objet dans l'image et donc ne permet pas la sélection du modèle correspondant. La phase finale de vérification est un calcul des différents scores qui portent sur les caractéristiques intrinsèques des primitives image et modèle.

Le système "Sticks, Blobs and Plates"

Le système présenté par Mulgaonkar *et al.* [Mulgaonkar *et al.*, 1984] exprime les modèles d'objets par des *tiges*, des *volumes* et des *plateaux*, illustrés en figure 1.23 et par les relations de position relative entre ces primitives qui sont de trois types : uniaire, binaire ou ternaire. Les relations unaires sont des seuils de tolérance qui permettent de prendre en compte les décallages entre les primitives image et modèle lors des appariements. Les relations binaires représentent les connexités existant entre les primitives. Par exemple, la connexion entre deux tiges s'exprime par le point de connexité et l'angle que forme les deux tiges ou encore la connexion entre deux plateaux se représente par le point de connexité et les trois angles α, β, δ qui donnent la position du plateau B par rapport au plateau A, comme l'illustre la figure 1.24. Les relations ternaires expriment les connexions des primitives trois par trois. L'image est décrite en terme de *silhouettes* qui sont les polygones 2D convexes représentant les objets de l'image. Le processus de reconnaissance traite séquentiellement les modèles de la base

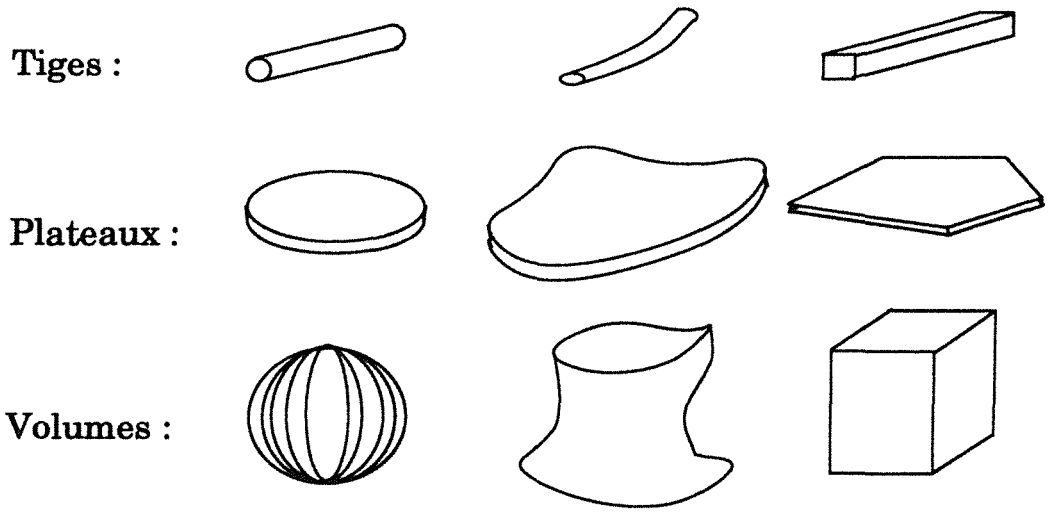


Figure 1.23. *Des exemples de tiges, plateaux et volumes [Mulgaonkar* et al., *1984].*

en créant pour chacun d'eux un arbre d'interpre'tation dont chaque niveau exprime la vérification de l'appariement d'une partie du modèle avec la silhouette concernée. La vérification revient à estimer le point de vue sous lequel la partie du modèle est observée dans l'image. Ainsi, à chaque niveau dans l'arbre d'interprétation, l'estimation du point de vue est affinée jusqu'à ce que l'appariement soit complet ou rejeté.

Mulgaonkar propose une sélection séquentielle des modèles à apparier car la base en contient peu. Mais, pour une base plus importante, Shapiro et Haralick [Shapiro and Haralick, 1982] proposent une classification structurelle hiérarchique des modèles des objets qui repose sur une mesure de distance entre graphes (isomorphisme de sous-graphes). Par ailleurs, la décomposition en silhouettes n'est pas évidente et aucun résultat ne vient étayer l'idée.

Le système "Sticks, Blobs and Plates", présenté par Mulgaonkar *et al.*, est une proposition théorique pour rapprocher les decriptions modèle et image.

Evaluation des systèmes "Organiser, Sélectionner, Vérifier"

Le principe de reconnaissance utilisé par tous les systèmes que nous venons de voir consiste à diriger la formation des hypothèses d'appariement à partir de l'informationm image. Le système doit opérer la formation des groupes image, la sélection des modèles susceptibles d'être appariés aux groupes image et, enfin, la vérification de ces hypothèses d'appariement. S'il y a des solutions pour effectuer la phase de vérification [Ayache and Faugeras, 1986], ce n'est pas le cas pour les phases d'organisation et de sélection. Aussi, Une solution proposée, pour assurer les phases d'organisation et de sélection, est de construire des graphes d'aspects.

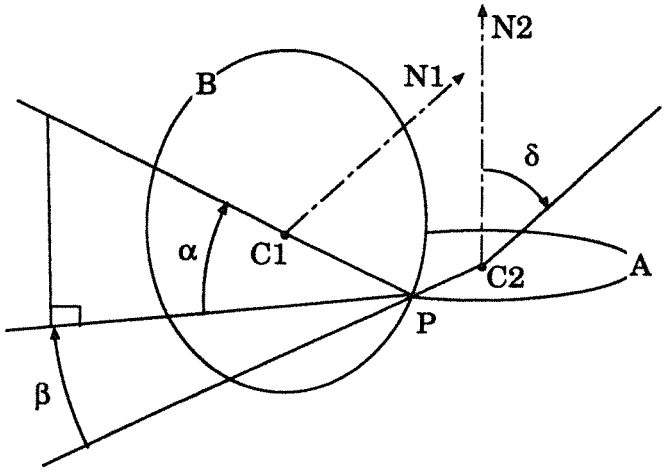


Figure 1.24. *La connexion entre deux plateaux [Mulgaonkar* et al., *1984].*

1.2.3 Les systèmes fondés sur les graphes d'aspects

Les travaux sur les graphes d'aspects et leur utilisation pour la reconnaissance sont fondés sur les études du comportement psychologique de la vision humaine[9]. Avec le système proposé par Ikeuchi, nous allons présenter une des références dans cette classe de systèmes.

Ce système [Ikeuchi, 1987] se concentre sur la modélisation des objets pour une utilisation directe lors de la reconnaissance. Pour chaque objet est généré l'ensemble des *apparences* 2D. Cet ensemble est automatiquement créé par l'observation des objets via un échantillonnage d'une sphère gaussienne étendue centrée en l'objet. Chaque point de vue, appelé *apparence*, est décrit en termes de faces. Puis ces apparences sont regroupées pour former les *aspects* des objets. Prenons, pour exemple, l'objet illustré en figure 1.25. Ses apparences sont illustrées par la figure 1.26.a : les primitives

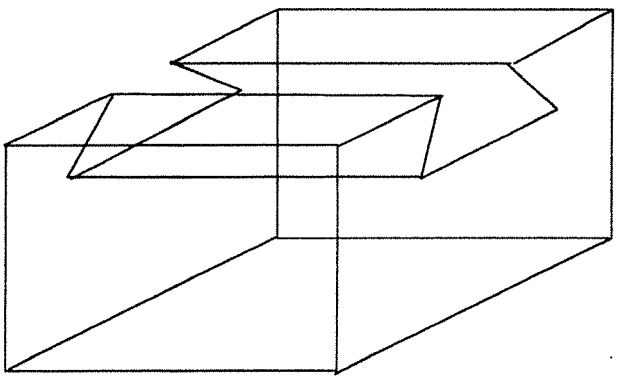
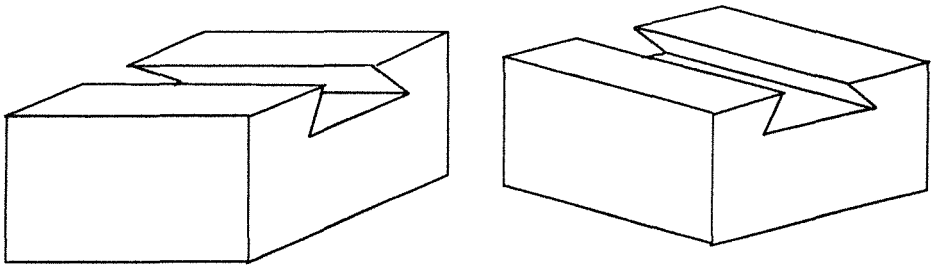
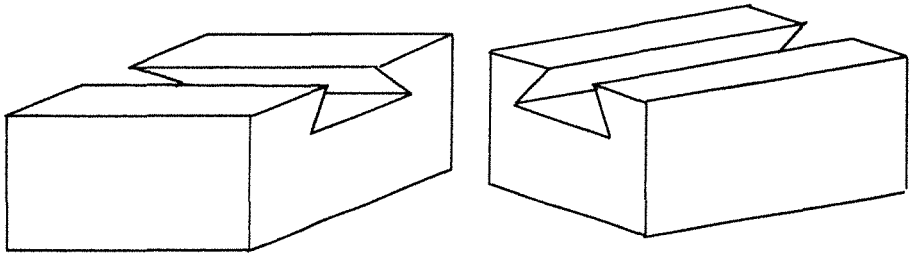


Figure 1.25. *Une vue synthétique d'un objet.*

9. Wertheimer blabla 1923



(a) **Deux apparences d'un même aspect.**



(b) **Deux aspects d'un même objet.**

Figure 1.26. *Des exemples d'aspects et d'apparences de l'objet de la figure 1.25.*

image observées sont les mêmes mais dans des positions différentes. Puis, cet ensemble d'apparences est partitionné en classes, appelées *aspects* : le passage d'un aspect à un autre est détecté par l'apparition d'une nouvelle primitive ou par la disparition d'une déjà présente. La figure 1.26.b illustre cette notion d'aspect : la différence de points de vue entraîne une différence entre les groupes de primitives observées.

L'objet est ensuite modélisé à partir de ses aspects sous la forme d'un arbre d'interprétation. Les premiers niveaux de cet arbre représentent la classification des aspects où les embranchements indiquent les tests de validation à effectuer lors de la reconnaissance. Ensuite, pour chacun des aspects, le concepteur indique les outils nécessaires pour définir complètement la position de l'objet ; c'est-à-dire qu'il définit les moyens nécessaires pour l'identification de l'apparence de l'objet observé. Ces outils étant les distributions des normales aux surfaces, les formes des surfaces,l'orientation des surfaces, etc., les caractéristiques qu'ils fournissent sont donc globales à un groupe de primitives. Aussi, ne supprotent-ils pas les occultations. L'arbre d'interprétation généré pour l'objet, de la figure 1.25 et modélisé par ses cinq aspects $(S_1, \ldots, S_5)$, est décrit en figure 1.27 : Les premiers niveaux (*classification*) permettent d'identifier l'aspect observé tandis que les niveaux suivant indiquent, pour chaque aspect, les outils permettant de définir au mieux l'apparence observée.

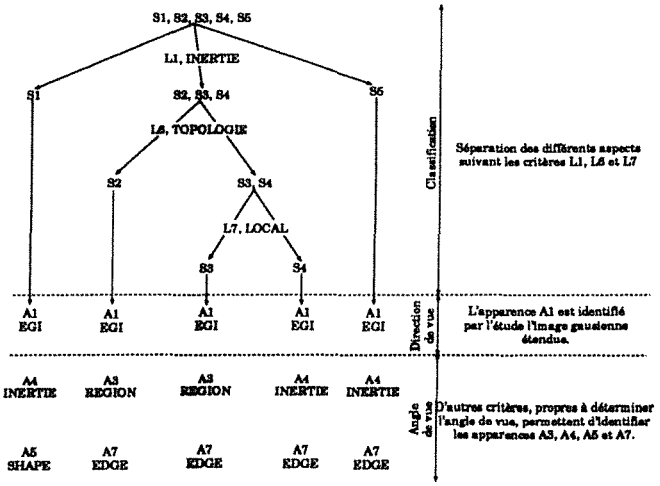


Figure 1.27. *L'arbre d'interprétation de l'objet de la figure 1.25.*

Evaluation

Le système est prévu pour manipuler un ensemble d'objets identiques. C'est l'utilisateur qui définit l'objet à manipuler et donc le modèle à utiliser. Il est possible de créer un arbre d'interprétation pour chaque objet car un seul est utilisé pour une manipulation. Il semble difficile de généraliser ce système à d'autres tâches qui comporteraient plusieurs objets différents pouvant s'occulter mutuellement.

Par ailleurs, le modèle n'est pas du tout générique. Il est important de noter que, malgré la dénomination *graphe d'aspects* 2D qui lui est attaché, ce système gère des informations 3D telles les images gaussiennes étendues, les moments des faces originales, les contours des faces originales, etc..

1.3 En résumé

Nous venons de décrire une grande partie des techniques et des systèmes de vision actuels. Nous les avons cataloguer en deux groupes pour différencier ceux qui recherchent *les occurrences d'un modèle d'objet* sélectionné lors d'une étape précédente, de ceux qui recherchent *tous les objets présents dans une image.*

Les techniques de recherche des occurrences d'un modèle

Ces techniques proposent des méthodes d'appariement tout à fait efficaces telles que la transformée de Hough ou les modèles élastiques. Mais elles présupposent que la formation des hypothèses d'appariement soit faite au paravant. La formation des hypothèses représente deux tâches qui ne sont pas forcément distinctes l'une de l'autre. D'une part, il faut former les groupes de primitives suivant les objets présents dans l'image. Et d'autre part, il faut sélectionner les modèles susceptibles de s'apparier aux

groupes image.

Pour répondre à ces deux étapes de formation d'hypothèses, diverses techniques sont utilisées et présentées sous la terminologie: *perception des objets.*

Les techniques de perception des objets

Parmi l'ensemble des techniques de perception des objets, nous avons décrit les plus représentatifs et nous les avons classés en trois grandes familles suivant le principe de formation des hypothèses utilisé:

- *Les techniques de prédiction à partir de modèles génériques* : Le système ACRONYM de Brooks propose de former les hypothèses à partir des modèles. Pour ce faire, les modèles d'objets sont partionnés hiérarchiquement et cette hiérarchie est utilisée pour guider la formation des groupes image. Mais cette technique semble limitée à des scènes qui comportent des caractéristiques suffisamment discriminantes pour éviter une explosion combinatoire de l'arbre de recherche lors de la phase de vérification des hypothèses. Par exemple, les vues aériennes d'aéroport sont tout à fait adaptées à ce genre de prédiction contrairement aux scènes d'intérieur de type robotique.
  TRIDENT est un système qui utilise également les modèles pour former les hypothèses d'appariement mais son fonctionnement est plus complet puisqu'il peut également les former à partir des primitives. Ce système a permis de mettre en avant le besoin d'une modélisation tout à la fois générique et souple. La généricité permet de prédire les appariements à partir des modèles. La souplesse signifie que les modèles ne doivent pas représenter des informations qui peuvent conduire à des erreurs difficilement détectable. Par exemple, la modélisation des scènes est à utiliser avec précaution. Par ailleurs, la formation des hypothèses à partir des primitives image est peu étudié et, en fait, il est supposé que l'on puissent découvrir assez rapidement une primtive suffisamment pertinente pour élaguer l'arbre de recherche développé par le processus de reconnaissance.

- *Les techniques de prédiction à partir des primitives image* : nous avons ensuite détaillé un certain nombre de systèmes qui utilisent les primitives image pour former les hypothèses. Tout comme nous l'avions suspecté avec TRIDENT, le coût de tels techniques est apparu comme l'un des obstacles majeurs. Différentes solutions aux niveau des différentes étapes de la formation et de la vérification des hypothèses ont été proposées. Mais si les techniques de vérification des hypothèses ont été beaucoup étudiées, il n'en est pas de même de la formation des hypothèses. Les seules techniques proposées sont ou peu satisfaisantes ou pas étudiées sur des images réelles.

- *Les techniques de modélisation des objets tels qu'ils sont perçus* : ces techniques cherchent à prendre en compte lors de la modélisation les informations susceptibles d'être pertinentes pour la formation des hypothèses d'appariement. Cette modélisation consiste à représenter les objets tels qu'ils sont perçus, c'est-à-dire par les différentes configurations obtenues par différents points de vues:

les *graphes d'aspects*. Le systèmes proposé est particulier puisqu'ils n'a pas pas besoin de tenir d'occultation possible (c'est une simplification importante) et que la sélection est assurée par la tâche imposée au robot. Bien que nombre de chercheurs étudient la possibilité d'utiliser ces graphes d'aspects pour assurer la formation des hypothèses d'appariement, un des principaux inconvénients couramment cités à est leur taille ; un objet peut comporter plusieurs millions d'appects [Ahuja *et al.*, 1992] !

Nous constatons que le problème de sélection des modèles et d'organisation des primitives image restent ouvert et qu'aucune direction étudiée jusque là n'a pu être confirmée ou infirmé par un banc de tests. Nous nous proposons donc d'étudier et de tester un des principes proposés pour la formation des hypothèses : *L'indexation structurelle des modèles d'objets*. L'indexation est un processus qui construit, à partir des modèles d'objets, un *index*, c'est-à-dire une structure qui opère une classification structurelle plane ou hiérarchique des modèles d'objets de manière à accélérer les phases de sélection et d'organisation. Dans ce cadre, le chapitre suivant présente une bibliographie d'un certain nombre de techniques utilisées pour l'indexation structurelle.

2
Les techniques d'indexation

L'*indexation* est une des techniques utilisées pour accélérer la sélection des modèles susceptibles de s'apparier aux groupes de primives représentant les objets observés dans l'image. L'objectif est d'enregistrer pour chaque modèle d'objet l'information susceptible d'être perçue dans les images à analyser. Cette information peut être de diverses natures telle que la fonction, la couleur ou la forme des objets. Dans ce chapitre, nous nous interesserons particulièrement à l'information contenue dans la structure des objets.

L'indexation structurelle donne lieu à une classificaion qui peut être plane ou hiérarchique. Les classifications planes supposent qu'un type d'information structurelle, tel que les jonctions de segments 2D, est suffisamment pertinent pour la sélection des modèles. En fait, la phase d'organisation est supposée stable, c'est-à-dire qu'elle fournit pour la même scène observée dans les mêmes conditions toujours les mêmes informations image et, dans ce cadre, l'index sert à la sélection des modèles. Avant d'introduire plus avant ces techniques, notons le fait que la phase d'organisation est supposée stable n'est pas vérifiée dans la pratique. Par contre, les classifications hiérarchiques conjecturent que l'information ne peut pas être complètement connue à l'avance et qu'il faut guider la construction des groupes image pertinents. Dans ce cas, l'index est utilisé pour fusionner les étapes de sélection des modèles et d'organisation des groupes image.

Définissons, dès à présent, les critères qui nous permettent de comparer les différentes méthodes d'indexation.

- *La facilité de mise à jour* : la connaissance d'un nouvel objet peut-elle être enregistrée au sein de l'index sans à avoir à reconstruire complètement celui-ci ? Si ce n'est pas le cas, le coût de la modélisation est fortement pénalisé.

- *La généralisation :* l'index permet-il une approche par affinement des groupes image tout en procédant à la classification des modèles ? Nous entendons par approche par affinement le fait de constituer des groupes image suivant l'ordre croissant des cardinalités. En d'autres termes, la généralisation traduit une hiérarchisation à la fois en termes de classes de modèles d'objets et en termes de description de groupes image.

- *La robustesse :* quel est le comportement de l'index en présence de bruits et d'occultations ?
- *La complexité :* le processus d'indexation, c'est-à-dire de classification des modèles, peut-il contrôler et optimiser l'espace mémoire utile à la classification ?

Ayant défini les critères d'évaluation des techniques d'indexation, ce chapitre va, en premier lieu, présenter les techniques de classifications numériques et conceptuelles ainsi que deux applications à la vision. Puis la section 2.2 présente les techniques de hachage utilisées en vision pour l'indexation. Enfin, avant de conclure, les techniques d'indexation fondées sur les graphes de décision sont décrites dans la section 2.3.

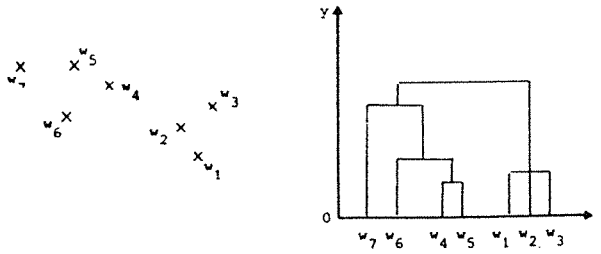
2.1 Les techniques de classifications

Parmi les classifications, nous distinguons les classifications numériques provenant de l'analyse de données et les classifications conceptuelles.

2.1.1 Les techniques d'analyse de données

Il existe deux catégories de techniques d'analyse de données : les méthodes ascendantes et les méthodes descendantes.

Les méthodes ascendantes traitent le problème en partant des données. Elles regroupent les données à classer suivant une notion de voisinage définie par une mesure de distance, jusqu'à obtenir une seule classe recouvrant l'ensemble de ces données. Ces regroupements forment un arbre, appelé dendogramme, où chaque niveau représente un partitionnement possible des données. La figure 2.1.b illustre la classification hié-



(a) (b)

Figure 2.1. *Un exemple d'arbre de classification obtenu par une technique ascendante [Diday* et al., *1982].*

rarchique obtenue à partir des données décrites en figure (2.1.a). Chaque feuille de l'arbre est une donnée initiale (w_i) tandis que la racine recouvre l'ensemble des données. L'arbre ainsi construit peut ensuite être étudié afin d'établir le niveau décrivant la meilleure classification. Ce choix est fait à partir d'un critère de distance interclasse. Ce principe nécessite la construction de toutes les classifications intermédiaires : des feuilles jusqu'à la classification répondant au critère de distance interclasse. Notons,

en plus, que l'introduction d'une nouvelle donnée entraîne la reconstruction de tout l'arbre de classification jusqu'à la classification souhaitée au sens du critère de distance interclasse établi qui entraine un coût en temps du processus de classification.

Quant aux méthodes descendantes, elles tentent de diminuer le coût élevé des techniques ascendantes. L'objectif est de créer des classes de manière itérative en cherchant la stabilité d'un critère. Une des techniques les plus connues est celle des nuées dynamiques. Cette méthode est décrite par la figure 2.2. Connaissant le nombre k

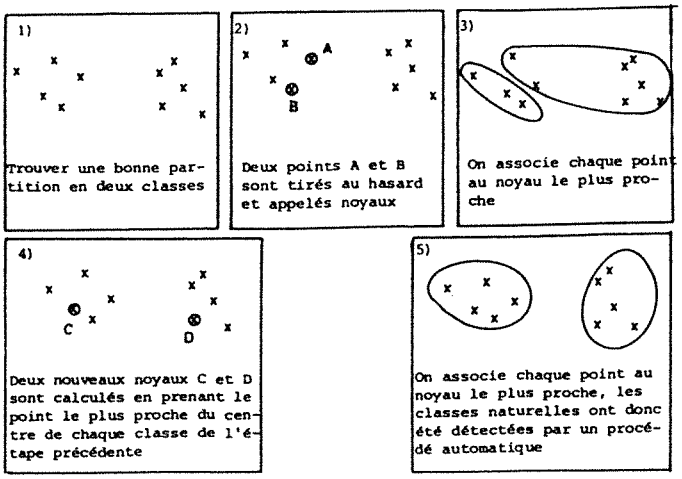


Figure 2.2. *Schéma de l'algorithme des nuées dynamiques [Diday* et al., *1982].*

de classes à créer, l'initialisation du processus de classification est obtenue en générant aléatoirement les centres des k classes. Puis chacune des données à classer s'associe au centre qui lui est le plus proche, suivant une mesure de distance définie auparavant. A partir de ces k classes ainsi construites, le processus calcule à nouveau les k centres et réitère ainsi jusqu'à stabilisation des centres.
Cette technique permet d'obtenir une classification de manière bien plus rapide surtout si le nombre de classes à construire est faible par rapport au nombre de données initiales. Elle peut être utilisée pour construire des classifications hiérarchiques comme l'illustre la figure 2.3. Cependant, le nombre de classes à créer est un des paramètres du problème.

Evaluation des techniques de classification numérique

Mise à jour	Généralisation	Robustesse	Complexité
Nulle	Correcte	???	Elevée

Les deux méthodes, ascendante et descendante, sont coûteuse en temps car elles ne sont pas incrémentielles : l'introduction d'une nouvelle donnée oblige la reconstruction de toute la classifation.

Par contre, la généralisation peut-être assurée par la construction d'un arbre de classification : le dendogramme.

La complexité n'est pas prise en compte par les méthodes de classification ascendantes. Mais, les techniques descendantes peuvent la contraindre en fixant le nombre de

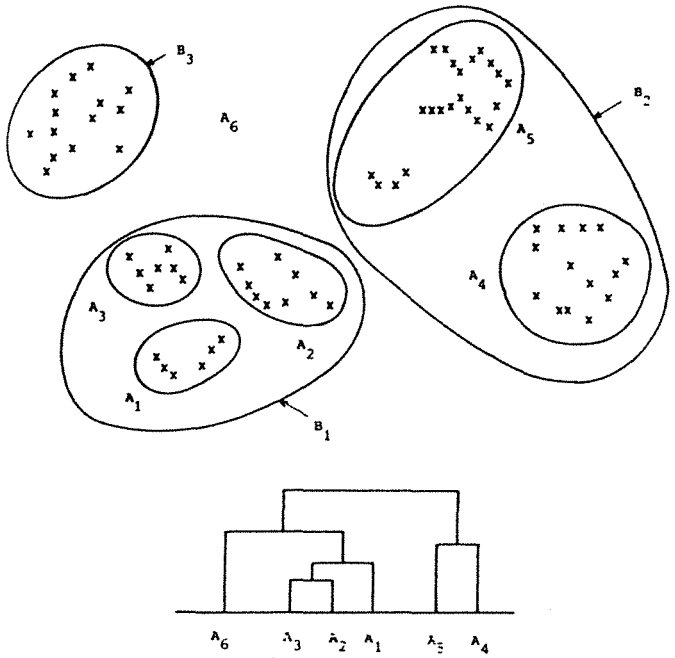


Figure 2.3. *Un exemple de dendogramme construit par une technique des nuées dynamiques [Diday* et al., *1982].*

classes à chaque niveau. Mais il n'existe pas de principe général qui permette d'évaluer le nombre de classes à créer.

2.1.2 Le regroupement perceptuel par classification

Wallace [Wallace and Kanade, 1990] [Wallace, 1989] présente une application d'une technique de classification par analyse de données pour le regroupement perceptuel. Les données sont des points images représentés par leurs coordonnées.

Le processus développé par Wallace se déroule en deux étapes : le module *NIHC*[1] et le module MDL[2].
Le module *NIHC* construit un arbre binaire de classifications par une technique ascendante. Les feuilles de l'arbre sont les données tandis que les noeuds sont construits en regroupant au fur et à mesure les données en minimisant la fonction récursive suivante :

$$E(u) = \begin{cases} 0 & si\ u = \emptyset \\ log(|C(u)| + E(\ell) + E(r) & sinon \end{cases}$$

r et l sont les fils droit et gauche du noeud u. $C(u)$ est la matrice de covariance et $\log(|C(u)|)$ ($|.|$ est le déterminant) est une approximation de l'entropie :

1. *N*on *I*terative *H*ierarchical *C*lustering
2. Minimum Description Length

$-\int f(x)\log f(x)dx \equiv log(|C|)$, où f est la gaussienne dont la matrice de covariance associée est C.

Une fois l'arbre créé, la deuxième étape est activée pour rechercher la classification qui minimise la fonction $MDL = I(C)+I(S/C)$ qui est également une mesure d'entropie au sens de la théorie de l'information. $I(C)$ exprime la dispersion de la classification C et $I(S/C)$ est la mesure de dispersion de l'ensemble de données S connaissannt la classification C. La figure 2.4 donne quelques exemples de classification.

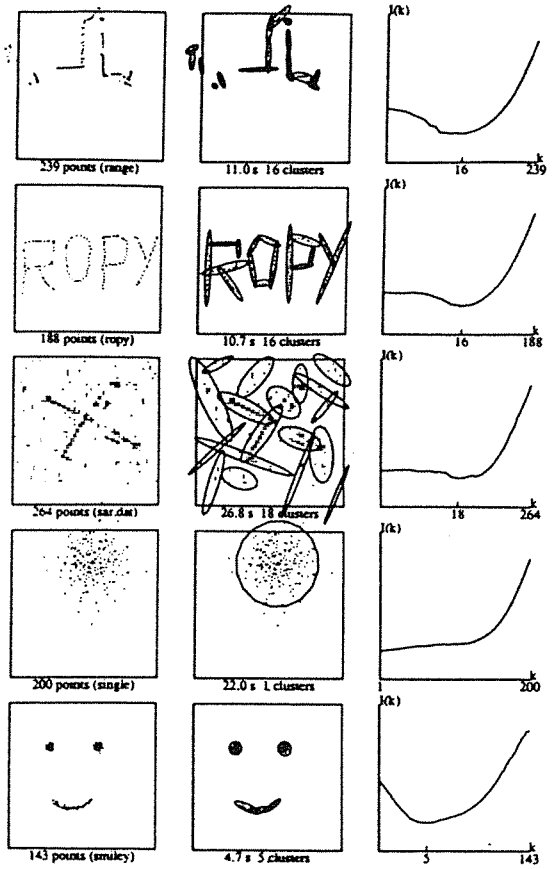


Figure 2.4. *plusieurs exemples de classification [Wallace, 1989].*

Evaluation

Notons que le but est le regroupement perceptuel automatique. Mais, l'utilisation, qui peut en être faite comme technique d'indexation, semble peu judicieuse car il ne serait pas raisonnable de modéliser les objets simplement par des vecteurs d'attributs. Un vecteur d'attributs purement géométriques ne peut exprimer la structure d'un objet qu'au risque d'obtenir un processus de reconnaissance qui risquerait d'être obligé de manipuler des informations numériques précises et donc fragiles quand il faudrait utiliser des informations plus robustes.

2.1.3 Une hiérarchie de graphes relationnels

Pour modéliser la forme des objets, Shapiro et Haralick [Shapiro and Haralick, 1982] proposent d'utiliser les *graphes relationnels* qui fournissent des descriptions structurelles. Les noeuds de ces garphes décrivent les primitives (arêtes, faces, volumes) par leurs caractéristiques intrinsèques tandis que les arcs expriment les relations entre ces primitives.
Chercher la similitude structurelle entre deux objets : c'est-à-dire chercher les parties présentant la même structure, revient à chercher le plus grand sous-graphe commun entre leurs deux graphes. Shapiro et Haralick [Shapiro, 1985] proposent une mesure D entre deux graphes relationnels A et B qui permet d'évaluer l'isomorphisme entre les deux graphes. Cette mesure $D(A, B)$ est une distance métrique qui permet de construire un arbre binaire décrivant une classification hiérarchique structurelle d'un ensemble de graphes relationels. L'algorithme du processus de classification est le suivant :

1. Construire la racine qui représente l'ensemble S des graphes.
2. Choisir dans S les deux graphes A et B tels que :

$$\sum_{G \in S} \min\{D(G, A), D(G, B)\}$$

   A et B sont les fils gauche et droit respectivement.
3. Construire les partitions P_A et P_B.
4. Si P_A (resp. P_B) est suffisamment grande pour être partitionner, recommencer les trois premières étapes avec P_A (resp. P_B).

Evaluation

L'utilisation de la description structurelle des objets est adaptée aux images fournies par les systèmes de caméras.

Par contre, le critère de mise à jour n'est pas respecté. En outre, le processus utilise nombre de paramètres qui sont fixés par l'utilisateur.

2.1.4 Les classifications conceptuelles

La classification conceptuelle regroupe en classes, appelées *concepts*, un ensemble d'objets hétérogènes. Un objet est décrit par une conjonction d'attributs de type qualitatif (nominal, ordinal), quantitatif (intervalle, proportion, exacte) ou hiérarchique (exemple en figure 2.5). Cette hétérogénéité des types d'attributs permet de représenter les objets de manière bien plus riche qu'en utilisant des attributs purement numériques. Par exemple, un train peut être défini par son nombre de wagons, la forme et la couleur de sa locomotive, la forme et la couleur de ces wagons, ainsi de

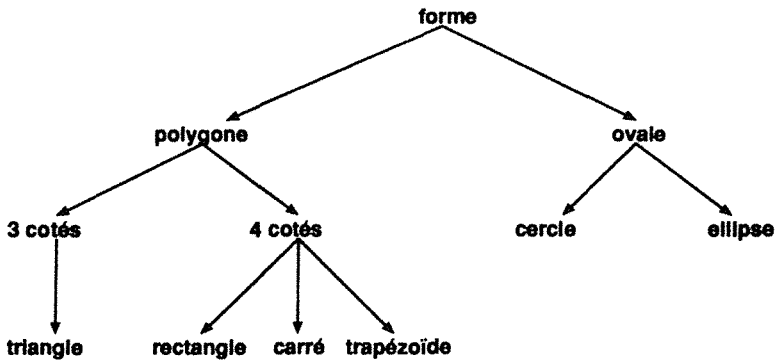


Figure 2.5. *L'attribut hiérarchique des formes.*

suite. Mais, les attributs nominaux, par exemple, ne sont pas mesurables comme le sont les attributs numériques ni comme le sont les attributs hiérarchiques. Il faut donc introduire une liste de critères de mesures. Michalski et Stepp [Michalski and Stepp, 1983] construisent des classifications suivant une technique conceptuellement proche des nuées dynamiques. Les paramètres de leur technique de classification automatique sont : les objets à classer, le nombre de classes à créer et la liste des critères de mesures pour chaque type d'attribut. Notons que cette classification n'est pas incrémentielle et donc le coût de cet algorithme s'en ressent.

Decaestecker [Decaestecker, 1989a] [Decaestecker, 1989b], Lebowitz [Lebowitz, 1987] [Lebowitz, 1988], Fisher [Fisher, 1987] proposent des versions incrémentielles des classifications conceptuelles hiérarchiques. Nous développons celle de Decaestecker, pour en comprendre le fonctionnement général. Les concepts et les objets doivent respecter deux règles fondammentales :

1. Les concepts d'un même niveau dans la hiérarchie doivent former une partition de l'ensemble des objets.
2. Le concept p est décrit par les attributs communs à tous les objets qu'il représente et qui ne sont pas présents dans les parents de p.

Chaque niveau dans la hiérarchie correspond à une partition $P = \{C_1, \ldots, C_n\}$. L'introduction d'un nouvel objet O se fait en deux phases :

- *Intégrer O dans P :*
  pour chaque concept C_j de P, le score $SC(O, C_j)$ est calculé. Ce score représente le nombre d'attributs en commun à C_j et à O.
- *Modifier P en p' et Intégrer O dans un des concepts de P' :*
  le score précédemment introduit nous permet d'ordonner par rapport à O les classes de la partition P :

$$SC(O, C_i) \geq SC(O, C_r) \geq \max_{k=1}^{n} SC(O, C_k)$$

Les classes C_i et C_r sont les deux classes s'appariant au mieux à l'objet O et sont donc les seules à être étudiée pour la modification de la partition P en P' :

- $p' = P$: la partition ne change pas,
- $p' = P \cup \{C_O\}$: le concept C_O, dont l'unique représentant est l'objet O, est ajouté à P,
- $p' = (P \backslash \{C_i, C_r\}) \cup \{C_{i,r}\}$: les deux concepts C_i et C_r sont fusionnés dans la partition P,
- $p' = (P \backslash \{C_i\}) \cup \{C_{i1}, \ldots, C_{ip}\}$: le concept C_i est supprimé de la partition P et ses concepts fils sont rattachés à son concept père.

Le choix de la meilleure partition pour p' se fait en optimisant la fonction suivante :

$$\max(SC(O, C_j, P')$$

.

Evaluation

Mise à jour	Généralisation	Robustesse	Complexité
Faible	Correct	???	Elevée

La mise à jour est faible car locale ; les modifications apportées ne concernent que les concepts les plus *proches* de l'objet introduit (C_i et C_r) alors que rien ne laisse entendre qu'un concept éloigné n'est pas concerné.

Outre cela, les attributs ne permettent pas de décrire la géométrie de n'importe quel objet. Par exemple, l'attribut forme est limitatif aux seules formes définies nominalement dans une hiérarchie et donc il n'est pas possible d'introduire un objet dont la forme ne peut pas être décrite de manière nominale. Une bien meilleure représentation d'un objet consiste en un graphe relationnel comme l'on introduit Haralick et Shapiro [Shapiro and Haralick, 1982].

2.2 Les tables de hachage

Les tables de hachage sont des outils d'indexation fort connus et utilisés pour le tri dans maints domaines : les systèmes d'exploitation, les bases de données, etc.. Une table de hachage représente un ensemble de tiroirs contenant les données à stocker. Les tiroirs sont accessibles par un jeu fini de clefs. L'exemple le plus simple est le dictionnaire qui stocke les mots en paquets suivant une seule clef qui est la première lettre de chaque mot. Les principales caractéristiques d'une table de hachage sont l'accessibilité aux données en temps constant et la possibilité d'ajouter et de supprimer une donnée sans en perturber son fonctionnement. Autrement dit, quelques soient les données et les clefs utilisées, le critère de mise à jour est respecté. Nous allons décrire trois index fondés sur les tables de hachage utilisant respectivement comme clefs les points, les chaînes et les graphes.

2.2.1 Les points

Considérons la recherche de l'appariements d'un modèle décrits par des points 3D avec une image décrite par des points 2D [Clemens and Jacobs, 1991] [Lamdan and Wolfson, 1988]. Le principe est de mettre en correspondance, pour chaque modèle, un minimum de points 3D avec des points image de manière à prédire la position du modèle dans la scène observée. Puis, chaque hypothèse est vérifiée en projetant le modèle sur l'image.

Ceci peut être fait par des techniques classiques qui n'utilisent pas d'index pour limiter la combinatoire lors de l'appariement entre groupes de points modèle et image. Dans ce cas, le coût est au pire de l'ordre de $M^G N^G$ avec M le nombre de points modèle, N le nombre de points image et G la taille du groupe. Pour les techniques qui utilisent les tables de hachage, Clemens et Jacobs ont prouvé qu'il faut obligatoirement utiliser, comme clefs, des groupes de quatre points au minimum. Les modèles sont représentés par des ensembles de groupes d'au minimum quatre points qui vont former les entrées dans la table de hachage. Les points d'un groupe doivent être coplanaires pour définir un invariant observable dans l'image et dans le modèle. Par contre, si les quatre points du modèle ne sont pas coplanaires, il n'existe pas d'invariant permettant de les indéxer. L'utilisation de groupes de points est donc limitée aux seuls objets possédants des groupes d'au moins quatre points coplanaires.

2.2.2 Les chaînes

Stein et Médioni [Stein and Médioni, 1990] proposent la reconnaissance d'objets observés par un capteur laser. Les modèles et les images des scènes sont 3D et sont décrits par les mêmes primitives.
D'une part, les discontinuités de profondeur et de surfaces sont des courbes 3D représentées par des chaînes de segments. Chaque chaîne de segments est décrite par ses courbures et ses torsions, comme l'illustre la figure 2.6. D'autre part, les surfaces sont

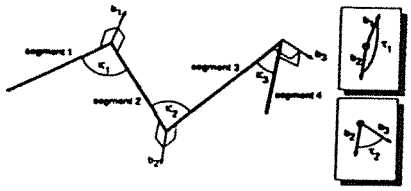


Figure 2.6. *Une chaîne de segments* 3D *:* κ_i *l'angle entre* $segment_i$ *et* $segment_{i+1}$, τ_i *l'angle entre la normale à* $segment_i$ *et* $segment_{i+1}$ *et la normale à* $segment_{i+1}$ *et* $segment_{i+2}$ *[Stein and Médioni, 1990].*

représentées par un ensemble de point 3D sur lesquels sont mesurées des coubes géodésiques. Ces courbes sont représentées de la même manière que les chaînes de segments (Figure 2.7). Stein et Médioni proposent donc une table de hachage mixte pour les courbes géodésiques et les discontinuités. Pour parer à l'instabilité de l'approximation polygonale, les chaînes de segments et les courbes géodésiques sont décomposées en sous-groupes. Par exemple, une chaîne de huit segments est décomposée en un en-

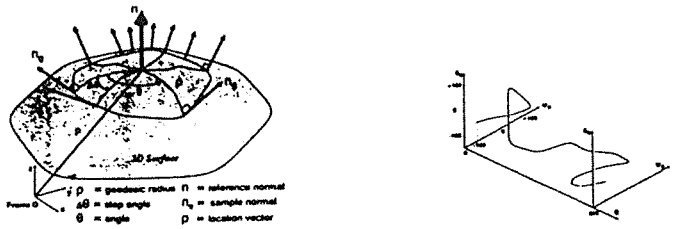


Figure 2.7. *(a)* Une courbe géodésique, *(b)* sa représentation [Stein and Médioni, 1990].

semble de deux chaînes de huit segments (une pour chaque direction), quatre chaînes de sept segments, six chaînes de six segments, etc..

Stein et Médioni ont le mérite de signaler que le coût de l'algorithme est élevé pour des scènes complexes ; c'est-à-dire des scènes comportant énormémant de discontinuités et de courbes géodésiques et dont les objets à identifier sont similaires. L'exemple qu'ils choisissent pour illustrer ce problème est présenté en figure 2.8.

Figure 2.8. *Une scène complexe à analyser [Stein and Médioni, 1990].*

2.2.3 Les graphes

Sossa et Horaud [Sossa and Horaud, 1992] utilisent une table de hachage pour indéxer la base de connaissance. Les objets sont décrits par leur *vues caractéristiques* 2D (CV). Ces vues caractéristiques sont modélisées sous la forme de graphes relationels et également décomposées en *figures fondamentales* (FF) qui sont des sous-graphes des vues caractéristiques. Les figures fondammentales se recouvrent partiellement. Ces recouvrements sont appelés des *intersections de figures fondammentales* (FFI). Chaque figure fondamentale ou intersection de figures fondamentales est décrite par le vecteur des coefficients $(c_0, \ldots, c_n)$ de sa forme $d_2$polynômiale. Cette représentation est utilisée comme clef pour une table de hachage des vues caractéristiques.
Compte tenu de la complexité du calcul des $d_2$polynômes, seuls les coeffiecients c_0 et c_{n-1} sont utilisés. La clef c_0 partionne l'ensemble des graphes FF et des FFI suivant le nombre de noeuds. Puis, au sein de chacune de ces classes, le coefficient c_{n-1} efffectue la distinction entre les graphes. La figure 2.9 illustre la table de hachage à deux niveaux. La reconnaisance se fait en décrivant l'image par ses figures fondamentales et les intersections de figures fondamentales associées.

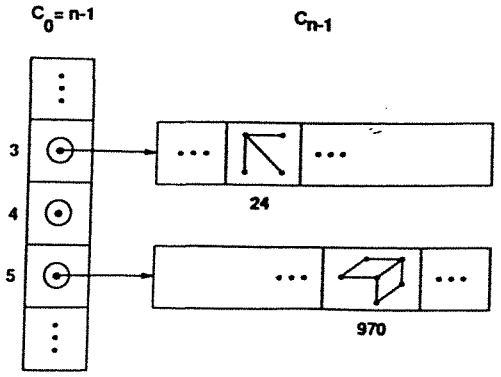


Figure 2.9. *La table de hachage à deux niveaux [Sossa and Horaud, 1992].*

Le fait d'utiliser une table de hachage permet de répondre favorablement au critère d'incrémentalité. La généralisation est faible mais les indices observés sont évolués puisqu'ils représentent des groupes de primitives. Malheureusement, les primitives utilisées sont les segments 2D qui ne sont pas stable à l'extraction.

2.2.4 Synthèse sur les tables de hachage

Mise à jour	Généralisation	Robustesse	Complexité
Correcte	Faible	Faible	Correcte

Contrairement aux classifications, les tables de hachage permettent l'incrémentalité. Mais, leur utilisation reste limitée par plusieurs problèmes :

- Comment tenir compte du bruit qui perturbe l'acquisition des caractéristiques image formant les clefs de hachage ?
- Comment sélectionner les indices et les primitives les plus pertinents ?
  Les indices et les primitives pertinents pour décrire un objet ne le sont pas forcément pour un autre.

En fait les critères de généralisation et de robutesse ne sont pas pris en compte. Nous proposons donc d'observer des index construits sur des indices plus descriptifs.

2.3 Les graphes de décision

Un graphe de décision représente les concepts qui caractérisent un ensemble de données. Ces concepts sont ordonnés de manière à former une classification hiérarchique. Les graphes de décision formalisent les connaissances d'une façon fort similaire à celles des classifications conceptuelles.

Parmi les techniques de construction, nous allons décrire les techniques par induction, les techniques pour les graphes de reconnaissance et les techniques de construction régies par un ensemble de règles.

2.3.1 La construction par induction

La méthode d'induction, proposée par [Quinlan, 1986], construit des arbres de décision à partir de jeux d'essai. Par exemple, la figure 2.10 décrit un jeu d'essai de 14

No.	Attributes				Class
	Outlook	Temperature	Humidity	Windy	
1	sunny	hot	high	false	N
2	sunny	hot	high	true	N
3	overcast	hot	high	false	P
4	rain	mild	high	false	P
5	rain	cool	normal	false	P
6	rain	cool	normal	true	N
7	overcast	cool	normal	true	P
8	sunny	mild	high	false	N
9	sunny	cool	normal	false	P
10	rain	mild	normal	false	P
11	sunny	mild	normal	true	P
12	overcast	mild	high	true	P
13	overcast	hot	normal	false	P
14	rain	mild	high	true	N

Figure 2.10. *Un ensemble d'entrainement [Quinlan, 1986].*

éléments décrits par quatre attributs (Environemet, Température Humidité et Vent) et par la classe d'appartenance (N et P). L'arbre de décision qui en découle est décrit par la figure 2.11. Les noeuds sont des groupes où se mélangent les éléments de classe

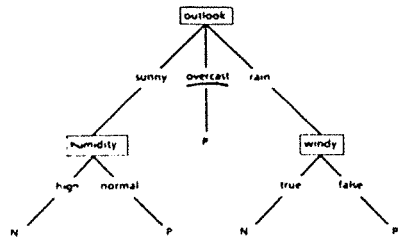


Figure 2.11. *Le graphe de décision qui correspond à l'ensemble de formation de la figure 2.10 [Quinlan, 1986].*

N avec ceux de classe P. A chaque noeud est associé l'attribut qui permet de scinder son groupe d'éléments en plusieurs groupes. Chaque arc sortant de ce noeud indique la valeur de l'attribut qui est choisie par les éléments du noeud fils. Ce processus est itéré jusqu'aux feuilles de l'arbre qui sont des groupes d'éléments de même classe (N ou P). L'objectif est donc de construire l'arbre le plus concis, ce que veut dire que le processus d'induction doit choisir pour chaque niveau le meilleur attribut de façon à limiter la taille de l'arbre.

ID3 [Quinlan, 1986] est un processus de construction par induction. Pour assurer un minimum de calculs, ID3 sélectionne aléatoirement un sous-ensemble du jeu d'essai, construit l'arbre associé et le teste avec le reste du jeu d'essai. Si toutes les données sont rangées dans l'arbre celui-ci est considéré comme correct. Sinon ID3 reprend un échantillon dans le jeu d'essai.
Le point capital de ce processus est la formation de l'arbre à partir de l'échantillonnage du jeu d'essai. En fait, il s'agit d'avoir un test qui permet pour chaque classe de sélectionner l'attribut le plus adéquat à converger vers les classes N et P. Ce test est une mesure d'entropie utilisée en théorie de l'information. En utilisant le jeu d'essai des éléments de classes N et P, ce test est le suivant :

$$\begin{cases} gain(A) & = & I(p,n) - E(A) \\ E(A) & = & \sum_{i=1}^{v} \frac{p_i+n_i}{p+n} I(p_i,n_i) \\ I(p,n) & = & -\frac{p}{p+n}\log_2\frac{p}{p+n} - \frac{n}{p+n}\log_2\frac{n}{p+n} \end{cases}$$

Dans cette formulation du test, il est supposé qu'il y ait en tout et pour tout deux classes P et N avec, respectivement, p et n éléments dans le jeu d'essai choisi. $I(p,n)$ est la mesure d'entropie qui croît quand la proportion $\frac{p}{n}$ tend vers 1. A est un attribut de domaine de valeurs $\{A_1,\ldots,A_v\}$. $E(A)$ est la mesure pondérée de l'entropie de l'attribut A suivant ces différentes valeurs : pour chaque valeur A_i il y a p_i éléments de classe P et n_i éléments de classe N. $gain(A)$ est le score associé à l'attribut A pour la classification des $n+p$ données du jeu d'essai.

Les graphes de décision peuvent également être représentés par des réseaux de neurones. Kononenko [Kononenko, 1989] utilise un reseau où tous les neurones sont connexes. Lors de la phase d'entrainement, les poids, valuant les synapses (les connexions entre les neuronnes) sont mis à jour suivant la formule de Bayes.

Ces techniques de contruction par induction permettent de tenir compte du bruit et des données incomplètes. Mais, tout comme pour les classifications, les attributs et les classes définies par un attribut discriminant ne permettent pas d'identifier de manière correcte les objets dans le cadre de la vision robotique.

2.3.2 Les graphes de reconnaissance

Drapper [Draper and Riseman, 1990] construit un graphe de reconnaissance de scènes. Ce graphe est à hiérarchies multiples. Chaque niveau est une représentation de la scène modélisée. Par exemple, dans la figure 2.12 le niveau A est une description 2D et le niveau B une représentation 3D. Un niveau est décrit par un ensemble d'états (les rectangles de la figure). Le passage d'un état à un autre se fait par vérification des sources de connaissances (les ellipses avec le symbole V de la figure). Par exemple, vérifier le parallélisme des segments. Le passage d'un niveau à un autre se fait par l'intermédiaire de sources de connaissances (les ellipses avec le symbole G dans la figure). Par exemple, l'analyse des points de fuite fournit des directions 3D à partir de segments 2D.

Ce graphe permet l'analyse de scènes, mais, ne présente pas une analyse en terme d'objets indépendant de leur contexte. Les scènes étudiées sont considérées statiques

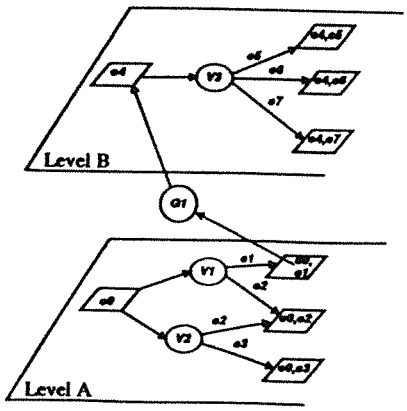


Figure 2.12. *Un exemple de graphe de reconnaissance [Draper and Riseman, 1990].*

ou peu différentes de celles modélisées par ce graphe de reconnaissance. En fait, c'est un outil de segmentation hiérarchique pour extraire des images des primitives évoluées.

2.3.3 Les règles de décision

Burns et Kitchen [Burns and Kitchen, 1987] [Burns and Riseman, 1992] nous proposent un index qui, tout à la fois, sert au regroupement des primitives image provenant d'un même objet et à la sélection des modèles susceptibles de s'apparier à ces groupes image. L'index est un graphe construit à partir de *prédictions* qui sont des conjonctions de relations. Les relations sont décrites par leurs valeurs numériques munies d'intervalles de tolérance. Les entrées du graphe d'indexation sont les relations considérées pertinentes telles que le parallélisme ou la connexité. Les prédictions qui forment le reste du graphe sont des *spécialisations* ou des *combinaisons* de prédictions antérieures. Dans la figure 2.13, la prédiction du carré est une spécialisation de la prédiction du parallélogramme tandis que la prédiction du prisme triangulaire est une combinaison de deux prédictions de parallélogramme avec une de triangle. Les spécialisations et les combinaisons p' d'une prédiction p sont obtenues en deux étapes. Tout d'abord, construire l'ensemble des dérivées de p en utilisant les trois règles suivantes.

1. le nombre d'instances d'objets associées à p' doit être inférieur au nombre d'instances associées à p.
2. chaque instance de p doit être recouverte par une instance de p'.
3. le nombre de prédictions p' attachées à p doit être faible.

Puis, sélectionner les dérivées pertinentes. A partir de l'ensemble des prédictions précédemment construites, les dérivées p' de la prédiction p sont sélectionnées de la manière suivante :

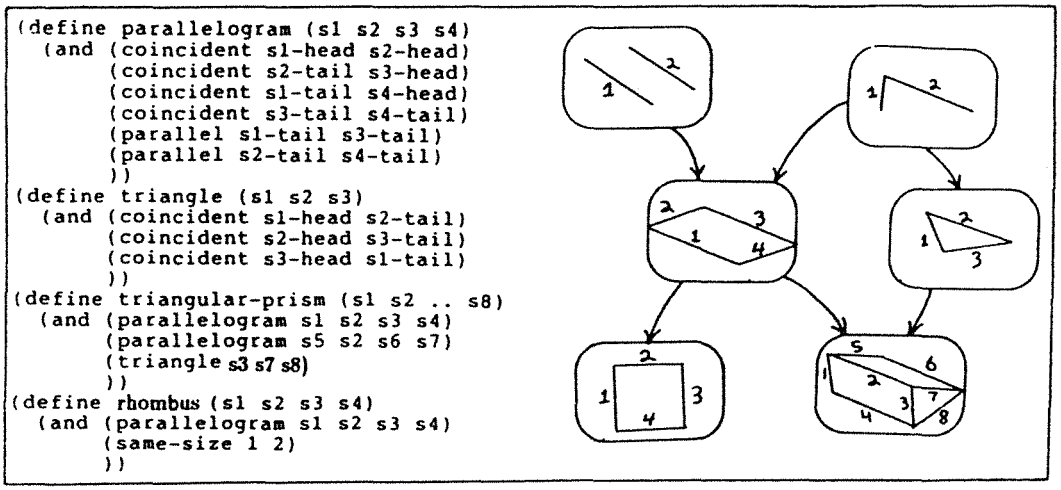


Figure 2.13. *Une spécialisation et une combinaison de prédictions [Burns and Kitchen, 1987].*

1. les futures dérivées p' sélectionnées sont celles qui recouvrent le plus grand nombre d'instances,
2. les dérivées sélectionnées doivent avoir un faible nombre d'instances.

La construction est itérative suivant une priorité bien établie entre les spécialisations et les combinaisons. En effet, de manière à observer des structures plus complexes que les simples relations entre primitives, les combinaisons sont construites avant les spécialisations. Le processus s'arrête quand peu d'instances de p sont laissées non recouvertes et que les dérivées non sélectionnées jusque là possèdent un trop grand nombre d'instances. Un exemple d'indexation est illustré par la figure 2.14.

Evaluation

Mise à jour	Généralisation	Robustesse	Complexité
Nulle	Correcte	Correcte	Moyenne

Le critère de mise à jour n'est pas considéré dans cette étude.

La généralisation est, par contre, le point majeur de cette hiérarchie de prédictions.

La robustesse est assurée en associant aux relations des intervalles de tolérance.

La complexité est contrôlée à l'aide d'un seuil sur le nombre de dérivées que peut avoir une prédiction.

Le choix de créer les combinaisons avant les spécialisations peut être contesté car certaines primitives image peuvent être manquantes lors de la reconnaissance. Il semble plus judicieux de vérifier une prédiction, c'est-à-dire créer ses spécialisations, avant de la combiner avec d'autres.

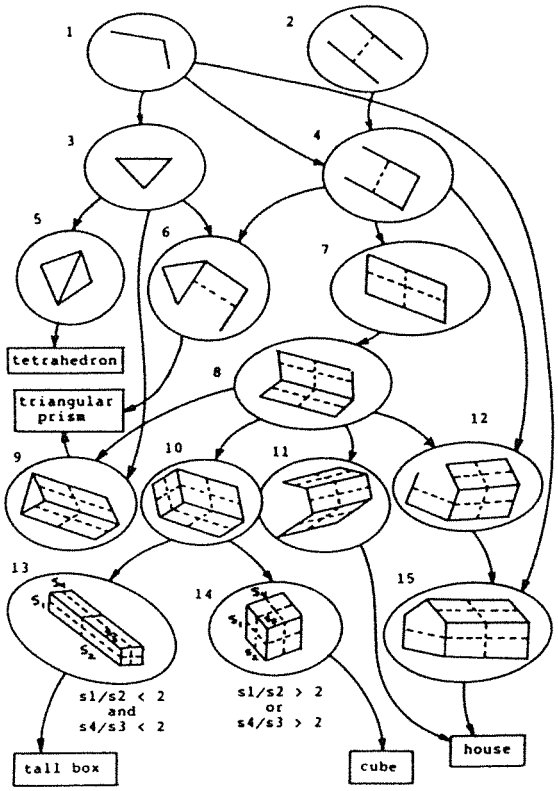


Figure 2.14. *Un exemple de graphe d'indexation [Burns and Kitchen, 1987].*

Le principal intérêt de cette technique est qu'elle créée un index qui permet de sélectionner les modèles tout en servant à la formation de primitives plus évoluées.

2.3.4 En résumé

	Mise à jour	Généralisation	Robustesse	Complexité
Classifications numériques	Nulle	Correcte	???	Elevée
Classifications conceptuelles	Faible	Correcte	???	Elevée
Tables de hachage	Correcte	Faible	Faible	Faible
Graphes de décision	Nulle	Correcte	Correcte	Moyenne

Les classifications numériques et conceptuelles sont lourdes à gérer car elles ne sont pas parfaitement incrémentielles. Le fait de modifier et principalement de remettre en cause des classes déjà existantes entraîne l'utilisation d'heuristiques qui empêchent d'être sûr que la classification incrémentielle d'un ensemble de données fournira le même résultat que la classification classique du même ensemble.

Quant à elles, les tables de hachages sont incrémentielles et leur réorganisation, à chaque introduction d'un nouvel objet, est donc efficace. Par contre, l'extraction de l'information image utile pour créer les clefs d'accès n'est pas garantie. Et la généralisation semble assez limitées par le nombre de clefs différentes qui peuvent être utilisées

pour décrire un objet.

Parmi les graphes de décision, les techniques de construction par induction ne sont pas adaptées. L'utilisation d'attributs indépendants est trop limitatif; la forme d'un objet ne peut être complètement décrite de cette manière.
Le graphe de reconnaissance ne permet pas de modèliser les objets mais plutôt sert de guide pour la phase d'organisation : Drapper a défini un modèle pour le regroupement perceptuel. Cependant, celui-ci semble être adapté à un type de scène (des scènes d'extérieur) et il n'est pas dit qu'il puisse être utilisé en d'autres circonstances.
Burns et Kitchen ont tenté de répondre, à la fois, aux deux problèmes d'organisation et de sélection. Ce travail est intéressant puisqu'il permet, à partir d'indices binaires (les relations de parallélisme et de connexité entre deux segments 2D), de construire des groupes de primitives tout en sélectionnant, au fur et à mesure, les modèles correspondants. Mais nombre de paramètres semble être laissés au choix de l'utilisateur.

La structure des objets est une information importante pour organiser une base de modèles d'objets. Mais la structure d'un objet est complexe car elle définie tout aussi bien les relations entre primitives que les caractéristiques intrinsèques des primitives et des relations. Aussi, pensons nous qu'il faut construire un index mixte : une hiérarchie *symbolique* fondée sur les relations entre les primitives combinée à des hiérarchies *numériques* reposant sur les caractéristiques intrinsèques. Nous utilisons une configuration de la partie symbolique qui est proche de celle utilisée par Burns et Kitchen. Nous nous attachons à permettre une mise à jour incrémentielle lors de l'indexation des modèles. Et contrairement à Burns et Kitchen, nous créons les spécialisations avant les combinaisons de façon à vérifier la validité de toute l'information structurelle d'un groupe de primitives avant de le fusionner avec d'autres.

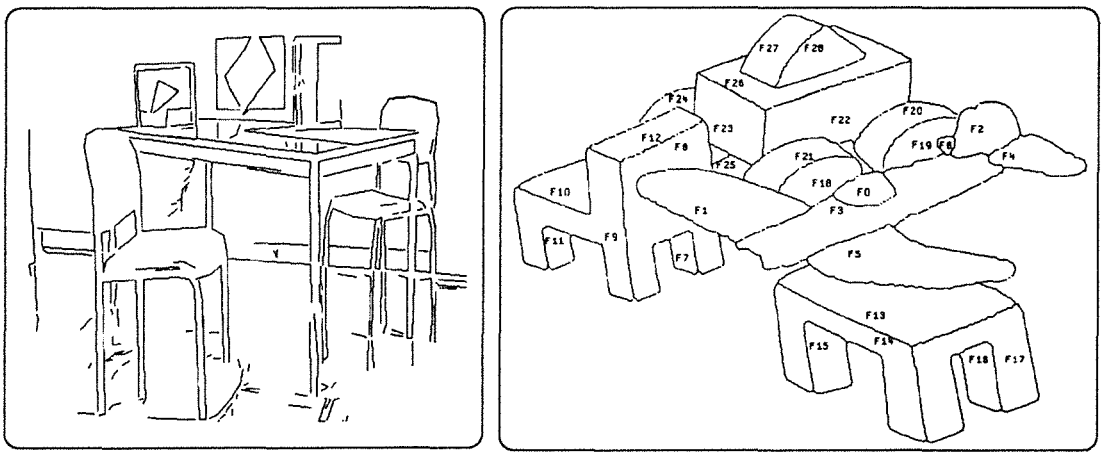
Le chapitre suivant est consacré au processus d'indexation et au graphe d'indexation qu'il fournit. Dans ce mémoire, nous ne développons pas la partie numérique de l'index. Nous présentons sa structure générale en conclusion.

3
L'indexation

Avant de décrire le processus d'indexation, ce chapitre présente brièvement la structure du système de vision MAGIC. Ceci nous permettera de définir correctement le type d'images utilisées et la place de l'index au sein du système. Puis, la section suivante introduira de manière intuitive le processus d'indexation et l'index qui en résulte. Ensuite, nous détaillerons les processus de filtrage et d'indexation.

3.1 La structure de MAGIC

Ce système a pour objectif la compréhension de scènes. Les scènes étudiées sont d'intérieur et constituées d'objets manufacturés. La figure (3.1) présente les deux types



(a) (b)

Figure 3.1. *(a) La segmentation en segment* 2D *d'une image saisie par caméra. (b) La segmentation en faces* 3D *d'une image* 3D *saisie par capteur laser.*

d'image que nous utilisons: *(a)* une description en segments 2D d'une acquisition caméra; *(b)* une description en faces d'une image 3D.

Dans la suite de cette section, nous présentons chacune des parties composant le système MAGIC, illustré par la figure (3.2), en omettant la partie apprentissage. Nous commençons donc par une présentation des données utilisées ainsi que les capteurs

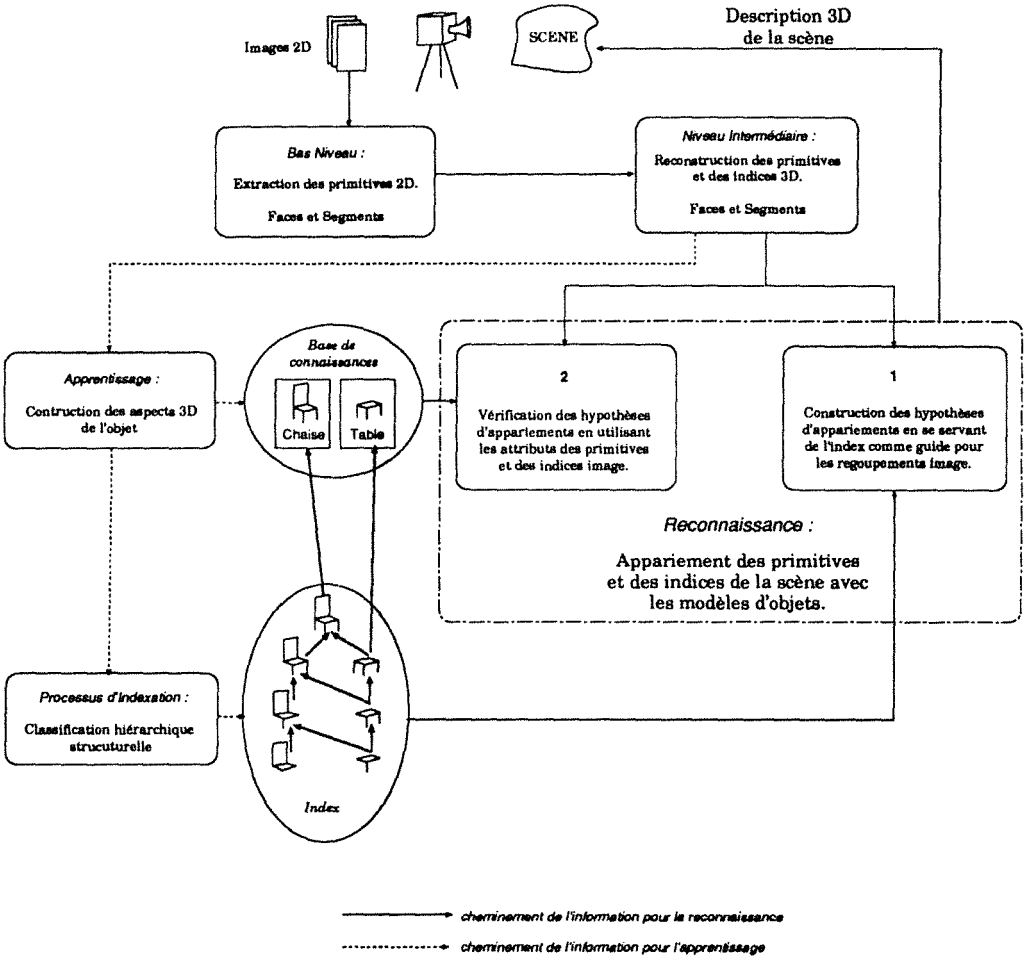


Figure 3.2. *La structure générale du système* MAGIC.

qui permettent leur acquisition. Les deux sections suivantes décrivent le *bas niveau* et le *niveau intermédiaire* qui composent l'étape d'extraction des primitives et des indices image. Puis nous présentons dans l'ordre : la base de connaissance, l'index avec le processus d'indexation et la reconnaissance. En conclusion, nous exposons notre contribution à l'élaboration de ce système.

3.1.1 Les données et leur acquisition

Les données décrivent les scènes observées. Elles dépendent du type de capteurs utilisés qui peuvent fournir des informations de type soit 2D soit 3D [Besl and Jain, 1985] [Faugeras, 1988].

L'information 2D

L'information 2D provient d'une image acquise par caméra. La projection de l'information visuelle de la scène sur le film de la caméra est une *projection perspective* déformée par les pertes dues à la traversée des différentes lentilles. Pourtant, la projection image est souvent modélisée par une projection perspective parfaite, modèle sténopé, comme l'illustre la figure (3.3). La mesure f indique la *distance focale* ; c'est-

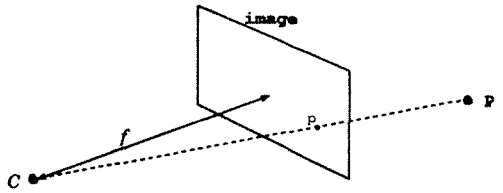


Figure 3.3. *Le sténopé d'une caméra.*

à-dire la distance entre le *centre optique* (C), et le plan image.

Pour être utilisée par un ordinateur, cette image est digitalisée, c'est-à-dire traduite sous forme d'une matrice d'intensité lumineuse dont l'unité est le *pixel*. Malheureusement, cette digitalisation ne fait qu'aggraver les pertes et les détériorations de l'information.

L'information 3D

Elle peut être acquise de deux manières :

- *Les capteurs passifs :* Un système de caméras fournit un ensemble d'images 2D. Cet ensemble est obtenu par acquisitions simultanées à partir de points de vue légèrement différents d'une même scène. C'est le principe de la *stéréovision* [Ballard and Brown, 1982] [Wrobel-Dautcourt, 1988] (cf. Fig. 0.1 à la page 6). Les coordonnées 3D d'un point P de la scène sont obtenues en calculant l'intersection des demi-droites $C1p1$, $C2p2$ et $C3p3$ où $p1$, $p2$ et $p3$ sont les images du point P et $C1$, $C2$ et $C3$ sont les centres optiques des caméras (cf. Fig. 3.4).

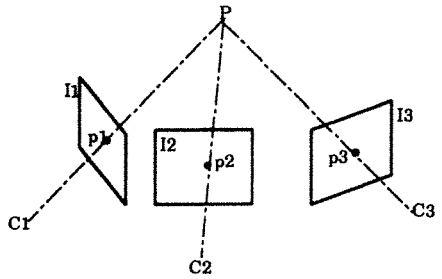


Figure 3.4. *Le principe de projection inverse.*

Une autre façon de procéder est d'observer les mouvements des objets de la scène ou le déplacement du système de caméras. Les données sont alors des sé-

quences temporelles d'images 2D. L'analyse du mouvement se fait suivant deux approches: le *flot optique* et le *mouvement discret par appariement de primitives*. Le flot optique est l'analyse du champ des vitesses d'une image à une autre en supposant que l'intensité lumineuse des points physiques est constante: $(I_t(x,y) = I_{t+\Delta t}(x+dx, y+dy))$. Cette contrainte d'invariance de l'intensité lumineuse des points physiques impose que la séquence d'images soit dense. Le flot optique est utilisé de manière qualitative, par exemple pour la détection des discontinuités de mouvements. La seconde approche extrait les primitives des images de la séquence, puis, elle les apparie. Ensuite, le mouvement 3D et la structure des objets de la scène sont calculés [Faugeras, 1988] [Bouthemy, 1988].

L'hypothèse d'invariance de la réfraction lumineuse d'un point physique en déplacement est une contrainte forte qui fragilise l'analyse par le flot optique. L'analyse du mouvement discret semble plus robuste car elle repose sur une segmentation en primitives plus globales et donc moins sensibles aux légères variations que les points d'intensité constante. Mais, il faut, tout comme dans le cas de la stéréovision, apparier les primitives qui doivent donc être invariantes dans leurs déplacements.

- *Les capteurs actifs :* Le système d'acquisition est composé d'un émetteur et d'un récepteur. L'information 3D est obtenue par *calcul du temps de vol* ou par *triangulation* [Jarvis, 1983].
  Le principe de calcul du temps de vol correspond à mesurer le temps écoulé entre l'émission et la réception d'un signal (cf. Fig. 3.5). Ce temps écoulé est

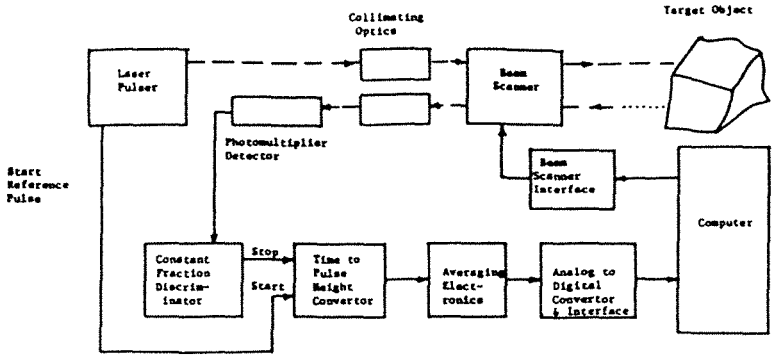


Figure 3.5. *Capteur* 3D *par temps de vol [Jarvis, 1983].*

proportionnel à la distance où se trouve l'objet qui a réfléchi le signal. Les systèmes fonctionnant par triangulation émettent un signal lumineux et observent sa déformation à la réception. Ce signal lumineux peut être un rayon [Fan *et al.*, 1989], une ligne [Smith and Kanade, 1985] ou une grille [Idesawa *et al.*, 1977] [Guisser *et al.*, 1991]. La distance entre les lieux d'émission et de réception étant connue, l'information 3D est reconstruite par triangulation géométrique (cf. Fig. 3.6).

Dans les deux cas, nous obtenons une image 3D, aussi appelée *image de distances*. Nous nommons, également, *pixel*, l'unité de la matrice associée à l'image 3D.

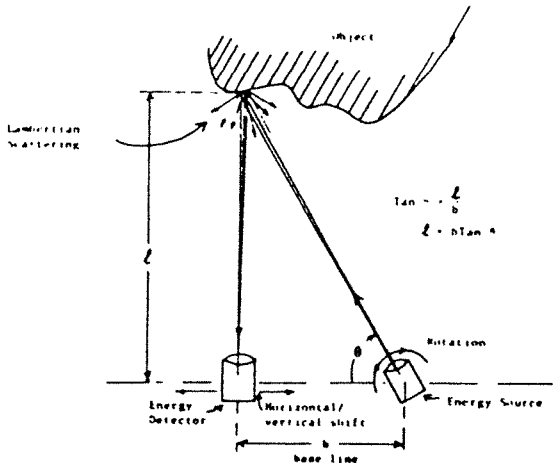


Figure 3.6. *Le principe de triangulation [Jarvis, 1983].*

Mais, l'information n'est pas complète car un signal émis peut être diffracté ou occulté, ce qui rend difficile l'acquisition de scènes fort encombrées. Par contre, la qualité de l'estimation de la profondeur, est actuellement meilleure qu'en utilisant un système de capteurs passifs.

3.1.2 Le bas niveau

Le type d'information diffère si les images sont 2D ou 3D. Des images 2D nous savons extraire les lignes de discontinuités et les zones uniformes en intensité lumineuse. Tandis que des images 3D nous obtenons une information de type surfacique.

La description primaire des images 2D

La description de l'image en termes de pixels ne correspond pas à une représentation des objets qui soit significative. Nous ne savons pas extraire du pixel *l'information sémantique* qui nous permet de nous référer à un modèle d'objet sans ambiguïté. Pour cette raison, nous cherchons à construire des *primitives* 2D plus évoluées. Ces primitives décrivent des caractéristiques propres à identifier les objets à partir de leurs silhouettes, leurs couleurs, leurs textures, etc.. Certaines de ces caractéristiques, comme les zones uniformes en intensité lumineuse sont décelables dans les images par regroupement de pixels voisins d'intensités homogènes. Nous parlons alors de segmentation en *régions*. D'autres caractéristiques, telle que la silhouette des objets, représentent les discontinuités d'intensité des images. Les pixels qui représentent ces discontinuités sont obtenues par une segmentation en *contours*. Ces pixels de contours peuvent être chaînés et convertis en *segments* 2D par une approximation polygonale.

La description primaire des images 3D

Les images 3D sont, également, segmentées en contours et régions [Besl and Jain, 1985]. Les contours sont les chaînes de pixels 3D qui représentent les courbes de discontinuités de surface et de profondeur [Fan, 1988] [Boufama, 1991]. Les régions sont les ensembles de pixels voisins de même type de surfaces. Il existe huit types de surfaces possibles [Besl and Jain, 1985] : *peak surfaces, flat surfaces, pit surfaces, minimal surfaces, ridge surfaces, saddle ridge surfaces, valley surfaces, saddle and valley surfaces.* Toute région est étiquetée par le type de surface de ces pixels.

3.1.3 Le niveau intermédiaire

Bien que le bas niveau ne fournissent pas une description primaire identique pour les images 2D et 3D, le niveau intermédiaire calcule, par des traitement différents, une description intermédiaire standard pour les images 2D et 3D.

Les traitements des images 2D sont plus complexes car il faut extraire des régions et des contours 2D, l'information surfacique de la scène observée pour obtenir une description similaire à la description primaire des images 3D. La stéréovision et l'analyse du mouvement sont les processus utilisés pour fournir cette description. Mais, les erreurs, dues aux bruits dans les images et aux occultations, sont suffisamment importantes pour limiter la qualité de la reconstruction surfacique. Aussi, les contours 2D et les régions 2D sont utilisés simultanément pour fournir une description 2D plus stable telle que les faces 2D [Chabbi and Masini, 1991]. Certains contours, dit occultants, contiennent une information surfacique qui est utilisée en stéréovision et en analyse de mouvement (R. Vaillant 1992).
Ces études sont extrèment complexes et leur intérêt est primordial, car l'acquisition par des capteurs actifs de données qui proviennent de scénes fort encombrées, est très délicate compte tenu du risque élevé de diffractions et d'occultations du signal émis. Le niveau intermédiaire construit des faces 3D [Fan, 1988] [Boufama, 1991] [Chabbi, 1992] à partir d'une description 3D de type surfacique et contour, obtenue par des capteurs passifs ou actifs. Cette description intermédiaire facilite la reconnaissance d'objets modélisés par des volumes, ce qui est couramment utilisé pour les objets manufacturés.

Les descriptions primaires et intermédiaires de notre système à capteurs passifs sont en cours de développement [Berger, 1991] [Chabbi, 1992]. Aussi, afin de mener à terme notre étude sur la reconnaissance et l'indexation, nous utilisons des images 3D obtenues par capteurs actifs. Ces images nous ont été offertes par le laboratoire de Gérard Médioni. La figure (3.7) représente la segmentation en faces [Boufama, 1991] de deux vues 3D, appelées *aspects* 3D d'un même avion.

3.1.4 La base de connaissance

La base de connaissance est une description topologique et structurelle de l'environnement du robot [Weymouth, 1986] [Thirion, 1989]. Cet environnement est un

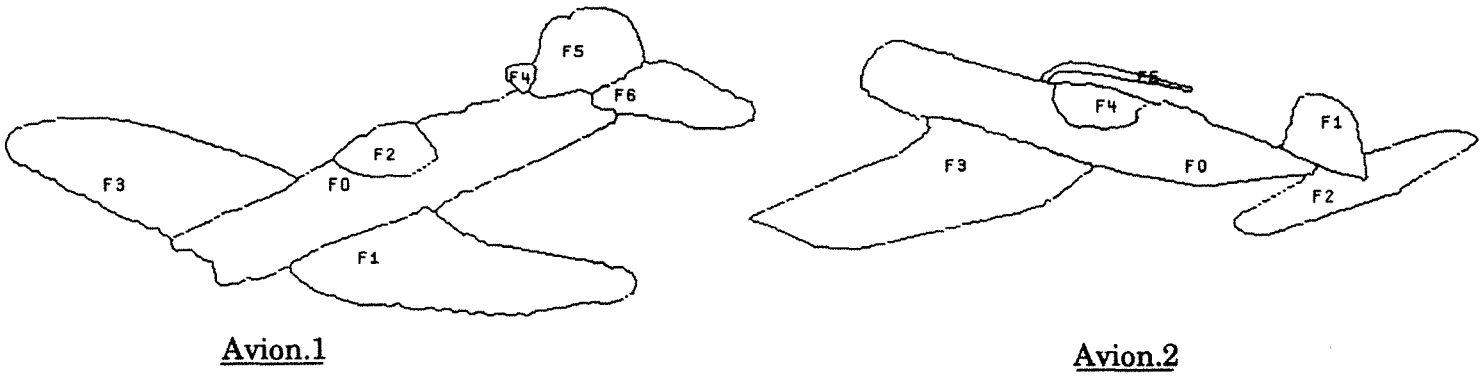


Figure 3.7. *La description de deux vues* 3D *du même avion.*

ensemble de *scènes* tel que des bureaux, des salles consoles, des salles machines, etc.. Ces scènes sont regroupées en *niveau de scènes* pour former des *bâtiments* qui eux-mêmes se regroupent pour former des *plans* [Glaser, 1992]. Les niveaux de scènes décrivent les relations topologiques entre les scènes d'un même étage d'un bâtiment. Les scènes sont décrites par les arrangements et les positions des objets physiques qui les composent ainsi que par leurs structures tels que les murs ou les fenêtres qui ne sont pas considérés comme des objets à proprement parler. Les objets physiques sont décris par des éléments géométriques de base, appelés *primitives*. Les primitives utilisées sont les *segments*, les *faces* et les *volumes* des objets [Thirion, 1989] [Glaser, 1992]. Cette description hiérarchique est illustrée par la figure (3.8).

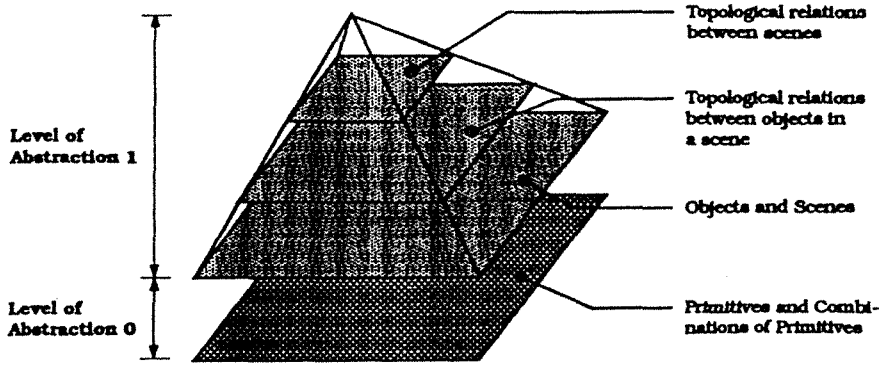


Figure 3.8. *La structure hiérarchique du modèle [Glaser, 1992].*

3.1.5 L'index et l'indexation

La base de connaissance que nous venons de décrire est imposante par sa taille. Il est donc nécessaire de lui adjoindre une architecture qui permet au processus de

reconnaissance de former ses hypothèses d'objets sans à avoir à parcourir toute la base de connaissance de manière séquentielle. Cette architecture, appelée *index*, se présente sous la forme d'un graphe qui exprime une classification structurelle hiérarchique des modèles d'objets [Paris, 1991]. Les objets modélisés sont des ensembles d'aspects 3D et, en fait, l'indexation se fait à partir de ces aspects 3D. Les entrées du graphe d'indexation sont les structures les plus simples, c'est-à-dire les relations qui existent entre les primitives.

Un de nos objectifs définis en introduction est l'autonomie du système de vision robotique. Aussi avons nous associé à l'index un processus d'indexation automatique; l'organisation de la classification structurelle hiérarchique, lors de l'acquisition d'un nouvel aspect, se fait sans aide extérieure. Ce processus est incrémentiel. Il réorganise l'index pour chaque nouvel aspect en utilisant le graphe d'indexation existant. Par exemple, les aspects *cube* et *arche* de la figure (3.9) sont indéxés comme l'illustre la

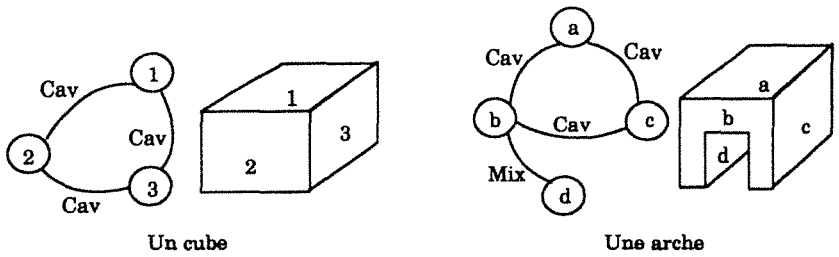


Figure 3.9. *Deux aspects à indexer.*

figure (3.10). Les aspects sont des graphes relationels qui expriment la structure. Les noeuds sont les primitives et les arcs les relations entre ces primitives. Ces relations sont la connexité concave entre deux faces 3D (Cav) et la relation mixte (Mix) qui exprime le lien d'occultation et de connexité à la fois entre deux faces. Par exemple, les faces 3D b et d de l'arche définissent une relation mixte.

Remarque 1 *Ces graphes seront par la suite représentés par des conjonctions d'indices. Le cube est ainsi décrit par* $\{Cav(1,2)\ \&Cav(1,3)\ \&Cav(2,3)\}$ *et l'arche par* $\{Cav(a,b)\ \&Cav(a,c)\ \&Cav(b,c)\ \&Mix(b,d)\}$.

Le processus d'indexation initialise l'index avec le cube (cf. Fig. 3.10). L'entrée du graphe d;indexation représente l'unique relation utilisée pour décrire le cube qui est la connexion concave entre deux faces. L'arc sortant de cette entrée la spécialise en un noeud qui décrit une face connexe de maniére concave à deux autres faces. Cette spécialisation représente tous les sous-graphes comptant trois primitives que l'on peut trouver dans le graphe du cube. Puis en fusionnant ces sous-graphes le processus d'indexation en vient à décrire complètement le cube en un troisième et dernier noeud. Les trois descriptions génériques sont appelées *structure génériques* et chaque sous-graphe du cube s'associant à une de ces structure générique est appelé *élément structurel*. Un noeud du graphe est un *concept*, c'est-à-dire une structure générique munies des éléments structurels qui lui sont isomorphes. Par exemple, le concept formant l'entrée de l'indexation du cube présente comme structure générique $Cav(x,y)$ et

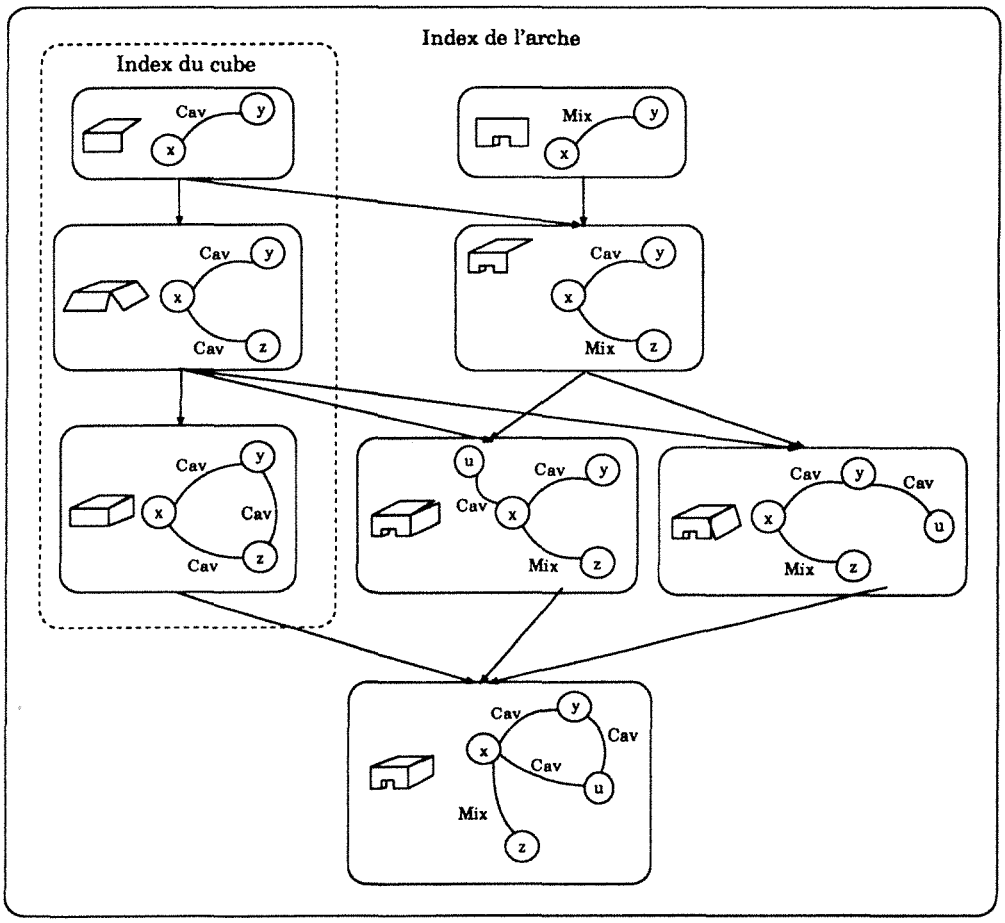


Figure 3.10. *L'index des aspects du cube et de l'arche.*

comme ensemble d'éléments structurels les sous-graphes du cube $Cav(1,2)$, $Cav(1,3)$ et $Cav(2,3)$ isomorphes à la structure générique.

Ensuite, le processus d'indexation enregistre l'arche. Comme le graphe du cube est sous-isomorphe au graphe de l'arche, l'index construit pour le cube est une partie de l'index de l'arche. Le processus a besoin de créer une nouvelle entrée pour la relation mixte. Mais en plus, la structure générique de la nouvelle entrée se fusionne avec celles des concepts créés par l'indexation du cube pour former de nouveaux concepts qui décrivent uniquement des sous-graphes de l'arche.

Le processus d'indexation fonctionne en deux étapes. La première étape crée les entrées nécessaires pour indexer le nouvelle aspect. La seconde étape construit les concepts qui représente tous les éléments structurels (les sous-graphes) de l'aspect indexé. Lors de cette deuxième étape, plusieurs éléments structurels peuvent être représentés par la même structure générique. Aussi, est-il nécessaire de rechercher tous les isomorphismes entre les éléments structurels et les structures génériques déjà créées.

Par la suite, les graphes des structures génériques et des éléments structurels seront représentés par des conjonctions d'indices, aussi, la recherche d'isomorphismes équi-

vaut à rechercher les *substitutions* de variables qui associent les éléments structurels aux structures génériques. Ces substitutions sont recherchées par un processus de filtrage entre un élément structurel et une structure générique en imposant que chaque constante de l'élement structurel (une primitive) intancie une et une seule variable de la structure générique.

3.1.6 La reconnaissance

Ce module permet de juger objectivement de la qualité des modules précédents. Il est composé de deux processus. Le premier utilise l'index pour identifier les objets de la scène en n'ayant que peu de connaissance sur celle-ci. Nous nous y intéressons tout particulièrement et nous le nommons *processus de reconnaissance*. Le deuxième utilise les informations du premier pour en déduire quelle scène est observée et pour en inférer les objets non encore identifiés. Ce processus ne sera pas plus détaillé ici. Mais le lecteur intéressé peut se référer à une étude complète de ce processus, dit *d'interprétation*, et du modèle associé dans [Thirion, 1989] [Glaser, 1992].

3.1.7 Synthèse

Nous venons de décrire la structure du système MAGIC élaboré par l'équipe MOVI. R. Mohr, G. Masini et E. Thirion [Thirion, 1989] ont construit la partie interprétation du module de reconnaissance ainsi que le modèle qui correspond. Actuellement, nous nous focalisons sur les deux thèmes majeurs en vision. Le premier est la construction de primitives évoluées [Chabbi, 1992] [Berger, 1991]. Le second est l'identification et la localisation des objets sans grande connaissance de la scène observée: le processus de reconnaissance utilisant l'index.

Le reste de ce chapitre est consacré au processus d'indexation et à l'index qui en découle.

3.2 Une approche intuitive

Nous allons nous servir d'un exemple simple (cf. Fig. 3.11). Ces aspects sont décrits par des *indices*. Ces indices expriment tous la même *relation* de connexité entre une face et un segment.

3.2.1 Description de l'index

Le processus d'indexation étant incrémentiel, nous considérons que l'aspect $Table.1$ est indexé en premier. L'index en résultant est le *graphe d'indexation* illustré par la figure (3.12). Il est composé de trois concepts dont une entrée. Cette entrée a pour structure générique la relation de connixté $\{Connexe(x, y)\}$. Chaque indice de l'aspect $Table.1$ est un élément structurel de cette entrée. Le lien entre un de ces éléments structurels et la structure générique se fait par une substitution. Par exemple, l'élément

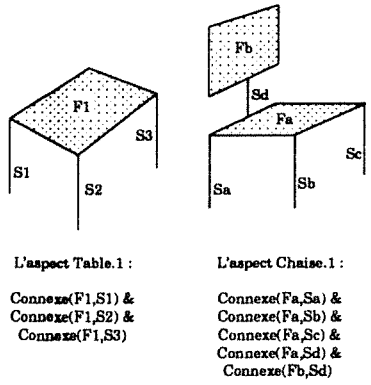


Figure 3.11. *les aspects Table.*1 *et Chaise.*1.

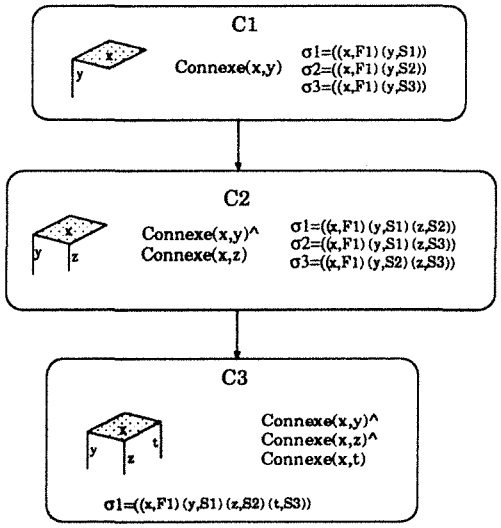


Figure 3.12. *l'index de l'aspect Table.*1.

structurel $\{Connexe(F1, S1)\}$ est filtré par la structure générique $\{Connexe(x, y)\}$ suivant la substitution $\sigma 1 = ((x, F1)(y, S1))$.
Le processus utilise les substitutions pour créer les concepts et les arcs du graphe. Aussi, chaque substitution qui n'a pas encore été utilisée est déclarée *active*. Ainsi dans notre cas, les substitutions $\sigma 1$, $\sigma 2$ et $\sigma 3$ du concept entrée $C1$ sont déclarées actives.
Les substitutions $\sigma 1$ et $\sigma 2$ de l'entrée représentent respectivement les éléments structurels $\{Connexe(F1, S1)\}$ et $\{Connexe(F1, S2)\}$. La fusion de ces deux éléments structurels $\{Connexe(F1, S1)\ \&\ Connexe(F1, S2)\}$ n'est pas une instance de la structure générique de l'entrée. Pour enregistrer ce nouvel élément structurel, le processus d'indexation crée donc un nouveau concept $C2$ de structure générique $\{Connexe(x, y)\ \&\ Connexe(x, z)\}$ et indique le lien avec l'élément structurel par la substitution $\sigma 1 = ((x, F1)\ (y, S1)\ (z, S2))$.
Le processus procède de même pour les fusions deux par deux des autres éléments

structurels associés au concept entrée $C1$. Ces fusions ont toutes pour structure générique celle du concept $C2$ et les trois substitutions de $C2$ les représentent toutes. Une fois toutes ces fusions faites, les substitutions du concept entrée $C1$ sont alors déclarées *désactivées* car elles viennent d'être utilisées. Par contre les nouvelles substitutions du concept $C2$ sont déclarées actives.
Ces nouvelles substitutions servent à créer le concept $C3$. Sa structure générique filtre la description complète de l'aspect *Table*.1 suivant la substitution $\sigma 1$. On dit que le concept $C3$ est *terminal* pour l'aspect *Table*.1.

Puis le processus indexe le deuxième aspect *Chaise*.1. Il va utliser les concepts déjà créés par l'indexation de l'aspect *Table*.1. La figure (3.13) illustre le graphe d'indexa-

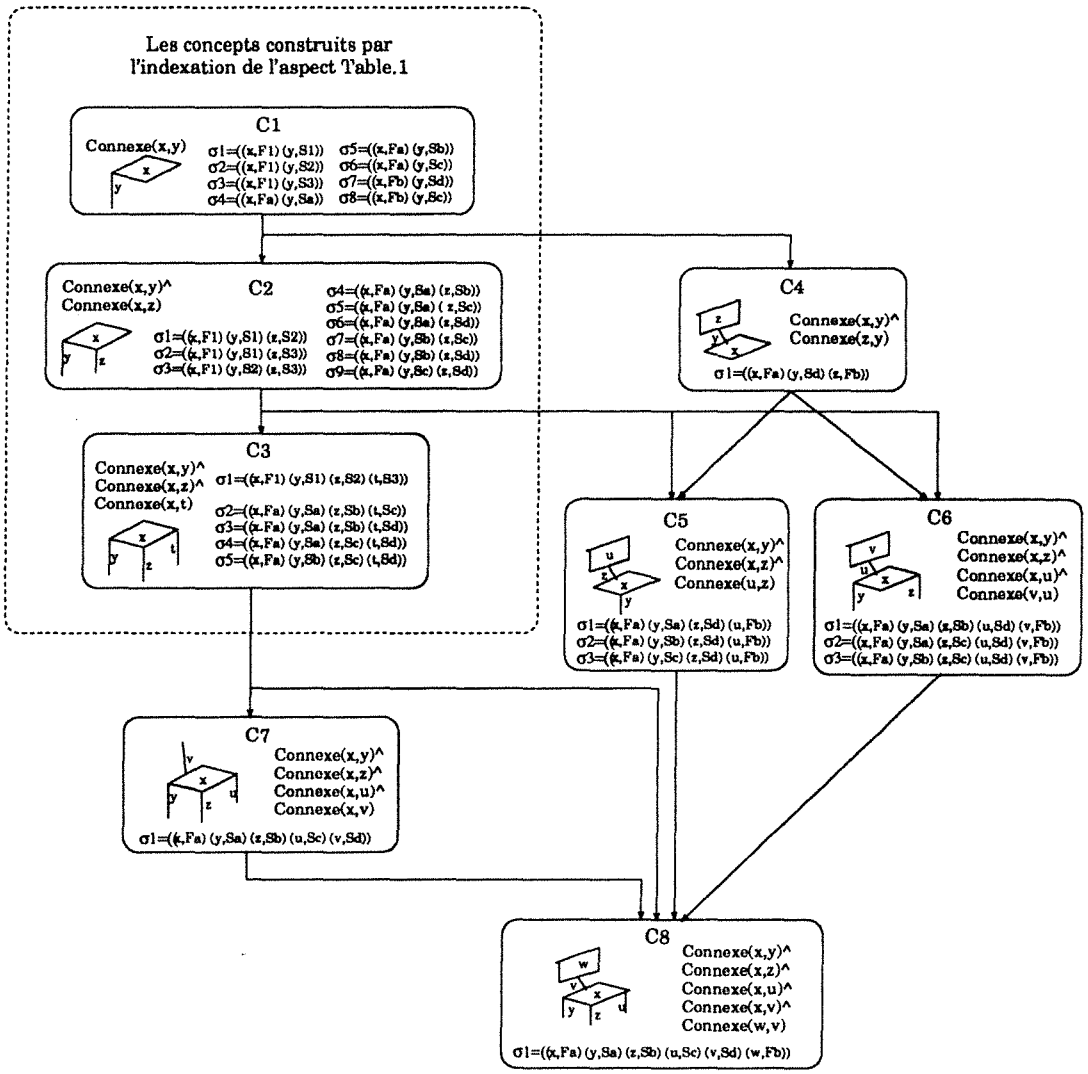


Figure 3.13. *L'indexation de Table*.1 *et Chaise*.1.

tion des aspects *Table*.1 et *Chaise*.1.
Il ne crée pas de nouveau concept entrée car il utilise également la relation de connexité entre une face et un segment. A cette entrée, il ajoute les cinq substitutions $\sigma 4, \ldots, \sigma 8$.

Tout comme précédemment chaque substitution représente un lien entre la structure générique de l'entrée et un indice de l'aspect.
Les différentes fusions possibles entre ces nouvelles substitutions actives définissent de nouveaux éléments structurels. Parmi ceux-ci, certains peuvent être filtrés par la structure générique du concept $C2$. Les substitutions $\sigma 4, \ldots, \sigma 9$ du concept $C2$ sont les liens entre ces éléments structurels et la structure générique de $C2$.
Toutefois, l'élément structurel $\{Connexe(Fa, Sd)\ \&\ Connexe(Fb, Sd)\}$, provenant de la fusion des substitutions $\sigma 5$ et $\sigma 8$ du concept entrée $C1$, n'est filtrée par aucun concept déjà créé par l'indexation de l'aspect $Table.1$. Le processus d'indexation construit alors le concept $C4$ dont la structure générique $\{Connexe(x, y)\ \&\ Connexe(z, y)\}$ filtre l'élément structurel suivant la substitution $\sigma 1 = ((x, Fa)\ (y, Sd)\ (z, Fb))$. Les substitutions $\sigma 4, \ldots, \sigma 8$ du concept $C1$ sont alors déclarées désactivées. Et les substitutions $\sigma 4, \ldots, \sigma 9$ du concept $C2$ et $\sigma 1$ du concept $C4$ sont déclarées actives.
Le processus itère ainsi jusqu'à ce que toutes les substitutions soient déclarées désactivées. Le résultat est un graphe d'indexation où les concepts $C1$, $C2$ et $C3$ présentent des structures génériques communes aux aspects $Table.1$ et $Chaise.1$ et où les concepts $C4, \ldots, C8$ présentent des structure génériques propres à l'aspect $Chaise.1$.

3.2.2 Une présentation de l'algorithme d'indexation

De manière plus formelle, observons la construction incrémentielle et automatique d'un graphe d'indexation. Le processus procède en deux étapes pour chacun des aspects :

Indexation (aspect) ;
Étape 1 : Instanciation des concepts entrée ;
L_sigmas = $\emptyset$; /* *Initaliser la liste des substitutions actives* */
Pour chaque relation R utilisée pour décrire l'aspect *Faire*
Si elle n'existe pas *alors* Construire la structure générique G correspondante ;
Pour chaque indice I exprimant une relation R dans l'aspect *Faire*
Construire la substitution σ représentant le lien entre G et I ;
Ajouter σ à L_sigmas ;
Fin-pour
Fin-Pour
Étape 2 : Construire les différents concepts ;
Tant que L_sigmas $\neq \emptyset$ *Faire*
Créer de nouveaux éléments structurels par fusion des substitutions actives ;
Pour chaque nouvel élément structurel *Faire*
Si un concept déjà existant le filtre *alors*
Construire la substitution correspondante ;
L'ajouter à L_sigmas ;
Sinon
Construire la structure générique et la substitution ;
Ajouter la substitution à L_sigmas ;
Fin-Si

Fin-Pour
Désactiver toutes les substitutions qui viennent d'être utilisées ;
Fin-Tq
Fin-Indexation

Dans cet algorithme général, il n'est pas défini comment sont créés les éléments structurels ni comment est vérifié qu'un concept filtre ou ne filtre pas un élément structurel.
La création d'un élément structurel correspond à la fusion de plusieurs substitutions actives si elles respectent une des trois règles suivantes, dans l'ordre indiqué :

1. *La règle d'Egalité* : Les éléments structurels correspondent à des substitutions actives et expriment des relations sur le même groupe de primitives de l'aspect indexé.

2. *La règle d'InClusion* : L'élement structurel d'une des substitutions actives exprime des relations sur un groupe de primitives qui recouvrent les groupes de primitives des éléments structurels de toutes les autres substitutions actives.

3. *La règle d'InterSection* : Les groupes de primitives des éléments structurels s'intersectent.

Nous allons maintenant observer de plus près le filtrage d'une structure générique vers un élément structurel. Puis, nous définirons plus exactement les règles de fusion des subsitutions actives. Ensuite, l'algorithme d'indexation sera vu plus en détail avec la preuve de la terminaison du processus et l'étude de la conplexité de l'index qu'il génère.

3.3 Le filtrage

Le processus de filtrage vérifie qu'un élément structurel de l'aspect indexé $\mathcal{A}$ est filtrée par l'un des concepts du graphe d'indexation.

Reprenons, par exemple, l'indexation de l'aspect $Chaise.1$ de la section précédente. Le processus va vérifier que la structure générique $G_{C_2} = \{Connexe(x,y)$ & $Connexe(x,z)\}$ du concept $C2$ filtre l'élément structurel $ES_{\sigma 4,\sigma 5} = \{Connexe(Fa,Sa)$ & $Connexe(Fa,Sb)\}$. Cet élément structurel a été engendré par la fusion des substitutions $\sigma 4$ et $\sigma 5$ du concept $C1$.
Le filtrage de l'élément structurel $ES_{\sigma 4,\sigma 5}$ par la structure générique G_{C_2} fournit en résultat la substitution $\sigma 4 = ((x,Fa)\ (y,Sa)\ (z,Sb))$ qui lie l'aspect $Chaise.1$ au concept $C2$.

En règle générale, nous noterons $ES_{\sigma_1,\ldots,\sigma_\ell}$ l'élément structurel provenant de la fusion des substitutions $\{\sigma_1, \ldots, \sigma_\ell\}$ et G_C la structure générique du concept C. Le processus de filtrage s'écrit alors :

$$\sigma G_C = ES_{\sigma_1,\ldots,\sigma_\ell}$$

sous la contrainte:

$$x \neq y \Rightarrow \sigma(x) \neq \sigma(y) \tag{3.1}$$

Cet équation peut être écrite sous la forme d'un système d'équations. Pour chaque équation d système, le terme gauche représente l'ensemble des termes de G_C définis par la même relation R_i et le terme droit l'ensemble des indices de $ES_{\sigma_1,\ldots,\sigma_\ell}$ qui expriment la même relation R_i. En fait, représenter l'équation (3.1) par un système d'équations revient à mettre en correspondance les termes de G_C avec les indices de $ES_{\sigma_1,\ldots,\sigma_\ell}$ qui définissent la même relation. Ceci permet de diminuer la conbinatoire. Vérifier que le nombre de termes est égale au nombre d'indices en chaque équation permet de s'assurer simplement que G_C peut filtrer $ES_{\sigma_1,\ldots,\sigma_\ell}$.

$$\begin{cases} \sigma R_1^1(x_1,\ldots,x_{n_1})\&\ldots\&R_1^{k_1}(x_1,\ldots,x_{n_{k_1}}) & = & R_1^1(p_1,\ldots,p_{n_1})\&\ldots\&R_1^{k_1}(p_1,\ldots,p_{n_{k_1}}) \\ \vdots & & \vdots \\ \sigma \underbrace{R_m^1(x_1,\ldots,x_{n_m})\&\ldots\&R_m^{k_m}(x_1,\ldots,x_{n_{k_m}})}_{G_C} & = & \underbrace{R_m^1(p_1,\ldots,p_{n_m})\&\ldots\&R_m^{k_m}(p_1,\ldots,p_{n_{k_m}})}_{ES_{\sigma_1,\ldots,\sigma_\ell}} \end{cases}$$

Ce système est simple mais reste fortement combinatoire en $O(\sum_{i=1}^{m} k_i!)$. L'utilisation de certaines heuristiques, propres aux contraintes définie par l'équation (??) du système, permet d'en diminuer le coût. l'algorithme que nous présentons utilise les notations suivantes:

σ	=	une des substitions solution de l'équation.
$V(G_C)$	=	$\{x_1 \ldots x_n\}$; l'ensemble des variables inconnues de G_C.
$\mathbb{P}$	=	$\{p_1 \ldots p_n\}$; l'ensemble des constantes de $ES_{\sigma_1,\ldots,\sigma_\ell}$;
	=	les primitives de l'aspect.
$N(x)$	=	l'ensemble des couples (n, R) tel que n relations R utilisent x.
$N(p)$	=	l'ensemble des couples (n, R) tel que n relations R utilisent p.
$v(x)$	=	l'ensemble des valeurs que peut prendre x.

Le processus de filtrage fournit la substitution σ qui indique que G_C filtre $ES_{\sigma_1,\ldots,\sigma_\ell}$. Les ensembles de valeurs $v(x)$ de chaque variable x sont tout d'abord initialisés aux constantes p_j telles que $N(x_i) = N(p_j)$.

- *Si* $v(x_i) = \{p_j\}$ *:* La variable x_i est instanciée par la constante p_j. Cette instanciation est ajoutée à σ. Puis, x_i de $V(G_C)$ et p_j des valeurs possibles pour les autres variables de $V(G_C)$.
- *Si* $(v(x_i) = \emptyset)$ *:* C'est un cas d'échec. σ est instancié à $\emptyset$ et le processus s'arrête car G_C ne peut pas filtrer $ES_{\sigma_1,\ldots,\sigma_\ell}$.
- *Sinon*, plusieurs constantes sont cohérentes et forment l'ensemble inital des valeurs $v(x_i)$ pour chaque variable x_i.

Après cette initialisation de $V(G_C)$ et de σ et s'il n'y a pas eu échec, le processus force l'instanciation de la première variable x_i de $V(G_C)$ avec la première constante de $v(x_i)$. Puis, il vérifie s'il s'agit d'une solution, en procédant de même avec les autres variables de $V(G_C)$. La constitution de la substitution résultat σ se fait donc par une recherche en profondeur de la solution. Une sauvegarde des états de σ et de $V(G_C)$ dans la pile Env est faite à chaque fois que l'instanciation d'une variable a été forcée. Ces sauvegardes sont utilisées pour effectuer les retour-arrières en cas d'échec.

Procédure instancier $(G_C, ES_{\sigma_1,\ldots,\sigma_\ell})$;

```
/* Initialiser σ et v(x_i) pour tout x_i ∈ V(G_C) */
σ ← ∅ ;
Env ← ∅ ;
/* Env est la pile des états correspondant aux différentes solutions possibles
représentées par les couples (V(G_C), σ) */

pour tout x_i ∈ V(G_C) faire
    N(x_i) ← {(n_i, R_j)/∃n_i relations R_j dans G_C utilisant x_i}
fin-pour
pour tout p_i ∈ IP faire
    N(p_i) ← {(n_i, R_j)/∃n_i relations R_j dans ES_{σ1,...,σℓ} utilisant p_i}
fin-pour
pour tout x_i ∈ V(G_C) faire
    v(x_i) = {p_j ∈ IP/N(p_j) = N(x_i)} ;
fin-pour

Mise_à_jour ;
si (σ = ∅) et (V(G_C) = ∅) alors exit ;
/* fin de l'initialisation */

tant que V(G_C) ≠ ∅ faire
    si v(x_i) = ∅ alors /* échec */ σ → ∅ ; exit ;
    /* forcer la variable x_i à prendre la valeur p_j */
    x_i ← premier (V(G_C)) ;
    p_j ← premier (V(x_i)) ;
    v(x_i) ← v(x_i)\{p_j} ;
    Vérifier (σ + (x_i, p_j)) ;
    si (σ = ∅) et (v(x_i) = ∅) alors /* échec */ exit ;
    sinon
        Emplier ((V(G_C), σ), Env) ;
        σ ← σ + (x_i, p_j) ;
        V(G_C) ← V(G_C)\{x_i} ;
        Supprimer (p_j) ; /* peut provoquer un échec */
        si σ = ∅ alors
            (V(G_C), σ) ← Dépiler (Env) ;
        sinon
            Mise_à_jour ; /* suppression des variables de V(G_C)
            dont l'ensemble des constantes contient une seule constante */
            si σ = ∅ alors (V(G_C), σ) ← Dépiler (Env) ;
```

fin-si
fin-si
fin-tq
fin-procédure instancier

La procédure *Mise_à_jour* est activée à chaque instanciation forcée: $\sigma(x_i) = p_j$. Elle supprime toutes les variables $x_k \neq x_i$ de $V(G_C)$ et ajoute à σ le couple (x_k, p_m) quand $x_k \neq x_i$ et $v(x_k = \{p_m\}$.

Procédure Mise_à_jour ;

pour tout $x_i \in V(G_C)$ *faire*
si $v(x_i) = \emptyset$ *alors*
/* *le système est incohérent: rechercher une autre solution* */
$\sigma \leftarrow \emptyset$; /* *échec* */
sinon si $v(x_i) = \{p_j\}$ *alors*
/* p_j *est la valeur de* x_i */
$\sigma \leftarrow \sigma + (x_i, p_j)$;
$V(G_C) \leftarrow V(G_C) \backslash \{x_i\}$;
Supprimer (p_j) ;
si $\sigma = \emptyset$ *alors* /* *échec* */
fin-si
fin-pour
Vérifier (σ) ;
fin-procédure Mise_à_jour

Les procédures auxiliaires sont :

- *Supprimer :*
  suppression d'une constante comme valeur possible pour les variables de $V(G_C)$.
- *Vérifier :*
  vérifier que σ est une solution du filtre de G_C vers $ES_{\sigma_1,\ldots,\sigma_\ell}$. La procédure *Substituer* remplace les variables x par leurs valeurs $\sigma(x)$. Nous affectons une valeur indéterminée $*$ pour les variables inconnues. La valeur $*$ est considérée identique à n'importe quelle autre valeur réelle.
  Si σ est une solution partielle ou complète de l'équation, *Substituer*(σ, G_C) doit être identique à $ES_{\sigma_1,\ldots,\sigma_\ell}$ en vérifiant, pour chaque indice non commutatif de $ES_{\sigma_1,\ldots,\sigma_\ell}$, que l'ordre des constantes est respecté par l'indice de *Substituer* (σ, G_C) correspondant.

Procédure Supprimer (p_j) ;
pour tout $x_i \in V(G_C)$ *faire*
$v(x_i) \leftarrow v(x_i) \backslash \{p_j\}$;
si $v(x_i) = \emptyset$ *alors* $\sigma \leftarrow \emptyset$; /* *échec* */

fin-pour
fin-procédure Supprimer

Procédure Vérifier (σ); *Substituer* $(\sigma, t(C)) = t(\sigma_1, \ldots, \sigma_\ell)$;

La complexité de l'algorithme: Le pire cas de cet algorithme correspond à une initialisation de la solution σ à vide; c'est-à-dire, qu'aucune variable n'est instanciée lors de l'initialisation $V(G_C)$ et donc $V(G_C) = \{x_1, \ldots, x_n\}$ et $\mathbb{P} = \{p_1, \ldots, p_n\}$. Ainsi pour chaque variable x_i, $v(x_i) = \{p_1, \ldots, p_n\}$.

Supposons que nous soyons à la $i^{ème}$ étape dans l'algorithme; c'est-à-dire que nous ayions un début de solution σ formé par l'instanciation des $(i-1)$ premières variables: développons cette $i^{ème}$ étape. Nous associons x_i à p_j (premier $(v(x_i))$ et nous vérifions la cohérence par rapport à σ (procédure *Vérifier*). Nous allons considérer le coût de cette vérification comme constant et de valeur a. Puis en supposant qu'il y ait cohérence, nous sauvegardons l'environnement (procédure *Empiler* dont le coût est =négligeable) avant d'ajouter (x_i, p_j) à σ et de supprimer x_i de $V(G_C)$. Ensuite nous supprimons (procédure *Supprimer*) p_j des valeurs possibles pour les $(n-i)$ variables restant à vérifier. Nous supposons le système toujours cohèrent $(\sigma \neq \emptyset)$ et donc procédons à la mise à jour de σ et $V(G_C)$. La procédure *Mise_à_jour* itère sur l'ensemble des variables de $V(G_C)$ de cardinalité $(n-i)$ à la $i^{ème}$ étape. Le pire cas suppose que pour toute variable x_k de $V(G_C)$, $v(x_k) \neq \emptyset$ et $v(x_k) \neq \{p_j\}$. La complexité de la procédure est donc $(n-i)$.

Le coût de la $i^{ème}$ correspond à la somme des différentes procédure expliquée ci-dessus: $a + (n-i) + (n-i) = 2(n-i) + a$. Aussi, le coût pour les n étape est le produit:

$$\prod_{i=1}^{n}(2(n-i)+a)$$

Mais dans les cas réelles, il est fréquent qu'à l'étape d'initialisation σ ne soit pas vide. Et donc le coût est fortement diminué.

3.4 Le processus d'indexation

Nous venons de décrire le processus de filtrage utilisé pour détecter si un concept C filtre un élément structurel, associé à l'aspect $\mathcal{A}$, provenant de la fusion des substitutions $(\sigma_1, \ldots, \sigma_\ell)$. Nous allons maintenant décrire comment former les concepts et les arcs correspondant à l'aspect $\mathcal{A}$ à indéxer.
Nous savons que le processus itère sur les substitutions actives en les fusionnant suivant les règles d'*Egalité*, d'*InClusion* et d'*Intersection*. Il crée les concepts pour les fusions n'ayant pas été filtrées par un des concepts existant. Cette section décrit en détail les règles utilisées et le processus d'indexation qui en découle. Puis, elle expose

les études faites sur la terminaison du processus et la complexité de l'index obtenu. Des exemples d'indexation conclurons cette section.

3.4.1 Définitions et notations

$\mathcal{A}$	=	un aspect 3D.
$I\!P$	=	$\{p_1, \ldots, p_n\}$; l'ensemble des constantes ;
	=	les primitives de l'aspect $\mathcal{A}$.
V	=	l'ensemble des variables ; $a, b, c, \cdots, x, y, z$.
$\Re$	=	l'ensemble des relations d'arité $n \geq 1$.
$\mathcal{I}$	=	$(\Re \times I\!P^n)$; l'ensemble des indices de l'aspect $\mathcal{A}$.
Ass	$\subset$	$(V \times I\!P)$; l'ensemble des associations (x_i, p_j).
σ	$\subset$	E ; une substitution ; par exemple, $((x_i, p_i)\ (x_j, p_j))$ telle que $x_i \neq x_j \Rightarrow p_i \neq p_j$.
Σ	=	$\mathcal{P}(Ass)$; l'ensemble des parties de Ass ;
	=	l'ensemble des substitutions possibles.
C	=	$(\&_i R_i, \{\sigma_j\}_j)$ où $R_i \in \Re$ et $\sigma_j \in \Sigma$;
	=	un concept est un couple ; le premier élément est une conjonction de relations représentant la structure générique du concept et le deuxième est l'ensemble des substitutions, résultat du filtrage des éléments structurels de l'aspect $\mathcal{A}$ indexé par la structure générale du concept C.
$\mathbb{C}$	$\subset$	$(\mathcal{P}(\Re) \times \mathcal{P}(\Sigma))$; c'est l'ensemble des concepts.
$\mathbb{C}_i$	=	un sous-ensemble de $\mathbb{C}$.
$\mathbb{C}_e$	=	$(R_i, \{\sigma_j\}_j)$; l'ensemble des concepts entrée ; un concept entrée a la particularité d'avoir une structure générique décrite par une seule relation et d'avoir, pour ensemble de substitutions, tous les indices intances de la relation R_i.

Remarque 2 *$\forall C = (\&_i R_i, \{\sigma_i\}_i) \in \mathbb{C}$, nous définissons ces trois notations :*
$\Sigma(C) = \{\sigma_i\}_i$; est l'ensemble des substitutions associées au concept C.
$\Re(C) = \&_i R_i$; est la conjonction des relations R_i exprimant la structure générique du concept.
$V(C) = V(\Sigma(C)) = V(\Re(C))$; est l'ensemble des variables utilisées dans la structure générique de C.

3.4.2 Les règles

Nous allons, dans cette section, décrire les règles utilisées pour l'indexation. Ce sont des règles d'inférence :

$$\frac{condition}{action}$$

La partie *condition* d'une des règles définie comment fusionner les substitutions actives et la partie *action* correspond à la création d'un nouveau concept si l'élément structurel crrespondant à la fusion n'est filtre par aucun concept existant. La partie action des règles, symbolisée par $\rightsquigarrow$, ne sera pas détaillé par la suite car le processus de filtrage vient d'être expliqué et la création d'un concept est assez évidente. Nous allons donc nous focaliser sur la partie action des règles.

Les conditions des trois règles sont fondées sur la notion de fusion d'él'ements structurels. Cette fusion s'opère à partir des substitutions actives directement sans rechercher à constituer les éléments structurels correspondant. En effet, nous avons vu dans lors de l'approche intuitive que les conditions de fusionnement portaient sur les groupes de primitives de l'aspect. Pour observer ces groupes de primitives nous allons utiliser deux fonctions de projection : une projetant une substitution sur son groupe de primitives et une projetant un ensemble de substitutions sur l'ensemble des groupes de primitives.

Définition 1

$$\begin{array}{rccl} Proj: & \Sigma & \rightarrow & \mathcal{P}(I\!P) \\ & \sigma & \mapsto & \{p \in I\!P \ / \ \exists! x \in V \ tq \ (x,p) \in \sigma\} \end{array}$$

Exemple: $\sigma = ((x, f_i)\ (y, t_j)) \Longrightarrow Proj(\sigma) = \{f_i, t_j\}$

Définition 2

$$\begin{array}{rccl} \wp roj: & \mathcal{P}(\Sigma) & \rightarrow & \mathcal{P}(\mathcal{P}(I\!P)) \\ & \Sigma_i & \mapsto & \{Proj(\sigma) \ / \ \sigma \in \Sigma_i\} \end{array}$$

Le résultat de cette fonction de projection est le multiensemble des projections de chacune des substitutions de Σ_i. C'est un multiensemble car deux substitutions peuvent avoir la même projection dans P.

Exemple: L'ensemble des substitutions$\{((x, f_i)\ (y, t_k)), ((u, t_k)\ (v, f_i))\}$, se projette en $\{\{f_i, t_k\}, \{t_k, f_i\}\}$.

Ayant défini ces deux fonctions, nous pouvons décrire la condition de chacune des trois règles suivant qu'elle projette l'ensemble des subtitions en un même ensemble de primitives, que la projection d'une des substitutions recouvrent les projections des autres ou encore que les projections des substitutions s'intersectent.

Après la description de ces règles, nous expliquerons leurs raisons d'être en même temps que nous détaillerons le processus d'indexation. , correspond à la création du concept C et de la substitution σ, sauf s'ils existent déjà !

Remarque 3 *Dans les définitions des règles nous noterons C_i le concept associé à la substitution σ_i. Mais cela ne signifiera pas forcément que deux concepts d'indices distincts seront différents.*

$$\sigma_i \neq \sigma_j \not\Rightarrow \quad C_i \neq C_j$$

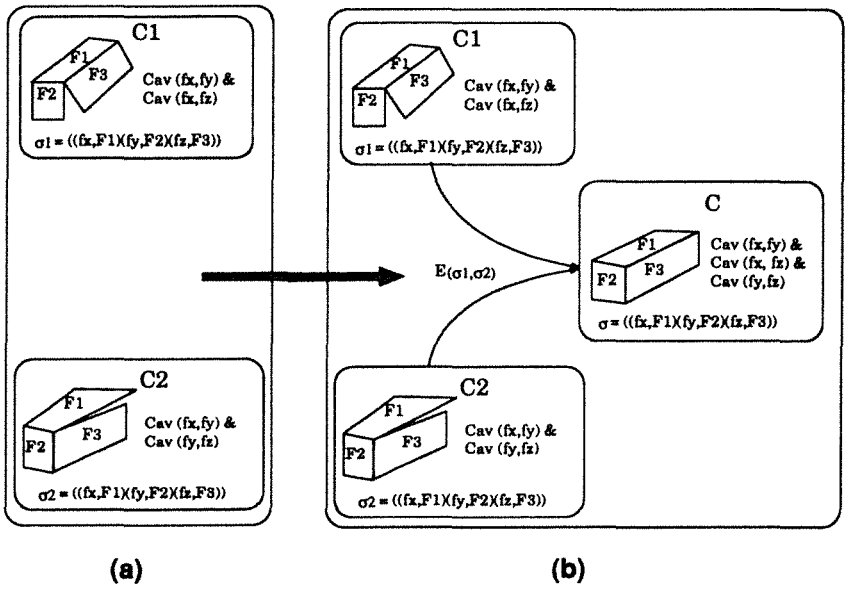
- *Egalité*: (E)

  Etant donné un aspect 3D $\mathcal{A}$, *nous sélectionnons toutes les substitutions associées à* $\mathcal{A}$. *De ces substitutions, on retient celles qui ont la même projection, c'est-à-dire celles qui décrivent un élément structurel associé à un même groupe de primitives de* $\mathcal{A}$. *Ces substitutions proviennent d'un ensemble de concepts ; c'est-à-dire qu'elles sont associées à un ensemble de structures génériques. A partir de ces substitutions et des structures génériques associées, nous construisons un nouveau concept : une nouvelle structure générique et une nouvelle substitution.*

Définition 3

$$\frac{\left[\begin{array}{c} Proj(\sigma_1) = Proj(\sigma_2) = \cdots = Proj(\sigma_k) \\ \forall i \in [1,k], \sigma_i \in \Sigma(C_i) \end{array}\right]}{\left[\begin{array}{c} \{\{C_1, C_2, \cdots, C_k\}\} \\ \sigma_1, \sigma_2, \cdots, \sigma_k \end{array}\right] \rightsquigarrow \left[\begin{array}{c} C\ \sigma\ tel\ que \\ Proj(\sigma) = Proj(\sigma_i) \forall i \in [1,k] \end{array}\right]}$$

Exemple:
Soit un aspect 3D $\mathcal{A}$ lié au concept $C1$ par la substitution $\sigma 1 = ((f_x, F_1)\ (f_y, F_2)\ (f_z, F_3))$ et au concept $C2$ par la substitution $\sigma 2 = ((f_x, F_1)\ (f_y, F_2)\ (f_z, F_3))$, comme illustré en figure (4.5).



(a) Le graphe d'indexation avant l'application de la règle E.
(b) Le graphe d'indexation après.

Figure 3.14. *Exemple d'application de la règle d'égalité.*

Nous constatons que $Proj(\sigma_1) = Proj(\sigma_2) = \{F_1, F_2, F_3\}$. Nous regroupons donc les deux structures génériques de C_1 et C_2 en une seule pour former le nouveau concept C, illustré en figure (4.5.b).

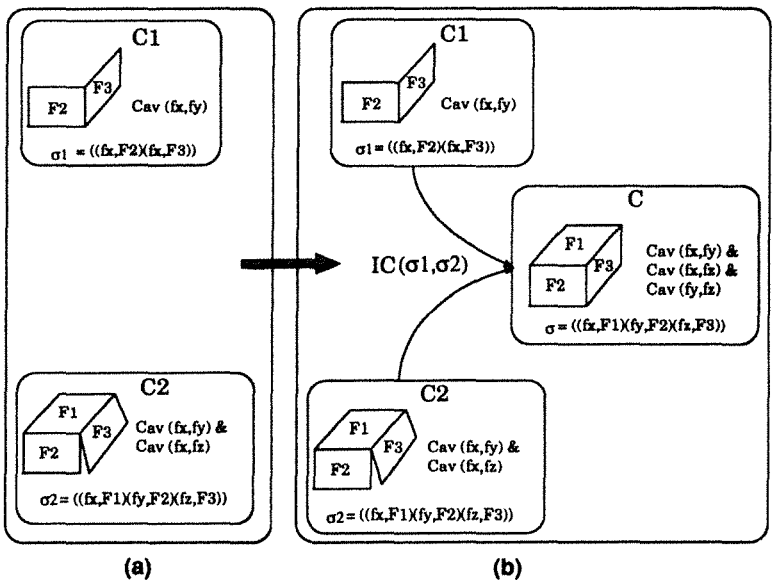
- *InClusion*: (IC)

  Soit un aspect 3D $\mathcal{A}$. *Nous agglomèrons les substitutions associées à* $\mathcal{A}$ *dont les projections sont strictement incluses dans un même groupe de primitives de* $\mathcal{A}$.

Définition 4

$$\frac{\left[\begin{array}{c} Proj(\sigma_i) \subset Proj(\sigma_j) \\ \sigma_i \in \Sigma(C_i), \forall i \in [1..k], j \notin [1..k] \end{array}\right] \quad \& \quad \left[\begin{array}{c} C_i \neq C_j \\ \forall i \in [1..k] \end{array}\right]}{\left[\begin{array}{c} \{\{C_1, \ldots, C_k, C_j\}\} \\ \sigma_1, \ldots, \sigma_k, \sigma_j \end{array}\right] \quad \rightsquigarrow \quad \left[\begin{array}{c} C\ \sigma\ tel\ que \\ Proj(\sigma) = Proj(\sigma_j) \end{array}\right]}$$

Exemple:
La figure (3.15.a) décrit un index composé de deux concepts $C1$ et $C2$ qui repré-



(a) Le graphe d'indexation avant l'application de la règle IC.
(b) Le graphe d'indexation après.

Figure 3.15. *Exemple d'application de la règle d'inclusion.*

sentent respectivement deux faces 3D formant une connexion cancave et deux faces 3D ayant une connexion concave avec une même face 3D. Chacun de ces concepts comprend une substitution l'associant au même aspect $\mathcal{A}$. La projection $Proj(\sigma 1) = \{F_1, F_2\}$ étant incluse dans la projection $Proj(\sigma 2) = \{F_1, F_2, F_3\}$,

la règle autorise le regroupement des deux structures génériques de $C1$ et $C2$ en une seule pour former le concept C, comme illustré en figure (3.15.b).

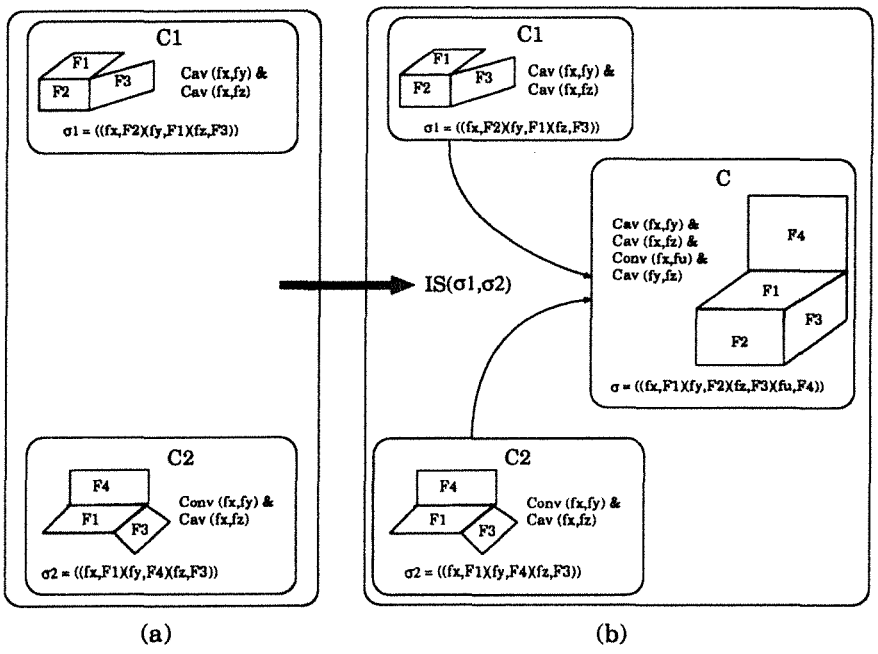
- *InterSection:* (IS)

  Soit un aspect 3D $\mathcal{A}$. *Parmi toutes les substitutions qui lui sont associées, nous regroupons deux par deux celles dont les projections ont des primitives en commun.*

Définition 5

$$\frac{\left[\begin{array}{c} Proj(\sigma_1) \cap Proj(\sigma_2) \neq \emptyset \\ \sigma_i \in \Sigma(C_i), i = 1,2 \end{array}\right]}{\left[\begin{array}{c} C_1, C_2 \\ \sigma_1, \sigma_2 \end{array}\right] \rightsquigarrow \left[\begin{array}{c} C\ \sigma\ tel\ que \\ Proj(\sigma) = Proj(\sigma_1) \cup Proj(\sigma_2) \end{array}\right]}$$

Exemple:
La figure (3.16.a) décrit un index composé de deux concepts $C1$ et $C2$ représen-



(a) Le graphe d'indexation avant l'application de la règle IS.
(b) Le graphe d'indexation après.

Figure 3.16. *Exemple d'application de la règle d'intersection.*

tant respectivement deux faces ayant une connexion concave avec une même face et une face ayant une connexion concave avec une face et une connexion convexe

avec une autre. Chacun de ces concepts comprend une substitution l'associant à un même aspect $\mathcal{A}$. Les projections $Proj(\sigma 1) = \{F_2, F_1, F_3\}$ et $Proj(\sigma 2) = \{F_1, F_4, F_3\}$ s'intersectant, la règle autorise le regroupement des deux structures génériques de $C1$ et $C2$ en une seule pour former le concept C, comme illustré en figure ??.b).

Détaillons maintenant le processus d'indexation utilisant ces règles, ce qui revient à définir le contrôle qui gère l'application de ces règles. En d'autres termes, définir la priorité entre ces règles d'inférence.

3.4.3 Contrôle d'application des règles

Il est fondamental d'appliquer les règles suivant un ordre qui regroupe l'information structurelle de manière cohérente. Nous voulons effectuer le regroupement de tous les éléments structurels définis sur un même groupe de primitives de l'aspect à indexer. Ceci est effectué par la règle d'égalité puisqu'elle opère le regroupement des substitutions dont les projections définissent le même groupe de primitives. Une fois ses regroupements faits, la règle d'inclusion permet d'assembler des éléments structurels dont l'un d'eux est défini sur une zone recouvrant les zones de tous les autres. Puis nous pouvons regrouper l'information structurelle provenant de différentes zones voisines dont l'information structurelle est complète puisque les règles d'égalité et d'inclusion ont été apppiquées auparavant. Nous pouvons donc établir l'ordre de priorité suivant :

$$E > IC > IS$$

Les règles d'égalité et d'inclusion sont d'arité quelconque (le nombre de substitutions formant un groupe pouvant être supérieur ou égale à 2), tandis que la règle d'intersection est binaire (les substitutions sont regroupées deux par deux). Ce choix pour les arités s'explique par le principe désiré pour le fonctionnement du processus de reconnaissance. Le processus de reconnaissance doit absoluement vérifier qu'une hypothèse d'appariement entre un groupe image et un modèle est correcte de la fusionner avec d'autres hypothèses. Mais, dans notre analyse, nous conjecturons que le bruit perturbe les données 3D dans le sens que certaines primitives peuvent être manquantes mais qu'elles ne peuvent pas être fausses. En conclusion de ce mémoire, nous nous proposons de disserter sur les techniques possibles pour prendre en compte l'instabilité des techniques de segmentation.

Nous autorisons les assemblages d'un nombre quelconque de substitutions lors de l'utilisation de la règle d'égalité ou d'inclusion car ces deux règles expriment le regroupement de toutes les informations structurelles provenant d'un même groupe de primitives. Si, lors de la reconnaissance, un groupe image correspond à une des informations structurelles, toutes les informations structurelles de ce groupe doivent être présentes puisque nous supposons les indices robustes à la détection. Par contre, la règle d'intersection représente la fusion de plusieurs groupes de primitives alors que rien ne nous autorise à croire que ces même groupes seront détectés lors de la reconnaissance

puisque des primitives peuvent être manquantes. En effet, lors de la reconnaissance, si une primitive n'a pas été détectée il est possible que la fusion de deux éléments structurels ne soit pas envisagée alors qu'elle l'a été lors de l'indexation. Aussi avons nous opté pour une fusion binaire des groupes pour permettre d'effectuer les fusions au fur et à mesure lors de la reconnaissance.

L'algorithme opère à partir des substitutions en les regroupant suivant les règles décrites précédemment pour créer de nouveaux concepts et de nouvelles substitutions. Pour assurer la terminaison, que nous allons prouver dans la section suivante, nous parlons de substitutions *actives* et *inactives*. Les substitutions actives sont celles utilisées par l'algorithme pour créer les nouveaux concepts et les nouvelles substitutions. Les subtitutions actives ayant été regroupées suivant une des règles sont, pour les itérations à venir, rendues inactives. Si ce groupe a donné lieu à une nouvelle substitution, cette dernière est activée pour les itérations suivantes.

Procédure Indexation $(\mathcal{A})$;
    /* *étape d'initialisation du processus:*
    nous construisons les substitutions associées aux concepts entrée */

    $L = \emptyset$; /* *l'ensemble des substitutions actives* */

    pour tout $C \in \mathcal{C}_e$ *faire*
        /* $\mathcal{C}_e$ *est l'ensemble des concepts entrée* */
        pour tout $I \in \mathcal{A}$ *faire*
            /* *la structure générique de* C *filtre l'indice* I */
            si $C \ll I$ *alors*
                créer σ *tq* $\sigma C = I$;
                $L \leftarrow L + \sigma$;
            fin-si
        fin-pour
    fin-pour
    répéter
        répéter
            $G \leftarrow$ *Egalites* (L);
            /* *former les groupes de substitutions actives suivant*
            la règle d'égalité (partie condition de la règle d'inférence)
            et supprimer dans L *les substitutions formant les groupes de* G */
            Construction (G);
            /* *processus de construction des nouveaux concepts*
            et des nouvelles substitutions */
            $G \leftarrow$ *Inclusions* (L);
            Construction (G);
        jusqu'à $(G = \emptyset)$;
        $G \leftarrow$ *Intersections* (L);
        Construction (G);
    jusqu'à $(G = \emptyset)$
fin procédure Indexation $(\mathcal{A})$

La procédure *Construction* procède suivant trois cas pour chacun des groupes de substitutions g_i de G:

1. *Si l'élément structurel défini par g_i est lié à un concept de l'index et s'il existe au sein de ce concept une substition représentant cet élément structurel.* Cette configuration a déjà été étudiée et elle n'a pas à être examinée à nouveau.

2. *Si l'élément structurel défini par g_i est lié à un concept de l'index mais qu'il n'existe pas de substitution le représentant.* La substitution correspondante est créée.

3. *Si l'élément structurel défini par g_i n'est filtré par aucun concept de l'index courant.* La structure générique et la substitution qui correspondent sont créés pour former un nouveau concept.

Procédure Construction (G);
/* $G = \{g_1, \ldots, g_n\}$ *est l'ensemble des groupes de substitutions formés suivant la condition d'une des règles* */

pour tout $g_i \in G$ *faire*
$F \leftarrow$ *Instanciation*$(\mathcal{C}, g_i)$;
/* *Instanciation utilise la procédure Instancier décrite au paravant et fournit en résultat un couple* $F = (C, \sigma)$ *tel que* $\sigma C = g_i$ */
si $F = nil$ *alors /* pas de concept filtrant* g_i */
créer C *tq* $C \ll g_i$;
créer σ *tq* $\sigma C = g_i$;
$L \leftarrow L + \sigma$;
sinon si $F = (C, nil)$ *alors*
/* *le concept existe mais pas la substitution* */
créer σ *tq* $\sigma C = g_i$;
$L \leftarrow L + \sigma$;
fin-si
/* *si* $F = (C, \sigma)$ *alors nous avons besoin de ne rien faire; ce cas a déjà été étudié* */
fin-pour
fin procédure Construction (G)

Il est primordial, avant toute autre chose, de vérifier la terminaison d'un tel processus.

3.4.4 Terminaison du processus de construction

Ce processus de construction itère sur l'ensemble des substitutions actives pour former de nouveaux concepts et de nouvelles substitutions par application d'une des règles, définies ci-dessus, suivant l'ordre indiqué.

Sans tenir compte du résultat que l'indexation fournie, une itération de ce processus peut être représentée par une fonction. Les arguments de cette fonction sont l'ensemble

des substitutions actives de l'étape courante, et la règle appliquée. Et le résultat est l'ensemble des substitutions actives pour l'étape suivante.

Fonction $\mathcal{F}$: Définissons une fonction schématisant le processus itératif d'indexation.

$$\begin{array}{rccc} \mathcal{F} & : & (\mathcal{X} \times \Sigma) & \rightarrow & \Sigma \\ & & (X, A_i) & \mapsto & A_{i+1} \end{array}$$

avec $\mathcal{X} = \{E, IC, IS\}$, A_i = l'ensemble des substitutions actives à l'étape i ($A_i \subset \Sigma_i$) et A_1 est l'ensemble des substitutions s'appliquant au concept entrée.

Remarque 4 *Soit Σ_i l'ensemble des substitutions actives et inactives existant à l'étape i tel que $\mathcal{F}(X, A_i) = A_i, \forall X \in \mathcal{X}$, nous disons alors que Σ_i est stable.*

Théorème 1 *La hiérarchie structurelle incrémentielle est obtenue par appels récursifs de la fonction $\mathcal{F}$ sur l'ensemble des substitutions actives A_1 jusqu'à l'obtention d'un ensemble A_η tel que Σ_η soit stable : $\exists \eta$ tel que $(\mathcal{F}(X, A_1))^* = A_\eta$*

Preuve : Pour vérifier ce théorème, il faut prouver *l'existence* de Σ_η. Deux possibilités se présentent :

- *Soit $|A_1| = 1$:*

  Il n'existe qu'une seule substitution associée à un unique concept entrée. Le concept entrée, représentant complètement cet aspect, est dit *terminal* pour celui-ci. En d'autre termes, l'aspect 3D, que l'on veut modéliser, est décrit par un seul indice. Aussi aucune règle n'est applicable ; Donc, $\Sigma_{A,e}$ est stable : $\Sigma_\eta = A_1 \diamond$

- *Soit $|A_1| > 1$:*

  L'existence de Σ_η signifie que $\mathcal{F}(X, A_\eta) = A_\eta \ \forall X \in \mathcal{X}$. Il faut donc démontrer qu'à partir d'un certain nombre d'applications de la fonction $\mathcal{F}$ l'ensemble des substitutions actives est obligatoirement réduit et atteint, dans le pire des cas, $|A_\eta| \leq 1$. En effet, si $|A_\eta| \leq 1$, aucune règle ne peut être appliquée. Pour faire cette démonstration nous allons observer, en premier lieu, le comportement de chacune des règles.

  - $\boxed{E}$

    Supposons qu'il y ait $\sigma_{i_1}, \ldots, \sigma_{i_k}$ substitutions dans A_i telles que $\forall j, \ell \in [1..k]$, $Proj(\sigma_{i_j}) = Proj(\sigma_{i_\ell})$.
    Les substitutions $\sigma_{i_1}, \ldots, \sigma_{i_k}$ servent à former une nouvelle substitution σ (et un nouveau concept). Puis, elles sont désactivées. Le cardinal du nouvel ensemble A_{i+1} de substitutions actives vaut donc $(|A_i| - k + 1)$ car les k substitutions $\sigma_1, \ldots, \sigma_k$ sont désactivées et la nouvelle est activée.
    Nous constatons que l'ensemble des substitutions actives ne peut que diminuer puisque la règle d'égalité revient, en fait, à former des classes de substitutions dans A_i et à remplacer chacune de ces classes par une seule substitution dans A_{i+1}.

- $\boxed{IC}$
  De même pour la règle IC, les ensembles se forment à partir des substitutions actives dont les projections sont incluses dans un même groupe de primitives. Alors, la règle IC opère également une classification des usbstitutions actives sur l'ensemble A_i. Le nombre de substitutions actives à l'étape $(i+1)$ est donc inférieur au nombre de susbtitutions actives à l'étape (i).
- $\boxed{IS}$
  Supposons $(\sigma_{i_1}, \sigma_{i_2}) \in A2$ telles que $Proj(\sigma_{i_1}) \cap Proj(\sigma_{i_2}) \neq \emptyset$.
  σ_{i_1} et σ_{i_2} servent à créer une nouvelle substitution et sont désactivée. Mais le nombre de substitutions actives ne décroit pas nécessairement car, dans le pire cas, nous pouvons avoir $\begin{pmatrix} m \\ 2 \end{pmatrix}$ groupes formés dans A_i. Ce qui désactive les m substitutions $\sigma_1, \ldots, \sigma_m$ de A_i, mais active les $\frac{1}{2}m(m+1)$ nouvelles substitutions.
  La question de terminaison, par observation de la diminution du nombre des substitutions actives, se pose donc pour la règle IS. Supposons qu'initialement, nous fixions les projetés des substitutions à deux primitives; c'est-à-dire que nous supposons utiliser des relations binaires. En appliquant la règle IS, les nouvelles substitutions ainsi construites présentent au minimum trois primitives.
  Par exemple, supposons que nous ayions σ_j et σ_ℓ avec $Proj(\sigma_j) = \{p_{j_1}, p_{j_2}\}$ et $Proj(\sigma_\ell) = \{p_{\ell_1}, p_{\ell_2}\}$.
  Si ces deux substitutions respectent les conditions suivantes[1], elles peuvent être fusionner pour créer une nouvelle substitution σ.

  $$Proj(\sigma_j) \cap Proj(\sigma_\ell) \neq \emptyset Proj(\sigma_j) \neq Proj(\sigma_\ell) Proj(\sigma_j) \not\subset Proj(\sigma_\ell) Proj(\sigma_\ell) \not\subset Proj(\sigma_j)$$

  Alors la nouvelle substitution σ est définie telle que:
  $|Proj(\sigma)| = |Proj(\sigma_j) \cup Proj(\sigma_\ell)| > 2$.
  Sachant que l'ensemble $\mathbb{P}$ des primitives est de taille fixe, il est évident que dans le pire des cas, à force de fusionner les groupes de primitives, il n'existera plus qu'une seule substitution portant sur l'ensemble des primitives de $\mathbb{P}$.

Ceci nous permet donc de conclure que le contrôle imposé sur les règles assure la terminaison de l'algorithme. $\diamond$

3.4.5 Complexité

Avant de commenter les résultats d'indexations de différentes bases de modèles et ceux du processus de reconnaissance utilisant ces différentes indexations, il est nécessaire d'estimer de la complexité du graphe d'indexation ; c'est-à-dire la *taille du graphe*

1. Nous supposons les règles E et IC déjà appliquées (l'ordre imposé)

d'indexation obtenu pour un ensemble d'aspects 3D donnés. La taille d'un graphe est définie par son nombre de noeuds (nombre de concepts Nbc), par sa profondeur (P) - car c'est un graphe orienté - et par le degré sortant moyen d'un concept ($\overline{DC}$). Le degré sortant d'un concept représente le nombre d'arcs sortant de ce concept.

Nous allons définir des valeurs de Nbc, P et $\overline{DC}$ dans le pire cas, le meilleur cas et en moyenne, pour l'indexation soit d'un seul aspect, soit de plusieurs. La taille maximale est obtenue en supposant que les substitutions se regroupent par deux et servent, à chaque fois, à créer de nouveaux concepts. Il est évident qu'une telle configuration du graphe d'indexation est complètement aberrante. Chaque concept représenterait un seul aspect par une unique substitution alors que le but de l'indexation est de construire une classification des modèles de la base de connaissance. Cela consiste à former des concepts qui représente des structures génériques à plusieurs aspects, afin de limiter la taille de la zone de recherche lors de la sélection des modéles. Autrement dit, lors de la reconnaissance, le pire cas revient à ne pas construire d'indexation mais à rechercher séquenciellement les modèles adéquats puis à organiser les primitives image suivant les hypothèses d'objet qui peuvent être nombreuses.
Quant au meilleur cas, il est défini par une taille minimale du graphe d'indexation qui est obtenue quand le processus d'indexation ne construit à chaque itération qu'un seul concept. Ceci est possible si nous considèrons les aspects décrits par une unique relation et qu'une conjonction quelconque d'un même nombre d'indices donne toujours lieu à la même structure[2]. Dans ce cas, le graphe d'indexation est également fort peu significatif puisque chacun de ses concepts est muni de substitutions associées à l'ensemble des aspects 3D indéxés. Il n'y a donc pas de classification et l'intérêt de l'indexation est complètement perdu.
Nous constatons donc que les coûts dans le pire cas et dans le meilleur cas donnent naissance à des graphes d'indexation tout à fait inexploitables pour une reconnaissance efficace. Mais, les expériences que nous avons faites, et que nous montrons dans la section suivante, sont loin de se comporter suivant un de ces deux cas. Il nous faut, en fait, avoir une estimation en moyenne pour pouvoir prétendre à une étude objective du processus. Malheureusement les calculs en moyenne ne sont pas réalisables compte-tenu du nombre de paramètres influant la construction du graphe d'indexation. En effet, la taille de l'index est directement liée :

- *au nombre d'éléments structurels contenus dans l'aspect à indéxer* :
  Chaque aspect à sa propre structure. Celle-ci peut-être plus ou moins complexe. Par exemple, l'aspect 3D d'une chaise engendrera beaucoup moins de concepts qu'une vue éclatée 3D d'une boîte de vitesses. Mais il est évident que ce paramètre ne peut-être fonction de quoi que ce soit sinon de l'aspect lui-même.

- *au taux d'appariement entre l'aspect à indéxer et les aspects déjà indéxés* :
  Les aspects ont des structures génériques en commun. Il se peut que certains d'entre eux aient une grande partie de leurs structures communes. Par exemple, la structure de l'aspect 3D *Table*.1 est une partie de la structure de l'aspect 3D *Chaise*.1 (cf. Fig. 3.11). Si nous indexons en premier *Chaise*.1, aucun nouveau

2. Ici, il est sous-entendu que la relation représentant ces indices est commutative.

concept ne sera créé lors de l'indexation de $Table.1$.
Le paramètre, indiquant le taux de nouveaux concepts par rapport au nouvel aspect à indéxer et, aussi, par rapport aux aspects déjà indéxés, ne suit aucune loi.

Aussi nous proposons une étude expérimentale de la complexité en moyenne dans la section suivante. Mais, tout d'abord, étudions le comportement dans le pire cas et le meilleur cas.

La complexité d'indexation d'un seul aspect.

Soit un aspect modélisé comportant n indices. Le meilleur cas représente la taille minimum du graphe d'indexation tandis que le pire cas exprime la borne supérieure de cette taille.

Le meilleur cas : Le meilleur cas correspond à $\boxed{Nbc = P = n \text{ et } \overline{DC} = 1}$. Ceci est possible quand les n indices, qui décrivent l'aspect, proviennent d'une même relation commutative. En effet, l'index aura une seule entrée représentant tous les indices. Puis les indices fusionnées deux à deux par la règle d'intersection et ces fusions sont représentées par un même concept. Il en va de même pour les itérations suivante du processus d'indexation jusqu'au concept terminal.

Le pire cas : Sachant que le calcul de la complexité dépend de l'algorithme de formation du graphe d'indexation, rappelons schématiquement celui-ci. Les concepts entrée représentent les différentes relations présentes dans l'ensemble des indices de l'aspect à indéxer. Nous supposons avoir, en entrée au graphe d'indexation, m concepts entrée tels que $m \leq n$. Ainsi à chaque concept entrée peut correspondre une ou plusieurs substitutions associées à l'aspect. Puis ces substitutions sont regroupées suivant l'une des règles : égalité, inclusion ou intersection. L'intersection assemble les substitutions deux par deux, tandis que l'égalité et l'inculsion forment des groupes de cardinaux $k \geq 2$. Mais dans le pire cas, on suppose k fixé à 2 pour former les groupes les plus petits possibles.

Le nombre de substitutions actives à l'entrée vaut donc n et le nombre de concepts associés $Nbc_1 = m$. A chaque application d'une règle est associé un niveau dans le graphe d'indexation. Le niveau juste après les entrées comporte au pire toutes les combinaisons deux par deux des n substitutions actives du niveau entrée ; c'est-à-dire $\begin{pmatrix} n \\ 2 \end{pmatrix}$ ce qui sous-entend que nous n'utilisons que la règle IS. Donc le nombre total de concepts créés jusqu'à ce niveau est $Nbc_2 = \begin{pmatrix} n \\ 2 \end{pmatrix} + m$. L'étape suivante définit au pire le regroupement des substitutions entrée trois par trois : $Nbc_3 = \begin{pmatrix} n \\ 3 \end{pmatrix} + Nbc_2$. A l'étape i : $Nbc_i = \sum_{j=2}^{i} \begin{pmatrix} n \\ j \end{pmatrix} + m$.

Ce qui implique :

- la profonfeur d'un tel graphe est $P = n$,
- le nombre total de concepts du graphe est :

$$Nbc = Nbc_n = \sum_{j=2}^{n} \left(\begin{array}{c} n \\ j \end{array} \right) + m = 2^n - (n+1) + m$$

P	$=$	n
Nbc	$\simeq$	$O(2^n)$
$\overline{DC}$	$=$	$\frac{\sum_{i=1}^{n-1}(n-i)}{n} = n$

La complexité d'indexation d'un ensemble d'aspects.

Supposons que k aspects $(\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_k)$ soient modélisés tels que :

- les $(i-1)$ premiers aspects sont indexés,
- n_i est le nombre d'indices décrivant l'aspect $\mathcal{A}_i$,
- m_i est le nombre de concepts entrée créés par l'indexation de $\mathcal{A}_i$.
- $Nbc_j^{\mathcal{A}_i}$ est le nombre de concepts créés à l'étape du processus quand les aspects $\mathcal{A}_1, \ldots, \mathcal{A}_i$ sont indexés.

Le meilleur cas : Le meilleur cas est défini quand tous les indices qui décrivent les aspects proviennent d'une même relation ($|\mathcal{C}_e| = 1$).

$$\Rightarrow \quad P = Nbc = \max_{i=1..k}(n_i) \text{ et } \overline{DC} = 1$$

Le pire cas :

$$\begin{aligned}
Nbc_1^{\mathcal{A}_i} &= Nbc_1^{\mathcal{A}_{i-1}} + m_i \\
Nbc_2^{\mathcal{A}_i} &= \left(\begin{array}{c} n_i \\ 2 \end{array} \right) + Nbc_2^{\mathcal{A}_{i-1}} + m_i \\
Nbc_3^{\mathcal{A}_i} &= \sum_{j=2}^{3} \left(\begin{array}{c} n_i \\ j \end{array} \right) + Nbc_3^{\mathcal{A}_{i-1}} + m_i \\
&= \sum_{l=1}^{i} \sum_{j=2}^{3} \left(\begin{array}{c} n_l \\ j \end{array} \right) + \sum_{j=1}^{i} m_j \\
&\vdots
\end{aligned}$$

si nous supposons $n_i = n, \forall i \in [1..k]$, nous obtenons :

$$Nbc_n^{A_i} = \sum_{l=1}^{i} \sum_{j=2}^{n} \binom{n}{j} + \sum_{j=1}^{i} m_j$$

Nous en déduisons, toujours suivant l'hypothèse $n_i = n$:

$$\begin{aligned} Nbc &= \sum_{l=1}^{k} \sum_{j=2}^{n} \binom{n}{j} + \sum_{j=1}^{k} m_j \\ &= \sum_{l=1}^{k} (2^n - 1 - n) + \sum_{j=1}^{k} m_j \\ &= k(2^n - 1 - n) + \sum_{j=1}^{k} m_j \end{aligned}$$

P	$=$	n
Nbc	$\simeq$	$O(k2^n)$
$\overline{DC}$	$=$	n

3.5 Expérimentations

Dans l'étude formelle de la complexité que nous venons de faire, nous n'avons pas envisagé le coût en moyenne qui est, somme toute, le plus intéressant à observer pour avoir une critique objective. En effet, le pire cas et le meilleur cas représentent respectivement les configurations suivantes :

- *Le pire cas* : chaque substitution est associée à un concept qui lui est propre et donc il n'y a aucun recouvrement de l'information entre tous les aspects. Autrement dit, chaque concept décrit une structure qui n'est pas générique puisqu'associée à un unique aspect par une unique substitution.
- *Le meilleur cas* : chaque niveau du graphe est défini par un unique concept représentant l'unique structure générique de ce niveau. Cette structure est donc partagée par tous les aspects. Le graphe ne joue plus son rôle discriminant entre les aspects.

Il est important de vérifier que le comportement en moyenne n'est pas voisin d'une de ces deux configurations. Nous allons effectuer des mesures du coût de l'indexation de manière expérimentale afin d'observer son comportement dans des cas que nous pouvons considérer comme courant. Nous allons procéder comme précédemmment par une analyse de l'indexation d'un seul aspect et ensuite de plusieurs aspects.

3.5.1 L'indexation d'un seul aspect

Les données que nous utilisons proviennent du laboratoire de G. Médioni [Fan *et al.*, 1989]. Ce sont des aspects 3D d'objets obtenues à partir d'un capteur actif :

triangulation entre un émetteur laser et une caméra. Ces images 3D des aspects sont ensuite segmentées en faces [Boufama, 1991] à l'aide d'un détecteur des contours 3D qui proviennent des discontinuités de surface et des discontinuités de courbure. Une fois la segmentation faite, nous recherchons les relations de connexité concave ou convexe liant deux faces ou encore les relations mixtes représentant un lien à la fois de connexité et d'occultation.

Nous avons testé l'indexation d'une base de connaissance qui ne contient qu'un seul objet décrit par un unique aspect. Nous avons observé cette indexation sur quatre objets différents : *Table*, *Chaise*, *Avion* et *Car*.

L'objet *Table*

Cet objet est décrit, dans la base de connaissance, par les deux aspects *table*.1 et *Table*.2 à partir des relations de connexités convexes ($Conv$), Connexité concave (Cav) et des relations qui représente à la fois une connexité et une occultation (Mix) (cf Fig. 3.17) :

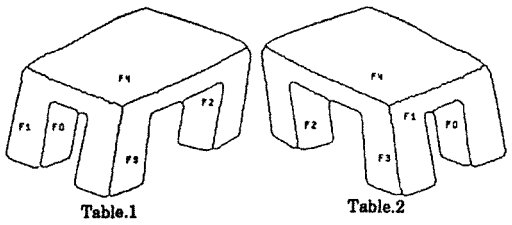


Figure 3.17. *Les deux aspects Table*.1 *et Table*.2 *de la table.*

$$Table.1 = \left\{ \begin{array}{ll} Mix(F0,F1) & \& \\ Cav(F1,F3) & \& \\ Cav(F1,F4) & \& \\ Mix(F2,F3) & \& \\ Cav(F3,F4) & \end{array} \right\} \quad Table.2 = \left\{ \begin{array}{ll} Mix(F0,F1) & \& \\ Cav(F1,F3) & \& \\ Cav(F1,F4) & \& \\ Mix(F2,F3) & \& \\ Concave(F3,F4) & \end{array} \right\}$$

Notons que ces deux aspects ont la même description structurelle. Ceci conduit à construire des index identiques pour les deux aspects. Le graphe d'indexation, illustré en figure (3.18), comporte 11 concepts. Même si l'information structurelle se révèle être importante lors de la reconnaissance, il est évident qu'elle n'en est pas moins insuffisante pour assurer une indexation correcte. Nous développerons ce point en fin de chapitre.

L'objet *Chaise*

L'objet *Chaise* est décrit par deux aspects également : *Chaise*.1 et *Chaise*.2 (cf. Fig. 3.19).

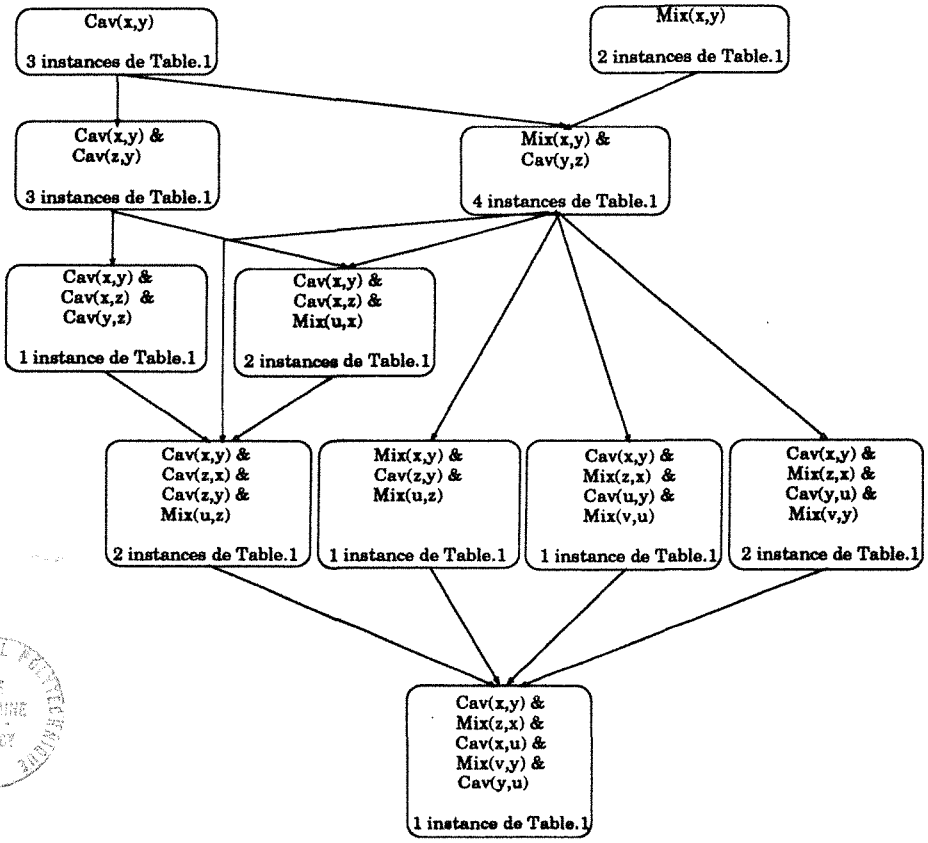


Figure 3.18. *L'index de l'aspect* 3D *Table.*1.

$$Chaise.1 = \left\{ \begin{array}{ll} Mix(F0,F1) & \& \\ Cav(F1,F3) & \& \\ Cav(F1,F4) & \& \\ Mix(F2,F3) & \& \\ Cav(F3,F4) & \& \\ Cav(F3,F5) & \& \\ Cav(F3,F6) & \& \\ Conv(F4,F5) & \& \\ Cav(F5,F6) & \end{array} \right\} \quad Chaise.2 = \left\{ \begin{array}{ll} Mix(F0,F1) & \& \\ Cav(F1,F3) & \& \\ Mix(F2,F3) & \& \\ Mix(F3,F4) & \& \\ Cav(F3,F5) & \end{array} \right\}$$

L'index de *Chaise.*1 comporte 73 concepts tandis que celui de *Chaise.*2 en comporte 17. Le fait, de contraindre la règle d'intersection à regrouper les éléments structurels deux par deux, entraine un coût élevé de concepts pour l'aspect *Chaise.*1.

L'objet *Avion*

L'objet *Avion* est décrit par les deux aspects *Avion.*1 et *Avion.*2 (cf. Fig. 3.20) :

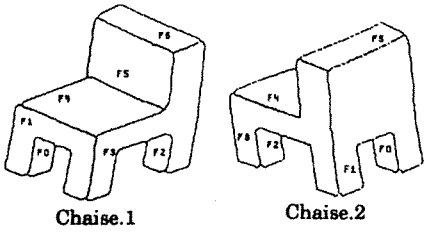


Figure 3.19. *Les deux aspects Chaise.1 et Chaise.2 de la chaise.*

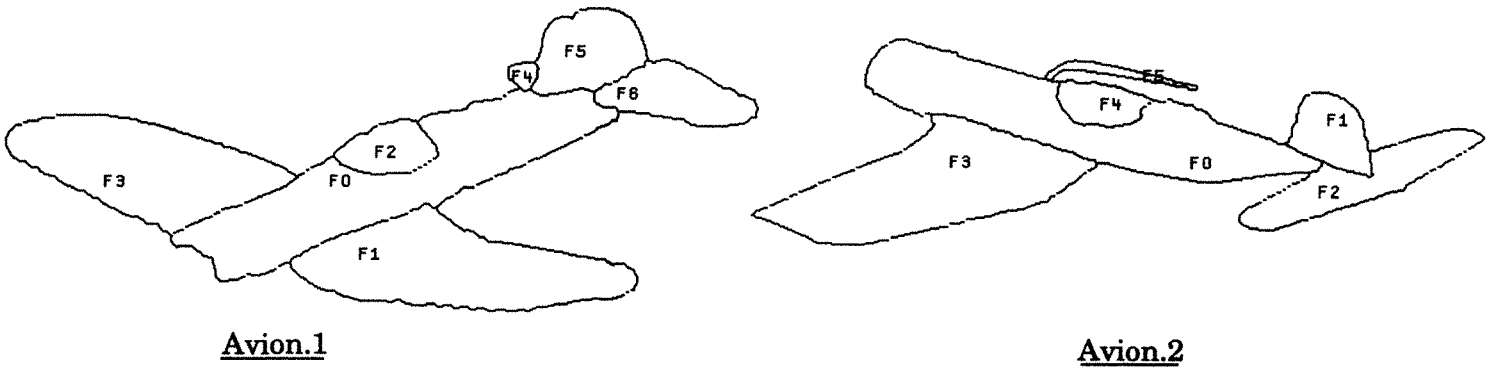


Figure 3.20. *Les deux aspects Avion.1etAvion.2 de la chaise.*

$$Avion.1 = \left\{ \begin{array}{ll} Conv(F0,F1) & \& \\ Conv(F0,F2) & \& \\ Conv(F0,F4) & \& \\ Conv(F0,F5) & \& \\ Conv(F0,F6) & \& \\ Conv(F5,F6) & \end{array} \right\} \quad Avion.2 = \left\{ \begin{array}{ll} Conv(F0,F1) & \& \\ Conv(F0,F2) & \& \\ Conv(F0,F3) & \& \\ Conv(F0,F4) & \& \\ Conv(F1,F2) & \end{array} \right\}$$

Le garphe d'indexation de l'aspect *Avion.1* comporte 9 concept et celui de l'aspect *Avion.2* en compte 7.

L'objet *Car*

L'objet *Car* est décrit par quatre aspects : *Car.1*, *Car.2*, *Car.3* et *Car.4* (cf. Fig. 3.21).

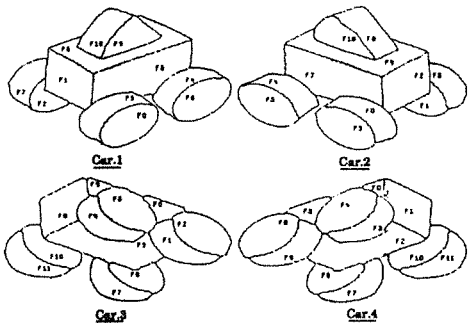


Figure 3.21. *Les quatre aspects Car.1,Car.2, Car.3 et Car.4 de la voiture.*

$$
Car.1 = \left\{ \begin{array}{ll} Cav(F1,F5) & \& \\ Cav(F1,F8) & \& \\ Cav(F5,F8) & \& \\ Cav(F3,F0) & \& \\ Cav(F4,F6) & \& \\ Cav(F2,F7) & \& \\ Cav(F9,F10) & \& \\ Conv(F3,F5) & \& \\ Conv(F4,F5) & \& \\ Mix(F8,F9) & \& \\ Mix(F8,F10) & \end{array} \right\}
\quad
Car.2 = \left\{ \begin{array}{ll} Cav(F3,F0) & \& \\ Cav(F5,F4) & \& \\ Cav(F1,F6) & \& \\ Cav(F2,F7) & \& \\ Cav(F2,F9) & \& \\ Cav(F7,F9) & \& \\ Cav(F8,F10) & \& \\ Conv(F7,F4) & \& \\ Conv(F7,F0) & \& \\ Mix(F9,F10) & \& \\ Mix(F9,F8) & \end{array} \right\}
$$

$$
Car.3 = \left\{ \begin{array}{ll} Cav(F1,F5) & \& \\ Cav(F0,F2) & \& \\ Cav(F4,F6) & \& \\ Cav(F7,F9) & \& \\ Cav(F3,F10) & \& \\ Cav(F10,F11) & \& \\ Conv(F5,F3) & \& \\ Conv(F5,F10) & \& \\ Conv(F2,F3) & \end{array} \right\}
\quad
Car.4 = \left\{ \begin{array}{ll} Cav(F0,F2) & \& \\ Cav(F1,F5) & \& \\ Cav(F3,F6) & \& \\ Cav(F4,F11) & \& \\ Cav(F7,F9) & \& \\ Cav(F10,F11) & \& \\ Conv(F2,F4) & \& \\ Conv(F4,F5) & \& \\ Conv(F5,F11) & \end{array} \right\}
$$

Le graphe d'indexation de $Car.1$ comporte 65 concepts, celui de $Car.2$ 65 également, celui de $Car.3$ 32 et celui de $Car.4$ 33.

Le coût

L'abaque (3.22) représente le coût des indexations obtenues à partir des aspects que nous venons de décrire. L'axe horizontal représente le nombre d'indices utilisés pour décrire l'aspect (n) tandis que l'axe vertical indique le nombre de concepts de l'index correspondant (Nbc). Nous indiquons les deux valeurs thérique du pire cas et du meilleur cas, calculés dans la section précédente. Nous ne fournissons pas de diagramme de la profondeur en fonction du nombre d'indices qui décrivent l'aspect indéxé car, comme dans le pire et le meilleur cas, la profondeur est proche du nombre

d'indices ($P \leq n$).

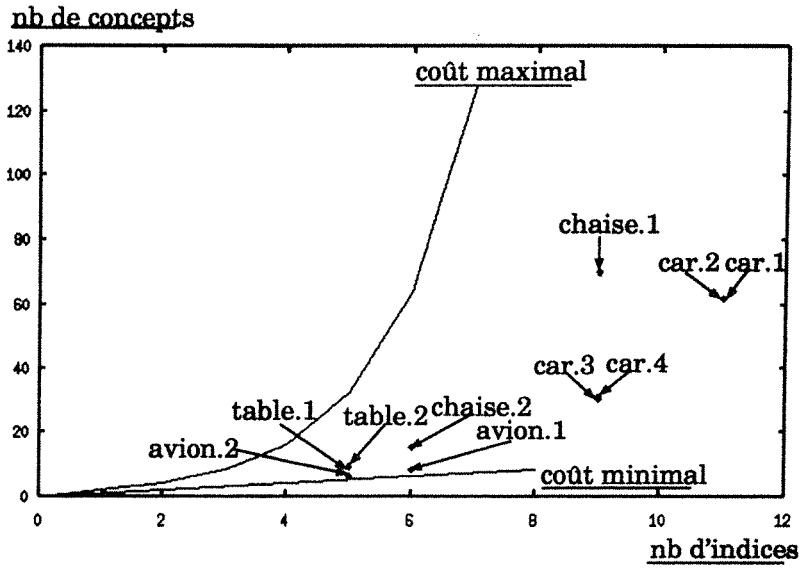


Figure 3.22. *Le nombre de concepts de l'index d'un aspect en fonction du nombre d'indice décrivant l'aspect.*

Ce graphique indique que le processus d'indexation n'a pas tendance à construire des index de taille trop importante ni trop petite comme nous pouvions le craindre. Malgré tout, le nombre de concepts par rapport au nombre d'indices qui décrivent l'aspect indéxé semble suivre une exponentielle. Ce point peut être génant si nous désirons indéxer un aspect décrit par un nombre important d'indices ou bien une base importante de modèles d'objets. Dans ce dernier cas, le nombre de concepts de l'indexation d'un ensemble d'aspects peut tendre vers le cumul du nombre de concepts de chacun des graphes obtenus par l'indexation séparée de chaque aspect. Autrement dit, nous devons vérifier que le coût en moyenne du graphe d'indexation d'un ensemble d'aspects ne vaut pas le coût dans le pire cas ($O(\sum_{i=1}^{k} 2^{n_i})$). Nous nous proposons donc d'expérimenter l'indexation de tous les aspects des quatre objets que nous venons de voir.

3.5.2 L'indexation de plusieurs aspects

Reprenons les dix aspects que nous venons d'indéxer séparément et construisons le graphe d'indexation de l'ensemble. Le graphe d'indexation de ces dix aspects est contitué 154 concepts et n'est bien sûr pas visualisable. Mais la figure (3.23.A) représente la classification structurelle que l'index exprime. Pour comprendre cette figure les notations suivantes pour les aspects sont utilisées :

- $C.i, \ldots, j$ désigne les aspects $Car.i$, ..., $Car.j$.
- $Ch.i, \ldots, j$ désigne les aspects $Chaise.i$ ... $Chaise.2$,

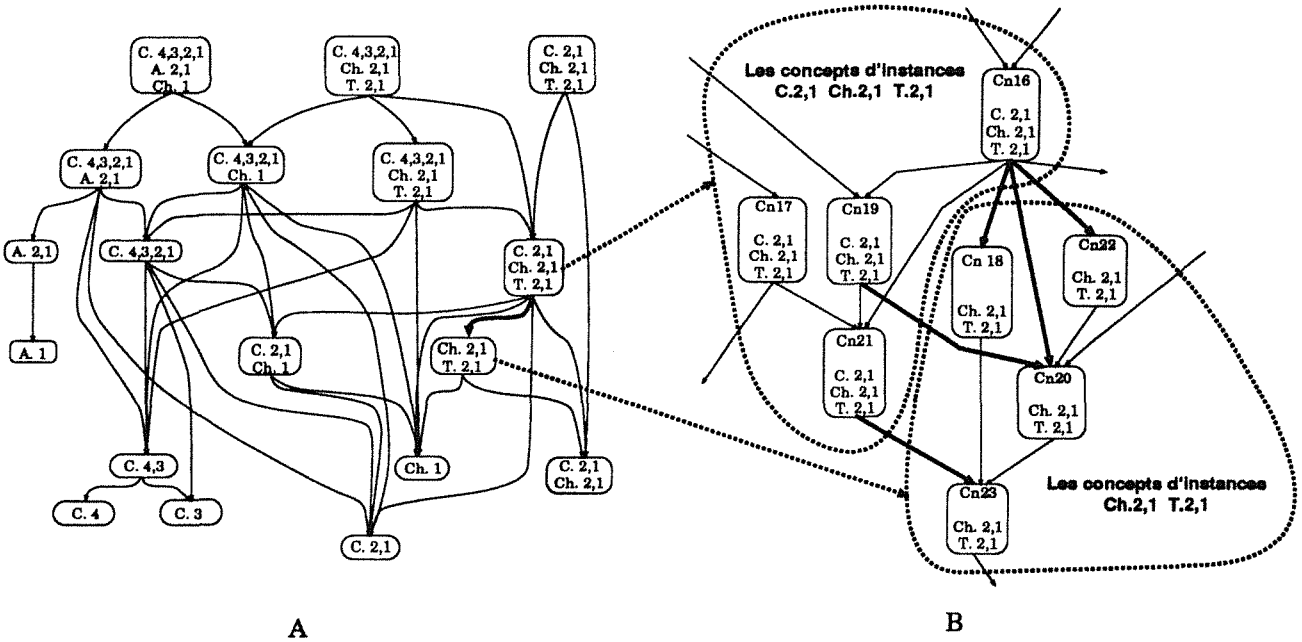


Figure 3.23. *La classification structurelle des dix aspects.*

- $A.i, \ldots, j$ désigne les aspects $Avion.1 \ldots Avion.j$,
- $T.i, \ldots, j$ les aspects $Table.i \ldots Table.j$.

Les noeuds du graphe de la figure représente les groupes de concepts dont les structures génériques sont instanciées par les aspects. Par exemple, le noeud $\{Ch.2, 1\ ; T.2, 1\}$ représente le groupe des concepts $Cn18$, $Cn20$, $Cn22$ et $Cn23$; la classe $\{C.2, 1\ ; Ch.2, 1\ ; T.2, 1\}$ les concepts $Cn16$, $Cn17$, $Cn19$ et $Cn21$. La figure (3.23.B) illustre les deux noeuds et la spécialisation structurelle qui lie ces deux groupes de concepts. Cette classification indique que les aspects $Chaise.1$, $Avion.1$, $Car.3$, $Car.4$ sont représentés par des concepts distincts alors que les autres aspects sont représentés par des concepts instanciés par plusieurs aspects à la fois. L'index est donc disciminant dans nombre de cas mais suffisamment précis pour différencier certains aspects entre eux.

Le coût

Le graphique (3.24) représente le nombre de concepts créés à chaque introduction d'un nouvel aspect. Ce graphique indique que le coût est largement inférieur au pire cas puisqu'en résultat nous avons un graphe constitué de 151 concepts sur une profondeur de 10 alors que le pire est défini, pour l'ensemble des aspects, par un graphe de $\sum_{i=1}^{10} 2^{n_i} = 5856$ concepts sur une profondeur de 11 ! De même ce graphe d'indexation est bien supérieur au coût minimal en nombre de concepts qui est, pour cet ensemble d'aspects, $\max_i(n_i) = 11$. Nous pouvons, d'ores et déja, présenter ces résultats comme positif.

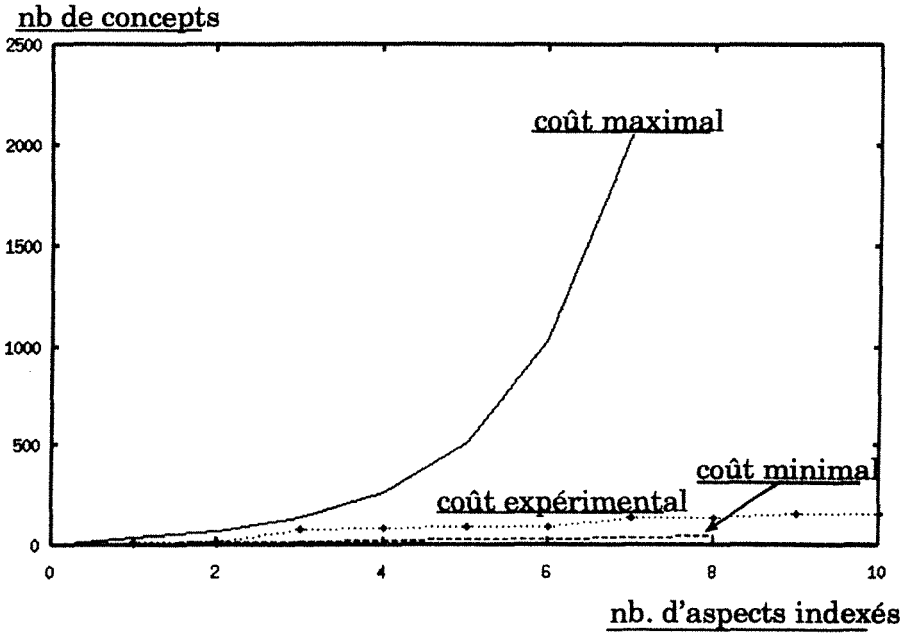


Figure 3.24. *Complexité pour l'indexation de plusieurs aspects.*

3.6 En résumé

L'étude expérimentale, pour être concluante quant au comportement en moyenne de la taille du graphe d'indexation, devrait être faite à partir d'une base de connaissance possédant énormément d'aspects 3D à indéxer afin de répondre aux critères des grands nombres. Mais, hélas, pour des raisons matérieles de tout ordre ces tests ne sont pas possibles, du moins à ce jour. De plus, mêmesi nous supposons ces tests possibles, le seul critére objectif est son utilisation lors de la reconnaissance. Mais, avant d'introduire le prochain chapitre traitant de la reconnaissance, énumérons les caractéristiques que nous avons découvertes au fur et à mesure dans ce chapitre :

- *L'information structurelle n'est pas suffisante pour indéxer complètement les aspects* : par exemple, $Table.1$ et $Table.2$ ou $Car.1$ et $Car.2$ sont décrits dans l'index par les même chemins : les même concepts et les même arcs.

- *Le coût en moyenne* : le coût de graphe semble tout à fait correct, au vues des résultats actuels, puisque proche de k^2 (où k est le nombre d'aspects indéxés). Mais compte tenu du comportement de la profondeur du graphe d'indexation, qui reste toujours voisine du nombre d'indices de l'aspect qui en posséde le plus ($P = \max_i(n_i)$), nous pouvons nous interroger sur le comportement du graphe d'indexation d'une base importante d'aspects.

  En effet, si tous les objets sont décrits avec un nombre à peu près identique d'indices, le graphe peut soit *s'étaler* soit présenter des concepts génériques à un trop fort nombre d'aspects.

- Dans le premier cas, à chaque introduction d'un nouvel aspect, le graphe va augmenter son nombre de concepts mais pas sa profondeur. Chaque niveau du graphe sera représenté par un nombre de plus en plus croissant de concepts ce qui n'est pas sans rappeler le pire cas de la taille du graphe calculé précédemment ! Ceci signifie que le graphe ne jouera plus son rôle de classificateur *hiérarchique* car chaque concept sera le représentant d'un nombre de plus en plus restreint d'aspects, dès les entrées.
- Dans le deuxième cas, les objets possèderons des structures très voisines et donc le graphe ne nous permettra plus de discriminer les aspects entre eux de manière suffisamment efficace pour être utiles lors de la reconnaissance. Ici le graphe aura un comportement proche du meilleur cas du calcul de la complexité ; ce qui signifie que chaque concept sera le représentant d'un trop grand nombre d'aspects.

Seule une étude faite à partir d'un nombre important d'aspects indéxés peut nous autoriser à tirer des conclusions objectives par rapport à ces deux critiques. Malgré tout, nous pouvons remarquer que les problèmes cités ci-dessus sont proches l'un de l'autre. En effet l'information structurelle n'est pas suffisante, mais, nécessaire. Les aspects que nous avons indéxés sont classés dans une hiérarchie de trois groupes :

- *Groupe 1 :* les classes concernants les aspects $Avion.1$ et $Avion.2$.
- *Groupe 2 :* les classes des aspects $Car.3$ et $Car.4$.
- *Groupe 3 :* les classes des aspects $Chaise.1$, $Chaise.2$, $Car.1$, $Car.2$, $Table.1$ et $Table.2$. Les aspects $Table.1$ et $Table.2$ ne sont pas représentés par des concepts qui leurs soient propres car leurs descriptions structurelles est d'un sous-ensemble de celles des aspects $Chaise.1$ et $Chaise.2$. L'aspect $Chaise.1$ est représenté par un concept terminal qui lui est propre. Par contre, le concept terminal pour l'aspect Chaise.2 est aussi associé à des éléments structurels des aspects $Car.1$ et $Car.2$. L'aspect $Car.1$ partage le même concept terminal avec l'aspect $Car.2$.

L'indexation structurelle que nous proposons, construit donc une *classification primaire* des aspects. Le terme *primiare* signifie que la classification n'est pas suffisamment détaillée pour permettre une reconnaissance correcte. C'est-à-dire qu'il faut la compléter par des informations complémentaire venant s'ajouter à cette classification. Nous nous proposons donc développer une indexation complémentaire au niveau de chaque concept pour classer les aspects suivants des critères d'ordre géométrique tel que le type de surface des faces.
L'indexation structurelle représente donc une classification primaire peu coûteuse et peu sensible au bruit, en rapport aux attributs géométriques de classification, et doit donc être utilisée en premier test. Nous reviendrons sur l'évolution à apporter à cette indexation en conclusion de ce mémoire.

4
La reconnaissance

Nous venons de décrire le processus d'indexation et l'index qui en résult. Nous allons maintenant présenter le processus de reconnaissance qui correspond. Il est en fait indispensable de déterminer le comportement de ce processus qui utilse le graphe d'indexation structurelle. Ce comportement définit la *qualité* de l'index et donc la *qualité* du processus d'indexation. La reconnaissance se fait en deux étapes qui peuvent être indépendante ou pas :

- *La Prédiction* génère les hypothèses d'appriements entre les groupes de primitives image de la scène observée et les modèles de la base de connaissance. Elle procède en deux phases : l'*Organisation* et la *Sélection*.
  L'organisation forme les groupes de primitives image correspondant à un même objet de la scène.
  La Sélection recherche les modèles de la base susceptibles de correspondre aux différents groupes image constitués lors de l'étape d'Organisation.

- La *Vérification* confirme ou infirme ces hypothèses d'appariement par une mise en correspondance des primitives image avec celles du modèle.

On appelle *arbre d'interprétation* le résultat du déroulement de ces deux étapes. Il est composé de noeuds qui représentent les associations *(primitive image, primitive modèle)*, et d'arcs qui forment les chemins de cohérences à travers les associations définies par les noeuds. Un chemin de l'arbre qui mène de la racine à une feuille, représente un appariement entre un groupe de primitives image et un modèle. La taille de cet arbre est trop importante si l'on n'utilise pas certaines contraintes pour la limiter [Grimson, 1990]. L'indexation, que nous avons proposée dans le chapitre précédent, doit nous servir pour élaguer cet arbre d'interprétation.

nous allons donc tester l'efficacité de l'index lors de la reconnaissance. C'est-à-dire vérifier que la classification structurelle hiérarchique peut servir de guide pour les phases d'organisation et de sélection. Aussi nous ne présenterons pas l'étape de Vérification du processus de reconnaissance mais uniquement celle de Prédiction (organisation + sélection). Les scènes étudiées sont des compositions à partir des images de distances qui décrivent les aspects 3D utilisés lors de l'indexation. Elles sont donc à mis chemin entre les images réelles - directement acquises par un capteur 3D - et les

images de synthèse. Leurs descriptions structurelles sont en termes de faces et d'indices tout comme celles des aspects.

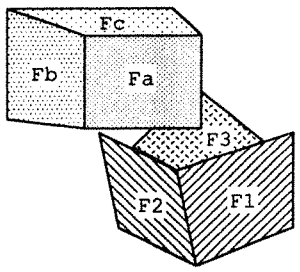
Ce chapitre va commencer par une explication intuitive du fonctionnement du processus de reconnaissance. Ensuite, les règles utilisées pour la reconnaissance sont étudiées en détail. Puis, l'algorithme est expliqué. Cet algorithme est expliqué par rapport à la taille de l'index utilisé. Ce chapitre se termine par des exemples de reconnaissance de scènes.

4.1 Le fonctionnement

La phase de prédiction du processus de reconnaissance consiste à former des groupes de primitives image et à les apparier à des modèles d'objets. Les groupes de primitives image les plus simples sont les indices qui définissent une relation entre primitives. Mais ces indices ne représentent pas des structures qui permettent de sélectionner un nombre restrient d'objets. L'étape de prédiction a donc besoin de construire des groupes plus complexes.
Nous proposons d'utiliser l'index pour construire les groupes image à partir des indices image tout en sélectionnant les modèles d'objets correspondant.

4.1.1 Intuitivement

Plus précisément, les indices image sont associés aux concepts entrée correspondant. Ces associations forment les *hypothèses* initiales. Par exemple, l'indice $Ortho(F1, F2)$, de la scène décrite en figure (4.1) est filtré par la structure générique $\{Ortho(x, y)\}$



Description de la scène :

Ortho (F1,F2) &
Ortho (F1,F3) &
Ortho (F2,F3) &
Ortho (Fa,Fb) &
Ortho (Fa,Fc) &
Ortho (Fb,Fc) &
Connexe (F1,F2) &
Connexe (Fa,Fb) &
Connexe (Fa,Fc) &
Connexe (Fb,Fc) &
Occulte (F1,F3) &
Occulte (F2,F3) &
Occulte (Fa,F3)

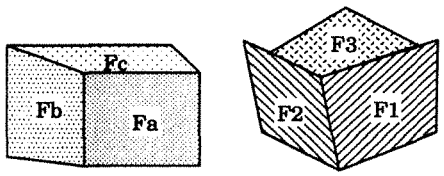
Figure 4.1. *Un exemple de scène : La relation Ortho indique la perpendicularité entre deux faces.*

du concept entrée $C1$ suivant l'hypothèse $H1 = ((x, F1)(y, F2))$. Puis à partir des spécialisations structurelles (les arcs du graphe d'indexation), le processus regroupe ces hypothèses afin d'en former de nouvelles associées à des concepts dont les stuctures génériques sont plus complexes.

Les hypothèses sont comparables aux substitutions, utilisées lors de l'indexation. Aussi à sa création une hypothèse est déclarée active. Et dès qu'elle a été utilisée, elle est déclarée inactive.
Ainsi le processus de reconnaissance itère jusqu'à ce qu'aucune fusion d'hypothèses ne soit possible. Les hypothèses restantes sont dites *finales*. Et elles expriment les couples associant un élément structurel de la scène à un concept de l'index.
Les hypothèses étant comparables aux substitutions, nous continuons la comparaison. Une hypothèse est en fait le lien entre un *élément structurel* et une *structure générique*. Si la notion de structure générique n'a pas changée, celle d'élément structurel ne se rattache plus à un modèle mais à l'image de la scène. Un élément structurel représente alors un groupe de primitive image que l'on peut apparier à au moins un aspect de la base de connaissance. Par la suite, au lieu de décrire les hypothèses comme des couples *(groupe image, modèle)*, nous les décrivons comme des couples *(groupe image, concept)* car les concepts sont les agents charnières avec les aspects indéxés.

4.1.2 Un exemple

Par exemple, considérons la scène illustrée par la figure (4.1) et la base de connaissance des deux aspects de la figure (4.2). L'index de la base est illustré par la figure



L'aspect A1 :

Ortho (Fa,Fb) &
Ortho (Fa,Fc) &
Ortho (Fb,Fc) &
Connexe (Fa,Fb) &
Connexe (Fa,Fc) &
Connexe (Fb,Fc)

L'aspect A2 :

Ortho (F1,F2) &
Ortho (F1,F3) &
Ortho (F2.F3) &
Connexe (F1,F2) &
Occulte (F1,F3) &
Occulte (F2,F3)

Figure 4.2. *La base de connaissance associée à l'exemple.*

(??). Il possède dix concepts et est de profondeur trois. Le concept $C9$ est terminal pour l'aspect $A1$ tandis que le concept $C10$ l'est pour l'aspect $A2$. Seul le concept $C4$ exprime une structure généque instanciée à la fois par $A1$ et $A2$.

L'arbre d'interprétation est illustré par la figure (4.4). Les hypothèses initiales, c'est-à-dire les indices image, sont classées suivant le type de relation. Au concept $C1$ sont associées les hypothèses $(H5, \ldots, H10)$, au concept $C2$ les hypothèses $(H1, \ldots, H4)$, au concept $C3$ les substitutions $(H11, \ldots, H13)$.
Ces hypothèses initiales sont déclarées actives.
Ensuite, le processus procède au regroument des hypothèses. Si l'on considère l'hypothèse $H1 = (\{F1, F2\}, C2)$, seul le concept $C4$ est une spécialisation de $C2$. Il faut donc que l'hypothèse $H1$ s'associe à une autre hypothèse pour former un élément structurel qui puisse être filtré par la structure générique $\{Ortho(x,y)\&Connexe(x,y)\}$

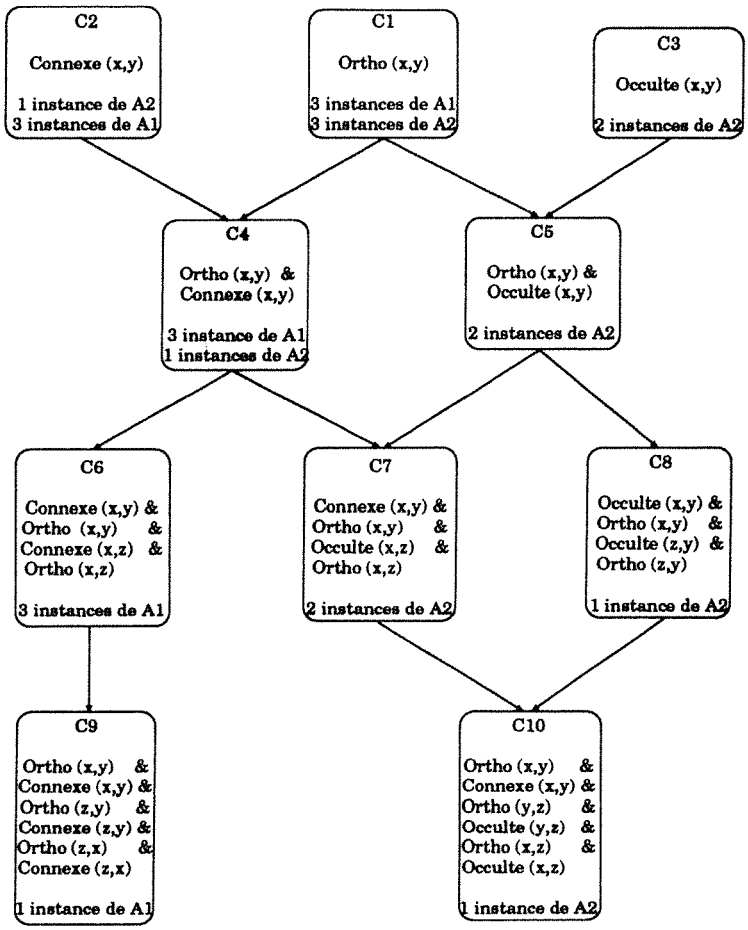


Figure 4.3. *L'indexation des aspects A1 et A2.*

du concept $C4$. Si l'hypothèse $H5 = (\{F1, F2\}, C1)$ est fusionnée à $H1$, l'élément structurel correspondant $\{Ortho(F1, F2)\&Connexe(F1, F2)\}$ est filtré par la structure générique du concept $C4$. Cet élément structurel définit en fait les deux structures génériques des deux concepts $C1$ et $C2$ qui se spécialise en la structure générique du concept $C4$.

Ainsi la fusion d'hypothèses correspond à rechercher le concept qui spécialise tous les concepts des hypothèses et qui filtre l'élément structurel défini par la fusion. Après la recherche de fusions, les hypothèses utilisées sont déclarées désactivées et les nouvelles créées sont déclarées actives. Le processus itère ainsi jusqu'à ce qu'aucune fusion ne soit possible. En résultat, le processus fournit les hypothèses $H13$, $H26$ et $H27$. L'hypothèse $H13 = (\{Fa, F3\}, C3)$ signifie que l'élément structurel $\{Occult(F1, F2)\}$ est supposé provenir d'une occurrence image d'un des deux aspects $A1$ ou $A2$. L'élément structurel de l'hypothèse $H26 = (\{Fa, Fb, Fc\}, C9)$ est présumé être une partie de l'aspect $A1$. De même l'élément structurel de l'hypothèse $H27 = (\{F1, F2, F3\}, C10)$ est présumé être une partie de l'aspect $A2$.

Nous constatons que les deux aspects de la scène ont été identifiés et localisés mais

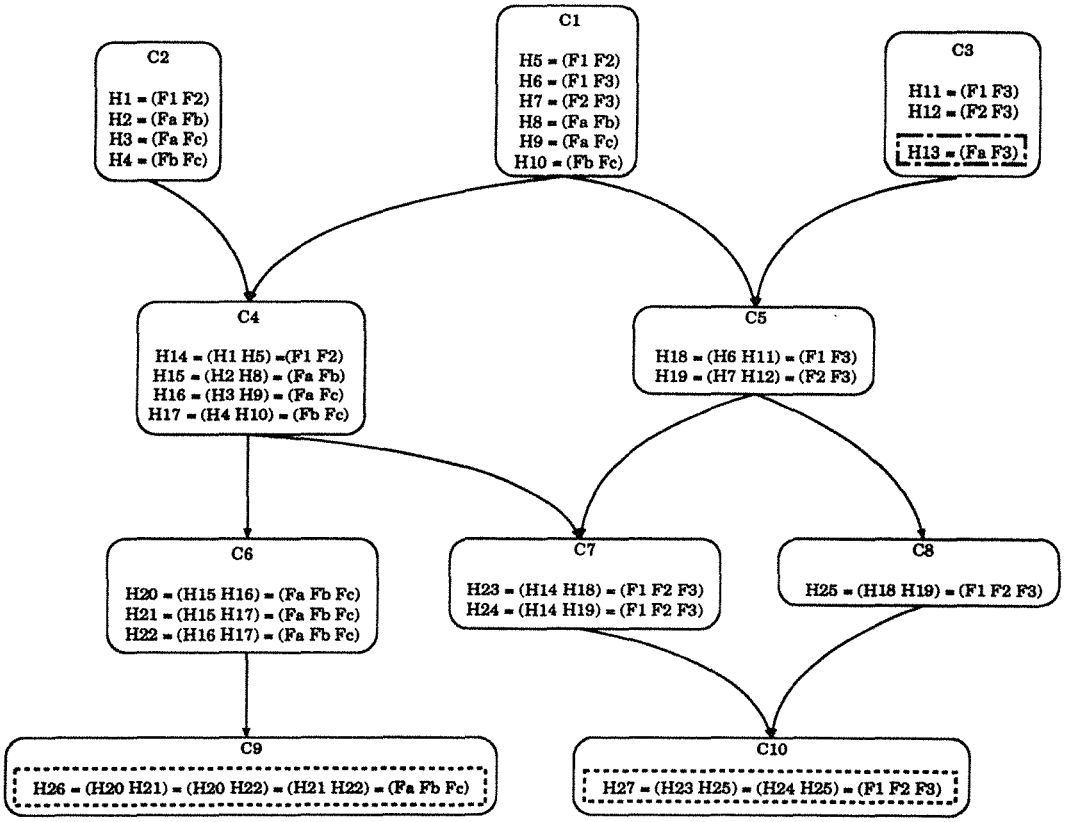


Figure 4.4. *L'arbre d'interprétation : les hypothèses finales sont encadrées en pointillé.*

qu'il persiste une fausse hypothèse, $H13$.

Comme les fusions sont validées par les concepts et les spécialisations structurelles de l'index, elles respectent les règles utilisées pour créer les concepts.

4.2 Les règles de fusion

Les règles d'*égalité*, d'*inclusion* et d'*intersection*, utilisées pour l'indexation, servent à fusionner les hypothèses. Lors de la reconnaissance, les concepts ne sont pas créés mais vérifiés parmi les groupes d'indices image.
Ainsi à la fusion de plusieurs hypothèses correspond un ensemble de concepts $\{C_i, \ldots, C_j\}$. Les spécialisations structurelles de cet ensemble de concepts $\{C_i, \ldots, C_j\}$ déterminent les concepts, désignés par $\{fils(C_i), \ldots, fils(C_j)\}$ susceptibles de filtrer l'élément structurel de la fusion.

- *Egalité*: (E)

  Etant donné une scène 3D $\mathcal{S}$, *nous sélectionnons toutes les hypothèses actives. De ces hypothèses, on retient celles qui ont la même projection, c'est-à-dire celles qui décrivent des éléments structurels associés à un même groupe de primitives.*

Ces hypothèses définissent un ensemble de concepts ; c'est-à-dire un ensemble de structures génériques. A partir de ces hypothèses et des structures génériques associées, nous construisons une nouvelle hypothèse.

Définition 6

$$\left[\begin{array}{c} Proj(H_1) = Proj(H_2) = \cdots = Proj(H_k) \\ \forall i \in [1,k], H_i = (\{p_{i_1}, \ldots, p_{i_n}\}, C_i) \end{array}\right] \& \left[\begin{array}{c} Trouver\ C\ tel\ que \\ C \in fils(C_i), \forall i \in [1,k] \end{array}\right]$$

$$\left[\begin{array}{c} \{\{C_1, C_2, \cdots, C_k\}\} \\ H_1, H_2, \cdots, H_k \end{array}\right] \rightsquigarrow \left[\begin{array}{c} Créer\ H\ tel\ que \\ HC = \wedge_{i=1}^{k} H_i C_i \end{array}\right]$$

Exemple:
Soit une scène 3D $\mathcal{S}$ liée au concept $C1$ par l'hypothèse $H1 = ((f_x, F_1)\ (f_y, F_2)\ (f_z, F_3))$ et au concept $C2$ par l'hypothèse $H2 = ((f_x, F_1)\ (f_y, F_2)\ (f_z, F_3))$, comme illustré en figure (4.5).

Nous constatons que $Proj(H_1) = Proj(H_2) = \{F_1, F_2, F_3\}$. Nous regroupons donc les deux éléments structurels des hypothèses $H1$ et $H2$ pour créer l'hypothèse H, illustrée par la figure (4.5.b).

- *InClusion*: (IC)

  Soit une scène 3D $\mathcal{S}$. *Nous agglomèrons les hypothèses dont les projections sont strictement incluses dans un même groupe de primitives de* $\mathcal{S}$.

Définition 7

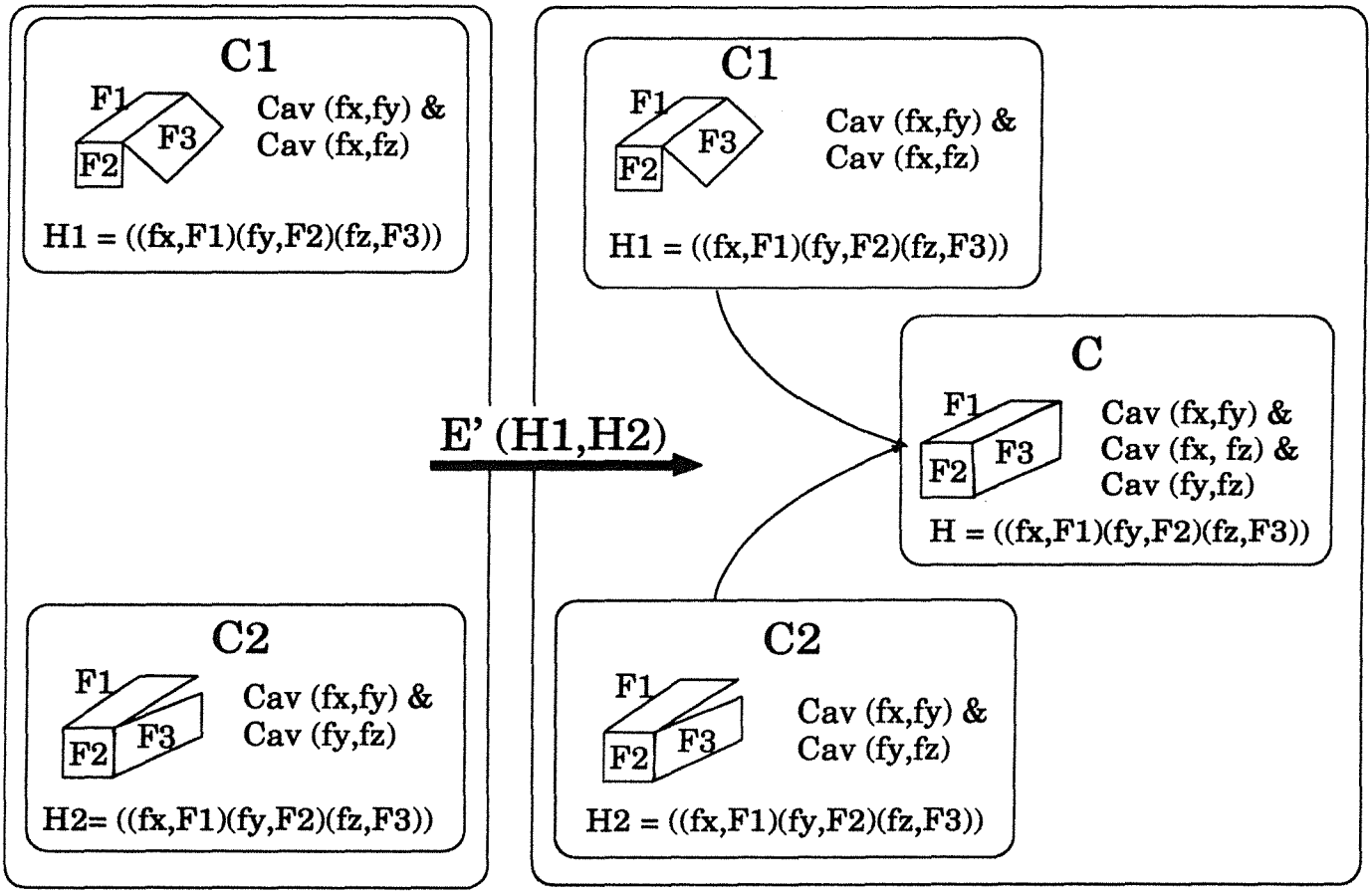
$$\left[\begin{array}{c} Proj(H_i) \subset Proj(H_j) \\ H_i = (\{p_{i_1}, \ldots, p_{i_n}\}, C_i), \\ \forall i \in [1..k], j \notin [1..k] \end{array}\right] \& \left[\begin{array}{c} C_i \neq C_j \\ \forall i \in [1..k] \end{array}\right] \& \left[\begin{array}{c} Trouver\ C\ tel\ que \\ C \in fils(C_i), \forall i \in [1,k] \end{array}\right]$$

$$\left[\begin{array}{c} \{\{C_1, \ldots, C_k, C_j\}\} \\ H_1, \ldots, H_k, H_j \end{array}\right] \rightsquigarrow \left[\begin{array}{c} Créer\ H\ tel\ que \\ HC = \wedge_{i=1}^{k} H_i C_i \end{array}\right]$$

Exemple:
En figure (4.6.a), le groupe de primitives image de l'hypothèses $H1$ est inclus dans celui de l'hypothèse $H2$. La règle d'inclusion autorise le regourpement en une hypothèse H associée au concept C (cf. Fig. 4.6.b).

- *InterSection*: (IS)

  Soit un aspect 3D $\mathcal{A}$. *Parmi toutes les substitutions qui lui sont associées, nous regroupons deux par deux celles dont les projections ont des primitives en commun.*



(a) Le graphe d'indexation avant l'application de la règle E.
(b) Le graphe d'indexation après.

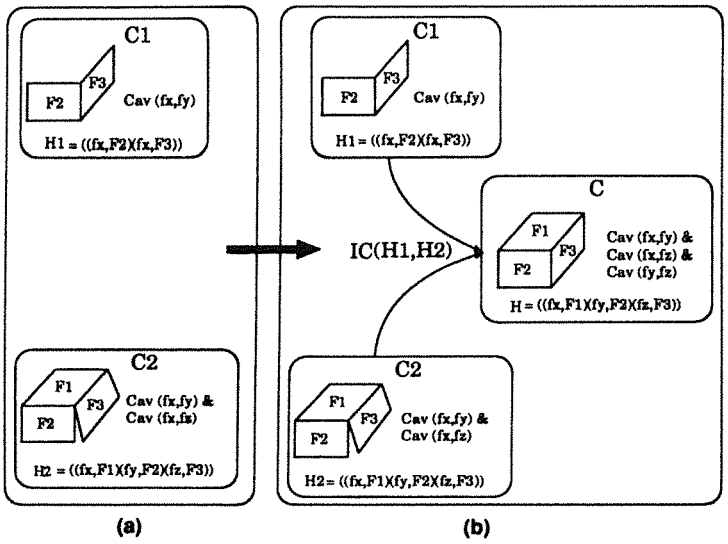
Figure 4.5. *Une exemple d'application de la règle d'égalité.*

Définition 8

$$\left[\begin{array}{c} Proj(H_1) \cap Proj(H_2) \neq \emptyset \\ H_i = (\{p_{i_1}, \ldots, p_{i_n}\}, C_i), i = 1,2 \end{array}\right] \& \left[\begin{array}{c} Trouver\ C\ tel\ que \\ C \in fils(C_1) \cap fils(C_2) \end{array}\right]$$

$$\left[\begin{array}{c} C_1, C_2 \\ H_1, H_2 \end{array}\right] \rightsquigarrow \left[\begin{array}{c} Créer\ H\ tel\ que \\ HC = H_1C_1 \& H_2C_2 \end{array}\right]$$

Exemple:
En figure (4.7.a), le groupe de primitives image de l'hypothèses $H1$ intersecte



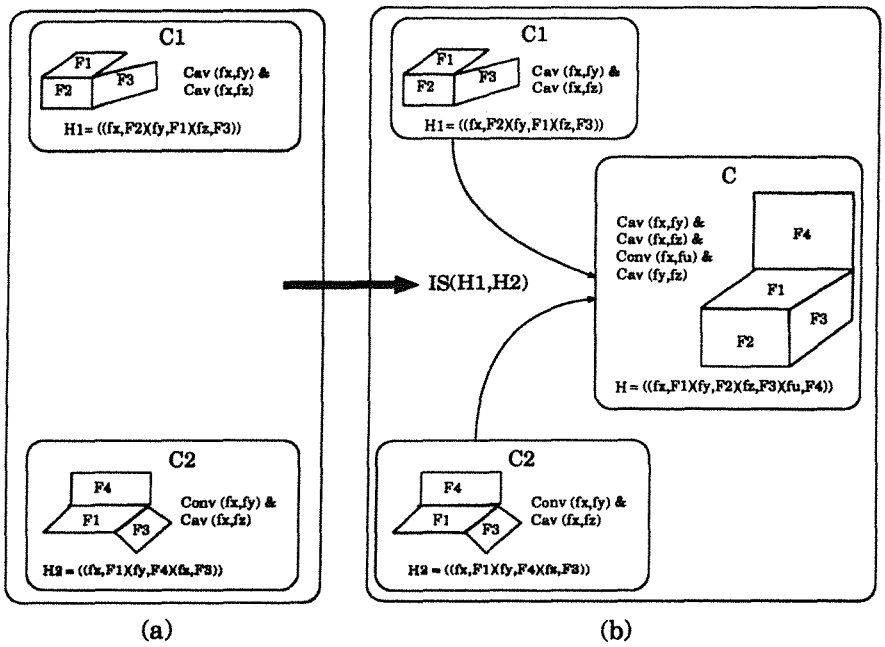
(a) Le graphe d'indexation avant l'application de la règle IC.
(b) Le graphe d'indexation après.

Figure 4.6. *Un exemple d'application de la règle d'inclusion.*

celui de l'hypothèse $H2$. La règle d'intersection autorise le regroupement en une hypothèse H associée au concept C (cf. Fig. 4.7.b).

Si les règles sont utilisées pour fusionner les hypothèses, il est fondamental de les appliquer suivant un ordre qui soit naturel par rapport au fonctionnement du processus de reconnaissance.
Les éléments structurels, définis sur un même groupe de primitives image, doivent forcément décrire le même aspect. Aussi doivent-ils, par l'intermédiaire du graphe d'indexation, être réunis en une seule hypothèse regroupant toute l'information structurelle. Si le graphe d'indexation n'autorise pas cette réunion, ceci signifie que ces hypothèses sont fausses car elles sont définies sur une zone décrivant des parties de différents aspects. Cette notion de réunion des hypothèses, définies sur un même groupe de primitives image, revient à définir la règle d'égalité prioritaire par rapport aux deux autres. De même la règle d'inclusion définie un regourpement de l'information structurelle pour un même groupe de primitives qui explique la priorité de la règle d'inclusion sur celle d'intersection :

$$E > IC > IS$$



(a) Le graphe d'indexation avant l'application de la règle IS.
(b) Le graphe d'indexation après.

Figure 4.7. *Un exemple d'application de la règle d'intersection.*

4.3 Ordonner les hypothèses

Avec l'exemple de la figure (4.1), le processus de reconnaissance obtient 3 hypothèses finales dont une $H13$ est fausse. Dans ce cas, il faut d'abord vérifier $H26$ et $H27$ avant $H13$ ce qui éliminera automatiquement l'hypothèse $H13$. Les hypothèses doivent donc être classées suivant la probabilité de *certitude* que l'on peut leur attribuer.

Une hypothèse décrivant un élément struturel complexe est plus sûre qu'une faisant référence à un élément structurel simple. Les hypothèses $H26$ et $H27$ décrivent des éléments structurels que l'on peut considérer de *complexité* égales et toutes deux supérieures à la complexité de l'élément structurel de l'hypothèse $H13$. La complexité d'un élément structurel peut être décrite par le nombre de primitives image le formant et la place du concept dans l'index. Les hypothèses faites sur un élément structurel sont d'autant plus sûres que cet élément structurel défini un groupe important de primitives image. La place d'un concept est évidente puisque les arcs sont orientés. Nous obtenons donc pour notre exemple :

$H26 = (\{Fa, Fb, Fc\}, C9)$: 3 Primitives concept terminal
$H27 = (\{F1, F2, F3\}, C10)$: 3 Primitives concept terminal
$H13 = (\{Fa, F3\}, C3)$: 2 Primitives concept entrée

$\Rightarrow p(H26) = p(H27) > p(H13)$
où $p(x)$ est la probabilité d'exactitude pour l'hypothèse x.

4.4 Algorithme

Le processus de reconnaissance que nous allons présenter est fort simple et beaucoup d'amélioration peuvent être faite. Mais son intérêt premier est la validation de la méthode d'indexation que nous avons proposé. Aussi préférons nous utiliser une version simplifiée de cet algorithme. Pour le décrire nous utilisons les notations suivantes :

$\mathcal{S}$	=	$\{I_1, \ldots, I_n\} \subset \mathcal{I}$; l'ensemble des indices décrivant la scène observée.
L_faux	=	La liste des agglomérats non autorisés.
L	=	La liste des hypothèses actives.

L'algorithme est identique à celui de l'indexation seule la procédure *Construction* est remplacée par la procédure *Conjecturer*.

Procédure Reconnaissance $(\mathcal{S})$;
$L\_faux \leftarrow \emptyset$;
$L \leftarrow \emptyset$;

pour tout $C \in \mathcal{C}_e$ *faire*
/* *Initialisation des hypothèses en association les indices de* $\mathcal{S}$ *aux concepts entrée* */
pour tout $I \in \mathcal{S}$ *faire*
créer σ tq $\sigma C = I$;
$L \leftarrow L + \sigma$;
fin-pour
fin-pour

Répéter
Répéter
$G \leftarrow$ *Egalités* (L) ;
/* *formation des réunions d'hypothèses suivant la règle d'égalité, en vérifiant qu'elles n'appartiennent pas à* L_faux. *Suppression des hypothèses de* G *dans* L */
Conjecturer (G) ;
$G \leftarrow$ *Inclusions* (L) ;
Conjecturer (G) ;
jusqu'à $(G = \emptyset)$;
$G \leftarrow$ *Intersections* (L) ;
Conjecturer (G) ;

jusqu'à $(G = \emptyset)$;
fin-procédure Reconnaissance ;

La procédure *Conjecturer* ne crée pas de concept mais uniquement les hypothèses liant les groupes image aux concepts du graphe. En particulier si à un ensemble d'hypothèses, obtenu par application de la partie condition d'une des règles d'inclusion ou d'intersection, ne correspond aucun concept ($((F = (C, \emptyset))$ et $(Test \neq Ok))$ ou $(F = \emptyset)$) alors le processus doit reconsidérer ces hypothèses comme actives ($L \leftarrow L \cup g_i$) pour d'autres regroupements possibles tout en notant que cet assemblage n'est plus valide ($L\_faux \leftarrow L\_faux + g_i$). D'où l'utilisation d'une liste des regroupements d'hypothèses non autorisés. Par contre, si la règle d'égalité est utilisée alors nous sommes en présence d'un ensemble de fausses hypothèses car elles sont fondées sur le même groupe image alors qu'aucun concept du graphe d'indexation ne confirme le regoupement de ces hypothèses. Ceci signifie que les éléments structurels correspondant proviennent de plusieurs aspects.

Procédure Conjecturer (G) ;
    /* $G = \{g_1, \ldots, g_n\}$ *est la liste des ensembles*
    d'hypothèses formés suivant une des règles */

    pour tout $g_i \in G$ *faire*
        $F \leftarrow$ *Instanciation* $(\mathcal{C}, g_i)$;
        /* *la même procédure que lors de l'indexation* */

        si $F = (C, \emptyset)$ *alors*
            /* *La structure générique du concept* C *est instanciée*
            par l'élément structurel de g_i */
            $Test \leftarrow$ *Vérifier* ($Concepts(g_i)$, Pères(C)) ;
            /* *Cette fonction vérifie que les concepts associés au groupe* g_i
            constituent un groupe d'antécédents pour C */
            si Test = Ok *alors* $L \leftarrow L + \sigma$;
            sinon
                $L\_faux \leftarrow L\_faux + g_i$;
                si (Règle = IC) *ou* (Règle = IS) *alors* $L \leftarrow L \cup g_i$;
            fin-si
        sinon si $(F = \emptyset)$ *alors*
            $L\_faux \leftarrow L\_faux + g_i$;
            si (Règle = IC) *ou* (Règle = IS) *alors* $L \leftarrow L \cup g_i$;
        sinon $L \leftarrow L + \sigma$;
    fin-pour
fin-procédure Conjecturer ;

4.5 Les caractéristiques de l'algorithme

L'efficacité du processus de reconnaissance dépend de sa complexité temporelle et de la qualité des résultats obtenus. Nous voulons que le processus soit efficace en temps et qu'il délivre un ensemble d'hypothèses finales qui soit correct et complet par rapport aux aspects composant la scène observée. Trois paramètres influent ces deux critères :

1. *Le taux de recouvrement :* les scènes observées sont composées d'objets pouvant s'occulter plus ou moins. Pour la reconnaissance de scènes où les objets sont séparés les uns des autres sur un fond uniforme, il est facile de former les groupes images correspondant à ces objets. En d'autres termes, l'étape d'organisation est, par le fait même, simplifiée et un des buts de notre indexation devient sans intérêt. Aussi allons nous observer des scènes encombrées d'objets multiples et variés.

2. *Le bruit dans les images :* nous supposons que les primitives peuvent être manquantes mais non erronées et que les indices proviennent de relations robustes à la détection. Pour la reconnaissance, ceci signifie que certains éléments structurels de l'image de distances de la scène observée ne seront pas détectable et donc qu'il y aura des pertes d'information mais pas détérioration de celle-ci. Ces pertes ne doivent pas être trop importante sinon la reconnaissance risquerait d'en souffrir fortement : il n'y aura plus suffisamment de faces et d'indice image pour construire des hypothèses sûres.
   La plus part des laboratoires de vision travaillaient voilà quelques années sur la modélisation et la reconnaissance d'ordre sémantique et/ou symbolique pensant pouvoir obtenir assez facilement des données 2D/3D parfaites et non manquantes. Mais depuis, ils ont orienté leurs recherches sur l'élimination du bruit lors de l'extraction de primitives car ce problème n'a été que trop sous-estimé. Bien que primordiale pour la vision, ce problème n'intervient pas dans notre étude et nous faisons donc des hypothèses sur les données. Nous en tenons, malgré tout, compte en utilisant des données réelles pour les aspects indéxés et *semi-réelles* pour les scènes reconnues.

3. *La taille de l'index :* le graphe d'indexation peut être de taille plus ou moins importante. Nous avons observé, dans le chapitre précédent, les différentes tailles que le graphe d'indexation peut avoir. Intuitivement, nous pouvons en déduire que cela aura une influence sur le processus de reconnaissance.

Le seul paramètre qui va donc nous intéresser sera la taille du graphe d'indexation. Nous allons tout d'abord considérer les effets du pire cas et du meilleur cas de la complexité du graphe d'indexation avant de considérer une taille de graphe plus raisonnable.

Le pire cas : Chaque aspect indéxé est représenté par un ensemble d'éléments structurels qui lui est propre. Aussi, le graphe d'indexation est fort discriminant. Chaque

structure générique de l'index représente un seul aspect. Mais, dans ces conditions, l'index ne sert à rien : parcourir un graphe d'indexation de taille $O(\sum_{i=1}^{k} 2^{n_i})$ (k aspects indéxés et n_i indices pour décrire l'aspect i : calcul du pire cas) revient à parcourir entièrement le garphe d'interprétation de taille $O(2^n)$ où n est le nombre d'inidces décrivant la scène.

Le meilleur cas : Dans le meilleur cas, le processus d'indexation construit à chaque niveau un seul concept associé à tous les aspects indéxés. Le graphe ne permet donc pas de discriminer les aspects entre eux. Tout comme dans le pire cas, l'indexation ne donne pas la classification structurelle attendue. Le processus de reconnaissance, associé à un index de cette configuration, ne peut donc mener l'étape de prédiction à son terme : seule l'Organisation est faite.

En moyenne : L'estimation en moyenne de la taille du graphe d'indexation a été faite expérimentalement. Nous allons donc utiliser l'indexation, qui servit à cette étude, pour observer le comportement du processus de reconnaissance. Nous allons chercher à savoir si la classification structurelle que représente l'index des dix aspects que nous avons construit dans le chapitre précédent, permet de guider correctement la formation des hypothèses.

4.6 Les expérimentations

Nous allons d'abord utiliser l'index des aspects décrits en figures (4.8) pour reconnaitre les scènes illustrées en (4.10) et (4.11). Puis les aspects de la figure (4.9) sont ajoutés à l'index et les résultats de la reconnaissance sur les scènes illustrées en figures (4.11) et (4.12), sont fournis.

4.6.1 *scene*.1

La scène *scene*.1, illustrée en figure (4.10), est composée des aspests 3D *Chaise*.2, *Avion*.1 et *Table*.2. L'index utilisé est construit pour les aspects illustrés en figure (4.8).

Résultats

Les groupe de primitives image	Les aspects conjecturés
($F0$ $F2$ $F3$ $F4$ $F5$)	*Avion*.1, *Avion*.2
($F13$ $F14$ $F15$ $F16$ $F17$)	*Chaise*.1, *Chaise*.2, *Table*.1, *Table*.2.
($F7$ $F8$ $F9$ $F10$ $F11$ $F12$)	*Chaise*.2

Les trois groupes de primitives représentant les objets de la scène sont détectés. Le groupe ($F7$ $F8$ $F9$ $F10$ $F11$ $F12$) est identifié correctement comment étant l'aspect

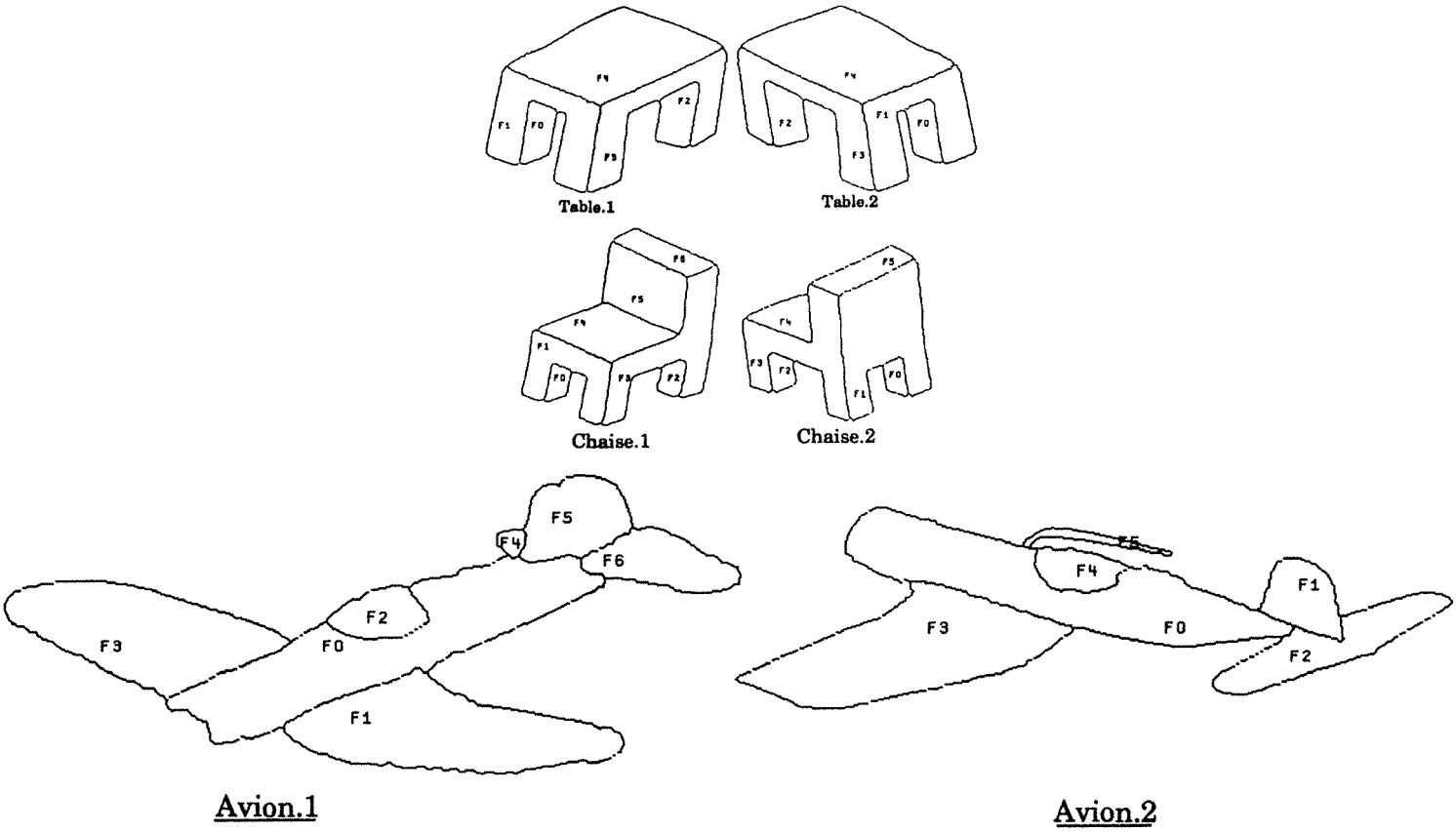


Figure 4.8. *Les aspects indéxées.*

Chaise.1. La structure décrite par le groupe ($F0$ $F2$ $F3$ $F4$ $F5$) (resp. ($F13$ $F14$ $F15$ $F16$ $F17$)) ne permet d'identifier exactement l'aspect correspondant.

4.6.2 *scene*.2

La scène *scene*.2, illustrée en figure (4.11), est composée des aspests 3D *Chaise*.2, *Avion*.1, *Table*.2 et *Car*.2.

Résultats

L'index utilisé est construit pour les aspects illustrés en figure (4.8).

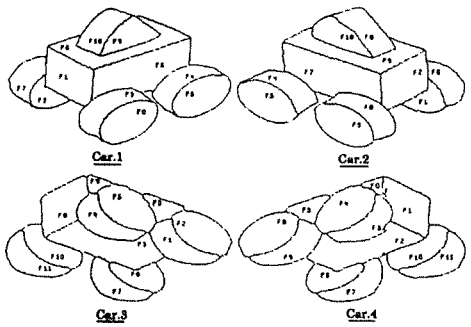


Figure 4.9. *Des aspects ajoutés à l'index existant.*

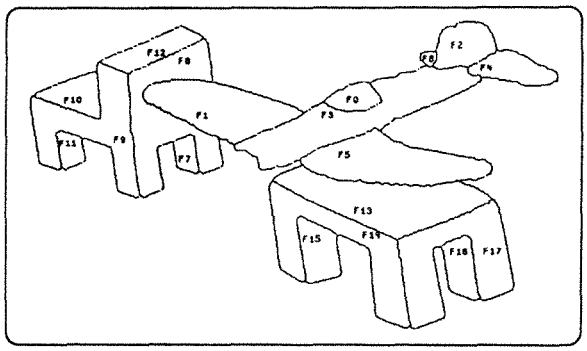


Figure 4.10. *scene*.1 *: composée des aspects Chaise*.1, *Avion*.1 *et Table*.1.

Les groupes de primitives image	Les aspects conjecturés
$(F0\ F2\ F3\ F4\ F5)$	*Avion*.1, *Avion*.2
$(F13\ F14\ F15\ F16\ F17)$	*Chaise*.1, *Chaise*.2, *Table*.1, *Table*.2
$(F7\ F8\ F9\ F10\ F11\ F12)$	*Chaise*.2
Huit groupes d'hypothèses provenant de $(F18 \cdots F28)$	*Table*.1, *Table*.2, *Chaise*.1 *Chaise*.2, *Avion*.1, *Avion*.2

La scène est identique à *scene*.1 à laquelle s'est ajoutée l'aspect *Car*.2. Le trois groupes représentant *Table*.2, *Chaise*.2 et *Avion*.1 sont identifiés comme précédemment. Mais l'aspect *Car*.2 n'ayant pas été modéliser et donc pas indexé, le processus de reconnaissance forme huit groupes de primitives qui recouvrent ($F7\ F8\ F9\ F10\ F11\ F12$). Chacun de ces groupes $(F_i \ldots F_j)$ est identifié comme un aspect différent des aspects des groupes recouvrant partiellement $(F_i \ldots F_j)$. Il est donc évident que ces regroupement et leurs identifications sont incohérents.

Maintenant, observons le processus de reconnaissance avec l'index précédent auquel s'est ajouté les aspects *Car*.1, *Car*.2, *Car*.3 et *Car*.4, illustrés par la figure (4.9).

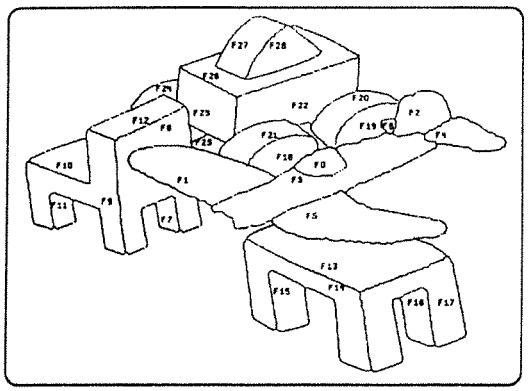


Figure 4.11. *scene.2 est composée des aspects Chaise.1, Avion.1, Table.1 et Car.1.*

Les groupes de primitves image	Les aspects conjecturés
(*F0 F2 F3 F4 F5*)	*plane.1*, *plane.2*
(*F13 F14 F15 F16 F17*)	*chair.1*, *chair.2*, *stool.1*, *stool.2.*
(*F7 F8 F9 F10 F11 F12*)	*chair.2*
(*F18 F19 F20 F21 F22 F23 F24 F26 F27 F28*)	*Car.1*, *Car.2*

Après l'indexation des aspects *Car.1*, *Car.2*, *Car.3* et *Car.4*, le processus de reconnaissance forme le groupe (*F18 F19 F20 F21 F22 F23 F24 F26 F27 F28*) mais ne peut distinguer entre les aspects *Car.1* et *Car.2* pour l'identification.

4.6.3 *scene.3*

La scène *scene.3*, illustrée en figure (4.12), est composée des aspests 3D *Chaise.1*, *Chaise.2*, *Avion.1*, *Table.1.*, *Table.2*, *Car.1* et *Car.3*. L'index est construit pour les aspects illustrés en figure (4.8) et (4.9).

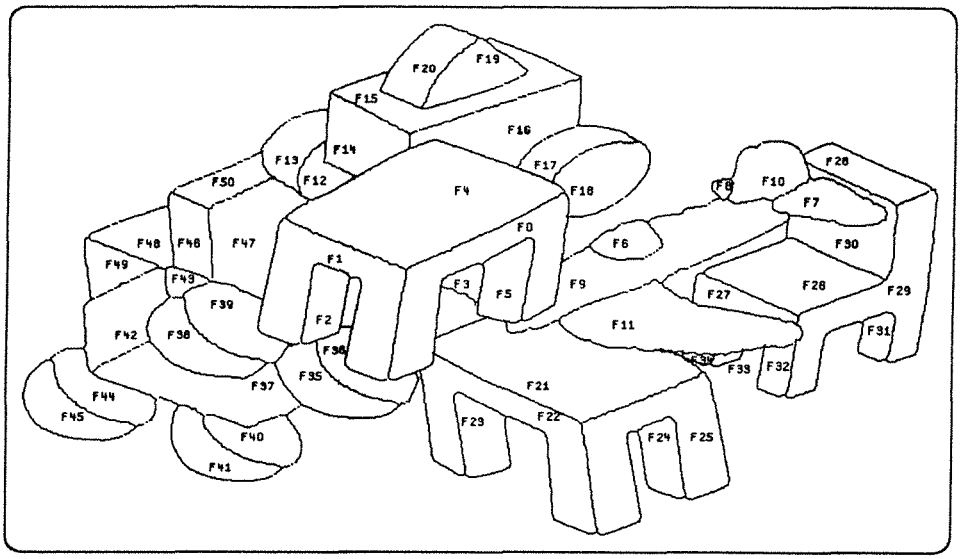


Figure 4.12. *scene.3 est composée des aspects Chaise.1, Avion.1 et Table.1.*

Résultats

Les groupes de primitives image	Les aspects conjecturés
(*F37 F40 F41 F42 F43 F44 F45*)	*Car.3, Car.4*
(*F14 F15 F16 F17 F18 F19 F20*)	*Car.1, Car.2*
(*F26 F27 F28 F29 F30 F31*)	*Chaise.1*
(*F21 F22 F23 F24 F25*)	*Table.1, Table.2* *Chaise.1, Chaise.2*
(*F0 F1 F2 F4 F5*)	*Table.1, Table.2* *Chaise.1, Chaise.2*
(*F6 F7 F9 F10 F11*)	*Avion.1, Avion.2*
(*F46 F47 F50*)	*Table.1, Table.2* *Chaise.1, Chaise.2*

Les groupes formés par le processus de reconnaissance correspondent aux différents aspects composant la scène. Seul l'aspect *Chaise.1* est correctement identifié.

4.7 En résumé

Le processus de reconnaissance forme correcte les groupes des aspects dont il a le modèle. L'étude de la scène *scene.2* indique que l'ajout d'un aspect non modélisé ne perturbe pas la reconnaissance des autres aspects de la scène.
Mais l'identification n'est pas toujours complète car pour un groupe de primitives il peut y avoir plusieurs hypothèses de modèles d'aspects. Ce manque de précision a plusieurs origines. D'une part, les aspects *Table.1* et *Table.2* étant décrits par des structures sous-isomorphes aux structures des aspects *Chaise.1* et *Chaise.2*, les concepts

terminaux pour les aspects $Table.1$ et $Table.2$ sont des concepts décrivant des structures également présentes dans les aspects $Chaise.1$ et $Chaise.2$. D'autre part, l'indexation ne décrit pas complètement la structure des aspects. Pour mener à bien l'identification, il faut en fait utiliser des informations de type métrique sur les faces et les indices.

5
Conclusions

Avant de conclure à propos de l'indexation, de son utilité pour la reconnaissance et du principe de fusion des étapes de *sélection* et d'*organisation* qu'elle implique, nous nous devons d'évaluer l'indexation à partir des expérimentations et des calculs de complexité des deux chapitres précédents.

5.1 Evaluation de l'indexation structurelle

Bien que les images observées ne soient que semi-réelles, les résultats des expériences de reconnaissance d'aspects permet de valider, du moins partiellement, l'indexation de la base des modèles sous forme de graphe de décision. En effet, on constate que l'occultation n'empêche pas la formation des groupes de primitives image. Et l'identification de ces groupes, si elle n'est pas complète, s'avère être une base tout à fait correcte pour l'utilisation d'information complémentaire sur la métrique des faces et des indices. En fait, les modèles d'aspects conjecturés au sien d'une même hypothèse, c'est-à-dire identifiant le même groupe de primitives-image, sont donc associés au même concept par différentes substitutions. Pour l'ensemble de ces modèles d'aspects et pour le groupe de primitives-image la structure générique du concept de l'hypothèse sert de référentiel pour des mesure complémentaires telles que les positions relatives ou le type de surface des primitives.
Nous avons également constaté que la reconnaissance d'une scène comportant un aspect non indexé (donc non modélisé) ne perturbe en rien la reconnaissance des autres aspects de la scène.

Quant au processus d'indexation, ses principaux atouts sont :

- l'ajout d'un aspect dans l'index se fait de façon purement incrémentielle.
- il n'y pas de paramètres de classification à fixer par l'utilisateur contrairement aux classifications numériques et conceptuelles ou aux graphes de décision.

Mais, il subsiste des inconvénients dus à la taille du graphe d'indexation qui peut être importante et le coût du processus de reconnaissance.

5.2 Les améliorations

Tel qu'est défini l'inex et les indices, et au vu des études précédentes, la taille du graphe d'indexation et du coût du processus de reconnaissance risquent d'être exponentiel. Ces coûts doivent être réduits sinon le graphe d'indexation risque de ne pas remplir son rôle comme guide lors du regroupement des primitives image en aspects identifiés dans la base des modèles. Par ailleurs, le processus de reconnaissance doit être le plus efficace possible en temps et les deux principaux facteurs de son coût sont le taux de recouvrement de la scène observée et la taille du graphe d'indexation. Il est primordial de minimiser la taille du graphe et d'envisager des architectures d'ordinateurs plus évoluées que celle utilisée à l'heure actuelle.

5.2.1 Restreindre la taille du graphe

Les règles E et IC regroupent les substitutions (ou les hypothèses) par k avec $k > 1$. Alors que la règle d'intersection IS les regroupe deux par deux. Aussi la première idée qui nous vient est d'utiliser une règle d'intersection n-aire et d'observer le comportement du processus de reconnaissance utlisant un tel graphe d'indexation. Ainsi la règle d'intersection regroupant les substitutions k par k se définit comme suit :

Définition 9 *La règle d'intersection pour l'indexation*

$$\frac{\left[\begin{array}{c} Proj(\sigma_1) \cap Proj(\sigma_2) \cap \ldots \cap Proj(\sigma_k) \neq \emptyset \\ \sigma_i \in \Sigma(C_i), \forall i \in [1..k] \end{array}\right]}{\left[\begin{array}{c} C_1, \ldots, C_k \\ \sigma_1, \ldots, \sigma_k \end{array}\right] \rightsquigarrow \left[\begin{array}{c} C \ \sigma \ tel \ que \\ Proj(\sigma) = \cup_{i=1}^{k} Proj(\sigma_i) \end{array}\right]}$$

Définition 10 *La règle d'intersection pour la reconnaissance*

$$\frac{\left[\begin{array}{c} Proj(\sigma_1) \cap Proj(\sigma_2) \cap \ldots \cap Proj(\sigma_k) \neq \emptyset \\ \sigma_i \in \Sigma(C_i), \forall i \in [1..k] \end{array}\right] \ \& \left[\begin{array}{c} Trouver \ C \ tel \ que \\ C \in fils(C_i) \forall i \in [1,k] \end{array}\right]}{\left[\begin{array}{c} C_1, \ldots, C_k \\ \sigma_1, \ldots, \sigma_k \end{array}\right] \rightsquigarrow \left[\begin{array}{c} Créer \ H \ te; \ que \\ HC = \wedge_{i=1}^{k} H_i C_i \end{array}\right]}$$

Notons que dans ce cas, le processus de reconnaissance doit être plus souple lors des formations des regroupements d'hypothèses. En effet, si, lors de l'indexation, la règle d'intersection est appliquée sur k substitutions pour construire un nouveau concept, lors de la reconnaissance, les k hypothèses correspondant aux k substitutions de l'indexation doivent se regrouper en une. Mais, compte tenu des occultations, il n'est pas certains que ces k hypothèses soient visibles et donc il faudra rechercher à apparier les hypothèses avec une partie seulement de la structure générique du concept construit

par application de la règle IS. Ceci revient en fait à développer le concept comme s'il avait été créé par des applications successives de la première version de la règle IS. Nous nous devons donc, avant de nous prononcer quant à l'efficacité de la règle d'intersection ainsi définie, observer des expérimentations portant sur un nombre important de scenes diverses et utilisant des bases de connaissance pour fournies et plus diverses que celles exposées dans le chapite précédent.

5.2.2 Accélérer la reconnaissance et l'indexation

Lors de la reconnaissance, chaque règle correspond à une action se déroulant en deux étapes : la condition et l'action. Premièrement, la condition de la règle forme des groupes d'hypothèses tels que leurs ensembles de primitives modèle soient identiques : c'est la condition de la règle d'égalité, ou qu'un ensemble recouvrent les autres : c'est la condition de la règle d'inclusion, ou encore qu'ils s'intersectent : c'est la condition de la règle d'intersection. Puis, si la condition est vérifiée, l'action proprement dite est déclanchée ; c'est-à-dire la construction d'une nouvelle hypothèse.

La seconde étape peut s'opérer de manière indépendante pour chacun des groupes d'hypothèses formés par la condition de la règle courante. Nous pouvons donc considérer que le traitement de chaque vérification est parallélisable. Ainsi, le processus peut fonctionner en deux cycles séquentiels. Premièrement, le regroupement des hypothèses actives : la partie condition des règles. Puis, pour chaque groupe, rechercher un concept de l'index dont la structure générique filtre l'élément structurel représentant ce groupe d'hypothèses : c'est la partie action des règles.

Mais remarquons tout de même que le coût reste exponentiel puisque le principal paramètre définissant l'efficacité du processus de reconnaissance provient de la taille du graphe. En effet si le graphe est de taille exponentielle, la phase de formation va être amenée à créer un nombre exponentiel de groupes d'hypothèses.

De même l'algorithme d'indexation procède en deux étapes :

- une phase de *fusion* des substitutions suivant la partie conditionelle de la règle courante,
- une phase de *construction* des concepts et des substitutions représentant ces fusions.

Pour chaque fusion de substitutions, la phase de construction effectue en premier la recherche d'un concept dont la structure générique filtre la fusion considérée. Si ce concept existe, l'élément structurel de la fusion devient une instance de ce concept sauf si la substitution existe déjà. En revanche si le concept n'existe pas, celui-ci et la substitution représentant le filtre sont créés.
Nous constatons que la phase de construction pour un groupe de substitutions ne se fait pas indépendamment de la phase de construction pourles autres groupes. En effet, il existe un lien entre les fusions réside :

- soit plusieurs éléments structurels de fusions sont filtrés par le même concept et ce concept n'est pas encore existant,

- soit plusieurs fusions définissent le même élément structurel.

Dans le premier cas, l'une des fusions va créer le concept tandis que l'autre l'utilise. Dans le deuxième cas, l'une des fusions crée la substitution et les autres l'utilise. Ceci nous amène à synchroniser les constructions de chacune des fusions afin de mettre à jour les concepts et les substitutions du graphe d'indexation.

5.3 Perspectives

A partir de ces diverses constatations, nous pouvons nous diriger vers une étude plus précise de l'indexation structurelle. Premièrement, il est évident que l'information utilisée pour décrire les aspects est insuffisante pour les distinguer tous parfaitement. Il faut utiliser, en complément, des informations plus précises qui comporte de attributs numérique telles que la position relatives des primitves, etc.. Ces informations peuvent être associées à chacun des concepts. En effet, la structure générique de chaque concept défini un référentiel idéal pour soit classer les substitutions lors de l'indexation, soit opérer l'appariment entre le groupe de primitives image et les modèles d'aspects lors de la reconnaissance.

Par ailleurs, nous nous devons d'étudier le comportement de l'indexation et de la reconnaissance à partir d'images réelles. Ce problème est très complexe car la segmentation en primitives n'est pas stables d'une acquisition à l'autre. La seule façon d'opérer correctement est de chercher en premier lieu à construire des primitives image stable pour l'indexation et la reconnaissance... Ce domaine est vaste en soi... Il fait référence à des domaines tels que la géométrie projective ou la géométrie différentielle.

Bibliographie

[Ahuja *et al.*, 1992] N. Ahuja, C. Dyer, O. Faugeras, K. Ikeuchi, R. Jain, J. Mundy, and A. Pentland. Workshop Panel Report : Why Aspect Graphs are not (yet) Pratical for Computer Vision. *Computer Vision and Image Processing*, 55(2):212–218, march 1992.

[Ayache and Faugeras, 1986] N. Ayache and O.D. Faugeras. HYPER: a New Approach for the Recognition and Positioning of 2D Objects. *IEEE Transactions on PAMI*, 8(1):44–54, 1986.

[Ballard and Brown, 1982] D.H. Ballard and C.M. Brown. *Computer Vision.* Prentice-Hall, Englewood Cliffs, New Jersey, 1982.

[Barrow and Tennenbaum, 1981] H.G. Barrow and J.M. Tennenbaum. Computational Vision. *Proceedings of IEEE*, 69:572–594, 1981.

[Berger, 1991] M.O. Berger. *Les contours actifs : modélisation, comportement et convergence.* Thèse de doctorat, Institut National Polytechnique de Lorraine, Vandœuvre-lès-Nancy, février 1991.

[Besl and Jain, 1985] P.J. Besl and R.C. Jain. Three-Dimensional Object Recognition. *Computing Surveys*, 17(1):936–958, 1985.

[Boufama, 1991] B. Boufama. Description et reconnaissance d'objets 3D à partir d'images de distances. Rapport de DEA, CRIN, 1991.

[Bouthemy, 1988] P. Bouthemy. Modèles et méthodes pour l'analyse du mouvement dans une séquence d'images. *Technique et Science Informatique*, 7(6):527–546, 1988.

[Brooks, 1981] R.A. Brooks. Symbolic Reasoning Among 3-D Models and 2-D Images. *Artificial Intelligence*, 17:285–348, 1981.

[Burns and Kitchen, 1987] J. B. Burns and L. J. Kitchen. An Approach to Recognition in 2D Images of 3D Objects from Large Model Bases. Technical Report COINS 87-85, University of Massachusetts at Amherst, 1987.

[Burns and Riseman, 1992] J.B. Burns and E.M. Riseman. Matching Complex Images to Multiple 3D Objects using View Description Networks. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Urbana Champaign, (USA)*, pages 328–334, 1992.

[Chabbi and Masini, 1991] H. Chabbi and G. Masini. A Combined Use of Regions and Segments to Construct Facets. In *International Conference on Image Analysis and Processing, Como (Italy)*, pages 334–338, september 1991.

[Chabbi, 1992] H. Chabbi. A New Approach for Finding 3D Planar Faces by Stereovision. In *IAPR International Workshop on Structural and Syntactic Pattern Recognition, Bern (Switzerland)*, 1992.

[Clemens and Jacobs, 1991] D.T. Clemens and D.W. Jacobs. Space and Time Bounds on Indexing 3-D Models from 2-D Images. *IEEE Transactions on PAMI*, 13(10):1007–1017, 1991.

[Clowes, 1971] M.B. Clowes. On Seeings Things. *Artificial Intelligence*, 2:79–116, 1971.

[Decaestecker, 1989a] C. Decaestecker. Formation incrémentale de concepts par un critère d'adéquation. In *Journée Française de l'Apprentissage*, pages 63–77, 1989.

[Decaestecker, 1989b] C. Decaestecker. Incremental Concept Formation with Attribut Selection. In K. Morik, editor, *European Working Session on Learning*, pages 49–58, December 1989.

[Diday *et al.*, 1982] E. Diday, J. Lemaire, J. Ponget, and F. Testu. *Eléments d'analyse de données.* Dunod, 1982.

[Draper and Riseman, 1990] B. A. Draper and E. M. Riseman. Learning 3D Object Recognition Strategies. In *Proceedings of 3rd International Conference on Computer Vision, Osaka (Japan)*, pages 320–324, 1990.

[Droulez, 1991] J. Droulez. Le mouvement à l'origine de l'intelligence? *Science et Vie*, 177:52–60, 1991.

[Duda and Hart, 1973] R.O. Duda and P.E. Hart. *Pattern Classification and Scene Analysis.* Wiley-InterScience, 1973.

[Fan *et al.*, 1989] T.J. Fan, G. Medioni, and R. Nevatia. Recognizing 3D Objects Using Surface Descriptions. *IEEE Transactions on PAMI*, 11(11):1140–1157, 1989.

[Fan, 1988] T.J. Fan. *Describing and Recognizing 3D Objects Using Surface Properties.* PhD thesis, Institute for Robotics and Intelligence Systems, University of Southern California, Los Angeles, California 90089-0273, august 1988.

[Faugeras, 1988] O.D. Faugeras. Quelques pas vers la vision artificielle en trois dimensions. *Technique et Science Informatique*, 7(6):547–590, 1988.

[Feldman, 1985] J. A. Feldman. Connectionist Models and Parallelism in High Level Vision. *Computer Vision, Graphics and Image Processing*, 31:178–200, 1985.

[Fisher, 1987] D.H. Fisher. Knowledge Acquisition via Incremental Conceptual Clustering. *Machine Learning*, 2:139–172, 1987.

[Flynn and Jain, 1991] P. J. Flynn and A. K. Jain. BONSAI: Object Recognition Using Constrained Search. *IEEE Transactions on PAMI*, 13(10):1066–1075, October 1991.

[Fréganc and Schalchli, 1991] Y. Fréganc and L. Schalchli. La conscience, un phénomène oscillatoire ? *Science et Vie*, 177:65–75, 1991.

[Fukushima, 1988] K. Fukushima. A Neural Netwok for Visual Pattern Recognition. *IEEE COMPUTER Magazine*, 21(3):65–75, March 1988.

[Glaser, 1992] N. Glaser. Modeling and Interpreting 3D Polyhedral Scenes. Rapport de stage, Centre de Recherche en Informatique de Nancy, 1992. Egalement : Diplomarbeit im Studienfach Informatik, Institut für Mathematische Maschinen und Datenverarbeitung, Friedrich-Alexander-Universität, Erlangen-Nürnberg, BR Deutschland.

[Grimson and Lozano-Pérez, 1987] W.E.L. Grimson and T. Lozano-Pérez. Localizing Overlapping Parts by Searching the Interpretation Tree. *IEEE Transactions on PAMI*, 9(4):469–482, 1987.

[Grimson, 1988] W. E. L. Grimson. The Combinatorics of Objects Recognition in Cluttered Environments using Constrained Search. *Artificial Intelligence*, 44:218–227, 1988.

[Grimson, 1990] W.E.L. Grimson. The Effect of Indexing on the Complexity of Object Recognition. In *Proceedings of 3rd International Conference on Computer Vision, Osaka (Japan)*, pages 644–651, 1990.

[Grimson, 1991] W.E.L. Grimson. The Combinatorics of Heuristic Search Termination for Object Recognition in Cluttered Environments. *IEEE Transactions on PAMI*, 13(9):920–935, September 1991.

[Guisser *et al.*, 1991] L. Guisser, R. Payrissat, and S. Castan. Perception 3D de surfaces d'objets par projection d'une grille. In *Actes 8ème Congrès AFCET de Reconnaissance des Formes et Intelligence Artificielle, Lyon*, volume 2, pages 771–789, 1991.

[Huffman, 1971] D.A. Huffman. Impossible Objects as Nonsense Sentences. *Machine Intelligence*, 6, 1971.

[Idesawa *et al.*, 1977] M. Idesawa, T. Yatagai, and T. Soma. Scanning Moiré Method and Automatic Measurement of 3D Shapes. *Applied Optics*, 16(8):2152–2162, August 1977.

[Ikeuchi, 1987] K. Ikeuchi. Precompiling a Geometrical Model into an Interpretation Tree for Object Recognition in Bin-picking tasks. In *Image Understanding, Los Angeles*, pages 321–339, february 1987.

[Jarvis, 1983] R.T. Jarvis. A Perspective on Range Finding Techniques for Computer Vision. *IEEE Transactions on PAMI*, 5(2):122–139, 1983.

[Kononenko, 1989] I. Kononenko. ID3, Sequential Bayes, Naïve Bayes and Bayesian Neural Networks. In *European Working Session on Learning*, pages 91–98, 1989.

[Lamdan and Wolfson, 1988] Y. Lamdan and H.J. Wolfson. Geometric Hashing: A General and Efficient Model-Based Recognition Scheme. *Proceedings of 2nd International Conference on Computer Vision, Tampa, FL (USA)*, pages 238–249, 1988.

[Lebowitz, 1987] M. Lebowitz. Experiments with Incremental Concept Formation: UNIMEM. *Machine Learning*, 2:103–138, 1987.

[Lebowitz, 1988] M. Lebowitz. Deferred Commitment in UNIMEM: Waiting to Learn. In *International Machine Learning*, pages 80–86, 1988.

[Lowe, 1985a] D. G. Lowe. *Perceptual Organization and Visual Recognition.* Kluwer Academic Publishers, Norwell, Massachusetts, 1985.

[Lowe, 1985b] D. G. Lowe. Visual Recognition from Spatial Correspondence and Perceptual Organization. In *Proceedings of 9th International Joint Conference on Artificial Intelligence, Los Angeles, CA (USA)*, pages 953–959, 1985.

[Lowe, 1986] D. Lowe. Three-Dimensional Object Recognition from Single Two-Dimensional Images. *Artificial Intelligence*, pages 355–395, 1986.

[Mackworth, 1973] A. K. Mackworth. Interpreting Pictures of Polyhedral Scenes. *Artificial Intelligence*, 4:121–137, 1973.

[Mackworth, 1977a] A. K. Mackworth. How to See a Simple World : an Exegis of Some Computer Programs for Scene Analysis. *Machine Intelligence*, 8:510–537, 1977.

[Mackworth, 1977b] A.K. Mackworth. Consistency in Networks of Relations. *Artificial Intelligence*, 8:99–118, 1977.

[Marr, 1982] D. Marr. *Vision.* W.H. Freeman, San Francisco, 1982.

[Masini and Zaroli, 1984] G. Masini and F. Zaroli. Présentation de TRIDENT : un système d'interprétation d'images tridimensionnelles. In *Actes 4ème Congrès AFCET de Reconnaissance des Formes et Intelligence Artificielle, Paris*, pages 179–189, 1984.

[Michalski and Stepp, 1983] R.S. Michalski and R.E. Stepp. Learning from Observation: Conceptual Clustering. In Tom M. Mitchell, editor, *Machine Learning.* Tioga Publishing Company, Palo Alto, California, 1983.

[Minsky, 1975] M. Minsky. A Framework for Representing Knowledge. In P. Winston, editor, *The Psychology of Computer Vision*, pages 211–281. McGraw Hill, New York, 1975.

[Mohr and Henderson, 1986] R. Mohr and T.C. Henderson. Arc and Path Consistency Revisited. *Artificial Intelligence*, 28:225–233, 1986.

[Mulgaonkar *et al.*, 1984] P. Mulgaonkar, L. G. Shapiro, and R. M. Haralick. Matching "Stick, Plates and Blobs" Objects Using Geometric and Relational Constraints. *Image and Vision Computing*, 2(2):85–98, may 1984.

[Paris and Boufama, 1992] S. Paris and B. Boufama. Concerning the Effect of the Indexing on the Recognition. In *IAPR International Workshop on Structural and Syntactic Pattern Recognition, Bern (Switzerland)*, August 1992.

[Paris and Masini, 1991] S. Paris and G. Masini. 3D Modeling: a Step Towards Self-government. In *International Conference on Image Analysis and Processing, Como (Italy)*, pages 221–225, 1991.

[Paris, 1991] S. Paris. Modélisation structurelle par apprentissage. In *Actes 8ème Congrès AFCET de Reconnaissance des Formes et Intelligence Artificielle, Lyon*, volume 3, pages 1345–1350, 1991.

[Quinlan, 1986] J.R. Quinlan. Induction of Decision Trees. *Machine Learning*, 1:81–106, August 1986.

[Shapiro and Haralick, 1982] L.G. Shapiro and R.M. Haralick. Organization of Relational Models of Scene Analysis. *IEEE Transactions on PAMI*, 4(6):595–602, 1982.

[Shapiro, 1985] L.G. Shapiro. A Fast Structural Matching Algorithm with Applications in Stereo Vision. In *Proceedings of 4th Scandinavian Conference on Image Analysis, Trondheim (Norway)*, pages 159–166, 1985.

[Smith and Kanade, 1985] D.R. Smith and T. Kanade. Autonomous Scene Description with Range Imagery. *Computer Vision, Graphics and Image Processing*, 31:322–334, 1985.

[Sossa and Horaud, 1992] H. Sossa and R. Horaud. Model Indexing: The Graph-hashing Approach. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Urbana Champaign, (USA)*, pages 811–814, 1992.

[Stark and Bowyer, 1991] L. Stark and K. Bowyer. Achieving Generalized Object Recognition through Reasoning about Association of Function to Structure. *IEEE Transactions on PAMI*, 13(10):1097–1104, 1991.

[Stark and Bowyer, 1992] L. Stark and K. Bowyer. Indexing Function-Based Categories for Generic Recognition. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Urbana Champaign, (USA)*, pages 795–797, 1992.

[Stein and Médioni, 1990] F. Stein and G. Médioni. Efficient Two Dimensional Object Recognition. In *Proceedings of 10th International Conference on Pattern Recognition, Atlantic City, NJ (USA)*, volume 1, pages 13–17, 1990.

[Mohr and Henderson, 1986] R. Mohr and T.C. Henderson. Arc and Path Consistency Revisited. *Artificial Intelligence*, 28:225–233, 1986.

[Mulgaonkar *et al.*, 1984] P. Mulgaonkar, L. G. Shapiro, and R. M. Haralick. Matching "Stick, Plates and Blobs" Objects Using Geometric and Relational Constraints. *Image and Vision Computing*, 2(2):85–98, may 1984.

[Paris and Boufama, 1992] S. Paris and B. Boufama. Concerning the Effect of the Indexing on the Recognition. In *IAPR International Workshop on Structural and Syntactic Pattern Recognition, Bern (Switzerland)*, August 1992.

[Paris and Masini, 1991] S. Paris and G. Masini. 3D Modeling: a Step Towards Self-government. In *International Conference on Image Analysis and Processing, Como (Italy)*, pages 221–225, 1991.

[Paris, 1991] S. Paris. Modélisation structurelle par apprentissage. In *Actes 8ème Congrès AFCET de Reconnaissance des Formes et Intelligence Artificielle, Lyon*, volume 3, pages 1345–1350, 1991.

[Quinlan, 1986] J.R. Quinlan. Induction of Decision Trees. *Machine Learning*, 1:81–106, August 1986.

[Shapiro and Haralick, 1982] L.G. Shapiro and R.M. Haralick. Organization of Relational Models of Scene Analysis. *IEEE Transactions on PAMI*, 4(6):595–602, 1982.

[Shapiro, 1985] L.G. Shapiro. A Fast Structural Matching Algorithm with Applications in Stereo Vision. In *Proceedings of 4th Scandinavian Conference on Image Analysis, Trondheim (Norway)*, pages 159–166, 1985.

[Smith and Kanade, 1985] D.R. Smith and T. Kanade. Autonomous Scene Description with Range Imagery. *Computer Vision, Graphics and Image Processing*, 31:322–334, 1985.

[Sossa and Horaud, 1992] H. Sossa and R. Horaud. Model Indexing: The Graph-hashing Approach. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Urbana Champaign, (USA)*, pages 811–814, 1992.

[Stark and Bowyer, 1991] L. Stark and K. Bowyer. Achieving Generalized Object Recognition through Reasoning about Association of Function to Structure. *IEEE Transactions on PAMI*, 13(10):1097–1104, 1991.

[Stark and Bowyer, 1992] L. Stark and K. Bowyer. Indexing Function-Based Categories for Generic Recognition. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Urbana Champaign, (USA)*, pages 795–797, 1992.

[Stein and Médioni, 1990] F. Stein and G. Médioni. Efficient Two Dimensional Object Recognition. In *Proceedings of 10th International Conference on Pattern Recognition, Atlantic City, NJ (USA)*, volume 1, pages 13–17, 1990.

[Stein and Médioni, 1992] F. Stein and G. Médioni. Structural Indexing : Efficient 3D Object Recognition. *IEEE Transactions on PAMI*, 14(2):125–145, february 1992.

[Suetens *et al.*, 1992] P. Suetens, P. Fua, and A. J. Hanson. Computational Strategies for Object Recognition. *Computing Surveys*, 24(1):5–61, march 1992.

[Thirion, 1989] E. Thirion. *Interprétation et apprentissage géométrique en vision par ordinateur.* Thèse de doctorat, Institut National Polytechnique de Lorraine, Vandœuvre-lès-Nancy, 1989.

[Thorpe, 1988] S. J. Thorpe. Traitement d'images chez l'homme. *Technique et Science Informatique*, 7(6):517–525, 1988.

[Thorpe, 1991] S. Thorpe. Vision Artificielle : un problème insoluble ? *Science et Vie*, 177:61–64, 1991.

[Vogel and Wong, 1979] M. A. Vogel and A. K. C. Wong. PFS Clustering Method. *IEEE Transactions on PAMI*, 1(3):237–244, 1979.

[Wallace and Kanade, 1990] R.S. Wallace and T. Kanade. Finding Natural Clusters having Minimum Description Length. In *Proceedings of 10th International Conference on Pattern Recognition, Atlantic City, NJ (USA)*, pages 438–442, June 1990.

[Wallace, 1989] R. S. Wallace. *Finding Natural Clusters through Entropy Minimization.* PhD thesis, School of Computer Science, Carnegie Mellon University, Pittsburg, PA 15213, June 1989.

[Waltz, 1975] D. Waltz. Understanding Line Drawings of Scenes with Shadows. In P.H. Winston, editor, *The Psychology of Computer Vision.* McGraw-Hill, New York, 1975.

[Weymouth, 1986] T.E. Weymouth. Using Object Description in a Schema Network for Machine Vision. COINS Technical Report 86–24, University of Massachusetts, 1986.

[Wrobel-Dautcourt, 1988] B. Wrobel-Dautcourt. *Perception de la distance par mise en correspondance de régions dans des images stéréoscopiques.* Thèse de doctorat, Institut National Polytechnique de Lorraine, 1988.

[Zaroli, 1983] F. Zaroli. *Réalisation d'un système d'analyse et d'interprétation d'images tridimensionnelles.* Thèse de 3ème Cycle, Université de Nancy 1, 1983.

[Zeki, 1991] S. Zeki. La construction des images par le cerveau. *La Recherche*, 1991.

[Zhao, 1991] F. Zhao. Machine Recognition as Representation and Search — A Survey. *International Journal of Pattern Recognition and Artificial Intelligence*, 5(5):715–747, 1991.

Résumé

La vision par ordinateur est amenée, pour des tâches de manipulation robotique, à identifer et à localiser les objets d'une scène observée. L'identification et la localisation, regroupées sous le terme-reconnaissance, implique une connaissance a priori des objets à manipuler. Cette connaissance se traduit en une base de modèles d'objets..

Pour des raisons d'accessibilité à la base de connaissance, nous proposons à travers ce mémoire d'étudier la classification des modèles suivant leur information structurelle. Cette classification structurelle, appelée *index* est produit automatiquement et incrémentiellement par un processus d'*indexation*.