Reconnaissance et modélisation d'objets 3D à l'aide d'invariants projectifs et affines

Bart Lamiroy

THÈSE

présentée par

Bart LAMIROY

pour obtenir le grade de **DOCTEUR**

de l'**INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE**

(Arrêté ministériel du 30 Mars 1992)

Spécialité : **Informatique**

Reconnaissance et modélisation d'objets 3D à l'aide d'invariants projectifs et affines

---

Date de soutenance : 8 juillet 1998

Composition du jury :

Président :	**Annick MONTANVERT**
Rapporteurs :	**Henri MAÎTRE**
	Marc RICHETIN
Examinateurs :	**Patrick GROS**
	Radu HORAUD
	Roger MOHR

Thèse préparée au sein du laboratoire GRAVIR–IMAG et INRIA Rhône-Alpes
sous la direction de Radu HORAUD et Patrick GROS

Une phrase d'esprit donne toujours l'impression que ce qui suit l'est aussi.

Remerciements

Je tiens tout d'abord à remercier Radu HORAUD pour m'avoir permis de faire mes « premiers pas » dans le domaine de la vision, il y a maintenant déjà quelques années, puis pour m'avoir fait confiance tout au long de ma présence dans l'équipe MOVI. Je remercie également, et de façon aussi enthousiaste, Patrick GROS pour ses encouragements, sa disponibilité et ses relexions pertinentes et constructives pendant ces années ainsi que pour son amitié et l'intérêt constant qu'il a porté à mon travail.

Je remercie vivement les personnes qui m'ont fait l'honneur d'avoir participé à mon jury : mes rapporteurs MM. Henri MAÎTRE et Marc RICHETIN pour leurs commentaires constructifs sur le manuscrit, MME. Annick MONTANVERT pour l'avoir présidé, ainsi que M. Roger MOHR d'avoir été examinateur.

Roger MOHR a également été un bon chef et collègue et je le remercie pour la confiance qu'il m'a accordée, sur le plan scientifique d'une part, et sur le plan technique et administratif d'autre part, lorsqu'il m'a permis de prendre de réelles responsabilités dans son équipe.

Merci aussi au personnel de l'INRIA Rhône-Alpes d'avoir assuré nos exceptionnelles conditions de travail, tout particulièrement notre assistante Danièle HERZOG, pour son efficacité et sa bonne humeur à toute épreuve.

Il est difficile de citer toutes les personnes que j'ai côtoyées et qui ont pu m'aider. Je tiens à remercier toute l'équipe MOVI pour l'ambiance chaleureuse qui règne dans son sein. Je la remercie plus particulièrement de m'avoir pardonné mes sauts d'humeur quand mes charges d'administration était lourdes, et pour l'échange scientifique constant et enrichissant qui la caractérise.

Merci en particulier à mon ami et co-bureau Jérôme BLANC pour nos nombreux fous rires. Il ne manquera pas de remarquer le clin d'œil que je lui fais dans ces remerciements. J'espère que l'amitié qui s'est construite pendant ces années perdurera maintenant que nos chemins se sont séparés.

Merci également à Sylvaine PICARD pour la collaboration fructueuse et les échanges scientifiques et épistémologiques qui ont beaucoup contribué à ce travail, et à Yves DUFOURNAUD qui, avec ses questions et rélexions techniques, a su me surprendre et qui m'a incité à aller toujours plus loin dans la réflexion scientifique.

Laurence, tu prends une place de choix dans ces remerciements. Tu m'aimes et tu me soutiens, et tu m'as fait réaliser, non sans mal, que la recherche n'est qu'un travail comme un autre et que la chose principale reste que notre famille grandissante passe toujours avant tout ... Je t'aime.

Sommaire

Table des figures

Liste des algorithmes

Introduction

L'INTERACTION automatisée d'un ordinateur avec son environnement a toujours été un des buts fondamentaux dans les domaines touchant à ce qu'on appelle parfois l'intelligence artificielle. La vision par ordinateur ne fait pas exception. En effet, à partir du moment où l'on peut imaginer qu'un système autonome puisse fournir une description sémantique de son environnement, une infinité d'applications intéressantes surgissent dans son sillage.

Il n'est donc pas surprenant que dans le domaine de la vision artificielle, l'obtention automatique d'une description sémantique ait été un des buts premiers de cette discipline. De multiples approches ont été proposées, sans pour autant fournir un environnement générique et exploitable de reconnaissance d'objets. Dans cette thèse, nous introduisons un nouveau mode opératoire qui pourra servir à résoudre une sous-classe de problèmes de reconnaissance : l'identification et la localisation dans une image d'instances d'objets connus et déjà observés. Nous aborderons une méthode de modélisation et de reconnaissance basée sur une segmentation préalable, et nous fournirons les preuves de son utilité dans une approche de reconnaissance générique dont elle est une instance particulière. Plus globalement, notre méthode fait partie des systèmes de reconnaissance par l'apparence, et elle s'appuie sur l'indexation de descripteurs locaux.

Ce travail n'a donc pas la prétention de conférer à un algorithme une « *intelligence* » de reconnaissance générique permettant de résoudre un problème persistant et difficile depuis des décennies. Il fournit néanmoins une base exploitable dans une approche empirique et constructive de la vision par ordinateur, et propose une nouvelle méthode pour aborder des aspects de la modélisation et la mise en correspondance dans le domaine de la reconnaissance calculatoire basée sur l'apparence.

Contexte historique

Aux débuts de la vision artificielle, on a cherché à trouver une représentation algorithmique, fonctionnelle et automatisable de la vision biologique. On a cru pendant longtemps – et ne le croit-on pas encore ? – que les processus de notre cortex visuel pourraient être analysés, formalisés et représentés sous forme d'algorithmes.

Très vite, on s'est rendu compte que cette représentation était complexe, et qu'elle dépendait d'un nombre important de facteurs. Alors sont apparus les problèmes de la seg-

mentation, de la représentation des données et de la modélisation des structures visuelles. Puis, s'est imposé le constat du non-avancement persistant dans le domaine [28, 85], et on est venu à la constatation que la notion de reconnaissance elle-même n'était pas définie. Forte de cette avancée, et consciente de l'origine neurobiologique de la vision, la recherche s'est orientée vers une définition utilisable de la vision et de la reconnaissance, aboutissant au fameux paradigme de MARR. Ce paradigme a réussi à mettre en évidence, une bonne fois pour toutes, les liens existant entre les représentations sémantiques du monde dans lequel nous évoluons, et les impulsions chromatiques qui forment une image. Dans la suite logique de ce formalisme ont émergé des approches, tentatives, contre-méthodes et autres représentations essayant de mettre en œuvre le fameux paradigme à quatre niveaux. Les uns avec un certain succès, les autres avec moins. L'étonnant étant que, au bout de quelques années d'activité effrénée, le constat était le suivant : les progrès des prétendus « sous-problèmes » de la segmentation, de la représentation des données et de la modélisation des structures visuelles avaient été considérables... mais l'automatisation de la reconnaissance au sens biologique du terme était loin d'être atteinte, malgré des structures hautement parallèles et optimisées qui avaient été mises en œuvre.

FIG. 1: *L'illusion de la reconnaissance.*

Des approches différentes ont alors commencé à émerger. En constatant qu'une copie du processus biologique complexe n'était guère réalisable puisque, d'une part, le processus lui-même n'était pas connu, et que, d'autre part, les technologies dont on disposait [1] étaient incapables de reproduire les connaissances partielles dont on disposait, on a formulé d'autres définitions de la reconnaissance. Plutôt que de laisser le système automatisé

1. ... et même celles dont on dispose actuellement !

fournir la sémantique d'une scène, le problème était allégé en cherchant à décrire des objets de façon plus concrète avec une terminologie plus « bas niveau ». Une nouvelle approche à la reconnaissance était développée, plus restrictive et algorithmique, et moins axée vers la reproduction des capacités visuelles biologiques. Dans ce contexte la reconnaissance était alors considérée comme une *mise en correspondance* entre indices visuels, et indices similaires d'un modèle auquel était ajoutée sa sémantique de façon autoritaire, et auquel on ne demandait plus au système automatique de la trouver. Nous utiliserons le terme « *géométrique* » pour ces approches. Elles essayaient principalement de résoudre un problème d'identification et d'alignement ; c'est-à-dire que, classiquement, on cherchait à identifier dans une image les instances d'un ou plusieurs modèles ainsi que leur position par rapport à la caméra.

Le fait de vouloir résoudre les deux problèmes (identification et positionnement) a introduit un certain nombre de questions et a contribué à une augmentation de la complexité qui était inutile pour une gamme d'applications pour lesquelles la position des objets 3D importait peu ; par exemple l'indexation et la consultation de base d'images par leur contenu. Dans les dernières années est donc apparu un nouveau type de modélisation : la modélisation par l'apparence. C'est dans cette approche que s'inscrit cette thèse.

Contexte scientifique

Jusqu'à récemment, c'est-à-dire, avant l'émergence des modélisations basées sur l'apparence, la reconnaissance dépendait d'une modélisation abstraite, imposée par celui qui concevait le système ayant à effectuer la reconnaissance. La principale difficulté, si ce n'était l'unique, était alors de trouver les liens entre ce qui était modélisé et ce qui était observé, et qui par définition était imparfait, bruité dans le processus d'acquisition (bruit électronique, saturation des capteurs, discrétisation du signal) et déformé par rapport au modèle de transition du 3D au 2D (distorsions optiques). Les plus grands besoins étaient alors d'obtenir une segmentation [2] précise, malgré le bruit, des occultations et des variations au sein d'une classe d'objets, et de modéliser le processus de formation de l'image afin de rendre compte des déformations projectives ou des variations de l'aspect visuel selon le point de vue. Dans les approches hiérarchiques, le besoin d'une segmentation hautement fiable était exacerbé par le fait qu'elle formait la clef de voûte de la structure et que sans elle aucune reconnaissance n'était envisageable. Les effets de perspective et de changement de point de vue étaient absorbés par la relative flexibilité de la description des objets. Les approches géométriques, quant à elles, toléraient dans leur conception des erreurs ou des défauts dans la détection des primitives. Par contre, cette prise en compte du processus de formation de l'image les rendait numériquement fragiles et les contraignait à prendre une modélisation rigide des objets afin que celle-ci puisse s'intégrer dans les concepts mathématiques de la géométrie projective.

Outre ces points-là, un autre aspect fondamental de ces approches réside dans le fait qu'elles nécessitent une modélisation explicite. Cette modélisation, généralement 3D, était

2. Par segmentation nous entendons extraction de tout indice de « bas niveau » : point, contour, segment, région...

une vision abstraite et parfaite de l'objet à reconnaître, et ne reflétait pas nécessairement le perçu, d'où une difficulté accrue de mise en correspondance des primitives observées et de celles modélisées.

Les méthodes de modélisation basées sur l'apparence tentent de résoudre toutes les carences de ces méthodes. Si les problèmes se situent au niveau de l'abstraction du modèle et l'extraction des indices, ainsi qu'au niveau du passage 2D-3D par la modélisation des caméras et la formation des images, pourquoi ne pas mettre modèle et image inconnue au même niveau ? En utilisant les images elles-mêmes comme des modèles, en essayant de les identifier à d'autres images, toute l'intégration de la formation de l'image et des difficultés liées au 3D est implicitement réalisée.

Exposé de l'approche

Le travail de cette thèse est présenté en deux parties principales, la première aborde le problème de la modélisation de la reconnaissance dans un but opérationnel, fournissant des algorithmes et des méthodes de représentation des données. La seconde fait une analyse algorithmique du problème de complexité, sous-jacent aux approches choisies.

Dans la première partie, nous proposons une nouvelle méthode de reconnaissance par indexation de descripteurs locaux. Elle fait partie des approches basées sur l'apparence, et modélise un objet 3D par plusieurs vues 2D. Dans chaque vue, nous calculons des indices locaux qui servent à caractériser des configurations géométriques représentatives pour l'image. La reconnaissance consistera donc à identifier des configurations d'une image inconnue à celles d'une image modèle (aussi appelée modèle, pour des raisons de commodité). Afin de faciliter cette identification nous calculons sur les configurations des valeurs invariantes (ou presque) qui nous permettront de les indexer. Cette indexation permettra, par la suite, de trouver rapidement les correspondants possibles des configurations d'une image inconnue.

Notre approche s'appuie sur une segmentation préalable en contours polygonaux de l'image. Ces contours sont regroupés par paires de segments adjacents pour lesquels on calcule l'angle et le rapport de leurs longueurs. Ces valeurs serviront de clé d'indexation et de mesure de similarité entre les configurations de deux images. La seule donnée de ces appariements ne suffit pas pour décider de l'appartenance d'une image à un modèle. Nous avons donc introduit une contrainte de vérification des mises en correspondance initiales basée sur le calcul du mouvement apparent entre l'image et son modèle présumé. La cohérence de ce mouvement avec les appariements permet de décider de quel modèle l'image observée est une instance.

Nous observons ensuite que cette vérification par cohérence du mouvement apparent est en fait indépendant de la façon dont la mise en correspondance des configurations locales est faite. Or, la principale différence des méthodes d'indexation de descripteurs locaux est justement la caractérisation permettant d'apparier des configurations. On en déduit donc que ce mouvement apparent peut être calculé pour toutes les approches, et peut servir de ce fait comme facteur unifiant pour les méthodes locales. Nous développons donc un schéma de reconnaissance dans lequel différentes approches coopèrent afin de

fournir une meilleure mise en correspondance et donc une meilleure reconnaissance.

Dans la deuxième partie de cette thèse, nous analysons la complexité algorithmique d'un processus générique de reconnaissance par indexation de descripteurs locaux. Nous constatons que l'idée, généralement répandue, que l'augmentation de la dimension de l'espace d'indexation réduit le temps de reconnaissance (puisqu'elle diminue le nombre de collisions lors d'une requête dans la base d'indexation) est fausse dans le cas où une gestion de l'erreur nécessite qu'un voisinage autour d'une clef d'indexation soit considéré, et non plus un unique endroit défini par cette clef. Dans ces cas-là, le coût d'accès à un voisinage de dimension n devient exorbitant par rapport au gain obtenu par la réduction des collisions. En étudiant le phénomène, nous constatons qu'il existe, pour une complexité d'images donnée, une taille des index garantissant l'optimalité du temps d'exécution. Nous finissons cette partie en proposant une façon de réduire la complexité en combinant des index contraints à une gestion d'erreur, et des index « discrets » permettant de ne pas accéder à une zone trop étendue dans l'espace d'indexation.

Contributions

Cette thèse comporte trois contributions principales.

- Elle développe un méthode de mise en correspondance d'images structurées par indexation de descripteurs locaux. Cette mise en correspondance est basée sur une modélisation par quasi-invariants d'une part, et une vérification de la cohérence globale par estimation du mouvement apparent. Sa conception autour d'un paradigme géométrique fort permet de garantir que l'approche est totalement indépendante de toute transformation rigide et changement d'échelle que pourrait subir l'image. De plus, la modélisation par descripteurs locaux basés sur des contours, garantit qu'elle est également robuste à des occultations et à des changements d'éclairage. Les descripteurs sont organisés dans une structure d'indexation dynamique que nous avons développée et qui permet un accès rapide aux valeurs stockées ainsi qu'aux valeurs similaires suivant un critère d'incertitude. C'est grâce à cette indexation que nous sommes en mesure de mettre en œuvre l'identification d'une image à partir de plusieurs modèles potentiels. La même structure d'indexation est par ailleurs utilisée pour calculer efficacement une approximation du mouvement apparent entre image et modèles, permettant d'exprimer une valeur de cohérence relative au modèle trouvé.

- Elle fournit une approche unifiant une partie des méthodes par indexation de descripteurs locaux connues, en les intégrant dans sa conception géométrique. Ainsi elle permet de faire coopérer différentes méthodes créant une approche de reconnaissance qui est applicable à une plus grande catégorie d'images.

- Elle analyse en détail les fondements algorithmiques des méthodes de reconnaissance par indexation de descripteurs locaux, et conclut que certaines conceptions *a priori* sont fausses dans les cas où l'indexation n'est pas exacte et qu'une gestion du bruit doit être mise en œuvre. Dans ces cas-là, elle montre que la complexité algorithmique est exponentielle dans la taille des descripteurs, ce qui limite sévèrement son

champ d'application pour des problèmes à grande dimension tels que, par exemple, les descripteurs photométriques intégrant la couleur. Elle montre que, pour qu'une méthode d'indexation soit efficace, la dimension optimale des descripteurs est donnée par leur nombre dans une image. Si l'on doit à tout prix réduire cette complexité, la seule solution est d'intégrer des dimensions ne nécessitant pas de gestion d'erreur.

Contenu des différents chapitres

L'exposé peut être divisé en trois parties principales. Le premier et le second chapitre tiennent lieu de motivation, situation de la problématique générale et état de l'art. Le troisième chapitre représente la première partie de notre travail, dans laquelle nous présentons une nouvelle approche pour aborder la reconnaissance. Dans le quatrième chapitre, et dernière partie de notre travail, nous fournissons une analyse plus algorithmique du problème de la reconnaissance par indexation dont notre méthode, proposée dans le chapitre 3, fait partie.

Chapitre 1 Dans le premier chapitre nous proposons un bref survol historique de l'évolution du problème de la reconnaissance. Commençant par la formalisation de D. MARR et les mises en œuvre qui en sont issues, et passant par les approches géométriques basées sur l'alignement d'une image composée de primitives de bas niveau, avec un modèle 3D de primitives similaires, nous analysons les deux écoles. Notre conclusion est que l'évolution naturelle des méthodes de reconnaissance s'oriente vers la modélisation par l'apparence.

Chapitre 2 Ce chapitre est dédié à la reconnaissance par apparence. En quoi consiste-t-elle ? Quels problèmes résout-elle ? Quelles sont les différentes approches qui la composent ? Nous ferons plus particulièrement, une distinction entre les méthodes globales et les méthodes locales, ces dernières formant le groupe dans lequel s'inscrit notre approche présentée dans le chapitre 3.

Chapitre 3 Nous abordons dans le troisième chapitre une nouvelle façon de mettre en œuvre la reconnaissance dans le contexte d'une approche locale. Nous partons pour cela de deux constatations. Dans une première étape, nous présentons un algorithme de mise en correspondance et de modélisation d'objets à l'aide de *quasi-invariants*. Cet algorithme a été développé par GROS dans sa thèse [35]. Nous faisons l'analyse de l'utilisation de ces quasi-invariants, que nous définirons, et nous constatons que cette méthode de mise en correspondance se généralise de façon naturelle en une méthode de reconnaissance. Cette méthode est basée sur l'indexation des descripteurs locaux utilisés dans la procédure d'appariement initial. Nous donnons le détail de cette approche et nous en fournissons une analyse par le biais d'une série d'expérimentations.

Chapitre 4 Ensuite, nous effectuons une comparaison entre notre méthode et les autres modélisations locales, et nous mettons en place une approche de la reconnaissance par

indexation de descripteurs locaux qui est générale. Elle permet d'intégrer toutes les approches actuelles dans un environnement général, basé sur une validation par cohérence géométrique. Nous validons notre approche par une série d'expérimentations sur des images réelles complexes.

Chapitre 5 Le dernier chapitre prend un point de vue radicalement différent du précédent. Dans cette troisième partie de la thèse nous nous intéressons aux aspects algorithmiques de l'approche de la reconnaissance que nous avons choisi. Toutes les méthodes qui sont basées sur l'indexation de descripteurs locaux, dont la nôtre fait partie, reproduisent un schéma algorithmique similaire. En faisant une analyse de la complexité sous-jacente de ce schéma, nous en venons à des résultats étonnants : un des arguments généralement utilisé pour ces approches, la rapidité de la reconnaissance, est mis à mal. Dans certains cas, un processus séquentiel parcourant tous les modèles peut s'avérer plus performant que l'application d'une indexation élaborée.

Annexes Trois annexes fournissent de plus amples détails concernant le calcul ou les démonstrations d'énoncés dans les parties précédentes. Elles peuvent être omises lors d'une lecture rapide, puisque leurs résultats sont repris dans le texte principal.

La première annexe aborde les optimisations que l'on peut appliquer aux structures de données évoquées dans le chapitre 3 lorsque elles sont représentées comme des arbres à profondeur finie. La seconde et troisième annexe concernent le quatrième chapitre, et consistent principalement en des détails des calculs relatifs à des résultats énoncés dans le texte.

Remarque concernant l'illustration choisie

Pourquoi ai-je choisi l'illustration des deux femmes ? Elle est certes sortie de son contexte[3], mais elle montre de façon succincte et extrêmement précise les deux méthodes d'aborder la reconnaissance évoquées précédemment.

Une personne, observant l'image pour la première fois, remarquera une femme. Si on lui donne l'information que le portrait d'une seconde femme s'y cache, elle mettra toutes ses connaissances concernant les aspects visuels du concept « femme » au service de la tâche de reconnaissance... et parviendra, non sans mal, à reconnaître le second portrait. Il me semble clair que la façon de procéder ici relève parfaitement de ce qu'a décrit MARR. Ayant fait l'exercice plusieurs fois, notre cobaye n'aura plus aucun mal à distinguer les deux figures, et qui plus est, lorsqu'on lui présentera par la suite la même image, il l'identifiera comme « *Le portrait de deux femmes* », sans pour autant procéder à l'identification des portraits. Ne s'agirait-il pas là d'une reconnaissance par l'apparence seule, sans plus utiliser la hiérarchisation des indices visuels ?

3. Elle figure, entre autres, dans un ouvrage de S. CONVEY donnant des règles d'or pour être un décideur et un dirigeant efficace où elle illustrait l'aptitude de constamment et rapidement savoir changer de point de vue [23]. Elle apparaît également dans bon nombre d'autres ouvrages. Sa véritable origine est inconnue.

Chapitre 1

Reconnaissance

DANS ce chapitre, nous effectuons un survol historique des méthodes de reconnaissance calculatoires que nous classons en deux types : celles principalement issues du paradigme de MARR, cherchant une justification biologique, et que nous appelons *hiérarchiques* ; et celles partant d'un formalisme purement mathématique, s'affranchissant de cet héritage, et que nous appelons *géométriques*. Nous confrontons les deux philosophies et nous analysons leurs acquis et leurs échecs, ce qui nous permet de postuler en conclusion de ce chapitre que l'évolution naturelle des méthodes de reconnaissance artificielle va vers la modélisation par apparence : modélisation qui ne nécessite plus de données 3D, mais qui les remplace par un ensemble de vues 2D, plus facilement maniables. Cette modélisation est abordée dans le chapitre suivant.

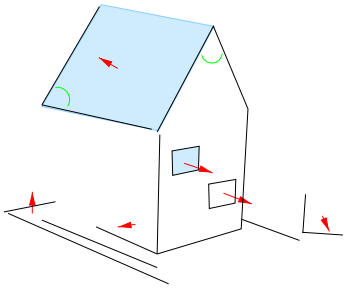
La section suivante aborde les deux paradigmes classiques. Nous ferons une analyse de leurs limites et des problèmes qu'ils soulèvent, en § 1.2. Nous finirons ce chapitre avec des exemples d'implémentation des deux paradigmes avant de conclure.

Il ne faut pas confondre la séparation que nous faisons ici avec une catégorisation absolue, et par définition dangereuse, des travaux existant en matière de reconnaissance. Outre tous les travaux en neuropsychologie et psychologie cognitive, qui sortent largement du domaine restreint de la reconnaissance computationnelle que nous abordons ici, un grand nombre de travaux traitant de la représentation et de la modélisation des données visuelles ne font pas partie des deux classes définies, bien qu'ils abordent des problèmes fondamentaux. L'étude des graphes d'aspect, introduite par KOENDERINK [54], fait, entre autres, partie de ce groupe. Nous détaillerons son lien avec les approches hiérarchiques et géométriques dans la section § 1.2.1.

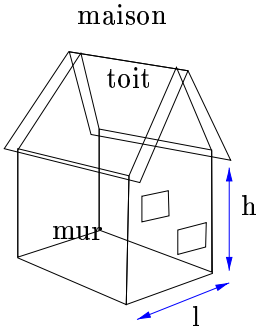
1.1 Définitions

On trouve l'une des premières tentatives de définition de la vision dans [1]. « *L'objet de la vue, c'est le visible. Or le visible est, en premier lieu, la couleur, et en second lieu, une espèce d'objet qu'il est possible de décrire par le discours, mais qui, en fait, n'a pas de nom ...* » [98]. Pour Aristote, la notion de perception visuelle, attribuée à *la vue*, est séparée de la reconnaissance qui est un acte de *l'intellect*. Par la suite, et dans un contexte plus analytique, d'autres se sont appliqués à trouver une définition plus adaptée à un traitement numérique.

image + première ébauche



ébauche $2\frac{1}{2}$D



modèle 3D

FIG. 1.1: *Les différents niveaux selon* MARR.

1.1.1 Une définition hiérarchique

David MARR [63] donne une définition de la vision qui lie intimement la représentation et le traitement de l'information visuelle comme deux facteurs interdépendants et interopérants. Il rejette la définition simple selon laquelle la vision serait « *[...] the* process *of discovering from images what is present in the world, and where it is.* », car il ne s'agit pas uniquement d'un processus au sens du traitement de l'information, mais d'une interaction fondamentale entre les structures de représentation et les actions qui permettent d'agir sur elles.

S'appuyant sur des résultats de neurophysiologie, il préconise de ne pas considérer uniquement l'aspect traitement, mais de le coupler à une représentation bien adaptée. Cette représentation se retrouve ensuite dans son célèbre partage en quatre niveaux : image, première ébauche (*primal sketch*), ébauche $2\frac{1}{2}$D ($2\frac{1}{2}$-D *sketch*) et la représentation 3D du modèle (*3-D model representation*). Dans ce contexte, la reconnaissance est le résultat d'une hiérarchie de processus et de représentations combinés entre les différents niveaux de représentation, les niveaux supérieurs ne se concevant pas sans les niveaux inférieurs. Cette définition a été pendant longtemps l'inspiration directe d'un nombre important de systèmes de vision et de reconnaissance. Nous en analyserons un exemple dans § 1.3.1 que nous confronterons à d'autres méthodes dans § 1.4. MARR n'est pas le seul à s'être basé sur une représentation de ce type. BIEDERMAN, par exemple, l'appelant *Recognition-by-components* (RBC) [12, 13], présente un schéma plus formel et plus simple, fondé sur les *ions géométriques* ou *géons* qu'il détecte à partir des informations de contours dans l'image uniquement. Un *géon* est un objet géométrique 3D élémentaire (il y en a 36 différents). Dans la modélisation que BIEDERMAN propose, chaque objet peut être décomposé en une structure formée de ces ions. La structure hiérarchique menant à l'identification des objets 3D se décompose alors selon le schéma FIG. 1.2 dont les cinq niveaux se décomposent ainsi :

1. *Edge Extraction* ou l'extraction des contours, dans laquellle on extrait de l'image les changements d'illumination et de texture pour former des contours.

2. *Detection of Nonaccidental Properties* et *Parsing at Regions of Concavity* ou la détection des configurations non-accidentelles et l'extraction des zones concaves, dans lesquelles on regroupe les contours pour mettre en exergue des propriétés comme l'alignement, le parallélisme *etc.*

3. *Activation of Geons and Relations* ou l'activation des géons et les relations entre eux, qui se base sur les informations précédemment obtenues pour formuler des hypothèses sur l'existence de *géons* dans des zones de l'image ainsi que les relations de position entre ces derniers.

4. *Activation of Object Models* ou l'activation des modèles d'objets, qui émet des hypothèses quant à la présence ou non de modèles, basées sur la présence des géons les composant.

5. *Object Identification* ou l'identification des objets, qui sélectionne les modèles effectivement présents dans la scène.

Les deux approches sont très semblables. Outre le fait qu'elles sont fondamentalement hiérarchiques, elles sont largement inspirées par des considérations neurobiologiques [1] et psycho-cognitives. Le schéma de MARR se transpose d'ailleurs parfaitement sur celui de BIEDERMAN lorsque l'on considère que l'identification des *géons* se fait par accumulation

1. Considérations qui se retrouvent même dans le choix de certains termes, comme, par exemple,« *activation* », qui rappelle le fonctionnement des neurones.

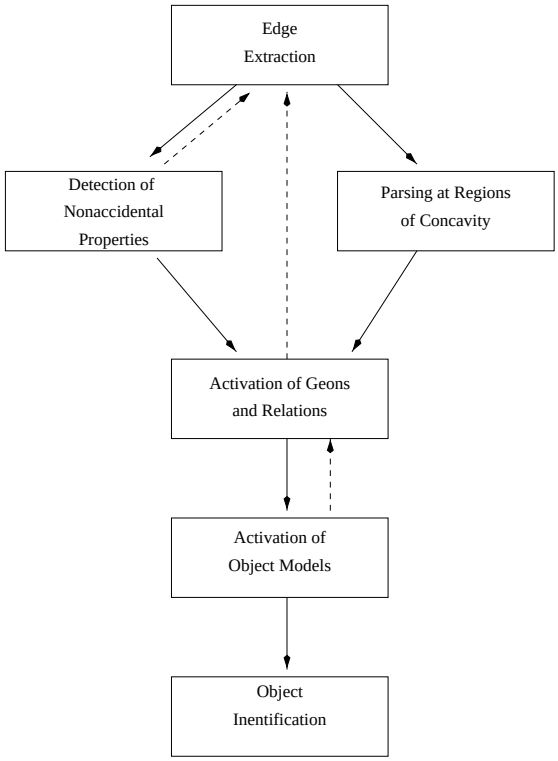


FIG. 1.2: *Les différentes phases pour la reconnaissance par composantes, proposée par* BIEDERMAN. *Schéma repris de [12], p. 118, fig. 2. Les flèches en pointillés représentent, une possibilité de retour d'information pour guider d'éventuelles nouvelles recherches.*

d'indices bas niveau, regroupés par leur configuration *non-accidentelle* qui laisse sous-entendre une configuration $2\frac{1}{2}$D, pour enfin former un regroupement 3D qui est le modèle.

1.1.2 Une définition géométrique

Plutôt que d'avoir une justification neurobiologique du problème, d'autres définitions essaient de fournir un cadre plus formel. En général, elles sont basées sur une approche plus géométrique et, historiquement, elles ont commencé par surgir au moment où on cherchait à s'éloigner du cadre précédent et à adopter un paradigme plus abstrait [16, 3].

Dans ce cas, on restreint la définition de la reconnaissance à un contexte plus pragmatique. Les auteurs se proposent plutôt de reconnaître des objets bien définis et parfaitement modélisés dans un environnement contrôlé. On formule dans ce cas la reconnaissance avec une définition géométrique (reprise de [29], chap. 11, p. 484) :

« *[Supposons] que chaque objet [à reconnaître] soit défini par un modèle* $M = (M_1, ..., M_n)$ *composé d'un nombre de primitives géométriques* M_i *[...], et qu'un nombre de capteurs visuels nous fournissent un ensemble de primitives similaires* $S = (S_1, ..., S_p)$ *pour la scène observée. Le problème de* reconnaissance *consiste alors à produire une liste de couples de*

primitives modèle/scène $R_n = ((M_1, S_{i_1}), ..., (M_n, S_{i_n}))$ *où les* S_{i_j} *sont soit des primitives de la scène, soit représentées par le symbole spécial* NIL, *qui indique que la primitive* M_j *du modèle n'est pas présent dans la scène.* »

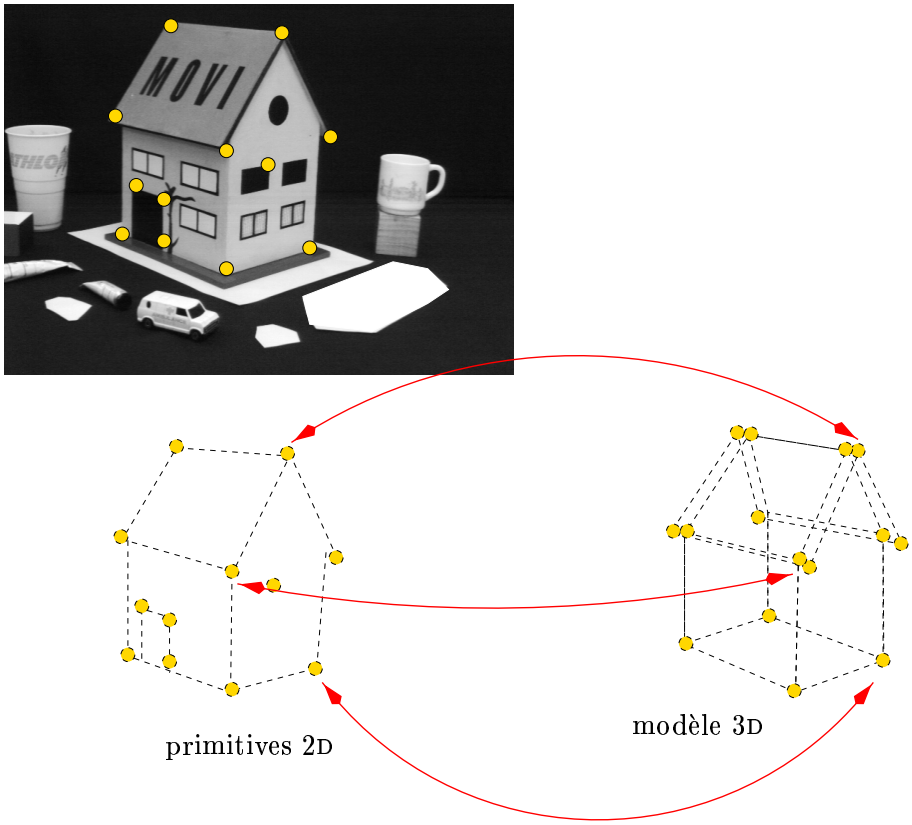


FIG. 1.3: *Représentation de la reconnaissance géométrique.*

Notons ici que nulle part on n'impose une méthode particulière de modélisation ou de mise en correspondance. Nous utilisons le terme « *géométrique* » uniquement dans le sens où l'on cherche à mettre en correspondance des primitives géométriques directement observables dans la scène avec des primitives d'un modèle, contrairement aux méthodes hiérarchiques où les mises en correspondance s'effectuent entre entités construites d'indices, supposés ou réels, mais pas nécessairement visibles dans l'image. Les moyens pour aboutir à l'appariement final n'en sont pas géométriques pour autant.

1.1.3 Définition générale

Au vu des deux définitions précédentes, apparemment orthogonales, est-il possible, ou même utile, d'essayer de trouver un facteur commun utilisable comme paradigme pour la reconnaissance? La première définition, en § 1.1.1, considère quasiment la reconnaissance comme un but en soi, sans préoccupation des buts finaux recherchés par celle-ci. La deuxième se cantonne à une optique de mise en correspondance stricte, ce qui la limite

à une reconnaissance d'instances parfaitement modélisées de modèles connus. Dans un contexte applicatif plus large, on pourrait vouloir s'affranchir du strict cadre de la mise en correspondance donné en § 1.1.2, surtout si l'on veut reconnaître des objets pour lesquels elle n'aurait pas de sens ; les objets déformables tomberaient dans cette catégorie, par exemple.

L'avantage des deux définitions, en revanche, est qu'elles fournissent implicitement un algorithme de mise en œuvre de la reconnaissance, en dehors du fait que leur validité peut être sujette à caution. Une définition plus générique devrait forcément s'abstenir d'imposer tel ou tel schéma de résolution du problème, et perdrait, par conséquent, son intérêt comme paradigme de la reconnaissance.

Nous en donnerons tout de même une qui nous semble juste. Elle traduit le fait que la reconnaissance artificielle se fait nécessairement dans un contexte spécifique et elle nous permet d'avancer que c'est le contexte ou le but recherché qui doit guider la mise en œuvre d'un algorithme de reconnaissance, et non pas l'inverse. On peut alors énoncer que *la reconnaissance consiste à attribuer une sémantique à une image ou à des parties de celle-ci, où la sémantique est régie par le but poursuivi.* Les deux définitions précédentes entrent parfaitement dans ce schéma : dans le cas de MARR, à chaque niveau de la hiérarchie, la reconnaissance s'effectue avec une optique différente et une sémantique liée aux primitives recherchées, produisant à la fin une notion de « reconnaissance » au sens de la vision humaine ; dans le cas géométrique, la sémantique finale vient de la mise en correspondance entre les primitives simples, le but final étant d'aligner au mieux deux instances d'un même objet.

1.2 Limites et problèmes

À partir de ces définitions de la reconnaissance, nous sommes maintenant en mesure d'énoncer un certain nombre de points importants surgissant de façon inhérente dans un contexte de reconnaissance automatisée.

1.2.1 La modélisation

Comment doit-on représenter les connaissances *a priori* pour les intégrer dans le processus de reconnaissance[2] ? Surtout, si l'on prend en compte le fait que le processus est régi par un but ou une application spécifique, la modélisation doit refléter l'information nécessaire à l'accomplissement de cette tâche.

MARR et BIEDERMAN proposent une modélisation hiérarchique justifiée par des connaissances neurobiologiques, où un modèle final est un ensemble structuré de modèles de plus bas niveau. Les avantages de cette représentation se trouvent surtout dans son aspect très intuitif et naturel et dans le fait qu'elle est suffisamment générale pour englober le spectre complet des observations visuelles. Et là, nous mettons en même temps l'accent sur sa faiblesse : son aspect générique ne permet pas de définir une méthodologie propre (à part

2. Le terme « processus » est pris ici comme notion d'action globale, et non pas au sens restreint rejeté par MARR.

le fait qu'elle doit refléter la hiérarchie et l'interaction entre les niveaux de représentation) et qui plus est, son fondement biologique la rend très inadaptée à une automatisation exploitable.

La modélisation géométrique s'affranchit de toute sémantique en ne considérant que des objets mathématiques qui n'ont pas nécessairement de lien avec la description de l'objet comme dans le cas précédent. Un objet 3D devient alors un ensemble d'entités géométriques, et son image est une projection par le biais d'une transformation de ces mêmes entités de 3D en 2D. Cette façon de définir des objets pour la reconnaissance perd de la généralité par rapport à l'autre méthode. En effet, ne peuvent être considérés que les objets pour lesquels l'extraction des informations géométriques soit suffisamment stable et répétable pour permettre une utilisation fiable. D'autre part, la perte totale de sémantique induit le fait que les modèles ne fournissent plus de description exploitable pour des définitions de classes visuelles ou pour former des groupements d'objets similaires dans le sens intuitif du terme. Par contre, son formalisme bien défini permet d'énoncer de façon triviale un algorithme de modélisation.

On voit clairement que la modélisation des données reflète bien le but que l'on cherche à atteindre. Dans le premier cas, la reconnaissance consiste à fournir une description de ce qu'on observe. Les modèles sont alors conçus pour répondre à ce besoin, et intègrent directement la notion de description. Dans le second cas, seuls l'identification et la localisation comptent. La description devient alors superflue, et on se munit d'outils formels permettant de mettre en œuvre une mise en correspondance et une localisation automatique.

Note C'est dans ce contexte que l'on peut situer la classe de travaux évoquées dans l'introduction de ce chapitre. La notion de *graphe d'aspect*, introduite par KOENDERINK [54], par exemple, cherche à résoudre par la modélisation des aspects visuels, le problème de perte d'information du passage d'une réalité 3D à une image 2D. À cause de cette perte d'information, il est possible d'observer un objet sous une infinité de projections possibles. Parmi les travaux issus de cette modélisation on trouve aussi bien des approches hiérarchiques que géométriques.

1.2.2 La méthodologie

De la même façon, et comme dual à la modélisation, on peut considérer le problème de la méthode de reconnaissance mise en œuvre (pour être plus proche de nos préoccupations, le terme *algorithme* serait plus adéquat).

La méthode qui est sous-jacente au paradigme hiérarchique est claire, bien qu'elle ne se formalise pas facilement dans le cadre d'une automatisation. L'algorithme de reconnaissance doit être hiérarchique, partant d'une extraction de bas niveau. Il collectionne des éléments de preuve pour la présence de tel ou tel objet d'un niveau supérieur, jusqu'à l'obtention de suffisamment de preuves pour confirmer ou infirmer la présence de tel ou tel modèle du niveau final connu. Dans son souci de rester proche de la vision biologique, le paradigme ne donne que des orientations génériques concernant l'organisation des données. Ces orientations ne se traduisent pas facilement en des termes algorithmiques fins.

D'ailleurs, si l'on observe les implémentations réelles qui seront évoquées ultérieurement, on peut noter que cette carence se retrouve dans la réalisation finale de l'algorithme. On s'y efforce de retarder la prise de décision relative à l'interaction entre niveaux au point d'en permettre la modification à tout moment [26].

La modélisation et la méthodologie géométriques tentent de résoudre ce problème en ne partant plus de considérations neurobiologiques, mais d'une tentative de « mise en équations » de la reconnaissance d'objets bien définis. La rigidité de l'approche en fait, là encore, sa force et sa faiblesse. D'une part, le problème de la méthodologie est entièrement résolu par le fait qu'il est complètement défini (il s'agit de recourir à une recherche de maximum dans un espace discret), bien que la mise en œuvre explicite de cette recherche reste non déterminée. D'autre part, le manque de flexibilité, et l'identification forte entre mise en correspondance et reconnaissance en fait une méthode mal adaptée comme système de reconnaissance générique.

Là également, le lien entre les objectifs et l'approche est très fort. Et par « transitivité » la méthodologie et la modélisation s'en trouvent intimement liées. Ceci nous conforte dans notre opinion que les systèmes de reconnaissance automatiques, dans l'état de l'art actuel, ne peuvent être dissociés d'un cadre opérationnel bien défini.

1.2.3 Le bruit

Un point important qui n'est pas directement abordé dans les définitions des deux approches est l'incertitude incontournable des mesures. En effet, beaucoup de facteurs interviennent dans le processus de formation d'une image, et contribuent chacun à détériorer le perçu par rapport à la réalité. Nous utiliserons terme générique de « *bruit* » pour désigner ces déformations visuelles. Comment peut-on caractériser l'influence sur la performance du système de reconnaissance de l'incertitude sur les mesures?

L'approche hiérarchique n'intègre pas directement une gestion d'erreur. Elle est suffisamment flexible dans sa méthode de description pour éventuellement absorber des fluctuations dans les positions ou les paramètres des primitives, mais le fait qu'elle repose fondamentalement sur des descriptions sémantiques la rend extrêmement fragile à l'absence d'indices ou à des indices « fantômes » qui peuvent apparaître lors d'une segmentation. En effet, à cause de la conception entièrement hiérarchique de la méthode, la défaillance du plus bas niveau met en défaut toute la construction.

Les problèmes liés à l'apparition ou la disparition d'indices sont intrinsèquement gérés par la méthode géométrique. Étant donné qu'elle peut se formuler comme une recherche de sous-ensembles maximaux, le nombre des indices, ou l'inexistence de correspondants de certains d'entre eux, importe peu. De plus, le cadre mathématique permet d'intégrer facilement des notions d'incertitude et de propagation d'erreur, ce qui peut mener à une mesure de confiance dans la mise en correspondance des primitives et dans la reconnaissance.

Pendant trop longtemps, l'extraction des indices a été considérée comme un problème à part, reléguée au domaine de la segmentation que l'on se donnait comme acquis pour des problèmes de reconnaissance. Nous pensons, au contraire, que l'imperfection des données de départ doit faire partie des hypothèses des méthodes de reconnaissance si l'on veut que celles-ci puissent être utilisées dans des conditions réelles.

1.2.4 Autres problèmes

D'autres problèmes ne sont pas directement abordés par ces deux classes de méthodes. Nous en citerons trois.

- Les deux techniques se basent explicitement sur une description formelle des objets à reconnaître. On peut légitimement se poser la question de savoir si une telle description est toujours disponible. Ou même si une description formelle est compatible avec les observations dans les images [21].

- Telles qu'elles sont définies, les approches ne répondent qu'à la question de savoir si un modèle M est présent dans une image I. Ce qui, suppose implicitement que ce modèle est unique ou fait partie d'un petit ensemble. Le problème plus général de savoir quel(s) modèle(s) est(sont) présent(s) dans une image I demande une organisation des données et une séparation entre l'identification et la localisation qui peut être difficile à mettre en œuvre.

- Quand on ne sait pas si un objet est présent dans une scène, comment veut-on que le système réagisse? Avec un franc oui ou non, avec un degré d'incertitude? Comment peut-on être sûr que l'objet trouvé est effectivement dans la scène ou seulement dû à une erreur d'appréciation du système?

Ces problèmes fondamentaux ne sont pas mis en exergue par les approches citées. Pourtant ils font partie intégrante d'un système de reconnaissance.

1.3 Des paradigmes aux implémentations

Le principal point commun entre les deux types d'approches est qu'elles essayent de mettre en correspondance directement (au sens large du terme) les informations 3D des modèles avec les informations 2D extraites des images (parfois même des informations provenant de capteurs 3D). Nous verrons ultérieurement, dans le chapitre 2, que de nouvelles approches tentent de mettre en place des méthodes basées sur une modélisation 2D. Afin de comprendre la justification de telles méthodes, nous étudierons d'abord un exemple de chaque paradigme décrit plus haut.

1.3.1 Application du paradigme de Marr

Le cas d'école par excellence est le système développée par Hanson et Riseman [40] : *Visions*, plus tard étendu par *Schema* de Draper *et al.* [26]. D'après les auteurs eux-mêmes, *[Visions] est une approche empirique au développement d'un système de vision générique.* Elle est entièrement centrée autour d'une représentation à niveaux telle que nous l'avons décrite précédemment. Le schéma à quatre couches est enrichi par l'éclatement en sous-couches plus spécialisées pour obtenir sept niveaux au total. La mise en œuvre consiste alors en deux zones de représentation des connaissances (une pour la modélisation *a priori*, une pour les informations visuelles détectées) sur lesquelles agit une bibliothèque d'algorithmes de traitement spécialisés, gérée par une structure de contrôle et de recherche.

En faisant abstraction des détails, la reconnaissance correspond à une sorte de vérification heuristique des modèles. Au début tous les modèles connus sont équiprobables. On procède progressivement à une sorte de *prédiction-vérification* sur les modèles les plus plausibles, en essayant d'obtenir des certitudes quant à la présence ou l'absence d'objets connus dans la scène. La partie *prédiction-vérification* est guidée de façon heuristique par l'organisation structurelle du modèle au sens des différents niveaux d'abstraction visuelle, et par les modules d'analyse de ces niveaux dont le système dispose.

La bibliographie dans cette catégorie est très riche, et l'approche basée sur la reconnaissance ou la modélisation par l'interaction de plusieurs niveaux a inspiré beaucoup d'auteurs [4, 40, 73, 26, 27]. Tous se sont appuyés sur une description hiérarchique des modèles, soit de façon « informelle » comme dans *Schema*, cité ci-dessus, soit en essayant d'introduire un formalisme plus rigide, en introduisant les cônes généralisés ou les *géons* [11, 25]

1.3.2 Exemples d'approches géométriques

Dans la définition géométrique de la reconnaissance, la notion de mise en correspondance de primitives tient un rôle essentiel. On peut dire que, globalement, ces méthodes procèdent à l'identification du modèle souhaité par un *alignement* de celui-ci avec l'image. On ne modélise plus seulement le modèle à partir duquel l'image a été formée, mais on prend en compte une modélisation *a priori* du processus de formation de l'image (généralement basée sur le modèle sténopé). Avec un certain nombre de *mises en correspondance* dépendant de cette modélisation, il est alors possible de calculer la transformation qui a projeté le modèle sur l'image. On appelle cette transformation *alignement* de l'un avec l'autre. La qualité de ce dernier permet alors de donner une mesure de validité à la mise en correspondance prise comme hypothèse initiale.

Toutes les méthodes inspirées par cette approche mettent donc en œuvre, d'une façon ou d'une autre, une technique de mise en correspondance et d'alignement. Elles se différencient principalement par leurs méthodes de recherche de l'ensemble optimal de correspondances, les plus classiques étant l'*isomorphisme de sous-graphes* [32], la *prédiction-vérification* avec souvent l'utilisation d'*arbres de recherche* [3, 16].

Étant donné que ces méthodes se situent dans un cadre algorithmique beaucoup plus formel et déterministe, il est possible de mesurer leur performance. Dans le cas général, on cherche à mettre en correspondance de façon optimale un sous-ensemble de caractéristiques de deux ensembles (l'image et le modèle) de respectivement n et m primitives. La taille de l'espace de solution est, dans ce cas [44] :

$$\sum_{i=1}^{n} C_n^i C_m^i i! = \sum_{i=1}^{n} A_n^{n-i} C_m^i$$

Pour la recherche d'un isomorphisme optimal entre sous-graphes, le problème devient $\mathcal{NP}$-complet [31], et des techniques d'optimisation combinatoire deviennent indispensables [42].

Les méthodes basées sur la prédiction-vérification procèdent à l'appariement et à l'alignement en même temps. Elles formulent une hypothèse de mise en correspondance mi-

nimale qui leur permet de calculer la transformation projetant le modèle dans l'image. À partir de cette projection il est possible de prédire la position d'autres primitives. La vérification de leur existence permet d'inférer d'autres mises en correspondance et de procéder de façon itérative. L'organisation en arbres de recherche permet de formuler des heuristiques de façon à ne pas explorer l'ensemble de l'espace des solutions.

1.4 Conclusion du chapitre

L'étude des approches considérées nous permet d'énoncer une série de conclusions utiles.

MARR fournit certes une analyse profonde qui permet de répondre dans des termes assez précis à la façon dont s'organise, à un métaniveau, la vision biologique, et, de ce point de vue, son approche peut donner des objectifs à très long terme à la vision artificielle, mais la théorie n'apporte pas de réponse quant à la façon dont cette organisation est mise en œuvre. De plus, d'autres travaux, comme ceux de BIEDERMAN[10], par exemple, montrent que cette mise en œuvre est un processus beaucoup plus complexe que la simple coopération hiérarchique que propose MARR. Par conséquent, tous les auteurs qui ont tenté de suivre cette voie [40, 26, 64] en sont venus à la conclusion que les moyens calculatoires qui doivent être mis en œuvre pour appliquer ce paradigme sont gigantesques.

L'approche géométrique s'affranchit de la contrainte biologique, et tente d'établir un cadre formel (lisez *mathématique*) pour aborder une partie du processus de vision. Son succès [3, 16] vient d'ailleurs du fait que l'approche ne cherche plus à produire un processus complet de vision artificielle, mais qu'elle cherche à résoudre des problèmes particuliers posés par la vision artificielle ; ce qui est fondamentalement différent. La méthode résout donc le problème de l'alignement, mais n'est pas très adaptée dans le cas d'une reconnaissance plus générale.

Les deux méthodes s'appuient d'ailleurs fortement sur une description structurée des modèles à reconnaître. Cette description n'est, cependant, pas nécessairement accessible, ni possible, comme le décrivent CHEN et MULGAONKAR dans [21].

Un dernier point, qui était déjà évoqué en § 1.2.4 concerne l'aptitude des deux méthodes à être intégrées dans un environnement où il est nécessaire d'identifier un modèle parmi un très grand nombre dans une image inconnue. Trop fortement basées sur une identification 1 à 1 d'une image à un modèle, avec peu de moyens pour « factoriser » de l'information, les modèles doivent tous être passés en revue pour être retenus ou rejetés.

La situation idéale serait donc

- d'avoir un formalisme fort, comme dans le cas de l'approche géométrique, ce qui permet de bien poser les problèmes et de fournir une méthodologie générale, quitte à ne résoudre qu'un problème spécifique de la vision artificielle ;
- de ne pas dépendre de modèles artificiels qui imposent de connaître les conditions de prise de vue, ou qui dépendraient d'une structuration préétablie ;

– de pouvoir adapter la méthode à de grandes quantités de modèles pour être en mesure de couvrir un domaine plus large qu'auparavant.

Une modélisation par *apparence* réunit ces propriétés, car elle se base sur l'apparence observée d'un modèle, et non plus sur une conception ou une représentation de sa réalité 3D. Elle permet ainsi de prendre en compte implicitement tous les paramètres de prise de vue et de reprojection. La mise en correspondance d'apparences de modèles et d'images inconnues peut alors être formalisée de façon similaire à celle utilisée par l'approche géométrique, sans pour autant avoir tous les problèmes liés au calcul de pose en 3D. En quelque sorte, la méthode par apparence sépare l'identification de la localisation 3D, ce qui lui permet d'être plus efficace. Nous consacrerons le chapitre suivant à cette modélisation.

Chapitre 2

Reconnaissance par apparence

DANS ce chapitre, nous définissons ce que nous entendons par *reconnaissance par l'apparence*. Nous fournirons une classification des principales méthodes existantes, nous exposerons leurs principaux avantages et inconvénients, et présenterons quelques exemples de mise en œuvre. Il est clair que les classifications qui seront données ne sont pas exhaustives, et qu'elles donneront seulement les tendances générales les plus courantes. Nous n'avons, également, pas essayé d'être exhaustif dans les références bibliographiques de ce chapitre. Nous nous sommes contentés de souligner les étapes les plus marquantes dans l'évolution de chaque classe en donnant quelques références parmi les plus citées [79, 66].

2.1 Définitions et caractéristiques

Nous entendons par *méthode de reconnaissance par l'apparence*, toute méthode de reconnaissance qui cherche à modéliser des objets (2D ou 3D) directement par leur image perçue et non pas par un modèle construit à partir d'une conception abstraite particulière. N'en font donc pas partie : les approches basées sur les graphes d'aspects [51, 52, 69], ou celles utilisant des modèles CAO [16, 3, 21].

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

Dans la majorité des applications qui implémentent cette approche, elle est couplée à une indexation, bien que ceci ne soit pas une condition obligatoire. Dans ce cas, la structure globale des algorithmes se présente comme montré dans le schéma 2.1

Algorithme 2.1 Squelette d'une reconnaissance par l'apparence s'aidant d'une indexation

```
Phase de modélisation :

     Paramètres d'entrée : IMAGE_MODÈLE[1], ..., IMAGE_MODÈLE[n]
     Paramètres de sortie : STRUCTURE_INDEXATION /* Contenant les n modèles */

     début
         pour i allant de 1 à n faire
         début
              Caractériser l'IMAGE_MODÈLE[i];
              Utiliser cette caractérisation pour indexer le modèle
                                              dans STRUCTURE_INDEXATION;
         fin

         retourner STRUCTURE_INDEXATION;
     fin

Phase de reconnaissance :

     Paramètres d'entrée : IMAGE_INCONNUE, STRUCTURE_INDEXATION
     Paramètres de sortie : IMAGE_MODÈLE

     début
         Caractériser IMAGE_INCONNUE;
         Utiliser cette caractérisation pour accéder au(x) modèle(s)
                      similairement indexé(s) dans STRUCTURE_INDEXATION;
         Éventuellement choisir le modèle le plus probant parmi ceux
                      trouvés;

         retourner ce modèle;
     fin
```

Il est important de noter que, lors de la phase de reconnaissance il n'est pas toujours nécessaire que l'image présentée soit une copie conforme d'un des modèles indexés. En effet, suivant la modélisation mise en œuvre, la tolérance sur la ressemblance peut être plus ou moins importante, permettant, le cas échéant, de couvrir un espace continu d'apparences avec seulement un nombre fini de modèles.

De façon générale, on distingue deux grandes classes dans les méthodes par l'apparence. L'une regroupe les approches qui considèrent une image comme une entité indivisible, donnant généralement lieu à un index unique par image, l'autre contient les méthodes qui modélisent les objets par un ensemble de caractéristiques hétérogènes désignant des parties plus ou moins grandes dans l'image. On parlera de *méthodes globales* pour les unes et de *méthodes locales* pour les autres. Leurs différences fondamentales sont détaillées dans les sections qui suivent.

2.2 Modélisation globale

Les modélisations globales sont principalement centrées autour d'une même approche, basée sur les « *images propres* ». Initialement développée par SIROVITCH et KIRBY [94] pour la reconnaissance de visages, et ensuite reprise et adaptée par un grand nombre d'auteurs [100, 70, 65, 7], la méthode ne considère plus les images comme des matrices $n \times m$ de valeurs de niveau de gris, mais comme des vecteurs de taille nm contenant ces mêmes valeurs. Ainsi, couvrant l'espace des apparences possibles d'un objet par k images modèles, il est possible de calculer une matrice de covariance Σ de dimension $nm \times nm$ qui capte la variation entre les différents modèles. La matrice est obtenue en calculant la moyenne $\bar{M}$ des vecteurs modèles $M_{i=1..k}$ et en posant :

$$\Sigma = \sum_{i=1}^{k} \left(M_i - \bar{M}\right)\left(M_i - \bar{M}\right)^T$$

Les *vecteurs propres* de cette matrice forment alors une base dans laquelle il est possible d'exprimer l'espace défini par les modèles. La base optimale en d dimensions, au sens des moindres carrés, est obtenue en prenant les vecteurs propres ayant les d plus grandes valeurs propres, ce qui permet de faire une réduction considérable de la dimension de l'espace sans perte de qualité visuelle (d est nettement inférieur à k), car cette décomposition est en fait une analyse en composantes principales. Une image est donc représentée comme un point dans l'espace de ces vecteurs propres, et la reconnaissance revient à faire une recherche du plus proche voisin dans l'espace à d dimensions considéré.

L'avantage principal de cette approche est sa réduction considérable des dimensions pour la représentation d'une image. Dans le cas d'une image 512×512 (vecteur de dimension $512^2 = 262144$, on prend typiquement les 10 plus grandes valeurs propres, ce qui revient à une réduction de dimension d'un facteur d'ordre 10^4. Ceci permet de réaliser des *indexations* efficaces et des recherches rapides, rendant cette méthode particulièrement adaptée pour des contraintes temps-réel [70]. Malheureusement ses inconvénients sont de taille :

1° le fait que l'approche soit globale la rend particulièrement sensible à des *occulta-*

tions partielles des objets observés, de changement du fond sur lequel l'objet se trouve, *etc.* ;

2° les *changement d'éclairage*, ou des transformations géométriques lors de la prise de vue influent directement sur la représentation finale de l'image.

La modélisation doit explicitement prendre en compte ces variations, soit en réalisant une segmentation préalable pour séparer un objet de son environnement, par exemple, soit en modélisant toute situation (d'éclairage, de positionnement, *etc.*) par des images modèles, ou en appliquant des méthodes de normalisation dans des cas spécifiques (reconnaissance de visages par exemple). Dans la plupart des cas, ces adaptations sont difficilement applicables pour la reconnaissance d'objets 3D génériques.

2.3 Modélisation locale

La classe des approches par modélisation locale est très hétéroclite. On admet généralement que son origine se trouve dans les travaux de LAMDAN et WOLFSON avec leur méthode de reconnaissance *Geometric hashing* [58], bien que des travaux antérieurs existent [53]. Des méthodes plus récentes, avec plus ou moins de liens explicites avec celle-ci sont [86, 71, 90, 60]. Dans ces approches, le modèle n'est plus l'image toute entière, comme dans la section précédente. Par contre, un modèle est représenté par un ensemble fini de configurations locales dans l'image. La notion de configuration est par ailleurs assez vague, et peut varier de simples points dans l'image [90], à des configurations assez complexes de segments ou de courbes [71, 60]. Ces configurations sont généralement sélectionnées suivant un critère de pertinence, et imposent donc une segmentation sur les images. Dans le cas de points, il pourrait s'agir de minima d'autocorrélation, par exemple, et dans le cas de contours ce pourraient être des maxima locaux de la norme du gradient.

Le but principal des méthodes locales est de mettre en correspondance des configurations d'une image inconnue avec des configurations de modèles déjà observés. Le taux de mise en correspondance et la cohérence géométrique entre celles-ci sont alors des mesures permettant de classer les modèles et de sélectionner celui qui est le plus ressemblant à l'image requête. Les méthodes diffèrent principalement dans leur façon de caractériser les configurations et dans leur manière de trier les modèles plausibles. Elles ont ceci en commun avec les approches géométriques du chapitre précédent qu'il s'agit intrinsèquement de trouver un sous-ensemble optimal d'appariement. Elles en diffèrent par le fait qu'il n'existe *a priori* pas de structuration entre les configurations. La comparaison s'arrête là, par contre. Afin d'accélérer cette mise en correspondance, les méthodes caractérisent les configurations par des descripteurs. Ces descripteurs sont choisis afin d'absorber une partie des déformations qui peuvent apparaître lorsque l'objet est observé dans des conditions différentes. On parlera donc souvent d'*invariant*[1] (suivant les cas des termes de *quasi-invariant* [60] ou de *semi-invariant* [71] sont utilisés) pour décrire cette propriété

1. Nous ne détaillerons pas ici, la théorie des invariants. Nous estimons que ce sujet nous éloignerait trop du but de ce chapitre. D'excellentes descriptions de la théorie peuvent êtres trouvées dans [86, 35]. Nous en aborderons une partie dans § 3.2.

des descripteurs. La comparaison des descripteurs d'une image et ceux d'un modèle permettent de faire une mise en correspondance initiale et grossière, tous modèles confondus, ou modèle par modèle, de façon efficace. Cette étape est souvent implémentée par une méthode d'indexation. Il est ensuite nécessaire de valider cette mise en correspondance par une vérification de cohérence au niveau des modèles trouvés. Cette vérification peut être aussi simple qu'un décompte des appariements par modèle [58], ou être un processus plus complexe [60, 71].

Par le fait qu'elles sont entièrement locales, ces méthodes sont extrêmement robustes à des occultations partielles des objets. Elles sont très riches, puisque, grâce à la mise en correspondance de configurations, elles fournissent la position du modèle dans l'image. C'est ce calcul de position qui les rend parfois moins performantes dans de très grandes bases de modèles.

Il est intéressant de noter l'approche de HUTTENLOCHER *et al.* basée sur la distance de HAUSDORFF [50, 49, 87]. Elle est locale dans le sens qu'elle modélise les images par des contours représentatifs qu'elle réussit à mettre en correspondance. L'appariement des contours ne se fait pas pixel par pixel, mais on identifie globalement un contour avec un autre. Cette méthode présente de très bons résultats de reconnaissance, mais n'est malheureusement pas adaptable à une mise en œuvre par indexation et est très consommatrice en temps de calcul.

2.4 Autres modélisations

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

2.4.1 Modélisation par histogrammes

Utilisée dans les premiers travaux cherchant à caractériser des images par leur distribution de couleurs par SWAIN et BALLARD [96], la modélisation par histogrammes a été implémentée avec succès pour l'accès à de grandes bases d'images par leur contenu couleur (*cf.* p. ex. [39, 76]). Elle a été récemment étendue par SCHIELE [88] pour d'autres descripteurs locaux. L'idée générale, et que l'on retrouve de façon simplifiée dans les travaux originaux, est de calculer, en tout point de l'image, un vecteur de descripteurs invariants, de manière similaire aux approches locales, à cette différence près qu'aucune segmentation préalable n'est effectuée. Une image modèle est donc représentée par son histogramme multidimensionnel de ces descripteurs.

Pour identifier une image à l'un des modèles connus, on calcule les mêmes types de descripteurs dans un ensemble quelconque de points. On obtient ainsi un « sous-histogramme »

que l'on peut comparer aux histogrammes stockés afin de déterminer le modèle le plus semblable.

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

2.4.2 Modélisations statistiques et probabilistes

Densités de probabilité Dans leurs travaux, HORNEGGER et NIEMANN [46] se basent sur une représentation par densité de probabilité de caractéristiques d'un objet pour le modéliser. La reconnaissance passe alors par une estimation de paramètres par maximum de vraisemblance, à partir d'un modèle probabiliste *a priori*. Les caractéristiques utilisées pour la reconnaissance sont considérés comme des variables aléatoires.

L'algorithme de modélisation et de reconnaissance se découpe alors en trois étapes principales. On suppose que k objets différents sont représentés par des fonctions de densité de probabilité paramétrées. Les caractéristiques et les paramètres de ces fonctions peuvent varier selon les méthodes et les objets. Dans la phase « d'apprentissage » les paramètres $B_\kappa, 1 \leq \kappa \leq k$ de la fonction de densité de probabilité *a priori* doivent être estimés à partir d'un ensemble de vues $\{^\rho O\}$ de départ pour chaque modèle κ.

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

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

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

Chaînes de MARKOV et décision statistique Les travaux de HERBIN[43] pourraient être classés sous différentes dénominations. Lui-même définit son approche comme la reconnaissance *en actes*. Il rejette notamment la réduction de la reconnaissance à un simple appariement empirique, et ne la conçoit que dans une activité cognitive guidée par le contexte. Il développe alors une modélisation qui s'appuie sur les graphes d'aspects, enrichies d'une structure probabiliste markovienne. La théorie asymptotique des tests d'hypothèses lui permet ensuite d'analyser le problème de discrimination des modèles markoviens.

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

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

Premièrement, l'approche s'appuie fortement sur les graphes d'aspects [55]. Des travaux menés dans le domaine, et notamment [30] soulèvent principalement leur complexité exorbitante (bien que polynômiale [78, 77]) et leur imprécision dans le cas de données réelles où leur dépendance d'une segmentation précise est trop forte. La solution proposée par HERBIN, consistant à « probabiliser » les graphes d'aspect ne résout pas ces problèmes.

Deuxièmement, l'auteur rejette avec force les approches empiriques basées sur la mise en correspondance d'indices visuels, telles que la plupart des méthodes décrites ou développées dans cette thèse. Son argument principal consiste à dire qu'un simple appariement ne véhicule pas l'*intelligence* indispensable à une tâche de reconnaissance de haut niveau, et que, de plus, il n'intègre pas la notion de *contexte* qui est fondamentale à son aboutissement. La validité de l'argumentation est indiscutable, mais il nous semble que les deux approches n'opèrent pas sur le même niveau, et la *reconnaissance en actes* s'appuie, lors de l'identification des nœuds entre eux, sur cette même mise en correspondance empirique. L'approche souffre donc d'une trop grande généralité et ne sera probablement exploitable que lorsqu'une méthode d'appariement ou d'identification « *empirique* » lui permettra de construire des graphes d'aspects et d'identifier des nœuds de celui-ci de façon robuste et fiable.

2.4.3 Caractérisation par graphes.

Dans sa thèse, SOSSA [95] propose de modéliser des objets 3D par un nombre limité de vues qui les définissent. Sa méthode est basée sur une approche de caractérisation provenant de la théorie des graphes. Elle utilise comme entrée des images segmentées en contours, approchés par des lignes polygonales. Ces lignes polygonales forment un graphe qui est caractérisé par son polynôme de deuxième *immanant*. La mise en équations de la topologie et de l'apparence du graphe, combinée à une indexation des caractéristiques de sous-graphes, permet à l'auteur de contourner le problème de la $\mathcal{NP}$-complétude de la recherche de sous-graphes isomorphes.

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

2.5 Exemples de méthodes par apparence

Afin de donner une vue de la diversité au sein de la classe des méthodes par apparence, nous fournirons dans cette section une description succincte de deux méthodes déjà citées. Elles peuvent être classées parmi le groupe des approches locales.

2.5.1 *Geometric hashing.*

L'approche du *Geometric hashing* que nous avons mentionnée précédemment [58], est l'un des précurseurs des méthodes locales basées sur l'apparence. À l'origine, la méthode dépendait encore de modélisations imposées, mais très vite des extensions ont été proposées pour prendre en compte des modèles « visuels ». Les images modèles sont représentées par un ensemble de points d'intérêt 2D. Toutes les combinaisons de trois points sont ensuite énumérées pour former une base affine du plan. Dans cette base $\mathcal{B}_i$, tous les autres points p s'expriment par leurs coordonnées $(p_x, p_y)_{\mathcal{B}_i}$, et cette expression est invariante à toute transformation affine appliquée à l'image. On met donc en place un espace d'indexation à deux dimensions qui reçoit, pour tout point caractérisé p une référence au modèle auquel il appartient à l'endroit indexé par $(p_x, p_y)_{\mathcal{B}_i}$.

Lors de la phase de reconnaissance, l'image inconnue subit un traitement similaire. On caractérise tous les points dans toutes les bases affines formées par trois points. Pour chaque caractérisation, la base d'indexation est consultée, et un modèle voit son compteur incrémenté chaque fois qu'une de ses références est accédée dans la table d'indexation. Le modèle le plus référencé est alors pris comme correspondant pour l'image inconnue. Nous reviendrons sur cette méthode en § 3.3.5.

2.5.2 Invariants locaux de luminance

Schmid [90] présente une méthode d'indexation basée sur le *jet local* de Koenderink [56]. Elle extrait un ensemble de points d'intérêt (minima d'autocorrélation) des images modèles, qui sont à leur tour caractérisés par un vecteur d'invariants de luminance. Les composantes de ce vecteur sont des combinaisons linéaires des dérivées partielles du signal, allant de l'ordre 1 à l'ordre 3. Elles sont invariantes à la rotation et à la translation, et absorbent un léger changement d'échelle. En implémentant une caractérisation multi-échelle, elle obtient ainsi un système de reconnaissance performant et rapide.

La comparaison d'une image inconnue et les modèles à reconnaître passe, là aussi, par une indexation des descripteurs et un vote parmi les modèles. Le paradigme de vote n'est plus simplement majoritaire comme dans le cas du *Geometric hashing*, mais il prend en compte uniquement des points pour lesquels un certain nombre de ses voisins ont été mis en correspondance avec le même modèle.

2.6 Conclusion du chapitre

Les méthodes par apparence réduisent le problème de la reconnaissance à une identification 2D-2D. Il existe des travaux qui proposent d'intégrer des informations 3D supplémentaires dans la phase de modélisation dans le cas d'applications précises [91, 62]. Ces modifications n'interfèrent en aucun cas avec le processus même de reconnaissance et sont à considérer comme de simples extensions pratiques du modèle, et montrent par la même occasion qu'il n'est pas toujours nécessaire d'avoir recours à une modélisation 3D explicite pour pouvoir utiliser des données tridimensionnelles. Du fait que les approches par l'apparence ne font pas intervenir de modélisation explicite, ces méthodes deviennent particulièrement adaptées pour la création automatique de modèles. Nous n'aborderons pas ce volet dans cette thèse bien que des travaux dans ce sens aient déjà été faits [35, 88].

La reconnaissance par l'apparence n'est certes pas aussi proche de la vision biologique que la modélisation de MARR, mais elle étend de façon significative l'approche géométrique. Plutôt que de répondre à la préoccupation principale de cette dernière (« *Où ce trouve tel objet précis dans l'image, et quelle est sa position par rapport à la caméra?* ») elle répond à une question plus générale : « *Est-ce que la scène observée comporte, dans sa totalité, ou en partie, des objets déjà observés (ou semblables) et quels sont-ils?* »

Les méthodes globales apportent leur réponse dans un contexte de reconnaissance et de caractérisation d'images et tant que telles, ou d'objets uniques, tels que des visages par exemple. Elles sont performantes, mais ne sont pas adaptées à des analyses plus fines d'identification et de localisation de multiples objets dans une même scène avec éventuellement un environnement non modélisé et variable.

Les méthodes locales ou par histogrammes répondent mieux à ce besoin, et les derniers résultats dans ce domaine sont remarquables [88, 90]. Leurs performances sont cependant très dépendantes du type de caractérisation locale choisie, qui reste, malgré tout, très proche du signal dans l'image. Nous allons donc proposer une méthode locale qui se base sur des informations d'un niveau d'abstraction juste supérieur : des contours et des segments. Elle permettra de formaliser un cadre plus générique, et intègrera les méthodes locales déjà évoquées pour les regrouper dans un formalisme géométrique. Cette approche sera abordée dans le chapitre suivant.

Chapitre 3

Indexation géométrique étendue

Dans ce chapitre, nous proposons une nouvelle méthode de reconnaissance par indexation de caractéristiques locales. Elle diffère des méthodes existantes (p. ex. [58, 71, 89, 90]), par le fait qu'elle intègre un formalisme géométrique. Ceci permet de l'étendre à d'autres approches sans modification, à condition que celles-ci soient basées sur des primitives géométriques, telles que des points ou des segments. Cette contrainte est beaucoup moins restreinte que celle proposée par d'autres auteurs (p. ex. [88] impose que les descripteurs soient statistiquement discriminants) puisque l'extraction de descripteurs locaux a toujours une base physique (et donc géométrique) dans l'image.

Après un bref rappel du contexte formel de la reconnaissance par informations locales dans § 3.1, nous reviendrons sur la méthode qui nous a servi de base de départ dans § 3.2 en détaillant l'algorithme de mise en correspondance entre deux images et en fournissant les justifications élémentaires du choix de la modélisation par quasi-invariants. Ensuite, nous expliciterons notre méthode en donnant l'algorithme détaillé d'indexation mis en œuvre. Nous l'illustrerons par des résultats avant de présenter une série d'améliorations et une généralisation à d'autres méthodes de reconnaissance dans § 4.2.

3.1 Contexte

Nous avons abordé dans le chapitre 2 plusieurs techniques récentes de reconnaissance locale [60, 71, 89, 90]. Chacune présente les avantages et les inconvénients que nous avons signalés, mais aucune ne prévoit une intégration facile d'autres types de descripteurs, à l'exception de SCHIELE, qui effleure le problème dans [88].

L'intégration d'autres descripteurs paraît, au vu des méthodes actuelles, le seul moyen de pallier leurs carences dans les différentes situations qui leur sont propres, et la seule façon d'endiguer les problèmes de complexité (*cf.* p. 136).

Nous présenterons donc, dans ce chapitre, un environnement unifiant permettant d'intégrer un large spectre de méthodes locales. Celui-ci sera construit autour de la méthode d'*indexation géométrique étendue*, présentée dans [60], sur laquelle nous reviendrons en détail par la suite.

Globalement, la reconnaissance par informations locales consiste à identifier une image inconnue avec un modèle connu. Pour ce faire, des zones de l'image sont mises en correspondance avec des zones équivalentes du modèle (*cf.* chapitre 2). Cette mise en correspondance est, *a priori*, un problème d'optimisation combinatoire difficile. L'introduction d'une structuration des zones locales permet de rendre le problème plus accessible par le biais d'une *indexation*. Cette structuration demande une caractérisation des zones locales que nous appelons *descripteurs*. La modélisation locale et la technique de reconnaissance qui y est associée se résument alors ainsi :

Algorithme 3.1 Reconnaissance locale générique

1° `extraction de configurations ou des supports pour le calcul des descripteurs locaux,`
2° `calcul des descripteurs, localement, sur les supports trouvés,`
3° `indexation des descripteurs,`
4° `mise en correspondance des descripteurs, et, par voie de conséquence, de leurs supports,`
5° `filtrage des mises en correspondance.`

Nous insistons sur le fait que les deux premières parties ont bien un caractère différent. La différenciation entre *localisation* et *caractérisation* est importante pour le déroulement de la généralisation abordée dans la section 4.2.

3.2 Mise en correspondance entre deux images

3.2.1 Description de la méthode

Dans [36, 37], Gros introduit une méthode de mise en correspondance d'images segmentées fondée sur le calcul d'invariants basés sur des transformations affines ou des similitudes. Les similitudes sont composées des isométries et des homothéties. Dans toute la suite, on ne considérera que les similitudes directes, composées d'une translation, d'une rotation et d'une homotétie. L'algorithme calcule des *quasi-invariants* à partir de configurations formées par deux segments adjacents (les *quasi-invariants* et leurs propriétés seront abordés dans la section suivante) qui sont l'angle entre les deux segments θ et le rapport de leurs longueurs ρ. Leur utilisation permet de mettre en correspondance des configurations sous différents points de vue, à condition que ces derniers ne soient pas trop éloignés. La méthode d'appariement de deux images suit alors l'algorithme 3.2, p. 50, avec l'hypothèse que le mouvement apparent entre deux images peut être approché par une similitude. Les étapes principales sont représentées dans la figure Fig. 3.1.

On peut noter que, bien que nous fassions le raisonnement avec des transformations

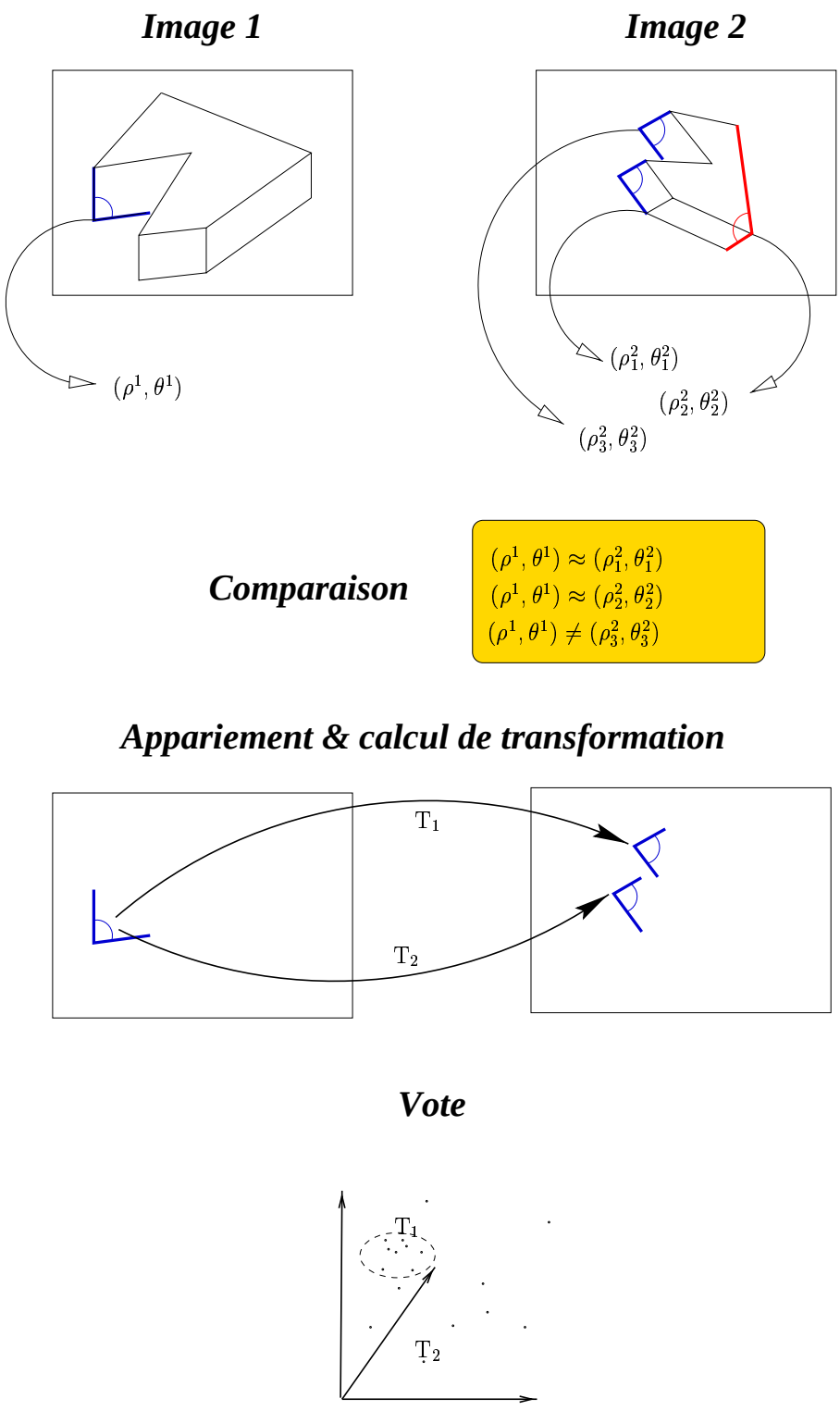


FIG. 3.1: *Algorithme de mise en correspondance de deux images.*

Algorithme 3.2 Mise en correspondance à l'aide de quasi-invariants

Paramètres d'entrée : IMAGE^1, IMAGE^2
Paramètres de sortie : LISTE (Cfg$_i^1$, Cfg$_j^2$)

début
    LISTE = liste vide;

    / Étape 1 : Segmentation */*
    Extraction de contours dans IMAGE^1 et IMAGE^2;
    Approximation polygonale des contours;

    $\mathcal{C}^1$ = ensemble de configurations formées par des couples de
        segments adjacents d'IMAGE^1;
    $\mathcal{C}^2$ = ensemble de configurations formées par des couples de
        segments adjacents d'IMAGE^2;

    / Étape 2 : Calcul des quasi-invariants */*
    pour chaque configuration $c_i \in \mathcal{C}^1$ ou $c_i \in \mathcal{C}^2$ faire
    début
        $\theta(c_i)$ = angle formée par les deux segments;
        $\rho(c_i)$ = rapport des longueurs des segments;
    fin

    / Étape 3 : Mise en correspondance */*
    pour tout couple de configurations $(c_i, k_j) \in \mathcal{C}^1 \times \mathcal{C}^2$ tels que
        $\theta(k_j) - \varepsilon < \theta(c_i) < \theta(k_j) + \varepsilon$ et que $\frac{1}{\eta}.\rho(k_j) < \rho(c_i) < \eta.\rho(k_j)$ faire
    début
        Calculer la similitude Σ_i^j paramétrée par (t_x, t_y, ρ, σ) qui
        projette c_i sur k_j;
    fin

    / Étape 4 : Estimation du mouvement apparent global */*
    Parmi toutes les transformations trouvées, sélectionner celles qui se
    regroupent dans l'amas le plus dense de rayon R (au plus) dans
    dans l'espace de transformation;

    LISTE = liste des couples de configurations ayant participé à la
        formation de cet amas;

    retourner LISTE;
fin

euclidiennes (similitudes), l'algorithme a été testé avec succès sur des configurations caractérisées par des invariants affines, et que tous les points abordés dans cette thèse peuvent être étendus à ces invariants. Il est toutefois nécessaire dans ce cas de bien prendre soin de reconsidérer le fondement théorique des *quasi-invariants* qui ne s'applique alors plus tel quel.

Dans la suite de cet exposé, nous ne nous intéressons pas aux premières étapes énumérées dans l'algorithme 3.2. Elles relèvent de la segmentation d'images, et nous éloigneraient trop de notre but principal. Sans vouloir pour autant éluder les difficultés de la segmentation, le lecteur intéressé est invité à lire [45], par exemple, qui décrit l'outil qui a permis de conduire les expériences décrites dans cette thèse. Nous donnerons, par contre, quelques justifications quant aux hypothèses, quant à l'utilisation de ρ et de θ ainsi que celles relatives au calcul de la transformation dans les sections suivantes. Plus de détails seront donnés lorsque nous aborderons l'*indexation géométrique étendue* proprement dite.

3.2.2 Modélisation par quasi-invariants

L'une des hypothèses principales de l'algorithme consiste à dire que les valeurs ρ et θ calculées lors de la deuxième étape permettent d'identifier des configurations sous différents angles de vue. On la justifie par la théorie des *quasi-invariants*. La modélisation par *quasi-invariants* intervient après le constat affligeant dans la théorie des invariants [22, 17, 68, 2] indiquant qu'il n'existe pas d'invariant pour des configurations quelconques dans le cas des projections de $\mathbb{P}^3$ dans $\mathbb{P}^2$, ou qu'alors, un tel invariant doit être constant sur toutes les configurations.

Les travaux empiriques[1] développés dans [8, 93, 75] ou le cadre formel fourni dans [14] montrent que certaines valeurs, telles que les ρ et θ utilisées par GROS, possèdent des propriétés permettant de les assimiler à des invariants projectifs lorsque le changement de point de vue entre deux images ne varie que faiblement. Cette propriété est formalisée dans la définition de *quasi-invariant*, donnée par BINFORD et LEVITT :

> Soient A un ensemble de configurations, V un ensemble de valeurs numériques, et $\rho : A \to V$ une fonction qui définit des classes d'équivalence sur A :
>
> $$a_1, a_2 \in A; a_1 \approx a_2 \Leftrightarrow \rho(a_2) = \rho(a_1).$$
>
> Une application $\phi : A \to V$ est un quasi-invariant de ρ en $\alpha \in A$ si elle est localement constante sur les classes d'équivalence de A, et si ϕ est localement équivalente à ρ.
>
> ϕ est localement constante, donc invariante, si le développement de TAYLOR pour ϕ en $\alpha \in A$ est constant jusqu'à l'ordre deux. ϕ est localement équivalente à ρ en $\alpha \in A$ si elle a le même développement de TAYLOR jusqu'à l'ordre un.

1. Ce terme n'enlevant en aucun cas de la valeur de ces travaux, mais dénotant simplement la démarche scientifique ayant été à l'origine des résultats présentés, contrairement aux travaux plus théoriques de BINFORD et LEVITT. Les travaux cités proposent par ailleurs des analyses rigoureuses en termes de calcul probabiliste.

Dans le cas de projections de $\mathbb{P}^3$ dans $\mathbb{P}^2$, ρ est typiquement une mesure sur la configuration 3D, et ϕ une mesure sur les points 3D projetés dans l'image 2D. La définition ci-dessus est rappelée dans le but de faire comprendre le comportement général des quasi-invariants : c'est-à-dire par rapport à un invariant à part entière, les quasi-invariants ne varient que très faiblement dans une zone de changement local de point de vue. Le développement complet de la théorie est donné dans [14], ainsi que la démonstration formelle de l'appartenance des rapports de distance et l'angle entre deux segments concurrents à la classe des quasi-invariants.

3.2.3 Invariants *vs.* quasi-invariants

Bien qu'il n'existe pas d'invariants de $\mathbb{P}^3$ dans $\mathbb{P}^2$ pour des configurations générales, il existe de nombreux cas particuliers pour lesquels des invariants projectifs peuvent être calculés [86, 35]. L'utilisation de quasi-invariants nous paraît néanmoins plus adaptée. Nous donnerons ici deux raisons pour ce choix.

Stabilité numérique Plusieurs travaux [67, 24, 59] montrent que le calcul d'invariants projectifs est sujet à de grandes instabilités numériques et que l'utilisation de métriques dans les espaces projectifs impose une étude rigoureuse et coûteuse du comportement des invariants. En général, l'utilisation de ces métriques accentue encore cette instabilité. Les quasi-invariants (qui, dans notre cas, sont de plus des invariants euclidiens ou affines dans le plan) permettent d'utiliser des métriques plus simples et sont numériquement moins fragiles.

Pouvoir descriptif Le prix de cette stabilité est la perte de l'invariance. La validité de la caractérisation par quasi-invariants n'est assurée que dans un faible spectre de changements de point de vue, tandis que les invariants réels couvrent tout l'espace des transformations projectives. C'est justement le fait qu'ils n'absorbent pas l'ensemble des transformations intervenant dans la formation de l'image, qui rend les quasi-invariants utiles. La contrainte d'invariance sous toute transformation projective est trop forte, car elle s'impose aussi dans des cas dégénérés. En utilisant cette propriété, Åström démontre dans [2] que toute courbe fermée est projectivement équivalente à ... un canard (*sic*!). Plus formellement, il démontre que, pour toute courbe fermée, et quelle que soit l'erreur de mesure $\varepsilon > 0$, il existe une transformation projective qui projette cette courbe sur un cercle avec une erreur $\kappa < \varepsilon$. Le fait que le calcul et la comparaison des invariants projectifs restent des opérations très délicates et numériquement fragiles [67, 24, 59], et que le bruit sur les mesures induit une trop grande variation des transformations possibles, a pour conséquence que projectivement, tout objet ressemble, sous un point de vue particulier, à un tout autre objet quelconque, ce qui est précisément l'argument de Burns [17] ou de Åström, entre autres.

Conclusion De ce fait, les invariants projectifs 3D-2D ne sont pas exploitables de façon robuste à des fins de reconnaissance. Puisque leur extraction dans des images 2D n'est pas

possible dans le cas général, leur utilisation passe nécessairement par des hypothèses sur leur configuration, telles que la coplanarité, la symétrie d'objets de révolution, *etc.* [86]

Soit ces hypothèses imposent qu'une segmentation préalable élaborée soit effectuée, soit elles font intervenir des configurations de taille assez élevée (p. ex. 5 points coplanaires) qui rendent l'approche très sensible à des occultations ou au bruit dans les images.

Les *quasi-invariants* ne présentent pas ces inconvénients, et demandent comme seule contrepartie de modéliser les objets 3D par une série limitée d'aspects. L'augmentation de la taille du modèle ainsi obtenu est largement compensée par l'augmentation de la robustesse et la facilité de manipulation des descripteurs.

3.2.4 Vote

Dans le contexte des quasi-invariants, on peut alors plus facilement comprendre l'hypothèse évoquée initialement concernant l'approximation par une similitude du mouvement apparent entre deux images. Elle est utilisée dans la quatrième, et dernière, étape de l'algorithme 3.2, où il s'agit de trouver une approximation du mouvement apparent global. Chaque mise en correspondance préalable de configurations par leurs *quasi-invariants* fournit un « vote » dans l'espace de paramètres des similitudes. Puisqu'elle contient suffisamment d'information pour permettre de calculer la transformation qui projette la première configuration sur la seconde (6 degrés de liberté pour les trois points de la configuration, tandis qu'une similitude en nécessite 4 pour être définie de façon non-ambiguë), chaque mise en correspondance fournit individuellement une transformation entre les deux images. Avec l'hypothèse qui permet de dire que le mouvement apparent global peut être approché par une similitude, et modulo les problèmes de bruit, on peut en déduire que des appariements corrects définiront une transformation identique, tandis que les mises en correspondance erronées définiront des transformations qui sont uniformément distribuées dans leur espace de paramètres lorsqu'elles ne sont pas corrélées. Il y a donc un effet d'accumulation des votes cohérents avec ce mouvement apparent global.

La justification vient du domaine des quasi-invariants. Effectivement, on sait que, dans le cas ou les objets observés se trouvent à une distance suffisamment éloignée de la caméra (la taille de l'objet doit être de l'ordre de 10% de sa distance à la caméra), la meilleure approximation du mouvement apparent est une homographie [97]. Par application du paradigme des quasi-invariants, on sait de plus que, pour des changements de point de vue modérés, les rapports de longueur ρ et les angles θ sont conservés. Or une homographie qui conserve les angles et les rapports de distance est une similitude.

Si l'on accepte l'hypothèse initiale, on prévoit donc que les mises en correspondances correctes se regroupent dans l'espace des paramètres des similitudes. La recherche de la transformation prédominante revient alors à une recherche d'un *amas*[2] dans cet espace de paramètres. La transformée de HOUGH généralisée [47, 5] est un excellent outil permettant de calculer rapidement le point d'accumulation de cet amas.

Quel est le but de cette étape ? Nous venons de mettre en correspondance des configurations qui partagent les même quasi-invariants (à un ε près). Or l'égalité de ces quasi-invariants ne garantit qu'une équivalence locale et pour une transformation arbitraire

2. Le terme anglais correspondant étant « *cluster* ».

(fixée par cet appariement par ailleurs). Or, deux configurations peuvent être localement similaires bien que ne correspondant pas physiquement au même objet. Cette situation est illustrée dans la FIG. 3.1 : parmi les trois configurations candidates, l'une est éliminée pour cause d'incompatibilité de ses quasi-invariants avec ceux de la requête, tandis que les deux autres ont des valeurs similaires avec ces derniers. On note, toutefois, que seule la première configuration donne un appariement correct.

La mise en correspondance des quasi-invariants introduit donc des erreurs inévitables, et dans [34], GRIMSON et HUTTENLOCHER montrent qu'à partir d'une certaine complexité d'images et d'un certain nombre de modèles, ces erreurs deviennent suffisamment nombreuses pour qu'un simple comptage des mises en correspondance ne puisse plus donner la réponse exacte. La vérification par mouvement apparent permet donc d'écarter ces appariements erronés et rend la méthode plus robuste.

3.3 Mise en correspondance entre images multiples

Après avoir donné un résumé de l'approche de mise en correspondance à l'aide de quasi-invariants, nous pouvons aborder le schéma de reconnaissance basé sur cette méthode. Dans ce qui suit, nous supposons que nous disposons d'une base de modèles auxquels nous voulons confronter une image inconnue. Le but est de fournir une mesure permettant d'indiquer lesquels de ces modèles représentent des objets présents dans cette image.

La méthode que nous présentons ici est directement inspirée de la méthode de *geometric hashing* de LAMDAN *et al.* [58, 57, 102] en ce qui concerne le principe général. Il est néanmoins nécessaire de bien différencier les deux méthodes pour les raisons qui seront évoquées dans la section 3.3.5. Nous utilisons des quasi-invariants géométriques extraits d'une image pour calculer une transformation prédominante entre modèles et image. La mesure de confiance associée à la transformation obtenue donne une mesure de reconnaissance entre l'image et le modèle en question. Notre approche trouve ses origines dans des travaux de GROS sur la mise en correspondance entre images présentés dans la section précédente.

3.3.1 Description de la méthode

L'idée directrice de l'algorithme de reconnaissance par *indexation géométrique étendue* est de confronter l'image à chacun des modèles et d'offrir un classement de ces modèles. Il est clair que le coût d'exécution ne permet pas de parcourir tous les modèles et de les comparer un à un à l'image inconnue. Le but est donc d'organiser les quasi-invariants de telle façon pour que ce coût soit réduit à celui équivalent à une mise en correspondance entre deux images.

LAMDAN *et al.* étaient les premiers à proposer un système qui répond à cette contrainte. Leur *Geometric hashing* (que nous traduirons par *indexation géométrique*) permet de stocker les descripteurs locaux des modèles, basés sur des invariants affines, dans une table d'indexation. Au moment de la reconnaissance, un nombre d'accès à la table d'indexation égal au nombre de descripteurs de l'image inconnue suffit à faire ressortir les modèles présents dans l'image. L'approche est décrite en détail dans la section 2.5.1.

Notre façon de procéder est similaire. Les *quasi-invariants* énoncés précédemment serviront de clés d'indexation. Et plutôt que de procéder à un simple vote majoritaire, nous avons recours à la transformée de HOUGH pour chaque mise en correspondance trouvée. Notre méthode se présente donc comme une extension directe des travaux décrits dans [36, 37].

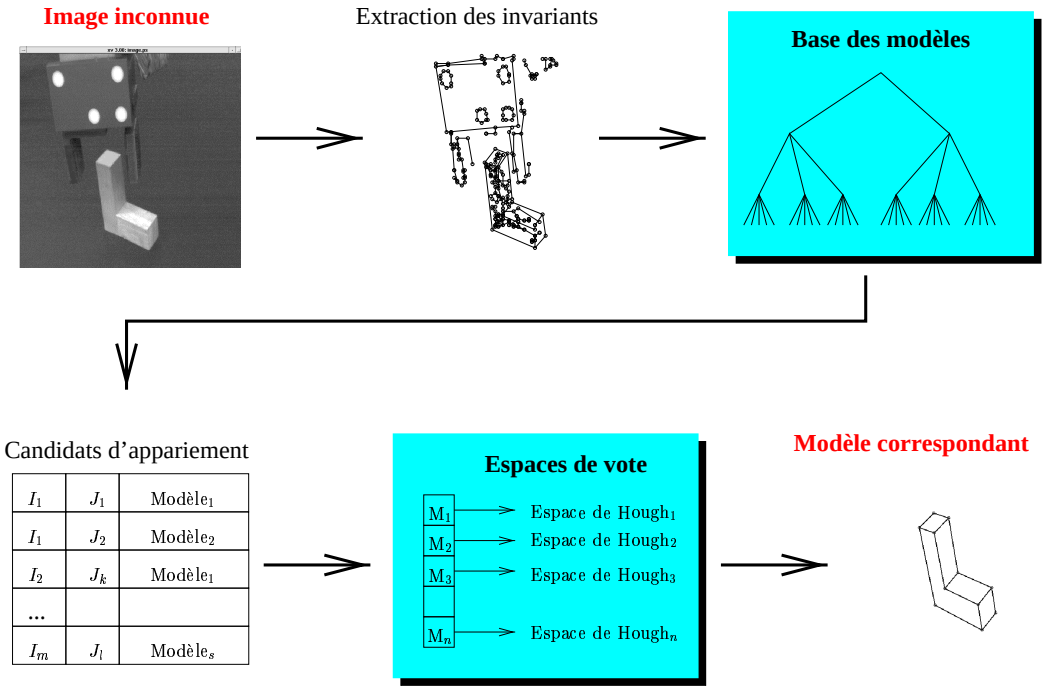


FIG. 3.2: *Algorithme de reconnaissance par* indexation géométrique étendue.

La figure FIG. 3.2 représente sommairement le déroulement de l'algorithme de reconnaissance. En résumé, les deux dernières étapes de l'algorithme 3.2 p. 50 sont transformées pour tenir compte de l'indexation des descripteurs. Cette indexation est obtenue en stockant dans une table bidimensionnelle les configurations de chaque modèle. La clé d'indexation de chaque configuration est obtenue en prenant :

$$\mathrm{f}(\theta,\rho) = \left(\left\lfloor \frac{\theta}{\theta_{\max}} k_\theta \right\rfloor , \left\lfloor \frac{\ln(\rho) + \ln(\rho_{\max})}{2\ln(\rho_{\max})} k_\rho \right\rfloor \right)$$

où k_θ et k_ρ représentent le nombre de paniers[3] disponibles en chaque variable, ou dimension, et où $\theta \in [0..\theta_{\max}[$ et $\rho \in \left]\frac{1}{\rho_{\max}}..\rho_{\max}\right[$ avec $\theta_{\max} = 360$ et $\rho_{\max} = 20$. L'utilisation du logarithme permet d'avoir une distribution des clés uniforme, et facilite la recherche de voisins dans l'espace d'indexation. Le choix des bornes sera expliqué en § 3.3.3, p. 59.

3. Nous utiliserons par la suite indifféremment les termes « *panier* » et « *case* » comme équivalent français des mots anglais usuels « *bucket* » ou « *bin* », à défaut de traduction établie.

L'algorithme 3.3 donne un aperçu de la méthode telle qu'elle a été mise en œuvre :

1. tous les descripteurs des modèles sont indexés suivant le schéma décrit ci-dessus, et stockés ensemble dans une table d'indexation ;
2. une nouvelle image est segmentée et ses descripteurs sont confrontés à ceux dans la base ;
3. chaque mise en correspondance ainsi obtenue est classée selon le modèle auquel appartient le deuxième descripteur ;
4. pour chaque modèle la transformation dominante est calculée comme dans l'algorithme 3.2 ;
5. les modèles sont triés par ordre décroissant de la densité de l'amas trouvé.

3.3.2 Modélisation des objets 3D

Étant donné que la méthode que nous considérons a un caractère fondamentalement 2D dans sa conception, et que l'objectif de la reconnaissance

est de manipuler des objets 3D, nous abordons ici la question de la modélisation et la représentation des objets 3D, afin qu'ils puissent être pris en compte par notre approche.

Nous avons déjà évoqué la caractéristique des quasi-invariants à absorber des changements de point de vue modérés. Cela veut dire que si nous indexons une vue d'un objet 3D, la méthode est, par sa conception même, capable d'identifier toute vue prise d'un point suffisamment proche de la prise initiale. Il semble donc intuitif de vouloir modéliser un objet 3D par une série de vues, un peu comme dans les approches basées sur les graphes d'aspects [51, 52, 69], bien que dans notre cas le problème soit moins formel.

Dans sa thèse [35], GROS propose une méthode de regroupement (*clustering*) permettant d'obtenir une série de vues modèles à partir d'un ensemble de points de vue d'un objet. Le seuil de rupture de cette méthode de regroupement est fonction du nombre de mises en correspondance que la méthode est capable de réaliser.

En ce qui concerne l'indexation et la reconnaissance, il s'agit d'un compromis entre différents facteurs. D'une part, le nombre de modèles augmente la qualité de la mise en correspondance, mais, d'autre part, cette augmentation est confrontée à des contraintes liées aux temps de calcul et à la consommation de mémoire. Plus les points de vue sont rapprochés, plus la modélisation prendra de la place, et plus il y aura de collisions dans les paniers d'indexation, puisqu'il y aura plus de modèles dans la base. Par contre, dans ce cas, il est possible d'être très strict sur la comparaison entre quasi-invariants ainsi que sur le calcul du mouvement apparent global, ce qui rendra la phase de vote plus rapide. D'un autre côté, le fait d'utiliser des points de vue plus espacés pour la modélisation diminuera bien le nombre de modèles et l'espace nécessaire. Ceci entraînera une baisse de collisions d'indexation. Puisqu'il faudra être plus lâche sur la comparaison des invariants et sur l'estimation du mouvement apparent, la phase de vote s'en trouvera affectée par une baisse de performance. Il s'agit donc de trouver une modélisation adaptée qui dépend des images traitées.

Algorithme 3.3 Indexation géométrique étendue

```
Paramètres d'entrée : IMAGE, MODÈLES[1..n]
                      /* L'image et les modèles sont segmentés et
                         on connaît leurs quasi-invariants ; les modèles
                         sont indexés dans une base */
Paramètres de sortie : LISTE (Cfg_i^1, Cfg_j^m)
                      /* m est l'index d'un modèle connu */

début
    LISTE = liste vide;

     /* Les étapes 1 et 2 de l'algorithme 3.2 sont supposées faites */

     /* Étape 3bis : Mise en correspondance */
    pour toutes les configurations c_i ∈ IMAGE faire
    début
        INDEX = indice d'accès dans la base d'indexation, f(θ(c_i), ρ(c_i));
        pour toutes les configurations k_j indexées à INDEX faire
        début
            Calculer la similitude Σ_i^j paramétrée par (t_x, t_y, ρ, σ) qui
            projette c_i sur k_j;
        fin
    fin

     /* Étape 4bis : Estimation du mouvement apparent global */
    Trier les Σ_i^j suivant le modèle auquel appartient k_j;

    pour tous les modèles m dans MODÈLES[1..n] faire
    début
        Parmi toutes les transformations trouvées pour lesquelles k_j
        appartient à m, sélectionner celles qui se regroupent dans l'espace
        de transformation de m dans l'amas de rayon R (au plus) qui
        soit le plus dense;
    fin

    Sélectionner le modèle m ayant l'amas le plus dense parmi les modèles;

    LISTE = liste des couples de configurations ayant participé à la
        formation de l'amas du modèle m;

    retourner LISTE;
fin
```

3.3.3 Indexation

Maintenant que nous disposons du cadre d'application de notre méthode, nous sommes en mesure d'étudier en détail la phase d'indexation des quasi-invariants de l'algorithme 3.3.

L'indexation doit permettre un accès rapide aux descripteurs qui y sont stockés et, par la même occasion, prendre en compte la variabilité des quasi-invariants. Plusieurs auteurs soutiennent l'hypothèse selon laquelle les clés d'indexation doivent être distribuées au mieux (c'est-à-dire, en s'approchant au mieux d'une distribution uniforme) parmi les paniers d'indexation. Cette observation est valable dans le contexte d'une indexation de valeurs exactes. Dans ces cas, la valeur du descripteur est unique au bruit de mesure près, et il est inutile, voire même encombrant, de considérer d'autres valeurs dans la proximité directe des points. Par conséquent, Il n'est pas nécessaire de préserver dans la table d'indexation la topologie de l'espace initial. Ce point de vue forme la base de nombreuses approches d'indexation et de stockage des données [84, 101, 38], basées sur le partitionnement des données, et il a été appliqué avec succès au *Geometric hashing* [33, 6].

Nous ne partageons pas ces points de vue pour trois raisons principales.

1° Des études récentes (*cf.* § 5.3) indiquent que les méthodes basées sur le partitionnement des données perdent de leur efficacité dans des espaces de grande dimension. Ceci gênerait l'extension de la méthode à d'autres types de descripteurs où la taille des descripteurs s'accroît sensiblement.

2° La variation des *quasi-invariants* ne garantit plus que le panier d'indexation accédé lors d'une requête soit identique à celui dans lequel le correspondant exact est indexé. Seule la prise en compte de la topologie de l'espace permettra de parcourir le voisinage du descripteur de façon efficace pour rechercher le correspondant de la requête.

3° La dernière raison est liée à la conception plus générale de la reconnaissance. Dans les approches liées aux invariants [86], et celles dérivant directement du *Geometric hashing*, le but est d'identifier rapidement la mise en correspondance correcte, et de formuler un vote pour le modèle concerné. Tout appariement superflu introduit un biais dans le vote. Le nombre de votes erronés doit donc être réduit au maximum, ce qui est traduit par l'organisation des données. Notre approche par quasi-invariants considère la mise en correspondance uniquement comme un filtrage des données clairement aberrantes. C'est en effet la phase de vote dans l'espace de HOUGH qui est responsable du tri final entre les faux appariements et les corrects. La modélisation rend explicitement compte de la variabilité des quasi-invariants et absorbe de ce fait l'incertitude qui peut par ailleurs exister sur la mesure des valeurs entrant en ligne de compte. Ceci allège donc les contraintes imposées par les autres méthodes, et rend l'approche également plus robuste.

Nous proposons donc de conserver la topologie de l'espace des invariants dans notre structure de stockage. La façon la plus évidente de réaliser ceci est de considérer un tableau multi-dimensionnel (dans notre cas, le nombre de dimensions est 2) qui discrétise de façon uniforme l'espace. Se posent alors deux questions liées à la mise en œuvre : comment gérer

un tableau de très grande taille en mémoire et qu'entend-on par discrétisation uniforme de l'espace ?

Organisation de la mémoire *A priori*, la méthode d'*indexation géométrique étendue* n'a besoin d'indexer que des informations qui pourraient être maintenues dans un simple tableau à deux dimensions. Néanmoins, si l'on prend en compte que nous prévoyons d'étendre notre méthode à d'autres descripteurs (*cf.* § 4.2.2) qui évoluent dans des espaces de plus grande dimension, il est nécessaire d'optimiser l'utilisation de la mémoire[4]. Nous avons donc opté pour une organisation de nos données dans un arbre de type *quadtree* (ou sa généralisation à des dimensions supérieures) à profondeur fixe. La raison de ce choix est principalement qu'une organisation de ce type préserve la topologie de l'espace, ce qui rend la recherche des voisins relativement aisée et qu'elle permet de n'allouer que les zones de la base des index qui sont effectivement utilisées. Chaque feuille terminale de l'arbre contient une liste de couples (I, M) d'invariants I avec leur modèle M associé.

L'utilisation d'un arbre à profondeur fixe vient du fait que la comparaison des quasi-invariants se fait à un seuil près. Comme indiqué dans § 3.3.2, une certaine variation de leurs valeurs est permise, et en-deçà de cette variation les quasi-invariants sont considérés comme égaux. Il est donc inutile de vouloir indexer leurs valeurs dans une structure à profondeur variable et donc à précision infinie.

Outre le fait qu'une organisation à précision infinie revient à appliquer un partitionnement des données, et non pas de l'espace, approche controversée du point de vue exprimé ci-dessus, l'utilisation d'arbres à profondeur finie permet de réduire la complexité de leur structure et d'optimiser les parcours et les recherches. Des détails de mise en œuvre sont donnés dans l'annexe A.

Indexation uniforme Un point important est l'uniformité des différentes dimensions d'indexation. Nous insistons sur le fait qu'il nous importe peu que les valeurs indexées soient distribuées de façon uniforme. Il est important en revanche, pour la comparaison de ces valeurs, que globalement, les clés d'indexation soient uniformément réparties dans leur espace par rapport aux requêtes. e que nous entendons par là est que, quelle que soit la case d'indexation, la probabilité qu'elle soit occupée doit être indépendante d'une connaissance *a priori* des modèles. Il se peut que pour des modèles particuliers certains paniers soient plus remplis que d'autres (et même de façon significative), mais cela n'enlève rien au fait que la probabilité *a priori* soit uniforme. La raison en est simple : si la répartition théorique des clés n'est pas uniforme, la comparaison des descripteurs par accès à ces cases d'indexation es n'a plus de sens, comme le montre MORIN dans sa thèse [67]. Une autre façon de formuler cette uniformité est de considérer l'incertitude ε autour des descripteurs. On fait l'hypothèse que la zone d'incertitude autour d'une valeur est connexe et que les différentes dimensions ne sont pas corrélées. Ainsi, pour une incertitude ε donnée, la valeur du descripteur $\mathcal{D}$ évolue dans une zone $A \leq \mathcal{D} \leq B$, où A et B dépendent du ε choisi et de la valeur de $\mathcal{D}$. Pour mieux exprimer cette dépendance, on notera $A = \mathrm{f}_{\min}(\mathcal{D}, \varepsilon)$, et

4. Nous n'avons pas étudié la possibilité de mettre en place un système de type SGBD avec stockage en mémoire secondaire. La problématique propre à ce sujet est un domaine de recherche à part entière.

$B = \mathrm{f}_{\max}(\mathcal{D}, \varepsilon)$

Ce qui nous intéresse alors, pour une valeur de descripteur $\mathcal{D}$ donnée, est de trouver tous les descripteurs indexés $\mathcal{D}_i$ tels que $\mathrm{f}_{\min}(\mathcal{D}, \varepsilon) \leq \mathcal{D}_i \leq \mathrm{f}_{\max}(\mathcal{D}, \varepsilon)$. Dans l'espace des descripteurs, le résultat de f n'est pas forcément linéaire en ε ou $\mathcal{D}$. Si l'on prend comme exemple la comparaison des ρ dans l'algorithme 3.2, on constate que $\mathrm{f}_{\min}(\rho, \varepsilon) = \frac{1}{\varepsilon}\rho$ et que $\mathrm{f}_{\max}(\rho, \varepsilon) = \varepsilon\rho$. À des fins d'optimisation du nombre d'accès à la base d'indexation, il serait intéressant qu'en termes d'index ceci se traduise en une simple opération linéaire : $I(\mathcal{D}) - I_\varepsilon \leq I(\mathcal{D}_i) \leq I(\mathcal{D}) + I_\varepsilon$. Pour la comparaison des rapports de longueur ρ, nous obtenons ceci en introduisant un logarithme dans le calcul des index. À partir d'une valeur pour le rapport des longueurs ρ, l'index I_ρ dans la dimension appropriée est :

$$I_\rho = \left\lfloor \frac{\ln(\rho) + \ln(\rho_{\max})}{2\ln(\rho_{\max})} k_\rho \right\rfloor$$

Ainsi, la recherche des invariants disposant d'une valeur $\tilde{\rho}$ telle que :

$$\frac{1}{\eta}.\rho < \tilde{\rho} < \eta.\rho$$

se limite à l'intervalle :

$$\left]\left\lfloor \frac{\ln(\frac{1}{\eta}.\rho) + \ln(\rho_{\max})}{2\ln(\rho_{\max})} k_\rho \right\rfloor \ldots \left\lfloor \frac{\ln(\eta.\rho) + \ln(\rho_{\max})}{2\ln(\rho_{\max})} k_\rho \right\rfloor\right[$$

ce qui, après réarrangement des termes, est égal à :

$$]I_\rho - k_\eta \ldots I_\rho + k_\eta[$$

avec :

$$k_\eta = \left\lfloor k_\rho \frac{\ln(\eta)}{2\ln(\rho_{\max})} \right\rfloor$$

constante pour une base et une précision donnée. La borne $\rho_{\max}$ qui est introduite est égale à 20 dans notre cas. Cette valeur est arbitraire, mais traduit à notre avis le rapport de longueur maximal que l'on peut espérer détecter dans une image segmentée de façon fiable dans les images 512×512 que nous utilisons principalement. Au delà, l'influence du bruit devient trop importante.

Accès aux données Avec cette organisation des données, l'accès à un invariant devient trivial : on calcule la clé d'indexation d'un invariant, et on récupère tous les couples $(I_k, \mathcal{M}_k)$ stockés à l'endroit indiqué par cette clé.

Soit Δ_ε une zone d'incertitude autour d'un descripteur requête, et soit I la fonction d'indexation. On appellera $\mathrm{I}(\Delta_\varepsilon)$ *l'image* de cette zone d'incertitude. Afin de prendre la zone Δ_ε en compte lors d'une mise en correspondance, il suffit donc de considérer tous les descripteurs indexés dans $\mathrm{I}(\Delta_\varepsilon)$.

La zone d'incertitude est paramétrée par deux facteurs : le bruit de mesure, et l'effet des quasi-invariants. Si on prend l'hypothèse que l'erreur commise sur la mesure de la longueur du segment ε est proportionnelle à cette longueur, on en déduit facilement que l'influence du bruit s'inscrit directement dans la façon de procéder de notre algorithme. Si on suppose que, pour les deux segments intervenants, la longueur réelle du premier est connue à $\varepsilon_1 = k.l_1$ près (où l_1 est sa longueur mesurée), et de façon similaire, celle du second est connue à $\varepsilon_2 = k.l_2$ près, on en déduit pour le quasi-invariant ρ que :

$$\frac{l_1 - \varepsilon_1}{l_2 + \varepsilon_2} \leq \frac{l_1}{l_2} \leq \frac{l_1 + \varepsilon_1}{l_2 - \varepsilon_2}$$

En réarrangeant les termes, on conclut que :

$$\frac{l_1}{l_2}\left(\frac{1-k}{1+k}\right) \leq \frac{l_1}{l_2} \leq \frac{l_1}{l_2}\left(\frac{1+k}{1-k}\right)$$

En posant $\eta = \frac{1+k}{1-k}$ on retrouve le formule $\rho.\frac{1}{\eta} \leq \rho \leq \rho.\eta$ utilisée dans l'algorithme précédemment décrit. De plus Ben-Arie [8] montre que la variation des quasi-invariants due au changement de point de vue présente le même comportement. Par exemple, en observant une configuration de deux segments adjacents sous tous les angles de vue possibles, la probabilité d'observer ρ_{obs} telle que $\rho_{\text{réel}}.\frac{1}{2} \leq \rho_{\text{obs}} \leq \rho_{\text{réel}}.2$ est de l'ordre de 86%. Dans notre cas où la variation du point de vue est plus modérée, le facteur η peut être choisi plus petit.

En prenant maintenant en compte notre façon de calculer des index décrite dans le paragraphe précédent, on voit que la zone d'incertitude autour de nos descripteurs se traduit par un hypercube dans notre espace d'indexation. Lors d'une requête, il suffit donc d'accéder également aux voisins pour prendre en compte toute variation modélisable de la valeur de nos quasi-invariants.

3.3.4 Vote

La phase de vote est la partie cruciale de notre méthode. Elle permet d'éliminer les appariements erronés parmi ceux trouvés lors de l'étape d'indexation. Elle s'appuie sur le fait que la théorie des *quasi-invariants* permet d'affirmer que localement, et pour des changements de point de vue modérés, le mouvement apparent entre deux images s'approche raisonnablement d'une similitude. Si on ajoute l'hypothèse supplémentaire que les objets 3D sont observés d'une distance suffisante pour que les changements de point de vue entre deux aspects n'induisent pas d'effet de perspective notable, on peut conclure que, pour un aspect donné, les mouvements apparents locaux entre configurations du modèle et ceux de l'image observée ne varient que faiblement. En d'autres termes, il est possible de vérifier la cohérence entre différentes mises en correspondance par le fait que les transformations locales qu'elles induisent devraient être très proches.

La vérification des appariements initiaux se réduit donc à trouver parmi eux, ceux qui définissent un mouvement apparent similaire. La transformée de Hough généralisée est un outil particulièrement adapté à nos besoins.

Justification Il est intéressant d'évoquer au passage les critiques sévères que GRIMSON et HUTTENLOCHER émettent dans [34] envers ce type d'approche. Ils y formulent trois objections majeures que nous reprenons ici.

1. Des vecteurs de transformations similaires aboutiront dans des paniers différents s'ils se trouvent de part et d'autre des frontières de discrétisation. Le problème est accru en cas d'incertitude sur les paramètres de cette transformation.

2. Dans des espaces de grande dimension, la table de vote peut devenir très grande, rendant ainsi la recherche d'amas fastidieuse.

3. La probabilité de formation aléatoire d'amas denses peut être élevée de part le fait que la discrétisation intègre le bruit en regroupant tous les événements aléatoires à l'intérieur d'un panier. Cette probabilité dépend du rapport du nombre de votes avec le nombre de cases dans l'espace de vote.

Nous écartons le premier argument par notre indexation à double niveau, décrite dans la deuxième partie de cette section, p. 64. L'introduction de deux niveaux d'indexation permet le chevauchement de paniers d'indexation, limitant ainsi l'effet évoqué. La figure FIG. 3.4 donne par ailleurs une bonne indication de l'erreur que l'on peut commettre.

Le second argument perd de sa validité si l'on considère que notre structure de vote repose sur une organisation arborescente de type *quadtree/octree* étendue à une dimension quelconque. Grâce à cette organisation, la structure peut être très creuse et prend une place quasiment minimale si la table a un faible taux de remplissage. La recherche de l'amas maximal peut alors être maintenue en ligne, avec une complexité linéaire dans le nombre de votes, ou bien elle peut être calculée *a posteriori*, avec une complexité linéaire dans le nombre d'urnes remplies. On remarquera au passage que le nombre d'urnes est toujours inférieur au nombre de votes.

La dernière objection peut être rejetée par l'observation que les configurations utilisées sont sémantiquement beaucoup plus riches. Nos configurations présentent deux degrés de liberté de plus que nécessaire au calcul du vote (qui doivent également être identiques, à une certaine tolérance près), et de plus nous ne considérons pas tout triplet de points présent dans l'image, mais nous imposons que ce triplet soit connexe par des segments, ce qui en réduit considérablement le nombre.

Cet ajout d'information contribue au fait que : premièrement le nombre de votes diminue de façon très significative, relevant les seuils sur la complexité et le nombre des modèles ; et deuxièmement, il introduit une modification de la distribution des événements aléatoires considérés dans [34] de sorte que la probabilité d'une aberration soit inférieure à celle utilisée dans l'analyse (*cf.* [8] pour une justification).

Pour illustrer la diminution du nombre de votes, la FIG. 3.3 montre un cas simple de mise en correspondance entre deux figures planes, identiques à une similitude près. Nous avons compté le nombre de votes dans l'espace de HOUGH que demanderait une mise en correspondance par différentes méthodes.

Dans le cas d'une transformée de HOUGH généralisée, toutes les configurations qui forment une base dans l'espace des similitudes (c'est-à-dire tout couple de points) sont

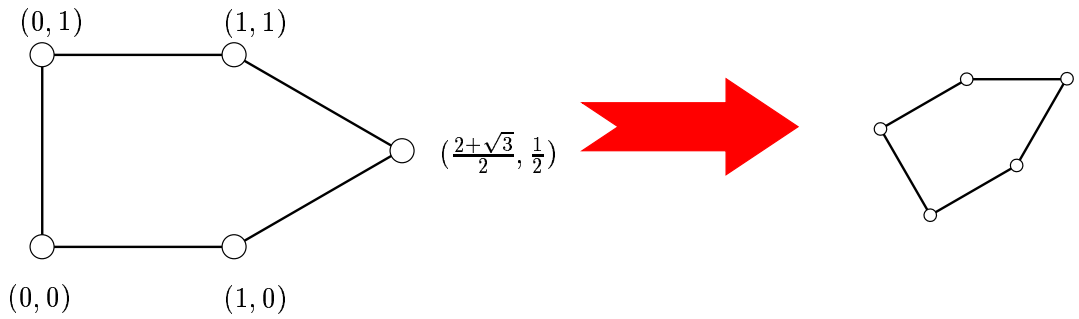


FIG. 3.3: *Configuration de test pour le dénombrement des votes.*

Type	Nb. de config.	Nb. d'invariants	Nb. de votes
HOUGH généralisé	20	N.A.	400
Geometric hashing	20	$20 \times (5-2) = 60$	142
Indexation géométrique étendue	5	$5 \times 2 = 10$	9

mises en correspondance avec toute configuration dans la deuxième image, ce qui donne n^2 votes pour $n = p(p-1)$ configurations, où p est le nombre de points dans l'image. La méthode de *Geometric hashing* dénombre la même quantité de configurations de base, mais calcule pour chaque configuration $p-2$ invariants qui sont utilisés comme filtre pour l'expression des votes. Notre méthode utilise les segments, qui sont généralement d'un nombre comparable au nombre de points, puis on énumère tous les couples de segments adjacents, ce qui donne un nombre de configurations de l'ordre de $3p$ ou $4p$ pour les images les plus complexes. Chaque configuration est alors décrite par deux invariants, ce qui fournit entre $2p$ et $8p$ invariants qui sont utilisés comme filtre pour l'expression des votes.

Revenons maintenant sur la principale difficulté dans les méthodes de type HOUGH : la recherche d'un point d'accumulation dans un espace à dimension quelconque. De plus, étant donné que l'espace des paramètres du groupe de transformations considéré (ici les similitudes) étant *a priori* infini, se pose aussi le problème de l'organisation des données.

L'organisation des données. Nous nous affranchissons de la taille de l'espace en considérant que les objets que nous observons évoluent dans un monde réel, avec des conditions de prises de vue réelles. De ce fait, les 3 paramètres dans le groupe des similitudes à support infini : la translation en x, T_x, la translation en y, T_y, et le changement d'échelle σ, sont contraints de respecter un certain « réalisme ». Ainsi, les translations sont limitées par la taille de l'image observée. Des transformations trop grandes projetteraient l'objet en dehors du champ de vision, ce qui est absurde. De même, on peut considérer que, pour un modèle et un objet à reconnaître dans des images de taille identique, un changement d'échelle supérieur à 3 ou inférieur à $\frac{1}{3}$ entraîne des effets de bord (notamment liés à la

segmentation) tels qu'une reconnaissance raisonnable ne peut être envisagée.

Maintenant que notre espace de recherche est devenu fini, il peut facilement être discrétisé et implanté comme une table multi-dimensionnelle. Chaque case de cette table devient alors une urne qui reçoit, au fur et à mesure de la vérification des appariements locaux, des votes pour l'intervalle de transformations qu'elles représentent [5].

Recherche de l'amas principal. Nous avons indiqué au début de cette section que l'hypothèse d'un mouvement apparent global approché par une similitude n'est valable que lorsque la déformation projective locale entre les deux vues est identique sur l'ensemble de l'image. Ceci n'étant jamais vérifié pour des objets 3D, les mouvements apparents locaux des mises en correspondance ont tendance à s'éloigner de cette approximation théorique. Cette dispersion est proportionnelle à l'inclinaison de l'objet et à sa taille par rapport à la distance de la caméra. De plus, des erreurs de mesure sur les configurations dues au bruit classique dans les images, induisent également des incertitudes sur les similitudes calculées. Afin de trouver les appariements résultant en une transformation globale similaire, il est nécessaire de prendre en compte cette diffusion lors de la recherche du point d'accumulation dans l'espace de vote.

Pour la recherche de l'amas d'accumulation dans l'espace discrétisé des transformations, nous devons inclure une zone autour de chaque emplacement lors du calcul de sa cardinalité. C'est alors cette cardinalité qui désignera la case ayant reçu le plus grand nombre de votes dans son voisinage.

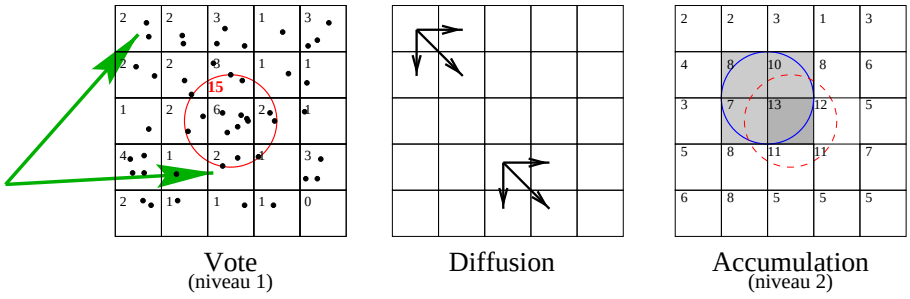


FIG. 3.4: *Système de vote à double niveau d'indexation.*

Nous avons développé une méthode de vote à double niveau d'indexation. Le premier niveau représente la discrétisation évoquée précédemment. Il reçoit les votes pour chaque mise en correspondance. Le deuxième niveau représente une discrétisation plus grossière, superposée à la première. Il représente les zones de recherche pour les amas d'accumulation. La particularité du deuxième niveau est qu'il possède le même nombre d'urnes que le niveau inférieur, malgré sa plus forte granularité. En effet, ses cases se trouvent partiellement superposées, permettant ainsi d'absorber les problèmes classiques de recherche de groupements dans des structures discrètes.

5. Nous utiliserons dans la suite indifféremment les termes « *espace de* HOUGH », « *espace des transformations* » ou « *espace de vote* ».

Au moment d'un vote, l'urne du premier niveau est incrémentée, et son état est communiqué à tous les paniers du second niveau auxquels elle appartient. Ainsi, les urnes du second niveau connaissent à tout moment le nombre de votes qu'elles contiennent, et il suffit, à la fin de la phase de vote, d'accéder au panier de ce niveau pour connaître le mouvement apparent prédominant, et les mises en correspondance qui le définissent. La finesse de granularité du premier niveau détermine la précision de positionnement de l'amas recherché de cette façon.

Le principe décrit ci-dessus est illustré dans la figure FIG. 3.4 dans le cas où la granularité des deux niveaux diffère d'un facteur deux. Dans cet exemple, on suppose que l'espace des transformations est de dimension $d = 2$, et que les paniers de second niveau sont indexés par leur case de premier niveau se trouvant en bas à droite. Un vote arrive dans une urne du premier niveau, elle communique son état aux cases de second niveau auxquelles elle appartient, et celles-ci mettent à jour leur décompte. Ainsi, à la fin de la phase de vote, les paniers de second niveau contiennent le nombre de votes qu'elles possèdent. Dans la figure, l'amas théorique est indiqué par le cercle rouge dans la partie « vote ». Il est repris dans la partie « accumulation » en pointillés. La zone trouvée par notre schéma est représentée en grisé.

3.3.5 Différences par rapport à des méthodes existantes

Nous abordons dans cette section une comparaison de notre méthode avec d'autres approches qui se basent sur des primitives semblables ou qui ont des mises en œuvre similaires. Le *geometric hashing* est une méthode d'indexation qui calcule des clés à partir d'invariants affines, les approches dérivant du *peaking effect* modélisent les objets avec les angles et rapports de longueurs entre segments, et la dernière méthode se base sur une cohérence géométrique.

Geometric hashing. L'approche de LAMDAN *et al.* [58, 57, 102] a déjà été évoquée plusieurs fois dans cette thèse, notamment dans la section § 2.5.1. À l'origine elle se base sur des invariants affines, tandis que nous utilisons principalement des invariants euclidiens (qui sont par la même occasion des *quasi-invariants* projectifs). Les extensions ultérieures de l'approche ont toujours gardé cette optique [81, 82, 99]. Une première différence fondamentale avec notre approche est que les invariants mis en jeu dépendent d'une base, laquelle doit être connue pour que les invariants aient une quelconque utilité. Ceci est indirectement dû au fait que la méthode n'est pas fondée sur la notion de « configuration », mais sur celle de « primitive ». On extrait des primitives des images, en général des points ou des droites, et on construit toutes les configurations possibles permettant de calculer les types d'invariants utilisés. Dans notre méthode ce sont les configurations elles-mêmes qui sont extraites, ce qui constitue une deuxième différence importante. La troisième différence consiste dans le fait que le mécanisme de vote du *geometric hashing* est juste un comptage de mises en correspondance entre les configurations, tandis que nous combinons ce vote avec une mesure de cohérence globale des appariements.

Les deux premiers points sont liés, puisque le fait d'exprimer la caractérisation des primitives en fonction d'une base arbitraire oblige à prendre en compte toutes les bases

possibles, ce qui mène à l'explosion combinatoire des configurations (*cf.* FIG. 3.3). Dans notre cas, les configurations contiennent suffisamment de degrés de liberté pour permettre une caractérisation indépendante. Outre l'avantage évoque lié à la complexité, cette méthode ne dépend pas de la présence de certaines primitives dans l'image (notamment celles qui forment la base), et est alors plus robuste aux occultations.

Un autre inconvénient de la création de configurations par l'énumération de combinaisons de primitives est que les configurations n'ont pas, dans ce cas, nécessairement une signification dans le modèle, tandis que notre approche permet de supposer que les segment utilisés font partie d'une caractéristique réelle du modèle.

Le fait que la méthode procède à un simple comptage de votes par modèle, et qu'elle n'impose pas de cohérence globale entre les votes la rend particulièrement sensible quand le nombre de modèles dépasse la dizaine ou lorsque les modèles sont trop complexes. Dans ces cas, le vote majoritaire simple a tendance à accumuler plus de bruit que d'information pertinente. Ceci mène alors à une situation où la réponse des modèles devient purement aléatoire [34]

Le « Peaking Effect ». Basés sur *l'effet de pic*, décrit par [8], les travaux de SHIMSHONI et PONCE [93], et ceux d'OLSON [75] mettent en œuvre une modélisation basée sur l'indexation des angles et les rapports de longueur de segments adjacents. Les auteurs prennent une approche résolument 2D-3D dans le contexte d'un calcul de pose. Ceci les contraint à utiliser uniquement des configurations d'au moins 4 points, afin de pouvoir déterminer cette pose 3D. Ils partent donc d'une modélisation *a priori* de leur modèle 3D, et précalculent la probabilité qu'une configuration de 4 points, caractérisée par ses angles et rapports de distance soit projetée en telle ou telle valeur. La donnée d'une mesure 2D leur permet ensuite, par indexation, de retrouver les correspondants 3D les plus probables (dans le contexte d'un seul modèle) et de calculer une transformation modèle-image définie par cette mise en correspondance locale, un peu comme nous calculons le mouvement apparent local. L'utilisation d'un vote de HOUGH pour valider la transformation globale permet alors d'aboutir.

Cette méthode est proche de celle que nous développons par le fait qu'elle s'appuie sur les mêmes fondements théoriques, bien que l'approche citée ici soit conceptuellement probabiliste et la nôtre géométrique. La principale différence est que les auteurs utilisent des données 3D avec une modélisation explicite, tandis que notre approche reste 2D avec une modélisation par l'apparence.

Reconnaissance non structurée. Il existe des méthodes de reconnaissance à partir de données géométriques non structurées [20]. Ces méthodes peuvent être considérées comme des extensions directes à la modélisation par l'apparence du paradigme géométrique évoqué dans le chapitre 1. L'auteur développe une méthode d'alignement de primitives entre une image et un modèle explicite. Le modèle peut être 2D ou 3D. Étant donné que les primitives utilisées sont des points, il est nécessaire de les enrichir d'information supplémentaire afin de pouvoir calculer, pour chaque mise en correspondance, la transformation d'alignement de l'image avec le modèle. Il ajoute ainsi, par exemple, une notion d'orientation dans le cas

d'une transformation rigide. Puisqu'il n'utilise pas de méthode de vote par transformée de HOUGH ou de discrétisation, il maintient des informations exactes concernant tous les mouvements apparents entre modèle et image, et peut à la fin du processus garantir l'exhaustivité des poses trouvées. La perte occasionnée par l'abandon de la structuration et le regroupement des primitives utilisées par les autres méthodes, contraint celle-ci à être exhaustive, ce qui lui confère une complexité élevée.

Récapitulatif Le tableau 3.1 suivant résume les particularités de chaque méthode décrite. La complexité est exprimée en fonction du nombre de primitives dans l'image.

TAB. 3.1: *Résumé des différentes méthodes citées.*

Méthode	Configurations	Structuration	Complexité	Cohérence
Geometric hashing	4 points	invariants affines	n^4	–
Peaking effect	3 segments adj.	quasi-invariants	n^2	Alignement 2D-3D
Non structuré	1 point enrichi	–	n^4	Alignement 2D-2D ou 2D-3D
Indexation géométrique étendue	2 segments adj.	quasi-invariants	n^2	Mouvement apparent 2D-2D

3.4 Exemples et résultats

Cette section regroupe des exemples de situations dans lesquelles nous avons testé notre approche. Une première expérimentation valide notre approche pour l'identification d'un objet 3D simple dans une image pour lequel on dispose d'une description CAO. La seconde montre qu'elle reste valable pour des objets plus complexes, mais pour lesquels la segmentation est facile et peu bruitée. Dans le cas d'objets simples, réels (c'est-à-dire, ne provenant pas de modélisation CAO), pour lesquels la segmentation n'est pas toujours optimale, on observe quelques défaillances de notre système que nous analysons.

3.4.1 Utilisation de modèles CAO

Dans cette expérimentation nous disposons d'un modèle 3D CAO d'un objet usiné que nous voulons reconnaître dans une scène afin de guider un robot. Nous procédons donc en trois étapes :

1. construction des vues-modèles à indexer,
2. indexation des vues-modèles,
3. confrontation de l'image inconnue.

Pour la première étape, nous avons projeté cet objet dans 200 positions différentes, correspondant à des points sur la sphère de vue espacés de façon régulière. Après application de la méthode de regroupement de GROS [35], que nous avons déjà évoquée, nous

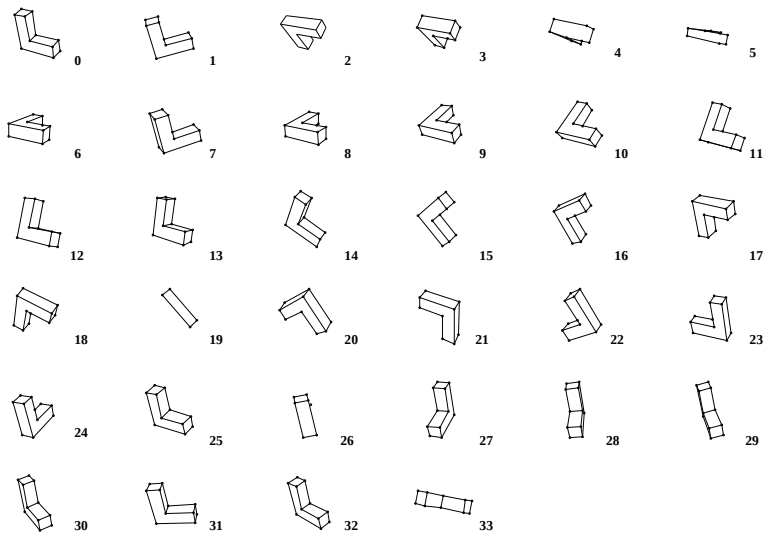


FIG. 3.5: *Les modèles* CAO *utilisés lors de l'expérimentation.*

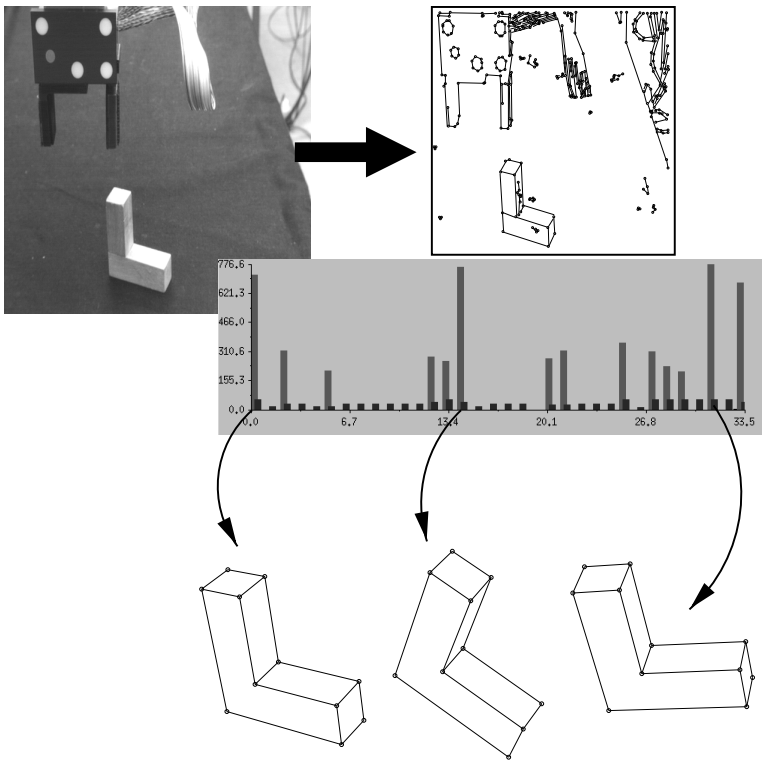


FIG. 3.6: *Reconnaissance de données réelles à partir de modèles* CAO.

avons obtenu 34 vues-modèles qui nous ont servi comme base de reconnaissance, et qui sont représentées en FIG. 3.5.

La seconde phase consiste à calculer les *quasi-invariants* de ces 34 vues, et à les indexer selon la méthode que nous avons décrite dans ce chapitre.

Dans FIG. 3.6 nous montrons le résultat de la reconnaissance de notre système. L'image représente l'objet modélisé en présence d'une pince de robot. L'histogramme qui y est présenté indique la réponse du système pour chaque modèle. En effet, chaque barre de cet histogramme représente le nombre de votes dans l'amas de plus grande densité pour chacun des 34 modèles, le numéro en abscisse correspondant à celui de FIG. 3.5. Nous avons représenté les modèles correspondant aux trois meilleurs scores : modèles 0, 14 et 31 de la figure FIG. 3.5. Tous étaient mis en correspondance avec l'objet recherché et représentaient des aspects 2D très proches. Le modèle ayant obtenu le quatrième meilleur score correspondait à un appariement erroné avec du bruit dans le fond de l'image. Le temps de reconnaissance était de l'ordre de l'une seconde sur une *UltraSparc1* à 140 Mhz. On remarquera que le bruit a tout de même réussi à engendrer un score élevé.

3.4.2 Influence de la phase de vote

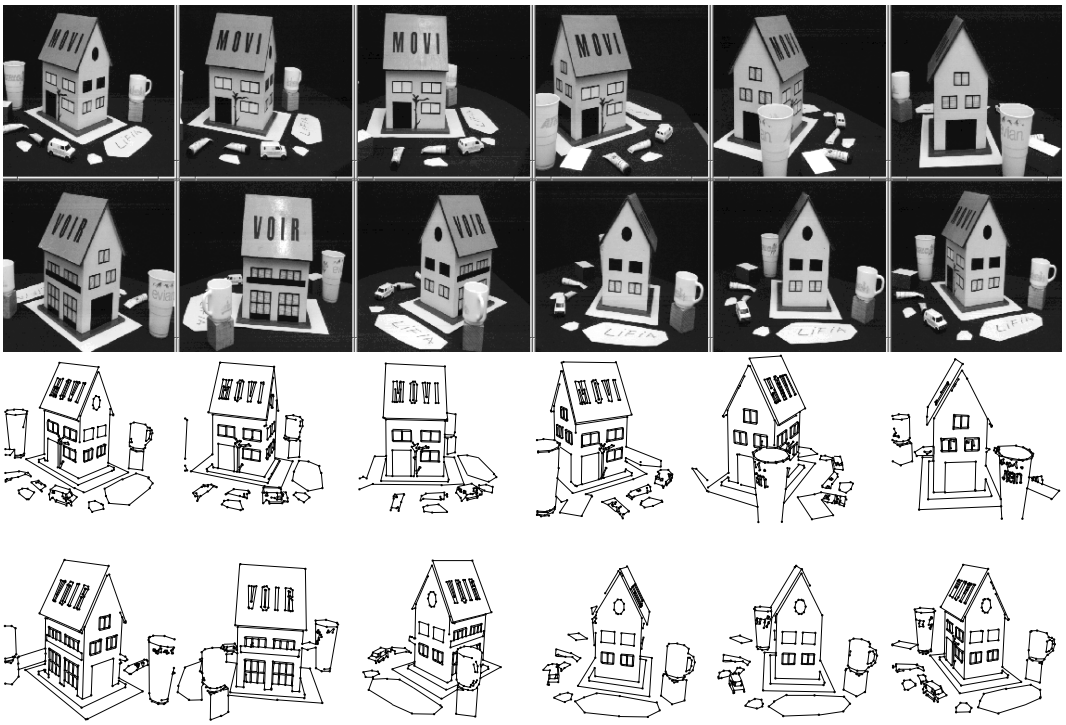


FIG. 3.7: *Images modèles utilisées.*

Nous montrons par cette expérience que la phase de vérification de la cohérence locale est fondamentale pour notre méthode, et par la même occasion nous montrons que notre approche fonctionne sur des modèles plus complets que dans le cas précédent. Nous avons

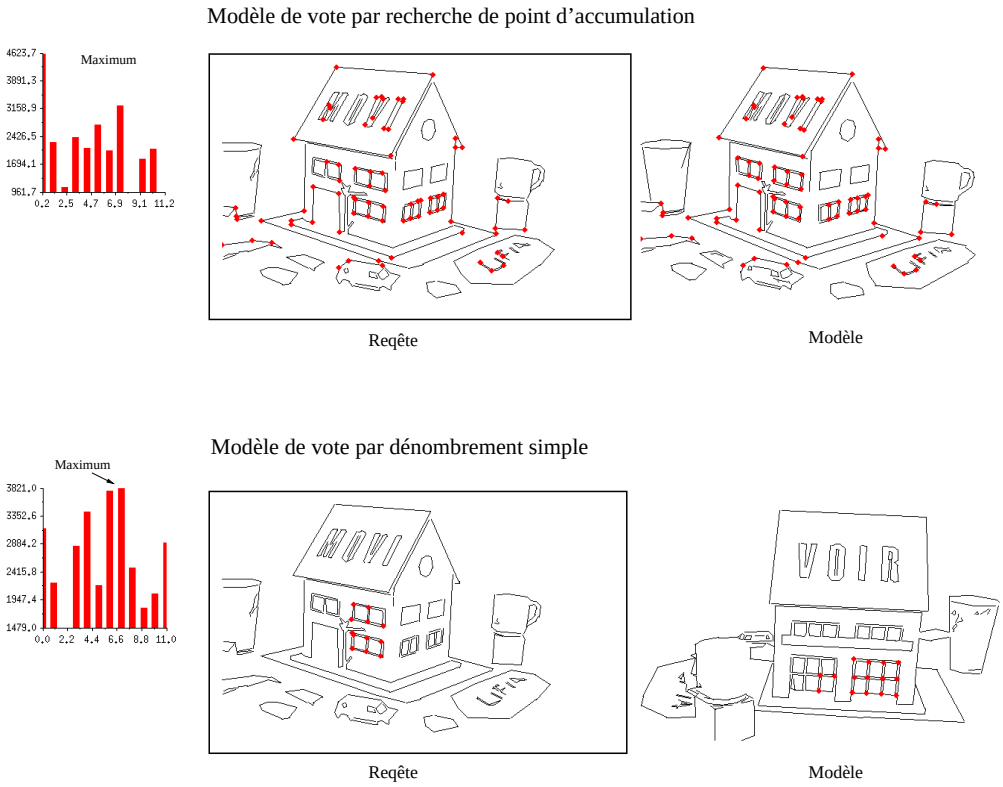


FIG. 3.8: *Influence de la phase de vote pour la reconnaissance.*

modélisé un objet 3D par 12 vues (*cf.* FIG. 3.7) qui correspondent à des prises de vue espacées de 30°. Ensuite nous avons présenté à notre système une image de cet objet, prise d'un point de vue intermédiaire (nous disposons d'une série de 120 vues, prises tous les 3°). Nous avons pris en considération le simple dénombrement des votes d'une part, et la recherche d'un point d'accumulation de l'autre. Dans l'exemple représenté par la figure FIG. 3.8, où l'on a seulement présenté une image à la base, la première partie montre la distribution des densités des zones d'accumulation pour les 12 modèles, et la mise en correspondance qui en résulte. La seconde partie montre la distribution du nombre de votes reçus par chaque modèle. On remarque que les modèles ayant obtenu le plus de votes ne sont pas nécessairement ceux qui ont obtenu l'amas d'accumulation le plus dense. En observant la mise en correspondance résultant du modèle qui a reçu le plus grand nombre de votes, on constate, premièrement que le modèle ne correspond pas à la requête, et que, deuxièmement, une structure répétitive est à l'origine du point d'accumulation. Ce sont notamment l'orthogonalité et la répétitivité des segments extraits au niveau des fenêtres permettent des mises en correspondance multiples avec une petite zone dans l'image requête.

Plus globalement, si on essaie de reconnaître chacune des 120 vues, avec comme modèle une vue tous les 30°, et si on représente les résultats de la reconnaissance sur un graphe, avec en abscisse l'indice de l'image et en ordonnée le modèle avec lequel elle est mise en correspondance, on devrait observer un graphe « *en escalier* », puisque le mouvement représenté par les vues est continu. C'est ce que nous vérifions avec la figure FIG. 3.9. Dans cette figure nous représentons simultanément les résultats de reconnaissance avec et sans contrainte géométrique. On observe aisément que la reconnaissance avec contrainte géométrique respecte relativement bien le schéma prédit (seules 22 images sur 120 – 17% – sont mal classées), tandis que celle sans contrainte donne un résultat constant : toutes les images sont identifiées à un seul modèle, celui qui contient le plus d'invariants, et qui, par la même occasion, génère le plus de votes.

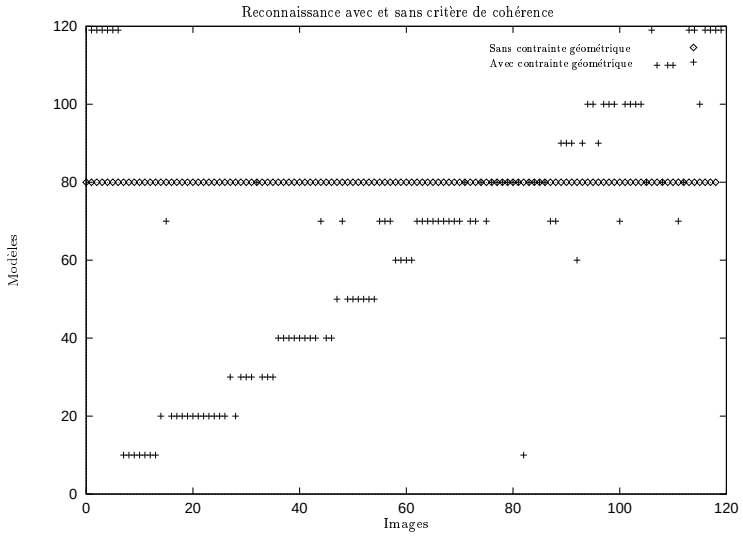


FIG. 3.9: *Reconnaissance avec et sans critère de cohérence.*

3.4.3 Identification parmi différents modèles

Jusqu'à maintenant, nous avons présenté des expériences faisant intervenir un modèle d'objet 3D unique. La base d'indexation contenait alors différentes vues 2D d'un seul objet, et la tâche consistait à le reconnaître dans une scène observée, éventuellement sous un angle de vue non modélisé. Dans cette partie nous introduisons un ensemble d'objets 3D différents, et nous essayons de les identifier dans une scène.

Nous reprenons la base de tests utilisée par SOSSA dans [95]. Elle est représentée dans la figure FIG. 3.10. Les images qui nous ont servi de requête figurent dans FIG. 3.11.

Nous avons recueilli les réponses de notre système pour chaque image de test. Dans 13 cas sur 16 (81%) le premier objet trouvé est correctement identifié dans la scène. Par contre, les résultats se détériorent rapidement lorsque nous voulons identifier tous les objets présents dans la scène. Les deux raisons principales de cette défaillance sont d'ordre différent. La première tient du fait que tous les aspects des objets ne sont pas modélisés. La deuxième raison est plus fondamentale. En effet, la majorité des fausses réponses est due au fait que certains modèles sont très ressemblants à d'autres. Considérons la situation montrée dans la figure FIG. 3.12

Suite à du bruit de segmentation, le coin grisé de l'objet présent dans l'image de droite a été mal extrait, et contient un grand nombre de petits segments. Les seuls segments représentatifs restants sont montrés en traits gras. Or, en analysant les modèles connus, on constate que trois objets présentent une configuration identique à l'objet que nous considérons. Cette expérience montre clairement une limite de notre système, car il ne dispose d'aucun moyen supplémentaire pour différencier les solutions proposées.

Cette situation est accentuée par le fait que les images ne présentent que très peu de d'invariants (une quinzaine au maximum) et qu'en somme, la transformation globale est calculée sur très peu de mises en correspondance. Ceci rend l'approche très instable pour de petits objets. Les mauvais classements sont donc principalement dus au fait que les objets sont assez pauvres et très bruités par rapport à leur taille. Les réponses erronées proviennent principalement du fait d'un alignement accidentel de bruit dans le fond avec deux ou trois invariants d'un modèle ce qui suffit, dans ce cas, à faire échouer notre méthode.

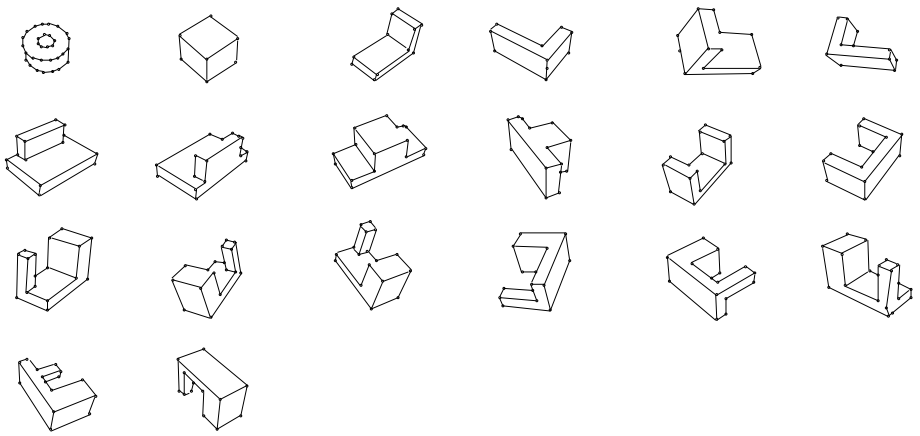


FIG. 3.10: *Images modèles pour une base hétérogène simple.*

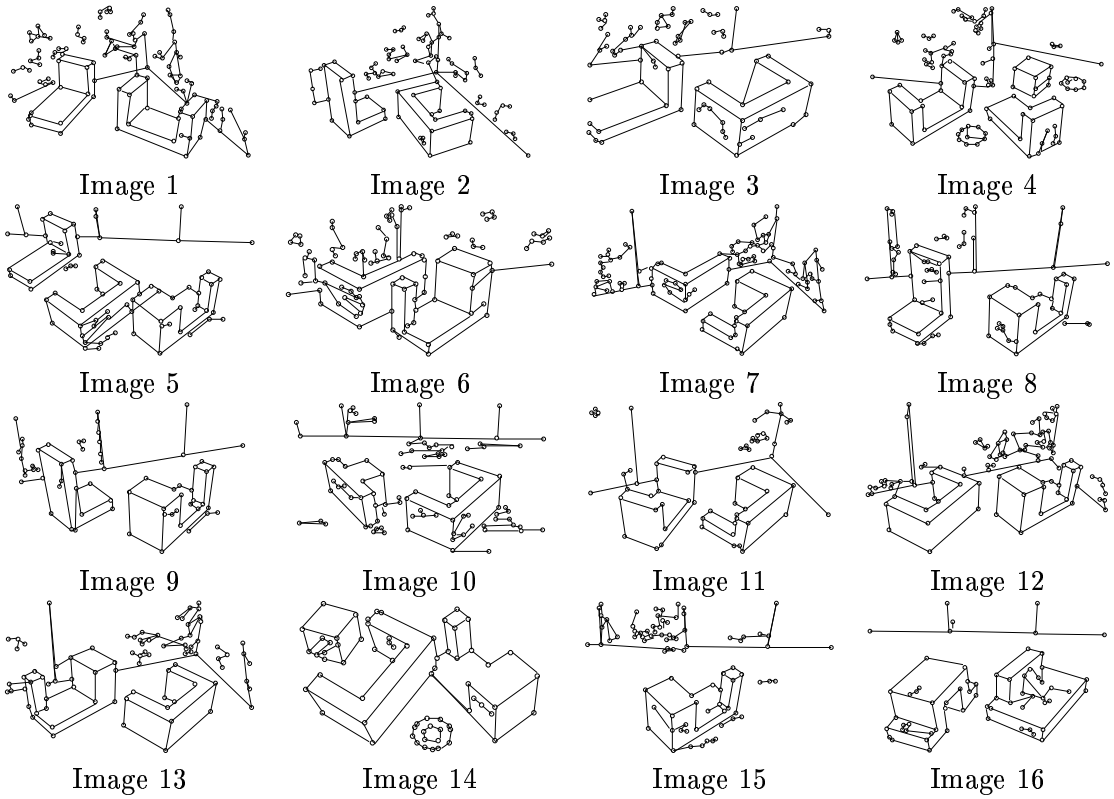


FIG. 3.11: *Images de test pour une base hétérogène simple.*

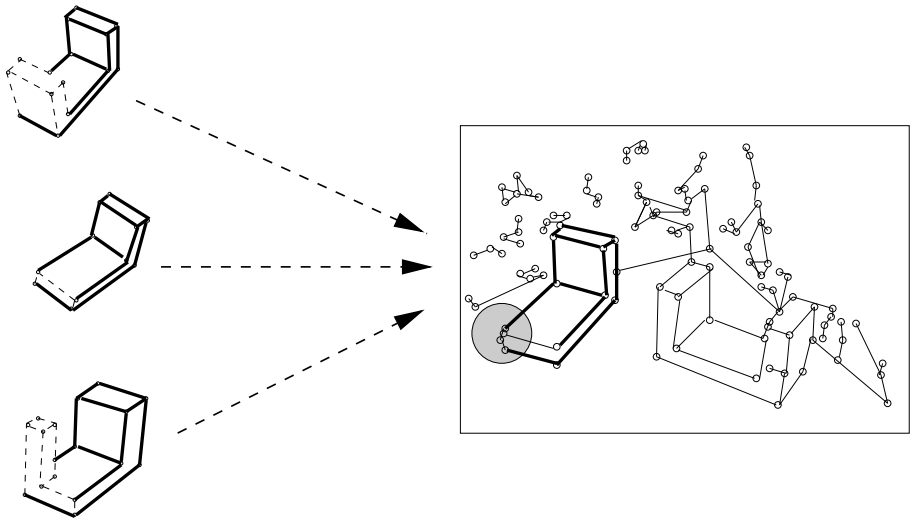


FIG. 3.12: *Exemple de défaillance du système ; les modèles sont trop ressemblants.*

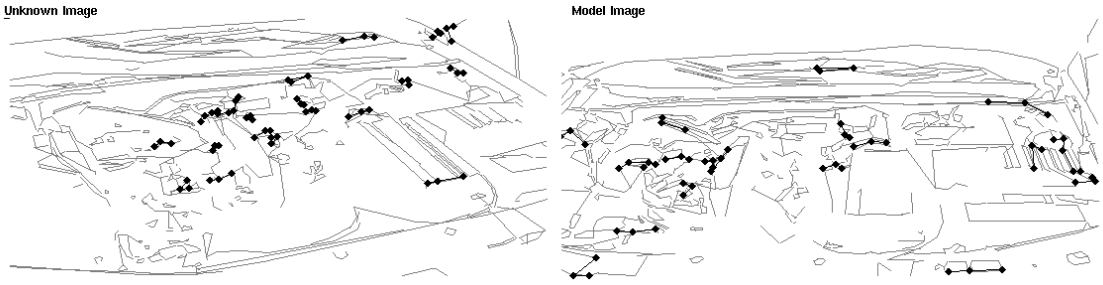


FIG. 3.13: *Exemple d'échec : les images sont trop bruitées.*

3.5 Limites de l'approche

Outre le problème évoqué dans la section précédente, et qui concerne les modèles trop simples (*cf.* FIG. 3.12). L'inverse est vérifié également. Les modèles trop complexes font aussi échouer la méthode.

Nous avons représenté, dans la figure FIG. 3.13, le résultat d'une simple tentative de mise en correspondance (c'est-à-dire que la base de modèles est réduite à une image, et que l'on s'intéresse aux couples de primitives qui ont contribué à l'amas le plus dense dans l'espace de vote). Il s'agit d'images d'un moteur de voiture, pris sous deux angles différents.

Le résultat est bien sûr géométriquement cohérent, mais on constate qu'il ne correspond pas à une mise en correspondance exacte. Ceci est principalement dû au fait que le bruit dans les images fait que la segmentation provoque l'apparition une grande quantité de petits segments, rendant l'utilisation de configurations simples, en « V », comme les nôtres, trop instables pour une reconnaissance. De plus, la mise en correspondance requiert de l'ordre de 7 à 8 minutes CPU sur une *UltraSparc30* à 250 Mhz.

3.6 Conclusion du chapitre

Dans ce chapitre nous avons abordé une nouvelle approche à la reconnaissance par apparence. Elle s'appuie sur une segmentation préalable des images en contours, les contours eux-mêmes étant approchés par des segments de droite.

Les images sont modélisées par un ensemble de configurations de segments adjacents. Ces configurations sont caractérisées par des quasi-invariants, ce qui leur permet d'être indexées dans une base de modèles. Ensuite, nous avons proposé une organisation de ces quasi-invariants afin de pouvoir déterminer rapidement, à partir de critères purement locaux, quels modèles peuvent être des candidats potentiels pour une image inconnue. Après ce premier tri, nous proposons une méthode de vérification de la cohérence géométrique globale de type *transformée de* HOUGH, qui permet d'éliminer des appariements initiaux erronés et qui classe les modèles en fonction de leur pertinence par rapport à l'image requête.

Un ensemble de tests nous a montré que l'approche est valable, et exploitable dans les cas où la segmentation est suffisamment propre pour fournir des données peu bruitées. Dès lors que la segmentation devient trop bruitée la méthode devient moins stable et donne plus souvent de mauvaises réponses. On constate également que pour des objets très complexes, les temps d'exécution augmentent sensiblement.

Chapitre 4

Généralisation à d'autres types de primitives

DANS le chapitre précédent, nous avons présenté une méthode de reconnaissance et de modélisation basée sur deux axes principaux: une description géométrique locale à partir de segments extraits de l'image, et un critère de cohérence globale qui s'appuie sur le calcul d'une approximation de mouvement apparent entre l'image inconnue et les modèles. Les résultats obtenus avec des images synthétiques et avec des modèles simples en § 3.4 ont montré que l'approche est valide, bien que peu exploitable dans le cas d'images réelles complexes. Dans ce chapitre, nous étendrons cette méthode pour qu'elle soit plus robuste, donc exploitable dans des cas réels, d'une part, et qu'elle fournisse un lien de collaboration entre méthodes existantes d'autre part.

La section § 4.1 introduira une extension des configurations locales utilisées précédemment qui permettra d'obtenir de meilleurs résultats de reconnaissance, sans pour autant augmenter le temps de traitement. La section § 4.2 montre que le paradigme de vérification globale par mouvement apparent peut être généralisé à d'autres types de configurations, dont nous en testons quelques uns. Elle se base sur un principe de collaboration qui permet de faire coopérer différentes modélisations locales existantes dans un même schéma global.

4.1 Extensions directes

Dans cette section nous abordons deux extensions de notre système de reconnaissance. Elles ont principalement pour but d'augmenter la vitesse de reconnaissance en introduisant des descripteurs plus riches, évoluant dans des espaces de plus grande dimension. Le but est de distribuer au mieux les index dans la base pour que moins de descripteurs répondent dans la première phase de mise en correspondance.

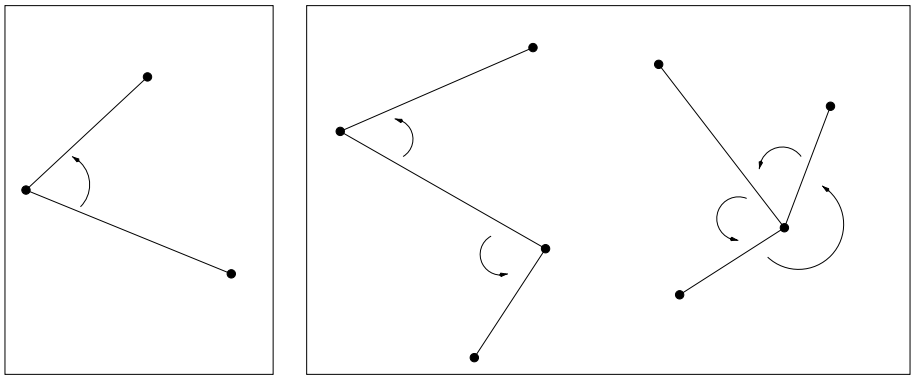


FIG. 4.1: *Configurations avec 3 sommets vs. configurations avec 4 sommets.*

4.1.1 Augmentation de la taille de la configuration

Jusqu'à maintenant, nous calculions nos quasi-invariants sur des configurations minimales de 3 points connectés par deux segments. Il est tout à fait possible de calculer les mêmes valeurs sur des configurations faisant intervenir 4 points connectés (*cf.* FIG. 4.1).

Suivant ces configurations (en « Z » ou en « Y »)[1] on peut calculer deux ou trois couples (ρ, θ), obtenant ainsi des descripteurs de dimension 4 ou 6.

Nous reprenons l'expérience décrite p. 71 dans laquelle nous utilisions 12 vues-modèles d'une maison d'une séquence de 120 vues obtenues en faisant un tour complet autour de l'objet 3D. Nous modélisons maintenant nos objets de trois façons : soit en utilisant uniquement des configurations en « Z », soit en utilisant uniquement des configurations en « Y », soit en utilisant les deux ensemble.

Influence sur le nombre d'invariants par modèle

FIG. 4.2 montre le nombre d'invariants pour chaque type de configuration et pour chaque image de la séquence. On note que le nombre d'invariants énumérés varie fortement selon le type de configuration utilisé. Intuitivement on voit que les configurations en « Y » doivent être plus rares que celles en « V », puisqu'elles ont une contrainte plus forte sur le point de concours des segments. Il est clair aussi que, dès qu'une figure présente des cycles dans son graphe de segments connectés, le nombre de configurations en « Z » augmente et devient plus grand que celui des configurations en « V ». Dans le cas où l'on utiliserait une combinaison des configurations « Z » et « Y », on doublerait approximativement le nombre d'invariants par image par rapport aux configurations en « V ».

Influence sur le temps d'exécution

Dans la figure FIG. 4.3 nous avons représenté le temps d'exécution nécessaire à la reconnaissance de chaque image. On constate tout de suite une très forte corrélation entre le

1. Dans la suite, nous référerons aux configurations initiales, calculées sur 3 points connectés, comme configurations en « V ».

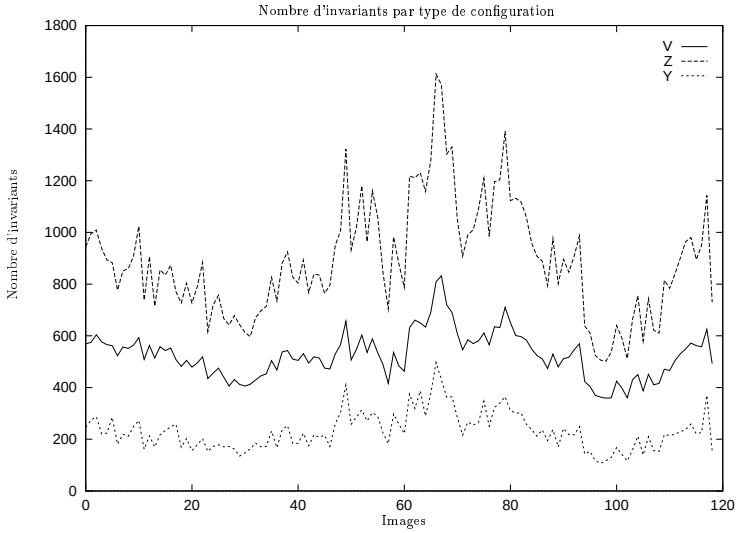


FIG. 4.2: *Le nombre d'invariants par modèle suivant la configuration utilisée ; de haut en bas « Z », « V », « Y ».*

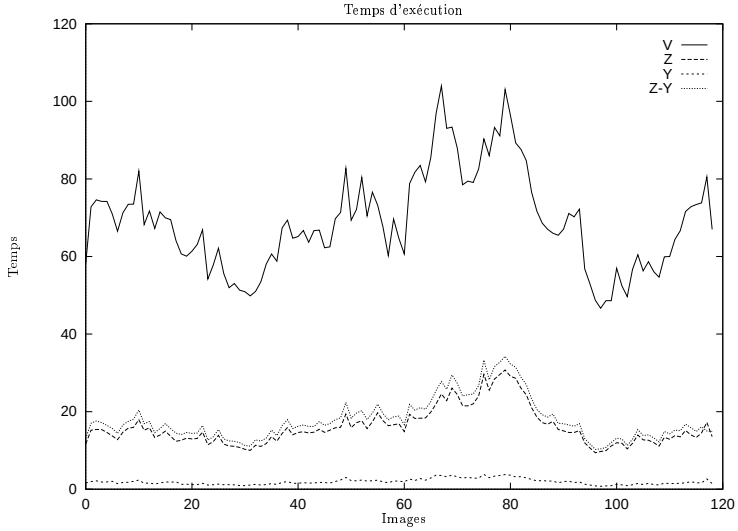


FIG. 4.3: *Temps d'exécution pour la reconnaissance avec des configurations à trois ou à quatre sommets, « V », « Z » et « Y ». Dans l'ordre, de haut en bas, les configurations « V », « Z-Y », « Z » et « Y ».*

nombre d'invariants et le temps d'exécution, les courbes représentant une forme similaire dans tous les cas. On note néanmoins que les configurations « Z » et « Y » ont des performances nettement supérieures à celles en « V ». Ceci est dû au fait que les invariants sont de plus grande dimension (4 pour « Z » et « Y » par rapport à 2 pour « V »). La relation exacte entre les performances en termes de complexité et temps de calcul sera détaillée dans le chapitre suivant. De plus, on constate que le temps d'exécution en utilisant à la fois des configurations en « Z » et en « Y » est la somme du temps pour les configurations prises individuellement.

On peut d'ores et déjà conclure que l'utilisation de ces nouvelles configurations diminue de façon sensible le temps d'exécution.

Influence sur la qualité de la reconnaissance

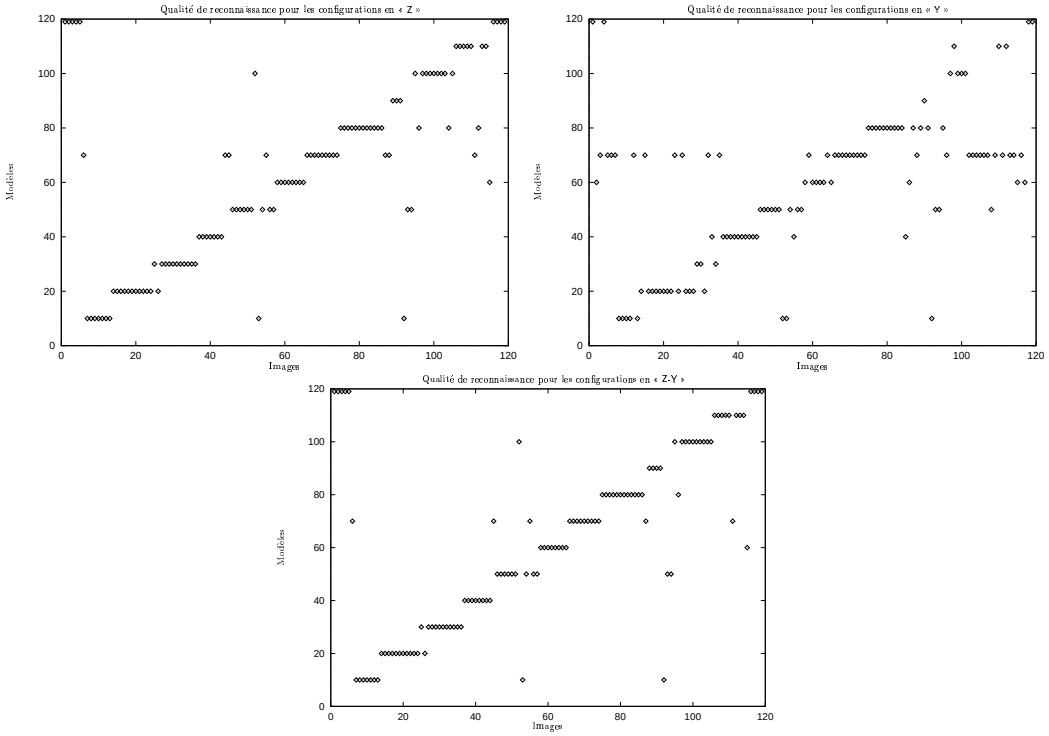


FIG. 4.4: *Influence des configurations à quatre sommets « Z » et « Y » sur la qualité de la reconnaissance. Dans l'ordre, de gauche à droite et de haut en bas, les configurations « Z », « Y » et « Z-Y » combinées.*

La figure FIG. 4.4 montre le résultat d'une expérience identique à celle décrite p. 71. On a confronté chacune des 120 images aux 12 modèles et on a représenté en ordonnée le modèle que le système nous a proposé comme correspondant, et ceci pour les trois types de description : par configurations en « Z », en « Y » et la combinaison des deux. On constate que l'utilisation des configurations en « Z » ne change en rien la qualité de reconnaissance obtenue précédemment. Par contre, les configurations en « Y » semblent beaucoup moins

performants pour ce type de reconnaissance. On obtient des résultats quasi-équivalents aux originaux en utilisant la combinaison des deux.

Conclusion

L'introduction d'invariants de plus grande dimension fait sensiblement décroître le temps d'exécution (d'un facteur 3 environ) sans pour autant affecter en quelque façon la qualité de la reconnaissance. Le gain de temps d'exécution se paie par une augmentation d'ordre équivalent de l'espace de stockage nécessaire pour la description des modèles et des images.

4.1.2 Partitionnement de l'espace des descripteurs

Une autre façon d'augmenter la taille des descripteurs est d'introduire de nouvelles valeurs calculées sur la configuration initiale. Étant donné que nos configurations ne présentent plus d'invariants géométriques pour des similitudes, nous introduisons une valeur invariante à une autre classe de transformations : les changements d'illumination. Inspirés des résultats annoncés par CARLSSON [19], nous avons classé nos configurations suivant l'orientation des segments qu'elles font intervenir. L'orientation est définie par la direction du gradient le long du segment, faisant en sorte que pour un parcours de la première extrémité à la seconde, le gradient soit toujours orienté vers la gauche. Une fois les segments orientés, les configurations peuvent être classées en quatre catégories pour celles faisant intervenir 3 points en V, et en huit pour celles avec 4 points en Z ou en Y (*cf.* FIG. 4.5).

Influence sur le temps d'exécution

FIG. 4.6 montre les résultats en temps d'exécution pour la même expérimentation que celle décrite dans la section précédente. On remarque tout de suite l'allure similaire à la courbe observée précédemment (FIG. 4.3) à cette différence près que l'échelle verticale a diminué d'un facteur 3.

FIG. 4.7 montre le gain obtenu. Elle représente le temps d'exécution sans orientation divisé par celui avec orientation sur le même échantillon. On constate que le gain est de l'ordre de 3 pour les configurations en « V », de l'ordre de de $3,5$ pour des configurations en « Z » et de l'ordre de $1,7$ pour les configurations en « Y ». Le gain est également de l'ordre de $3,5$ pour l'utilisation combinée de « Z-Y », qui n'est pas représentée sur cette figure.

Influence sur la qualité de la reconnaissance

Si on reprend le même schéma expérimental que celui de la section précédente, on constate que l'introduction de l'orientation augmente la qualité de la reconnaissance (FIG. 4.8) de façon sensible.

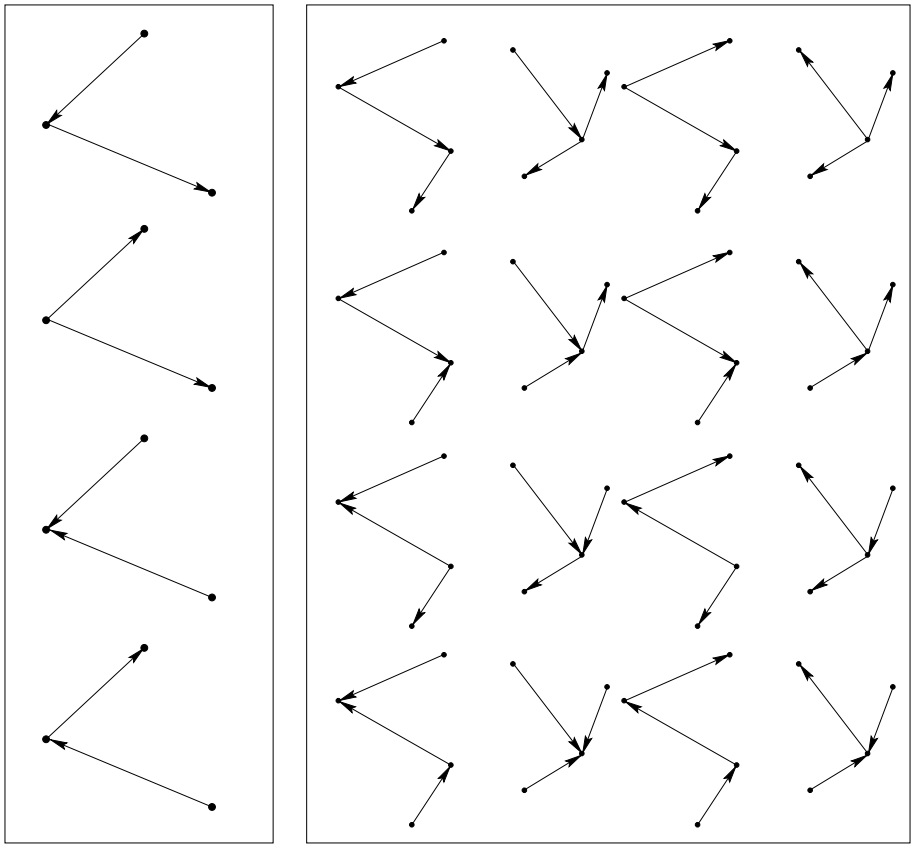


FIG. 4.5: *Configurations orientées avec 3 sommets vs. configurations orientées avec 4 sommets.*

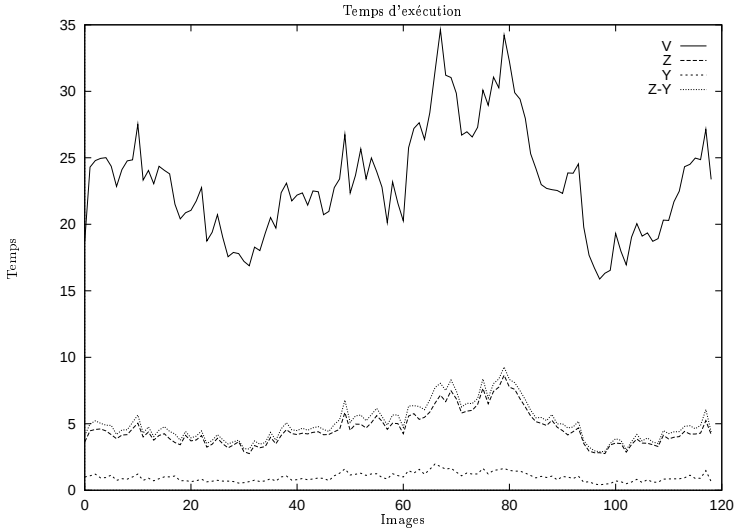


FIG. 4.6: *Temps d'exécution pour la reconnaissance avec des configurations à trois ou à quatre sommets, « V », « Z » et « Y » avec orientation. Dans l'ordre, de haut en bas, les configurations « V », « Z-Y », « Z » et « Y ».*

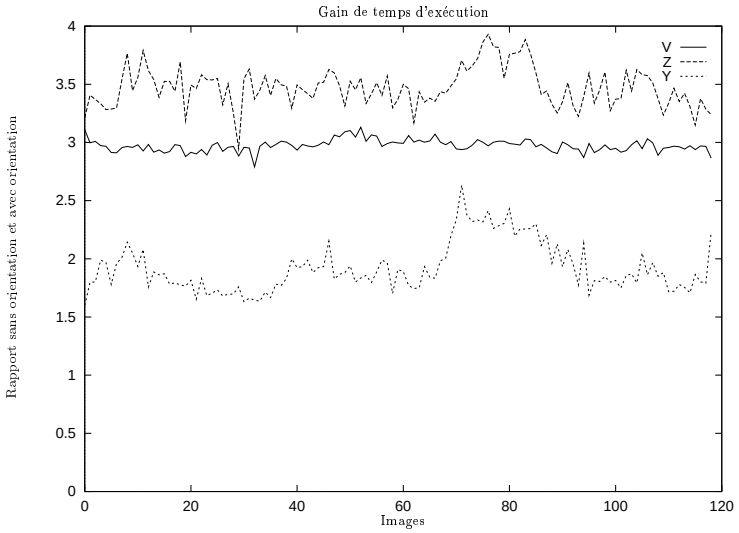


FIG. 4.7: *Gain en temps d'exécution entre la reconnaissance sans orientation et avec orientation.*

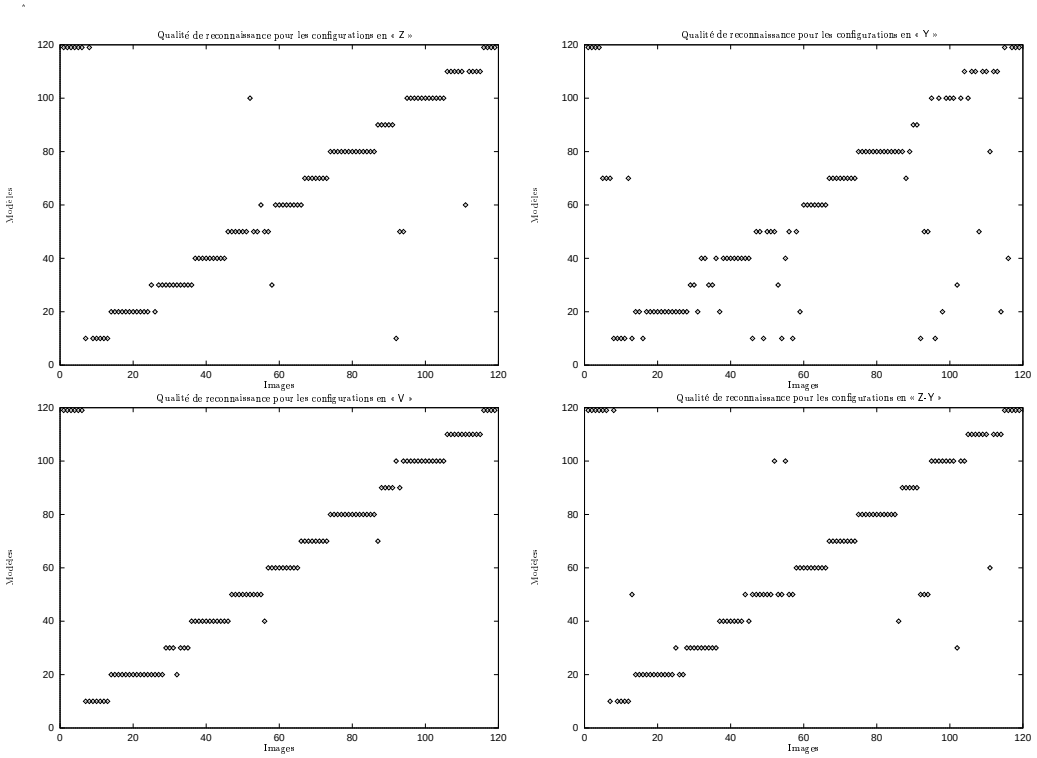


FIG. 4.8: *Influence des configurations à trois sommet « V » et à quatre sommets « Z » et « Y » avec orientation sur la qualité de la reconnaissance. Dans l'ordre, de gauche à droite et de haut en bas, les configurations « Z », « Y », « V » et « Z-Y » combinées.*

Conclusion

Il est clair que l'utilisation de l'orientation a une influence importante sur la reconnaissance. D'une part, elle permet de réduire le temps de reconnaissance d'un facteur 2 à 3, et par la même occasion elle augmente le pouvoir de discrimination de la méthode. Par la suite de l'exposé nous utiliserons implicitement une information sur l'orientation des segments de droites utilisés. L'utilisation de l'orientation élimine la quasi-totalité des erreurs de la modélisation en « V » et diminue sensiblement les erreurs dans les autres cas.

4.2 Vers une généralisation

La méthode présentée dans ce chapitre fonctionne bien sur de petites bases d'objets de quelques dizaines, voire d'une centaine de modèles, avec des images moyennement complexes d'approximativement moins de 800 quasi-invariants par image. Au delà de ces valeurs, les temps d'exécution deviennent exorbitants, bien que la qualité de la reconnaissance n'en souffre pas. Parallèlement, on constate qu'il existe des catégories d'images sur lesquelles l'approche ne fonctionne pas. Ce sont principalement des situations où la segmentation n'a plus de sens car elle ne véhicule plus d'information sémantique (*cf.* FIG. 4.9),

ou parce qu'elle devient trop bruitée (*cf.* FIG. 4.10), bien que la sémantique sous-jacente reste conservée.

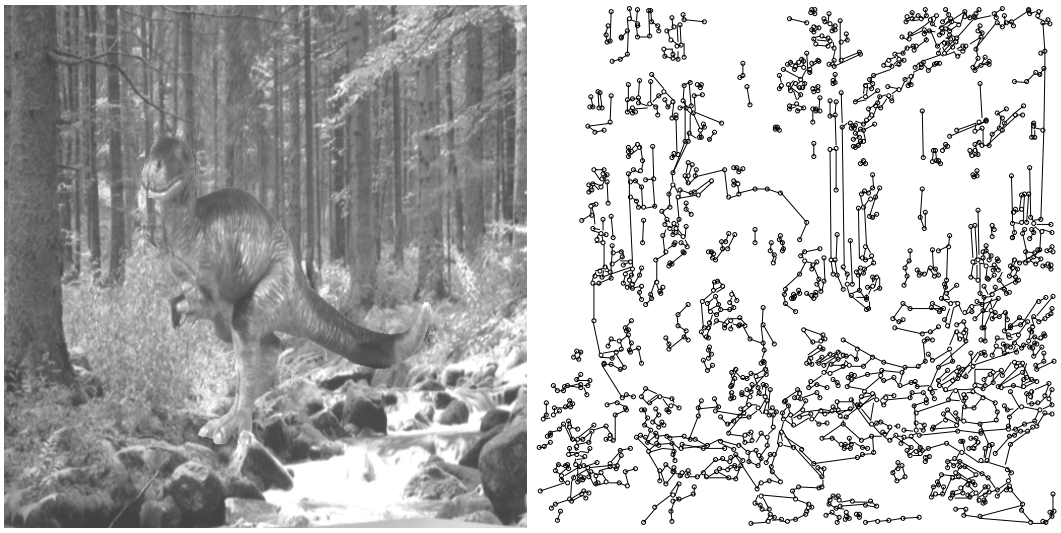


FIG. 4.9: *Exemple d'image où la segmentation perd son information sémantique.*

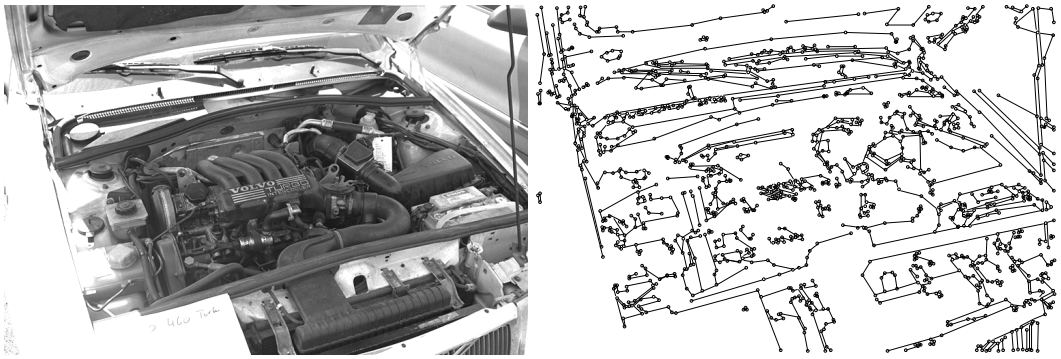


FIG. 4.10: *Exemple d'image où la segmentation devient trop bruitée.*

D'un autre côté, on observe l'émergence de méthodes similaires avec d'excellentes performances [90, 88], mais qui souffrent d'un manque de généralité. Dans les sections qui suivent, nous allons analyser les autres méthodes existantes, et nous étudierons leur intégration avec notre paradigme géométrique pour en déduire un schéma unificateur qui permettra de faire coopérer différentes approches, augmentant ainsi leur champ d'application à d'autres types d'images.

4.2.1 Motivation

Nous avons abordé les forces et faiblesses de notre méthode dans les parties précédentes. Dans le chapitre 2, nous avons également vu que d'autres méthodes existent qui, elles, ne

présentent pas les mêmes avantages ou inconvénients. Il serait donc intéressant de voir comment faire coopérer plusieurs méthodes afin de combler les lacunes des unes par les avantages des autres.

Notre méthode d'*indexation géométrique étendue* se distingue par l'introduction d'une contrainte géométrique globale pour valider les résultats de l'indexation. Par contre, sa faiblesse principale réside dans le fait que les descripteurs locaux manquent de puissance descriptive.

D'une manière générale, les méthodes de modélisation locale représentent des images par des descripteurs (quasi-)invariants. Les descripteurs dépendent du support sur lequel ils sont calculés. L'idée principale que nous allons introduire dans cette partie part de deux constatations.

1° Notre méthode fonctionne très bien sur des images structurées où les segments véhiculent la morphologie implicite de la scène. Elle dépend malheureusement et de la qualité de la segmentation et de l'à-propos des images dans ce contexte. Les autres méthodes se basent principalement sur des informations locales du signal, réduisant la dépendance d'une segmentation complexe, et fournissant par ailleurs d'excellents résultats dans des situations difficiles.

2° Le manque de structuration géométrique des autres méthodes les rend moins bien adaptées à des extensions relatives à la recherche dans des bases par le contenu. Ceci est dû aussi au fait qu'elles dépendent trop fortement du signal des images initiales.

Notre but est de fournir une modélisation regroupant toutes ces méthodes en les faisant coopérer. Ainsi, telle ou telle approche serait prédominante dans son domaine de prédilection, tandis que dans des cas de figure où aucun des systèmes individuels ne donne de performances satisfaisantes, l'union de leurs efforts permettrait de faire émerger un nouveau schéma de reconnaissance.

4.2.2 Intégration d'autres méthodes de reconnaissance

Notre paradigme de cohérence géométrique, fournie par la transformée de HOUGH servira de facteur commun entre les différentes méthodes. Il sera donc nécessaire de faire entrer les autres méthodes dans cette structure. Globalement, l'inclusion d'autres approches consistera en deux étapes, schématisées dans la figure FIG. 4.11, et présentées successivement dans la suite.

1° Permettre le calcul d'une similitude à partir d'une mise en correspondance de deux configurations, et intégrer de la méthode dans le paradigme de HOUGH. En faisant participer les appariements des différentes méthodes dans le même espace de vote, on crée ainsi une accumulation d'indices provenant de différentes méthodes de modélisation. Dans FIG. 4.11, ceci est représenté par les deux colonnes à gauche et à droite, chacune désignant une méthode autonome, mais qui participent au vote dans un même espace (partie basse du schéma).

2° Rechercher des combinaisons géométriques entre le support nouvellement intégré et les supports existants. Ceci a pour but de fournir une description combinée entre les

différentes configurations locales, fournissant des mises en correspondances dans les cas où les méthodes isolées n'en fournissent pas ou peu. Dans FIG. 4.11 cette étape se retrouve dans la colonne du milieu, où l'on crée de nouveaux invariants à partir des configurations utilisées par les autres méthodes. Le processus de reconnaissance qui en résulte produit des votes dans le même espace qu'auparavant.

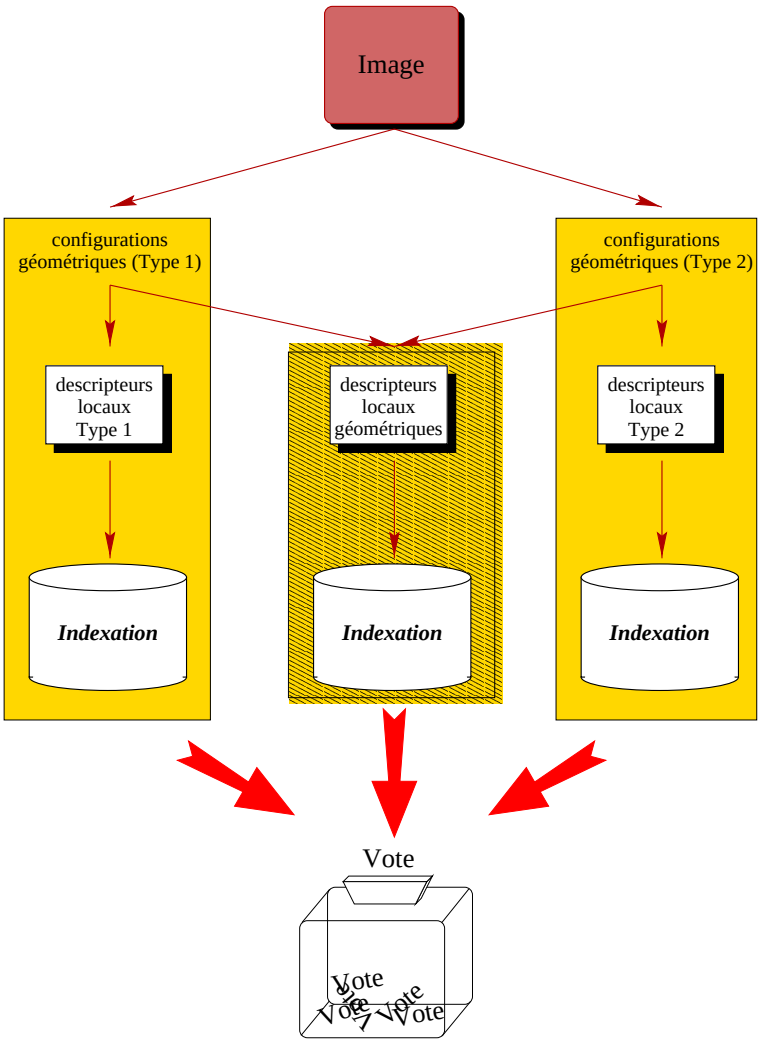


FIG. 4.11: *Principe de coopération entre différentes méthodes locales.*

4.2.2.1 La notion de « configuration »

La notion de configuration permet de calculer une transformation locale entre deux images lorsqu'il existe une mise en correspondance. Afin de pouvoir faire participer différentes méthodes dans un même espace de vote, il est nécessaire de caractériser ces configurations et la façon dont elles définissent ce mouvement local.

Dans l'approche que nous avons présentée en § 3.3, le nombre de degrés de liberté d'une mise en correspondance de configurations était suffisamment élevé pour permettre le calcul d'une similitude locale. Formellement, une similitude Σ est définie par 4 paramètres : $(t_x, t_y, \alpha, \sigma)$; deux pour la translation t_x et t_y, un pour la rotation α et un pour le changement d'échelle σ.

$$\Sigma\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right) = \sigma \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \tag{4.1}$$

Il est donc nécessaire qu'une configuration ait au moins 4 degrés de liberté afin de pouvoir déterminer la transformation locale entre deux configurations. Dans notre cas, par exemple, une configuration consiste en au moins trois points, ce qui donne 6 degrés de liberté.

Dans le contexte de l'intégration avec d'autres approches, une configuration doit être suffisamment riche pour permettre le calcul de cette transformation. Il est clair qu'une modélisation locale faisant intervenir au moins deux points suffit à nos besoins. Malheureusement, des approches existent où les descripteurs sont calculés sur des points dans l'image, fournissant ainsi un nombre insuffisant de degrés de liberté.

Dans ces cas nous proposons de calculer localement des informations non-invariantes qui permettront d'augmenter le nombre de degrés de liberté. Étant donné que les coordonnées du point permettent de récupérer facilement la translation, il reste donc seulement la rotation et le changement d'échelle à évaluer.

L'orientation du gradient introduit un degré de liberté qui est variant par rotation et qui permet facilement de calculer la rotation entre deux images.

Le facteur d'échelle est moins facile à trouver. Nous ne proposons pas de méthode qui soit applicable dans tous les cas, bien qu'il existe des mesures locales qui varient avec le changement d'échelle. Des mesures telles que la courbure ou l'auto-corrélation pourraient permettre de calculer la différence d'échelle entre deux points correspondants. Elles sont malheureusement trop instables du point de vue numérique pour être utilisables dans des cas réels. Dans l'exemple d'intégration que nous traitons, l'échelle est retrouvée grâce à une mise en correspondance multi-échelle. Chaque descripteur comportant une information quant à l'échelle à laquelle il a été calculé, une simple comparaison entre les valeurs mises en correspondance fournit le changement d'échelle entre les deux. Bien que cette approche soit spécifique à la méthode mise en œuvre, on peut néanmoins constater que des méthodes de reconnaissance doivent intégrer, d'une façon ou d'une autre, une tolérance aux variations de l'échelle. En étudiant cette mise en œuvre, il est en général possible de retrouver l'information recherchée.

4.2.2.2 Coopération entre méthodes existantes

Nous disposons donc d'un cadre regroupant toutes les méthodes locales dans un même contexte d'application. Le fait qu'elles puissent coopérer en exprimant leurs votes dans le même espace permet de développer un système qui choisit naturellement la modélisation la plus appropriée. Dans FIG. 4.12 on voit clairement ce qui se passe. Deux méthodes

agissant sur les mêmes données répondent de façon différente : la première identifie bien le mouvement apparent entre l'image et le modèle, tandis que la seconde ne réussit pas vraiment à faire ressortir un franc centre d'accumulation, et l'amas trouvé n'est pas correct. En faisant voter les deux approches dans le même espace on trouve la transformation correcte avec une densité plus élevée, et on est en mesure d'écarter l'hypothèse proposée par la deuxième méthode.

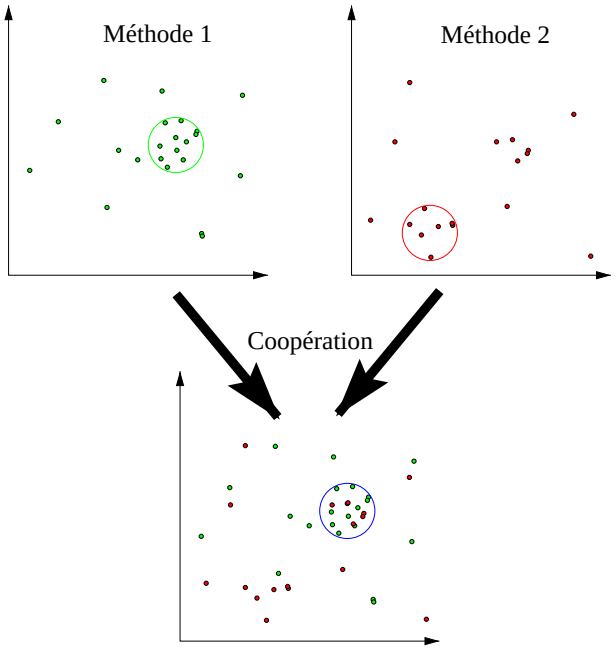


FIG. 4.12: *Coopération de deux méthodes locales par partage de leur espace de vote.*

On peut supposer (et nous le montrons dans la section § 4.2.4.1) que cette façon de procéder donnera encore de bons résultats si les deux méthodes ne réussissent pas à fournir une transformation correcte et lorsque les amas initialement trouvés ne sont pas trop denses. Il nous paraît évident que cette supposition ne suffit pas pour garantir le bon fonctionnement de la méthode dans tous les cas. Afin de proposer un schéma plus solide, nous introduisons de nouveaux descripteurs, basés sur les configurations utilisées dans les deux méthodes considérées.

4.2.2.3 Introduction de nouveaux descripteurs

L'introduction de nouveaux descripteurs a pour but de renforcer la coopération entre méthodes lorsqu'elles ont du mal à trouver un point d'accumulation dans l'espace des transformations. Quand une méthode locale ne parvient plus à remplir sa fonction, ceci ne peut être dû qu'à deux facteurs. Soit les configurations extraites ne sont plus fiables (par manque de précision, ou à cause de défauts de répétabilité, par exemple), soit les descripteurs calculés ne véhiculent plus d'information pertinente, ce qui fait échouer la mise en correspondance.

On peut considérer que toute l'information véhiculée par les descripteurs des méthodes intervenant est captée par l'algorithme de mise en correspondance modifié, décrit en § 4.2.2.2. Il est donc inutile de reconsidérer ces descripteurs ici. De plus, les configurations propres à chaque méthode ont été caractérisées par ces descripteurs et mises en correspondance. La seule information restante qui soit exploitable, est la coexistence de configurations du premier type (appartenant à la première méthode) avec celles du second type (appartenant à la seconde méthode). Nous proposons donc de construire des configurations hybrides, basées sur les configurations existantes. Puisque ces configurations peuvent beaucoup varier d'une méthode à l'autre, nous ne sommes pas en mesure de fournir une formulation générale pour la création de telles configurations. Nous fournirons un exemple détaillé d'un cas concret en § 4.2.3.3.

Il est néanmoins possible de formuler deux contraintes sur les configurations et leurs descripteurs.

1° Les configurations de départ étant individuellement suffisamment riches pour calculer la transformation nécessaire pour exprimer un vote, la fusion des deux possède nécessairement (sauf cas dégénérés) un nombre de degrés de liberté confortable pour l'énumération de quasi-invariants géométriques. Il nous semble donc naturel de continuer d'utiliser ce formalisme comme base pour les nouveaux descripteurs.

2° Si les configurations sont suffisamment riches, il peut être intéressant de réduire le nombre de degrés de liberté. Cette réduction peut augmenter la robustesse du processus si elle est associée aux parties les plus fragiles des configurations.

Ces réflexions étant relativement génériques et vagues, nous les mettrons en œuvre dans § 4.2.3.3 afin d'illustrer leur application.

4.2.3 Exemple de mise en œuvre

Nous avons réalisé l'intégration de notre méthode avec celle proposée par SCHMID, décrite dans sa thèse [90]. Nous rappelons ici brièvement la méthode en question, et nous abordons ensuite point par point les étapes qui ont mené à la coopération totale des deux approches.

4.2.3.1 Description de la méthode intégrée

La méthode de SCHMID s'appuie sur l'extraction de points d'intérêt sur lesquels on calcule des informations basées sur le signal en ces points. L'algorithme 4.1 donne le déroulement général de l'algorithme d'indexation.

Pour la reconnaissance, l'auteur utilise un critère semi-local pour filtrer des appariements initiaux donnés par l'indexation. Elle impose que, dans un voisinage d'un point apparié, 3 des 5 plus proches voisins soient également appariés. Avec cette approche, elle obtient des taux de reconnaissance supérieurs à 99%.

Il est à noter que les invariants de luminance calculés ne sont invariants qu'aux transformations rigides (translation et rotation). Afin d'absorber d'éventuels changements d'échelle, l'auteur a introduit une mise en correspondance multi-échelle. Elle calcule les descripteurs

Algorithme 4.1 Méthode d'indexation de SCHMID

```
Paramètres d'entrée : IMAGE, BASE_DE_MODÈLES
Paramètres de sortie : BASE_DE_MODÈLES avec les descripteurs de IMAGE

début

    LISTE = Liste de points d'intérêt par application du détecteur
            de HARRIS[41];

    pour tous les I, éléments de LISTE faire
    début
        Calculer, en I, les dérivées partielles du signal
           jusqu'à l'ordre 3;

        Combiner ces dérivées pour obtenir le descripteur D_I
        à neuf dimensions suivant :
        (cf. [90], p. 36, pour les notations*)
```

$$D_I = \begin{bmatrix} L \\ L_i L_j \\ L_i L_{ij} L_j \\ L_{ii} \\ L_{ij} L_{ji} \\ \varepsilon_{ij} \left(L_{jkl} L_i L_k L_l - L_{jkk} L_i L_l L_l \right) \\ L_{iij} L_j L_k L_k - L_{ijk} L_i L_j L_k \\ \varepsilon_{ij} L_{jkl} L_i L_k L_l \\ L_{ijk} L_i L_j L_k \end{bmatrix}$$

```
        Utiliser D_I comme index dans BASE_DE_MODÈLES et
              ajouter une référence de I dans la base;

    fin

fin
```

*Les composantes du vecteur D_I ci-dessus sont exprimés en notation d'EINSTEIN, et représentent des sommes de combinaisons de dérivées partielles sur le signal de luminance L de l'image. Par exemple on a :

$$L_{ij} = \sum_i \sum_j \frac{\partial^2 L}{\partial_i \partial_j} = \frac{\partial^2 L}{\partial_x \partial_x} + \frac{\partial^2 L}{\partial_x \partial_y} + \frac{\partial^2 L}{\partial_y \partial_x} + \frac{\partial^2 L}{\partial_y \partial_y}$$

ou encore :

$$L_i L_j = \sum_i \sum_j \frac{\partial L}{\partial_i} \frac{\partial L}{\partial_j} = \frac{\partial L}{\partial_x} \frac{\partial L}{\partial_x} + 2 \frac{\partial L}{\partial_x} \frac{\partial L}{\partial_y} + \frac{\partial L}{\partial_y} \frac{\partial L}{\partial_y}$$

ε_{ij} représente le tenseur canonique anti-symétrique : $\varepsilon_{12} = -\varepsilon_{21} = 1$ et $\varepsilon_{11} = \varepsilon_{22} = 0$.

à plusieurs niveaux de résolution. En modifiant le support de calcul d'un facteur $1,2$ par niveau, elle est capable de traiter des similitudes pour des changements d'échelle allant de $\frac{1}{2}$ à 2.

4.2.3.2 Intégration dans le paradigme de HOUGH

Il s'agit ici d'obtenir les paramètres de translation, de rotation et de changement d'échelle à partir d'une mise en correspondance de points d'intérêt. La translation étant triviale à calculer, nous obtiendrons la rotation par calcul de la différence de l'orientation du gradient dans les points calculés. Le changement d'échelle est donné par l'information fournie par la mise en correspondance multi-échelle décrite par SCHMID. La valeur obtenue est précise à un facteur $1,2$ près. Ce facteur est suffisamment précis pour notre algorithme de vote.

4.2.3.3 Introduction de descripteurs hybrides

Supposons que ni l'approche originale ni la méthode de SCHMID ne réussissent à reconnaître des objets. Cela veut dire que l'approche initiale n'a pas pu extraire des configurations significatives, et que les descripteurs de la seconde étaient sujets à des changements d'illumination ou à des effets 3D trop importants. Afin d'essayer de retrouver au mieux l'information perdue nous proposons deux nouvelles configurations qui nous permettent de calculer des quasi-invariants, dans l'espoir qu'ils véhiculent la description nécessaire pour procéder à une reconnaissance.

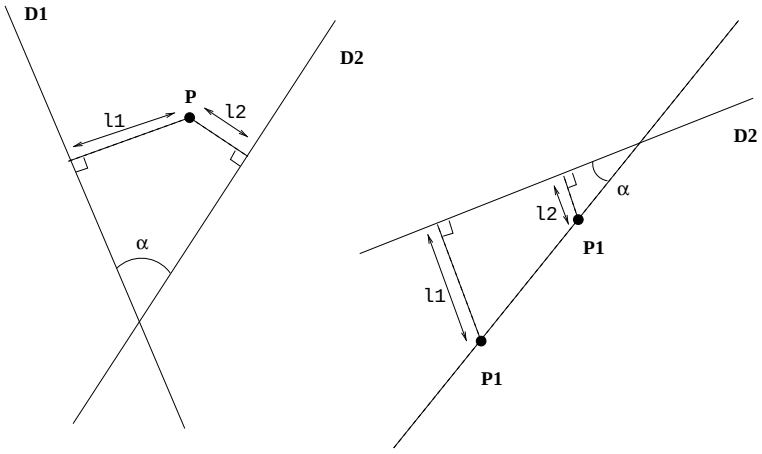


FIG. 4.13: *Configurations hybrides issues de la coopération de notre méthode avec celle de* SCHMID.

En ne gardant que les informations les plus robustes et répétables au niveau de la segmentation dans les configurations des deux approches, nous réduisons au mieux toute influence de bruit parasite. De ce fait, les segments sont « épurés » pour n'en garder que

la droite orientée qu'ils définissent. Les points d'intérêt sont gardés tels quels, sans information adjointe supplémentaire. Les nouvelles configurations sont constituées soit de deux droites obtenues par ce procédé plus un point, soit de deux points et une droite. Elles possèdent chacune 6 degrés de liberté, permettant de calculer 2 invariants indépendants (pour les similitudes). Ces invariants sont l'angle α et le rapport des distances $\frac{l1}{l2}$ représentés dans la figure FIG. 4.13. Dans le cas d'une configuration avec un point et deux droites, l'angle est celui formé par les deux droites dans le secteur où se trouve le point, et $l1$ (resp. $l2$) est la distance du point à la première droite (resp. seconde droite). Dans le cas d'une configuration avec une droite et deux points, α correspond au plus petit angle entre la droite donnée et celle passant par les deux points. $l1$ (resp. $l2$) est la distance du premier point (resp. second point) à la droite donnée.

En ayant choisi ces invariants tels qu'ils soient quasi-invariants projectifs, les configurations s'intègrent naturellement dans notre approche.

Afin de résoudre les problèmes d'explosion combinatoire du nombre de configurations, nous avons défini une notion de *proximité*. Cette proximité a l'avantage d'être indépendante aux changements d'échelle qui pourraient intervenir. En effet nous considérons uniquement les points et les droites issues de segments qui se trouvent dans une zone délimitée par une ellipse, centrée sur l'un des segments initiaux intervenants. Si on prend l'exemple

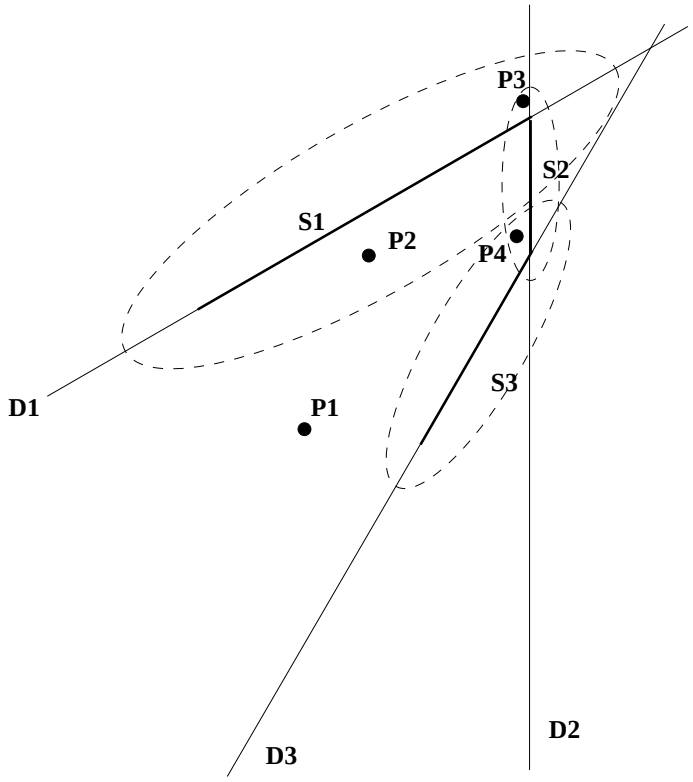


FIG. 4.14: *Schéma de regroupement des primitives pour la formation de configurations hybrides.*

présenté dans la figure FIG. 4.14, on voit alors que seules les 7 configurations $(D1, P2, P3)$, $(D1, D2, P2)$, $(D1, D2, P3)$, $(D1, D2, P4)$, $(D2, P3, P4)$, $(D2, D3, P3)$ $(D2, D3, P4)$ seront utilisées, tandis qu'il existe, au total, 30 combinaisons de points et de droites possibles.

4.2.4 Expériences et résultats

Le but de cette section n'est pas de fournir une comparaison formelle entre les performances de reconnaissance dans le cas d'une modélisation étendue, utilisant la coopération, et une modélisation simple, telle que nous l'avons décrite dans le cas des configurations connexes dans le chapitre précédent et dans la section § 4.1. Une telle comparaison n'aurait pas tellement de sens, dans la mesure où, confrontées à des bases de modèles identiques, ces modélisations pourraient, dans quelques cas, effectivement trouver un modèle correct pour une image inconnue, mais une étude plus approfondie montrerait rapidement que ceci serait basé sur le hasard plutôt que sur une mise en correspondance valable.

Dans ce qui suit, nous montrerons quelques exemples de modèles que notre système est capable de gérer. Dans toutes les expérimentations nous donnerons la base de modèles utilisée, les requêtes correctement traitées, et des requêtes incorrectement traitées. Tous les modèles identifiés par notre méthode le sont sur une base cohérente et géométriquement valable. Nous fournirons, ici et là, quelques exemples de mise en correspondance obtenue après identification.

Concernant l'environnement de test

Dans les deux séries de tests qui vont suivre, nous n'avons pas fourni de modélisation 3D complète de chaque objet stocké. Pire encore, chaque objet, à une exception près, n'est présent qu'une seule fois dans chaque base. De plus, les bases de modèles sont relativement petites (de l'ordre d'une dizaine de modèles). Ces deux restrictions proviennent d'une même limitation forte de notre approche : les ressources physiques nécessaires. Chaque série de test requiert entre 200 et 300Mo de mémoire vive, malgré les soins que nous avons apportés à notre implémentation[2], et nécessite quelques minutes de calcul sur une *UltraSparc30*. Une augmentation trop importante du nombre de modèles de la complexité présentée ici est actuellement donc difficilement concevable.

Dans cette même optique, il n'a pas de sens de parler de « *taux de reconnaissance* » puisque nous ne disposons pas de modèle complet dans notre base. Nous fournirons donc uniquement des exemples de réussite et d'échec.

4.2.4.1 Validation de la collaboration des méthodes

Dans la section § 4.2.2.2, nous avons introduit la notion de coopération entre méthodes pour combiner différentes approches locales. L'extension « Z-Y » en était déjà un exemple, la méthode utilisant des configurations hybrides en est un autre.

2. Nous avons de fortes présomptions qu'il est possible de baisser la taille nécessaire d'au moins un facteur 2 à 5, et peut-être plus. Par contre, une telle réduction irait de pair avec une augmentation du temps d'exécution

En observant la différence en performance de reconnaissance pour les méthodes « Z », « Y » et « Z-Y », exposée dans les figures FIG. 4.8 on constate que, dans l'absolu, la méthode de collaboration se comporte légèrement moins bien que l'une de ses composantes, notamment « Z ». On peut donc se demander si trop de votes bruités n'introduisent pas un biais négatif dans le résultat final. En effet, dans l'expérimentation évoquée, la méthode « Z-Y » introduit 4 fausses réponses nouvelles, par rapport aux 9 initiales de la méthode « Z ».

Au vu des données brutes on peut faire deux constats. Soit la coopération est trop sensible au bruit, et alors le fait d'introduire de nouvelles méthodes pourra être nuisible, soit l'information globale utilisée pour faire la reconnaissance était insuffisante pour faire ressortir le modèle correct. Dans ce dernier cas l'ajout de nouvelles informations devrait pallier ce manque. C'est ce que nous avons fait dans l'expérimentation qui suit. Nous avons fait collaborer les trois types de configuration suivants: « Z », « Y » et la configuration hybride comportant deux segments et un point d'intérêt, à laquelle nous référerons comme « SSP ». Nous avons obtenu les résultats représentés dans la figure FIG. 4.15.

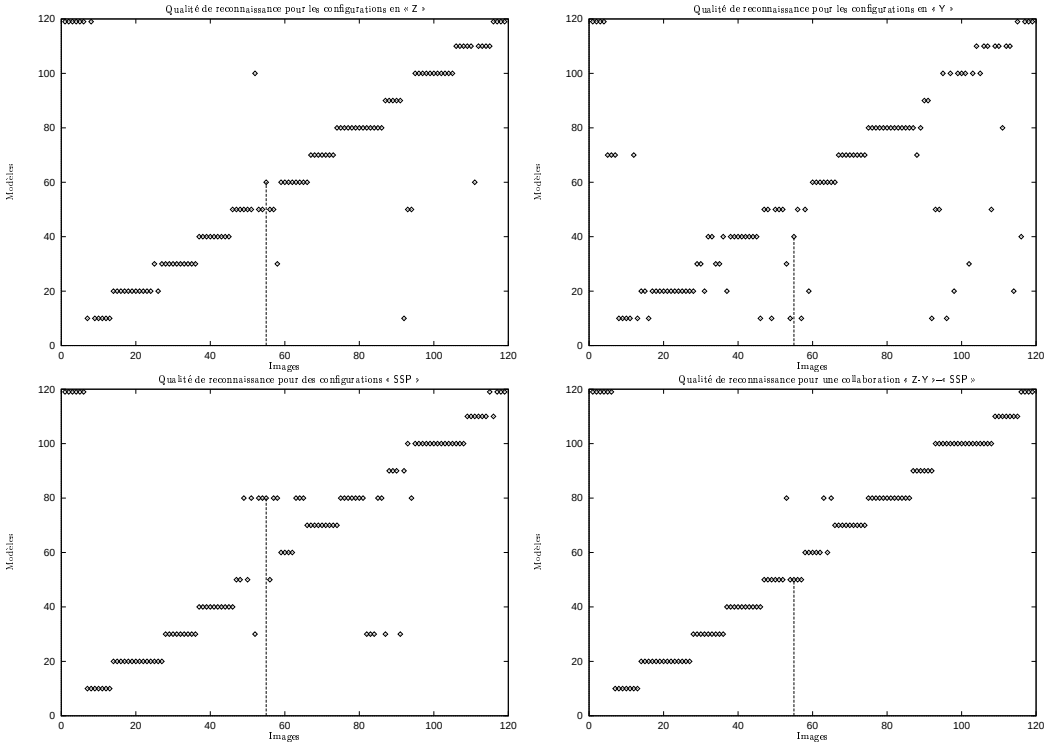


FIG. 4.15: *Collaboration entre « Z », « Y » et « SSP ».*

Le résultat est clairement meilleur que ce que tout ce que nous avons obtenu jusqu'à maintenant, bien qu'en individuel, chaque méthode ait de moins bonnes performances.

Il est intéressant d'observer le comportement des 3 méthodes individuelles dans le cas de l'image 55, par exemple. La méthode en « Z » l'associe au modèle 60, la méthode en « Y » l'associe au modèle 40, tandis que « SPP » la reconnaît comme une instance du

modèle 80. Les trois méthodes fournissent donc un résultat erroné. La collaboration des trois, en revanche, reconnaît bien le modèle 50 comme étant le plus proche de l'image en question.

La collaboration entre méthodes par expression de votes dans un même espace est donc bien réelle. Bien que dans des cas de bruit persistant et géométriquement cohérent, l'approche puisse échouer.

4.2.4.2 Identification de moteurs de voitures

Dans cette expérimentation, la base de modèles consiste en 9 vues de moteurs de voitures différentes. Nous avons présenté à notre système une série d'autres vues des mêmes moteurs, mais prises d'un angle différent.

Les figures FIG. 4.16 et FIG. 4.17 montrent le type de requête que nous sommes en mesure de satisfaire. À droite figurent les images formant la base de modèles. À gauche se trouvent des exemples d'images correctement identifiées par notre système.

On observe que l'on peut s'autoriser des changements de point de vue de 10 à 20° sans nuire à la qualité de reconnaissance. On montre dans la figure FIG. 4.18 le type de mise en correspondance de primitives que l'on obtient alors.

Afin de montrer l'aspect non-aléatoire de la reconnaissance, nous représentons dans le tableau suivant le nombre de votes reçu par les deux meilleurs modèles répondant à chaque requête représentée dans la figure FIG. 4.16. Les requêtes sont numérotées selon le parcours haut-bas, gauche-droite.

Requêtes	Premier choix		Second choix	
requête 1	714	(modèle 1)	568	(80%)
requête 2	896	(modèle 1)	392	(44%)
requête 3	116	(modèle 2)	40	(34%)
requête 4	857	(modèle 2)	477	(56%)
requête 5	670	(modèle 3)	428	(64%)
requête 6	**327**	**(modèle 3)**	**226**	**(69%)**
requête 7	184	(modèle 4)	144	(78%)
requête 8	60	(modèle 5)	52	(87%)
requête 9	**230**	**(modèle 6)**	**159**	**(69%)**
requête 10	**474**	**(modèle 7)**	**265**	**(56%)**
requête 11	106	(modèle 7)	71	(67%)
requête 12	440	(modèle 8)	318	(72%)
requête 13	341	(modèle 8)	273	(80%)

Dans la plupart des cas, le second reçoit 30 à 70% de votes en moins que le premier. Ce qui montre que la sélection du meilleur modèle dans cet exemple est sans ambiguïté. Nous avons, par ailleurs, repris les mises en correspondance obtenues pour le premier et le second choix des requêtes en gras. Ces résultats sont représenté dans les figures FIG. 4.19 à FIG. 4.21, pp. 99–100.

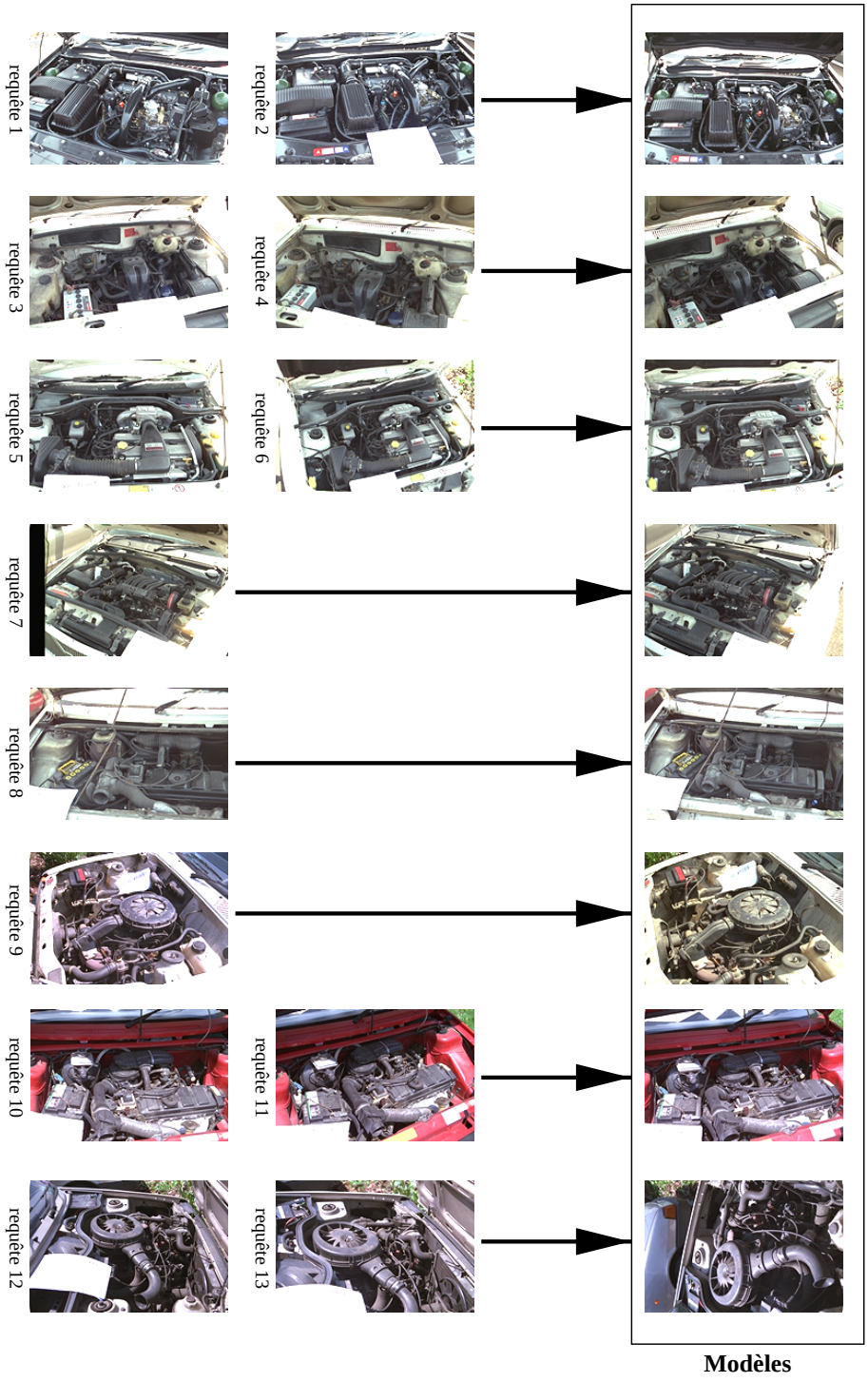


FIG. 4.16: *Exemples de requêtes réussies. Les modèles correspondants sont représentés à droite.*

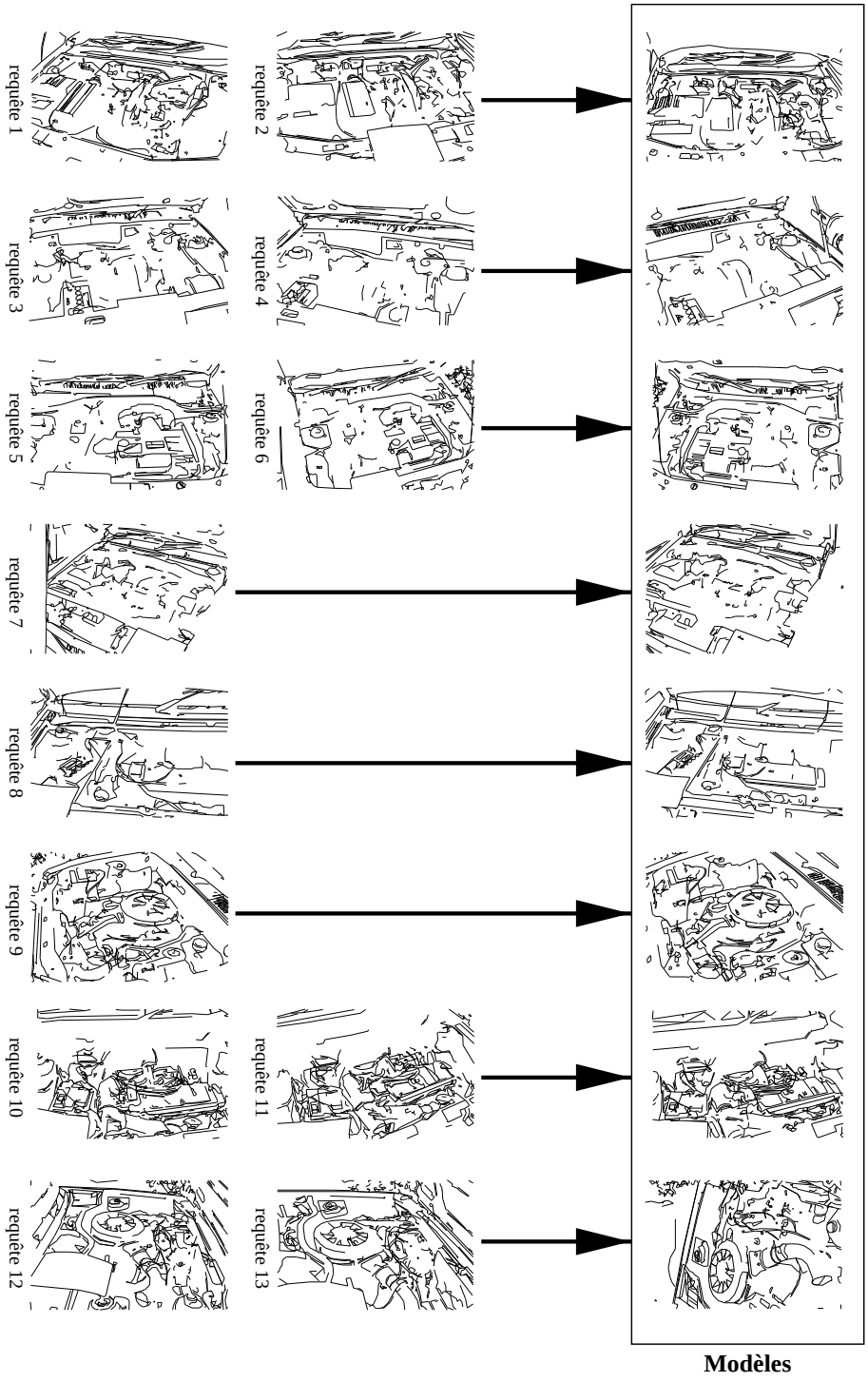


FIG. 4.17: *Exemples de requêtes réussies. Les modèles correspondants sont représentés à droite.*

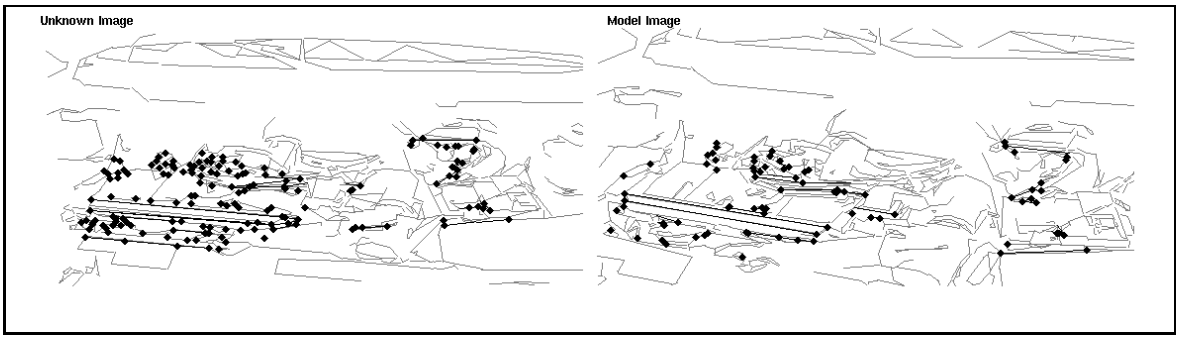


FIG. 4.18: *Mise en correspondance de primitives entre une image et son modèle (moteurs de voiture).*

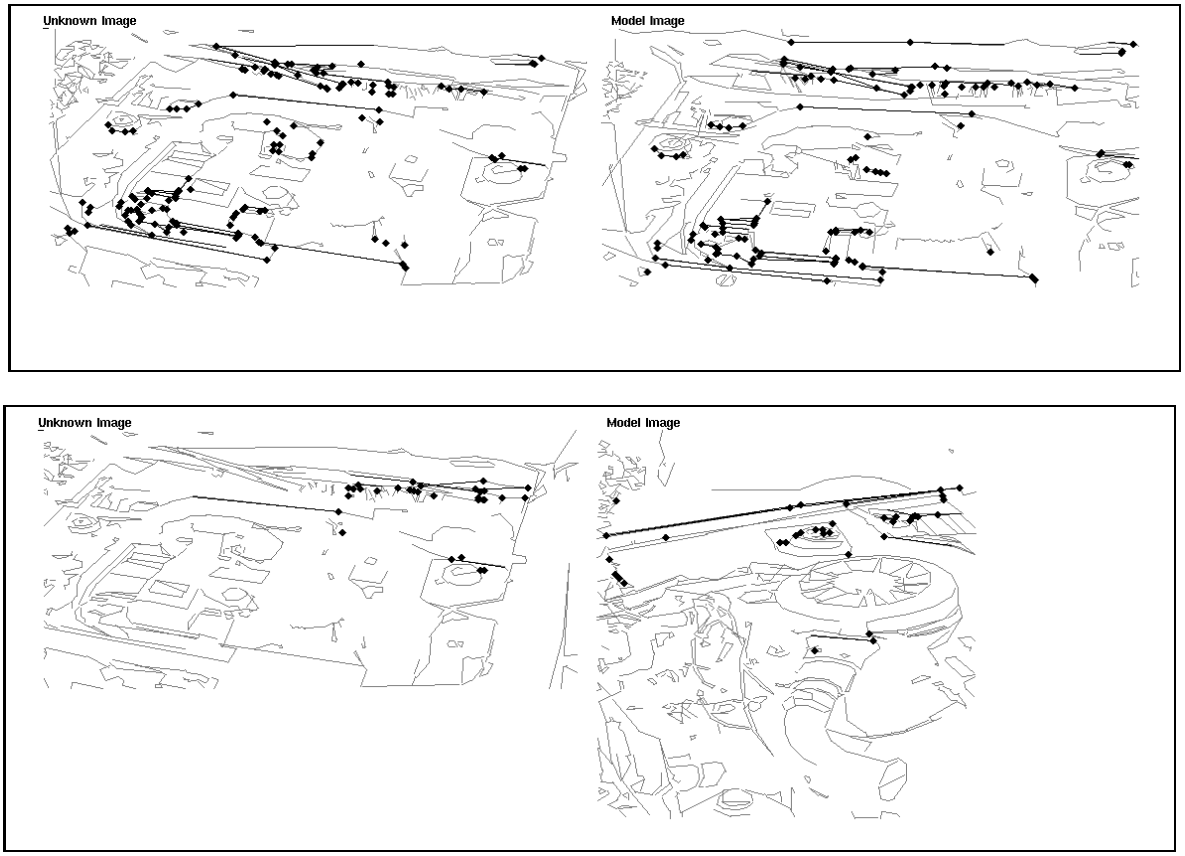


FIG. 4.19: *Requête 6 : mise en correspondance obtenue pour le premier et second choix.*

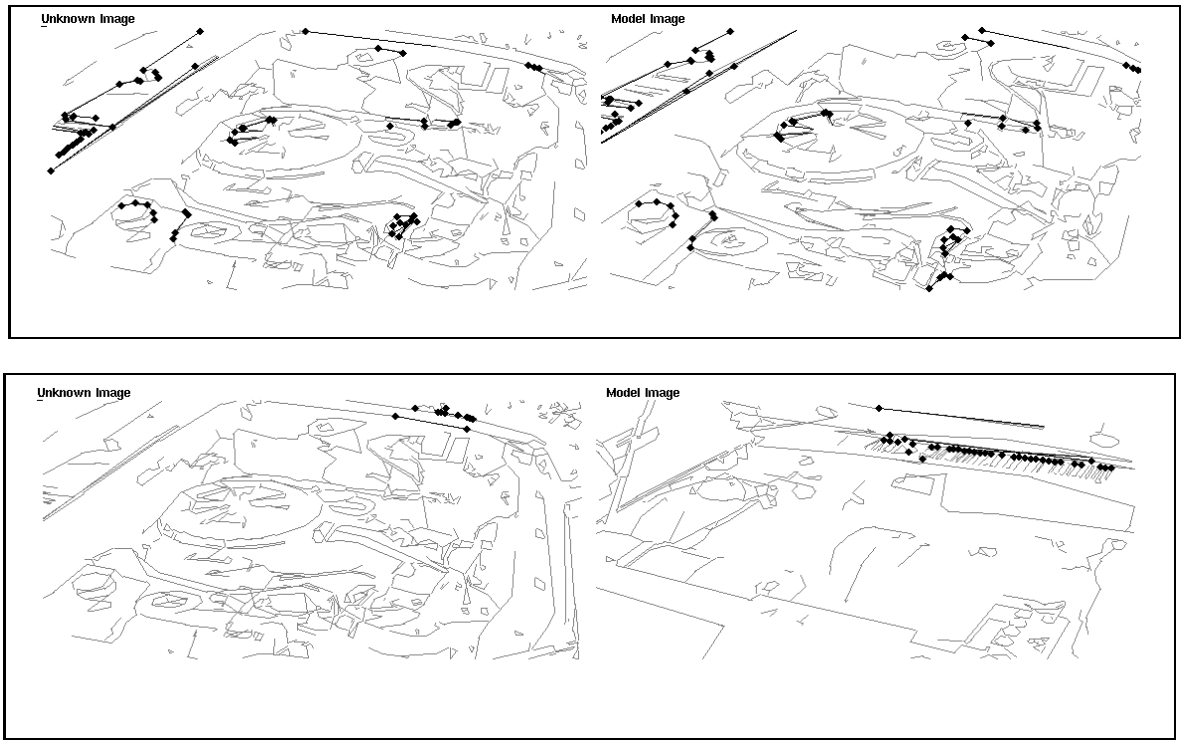


FIG. 4.20: *Requête 9 : mise en correspondance obtenue pour le premier et second choix.*

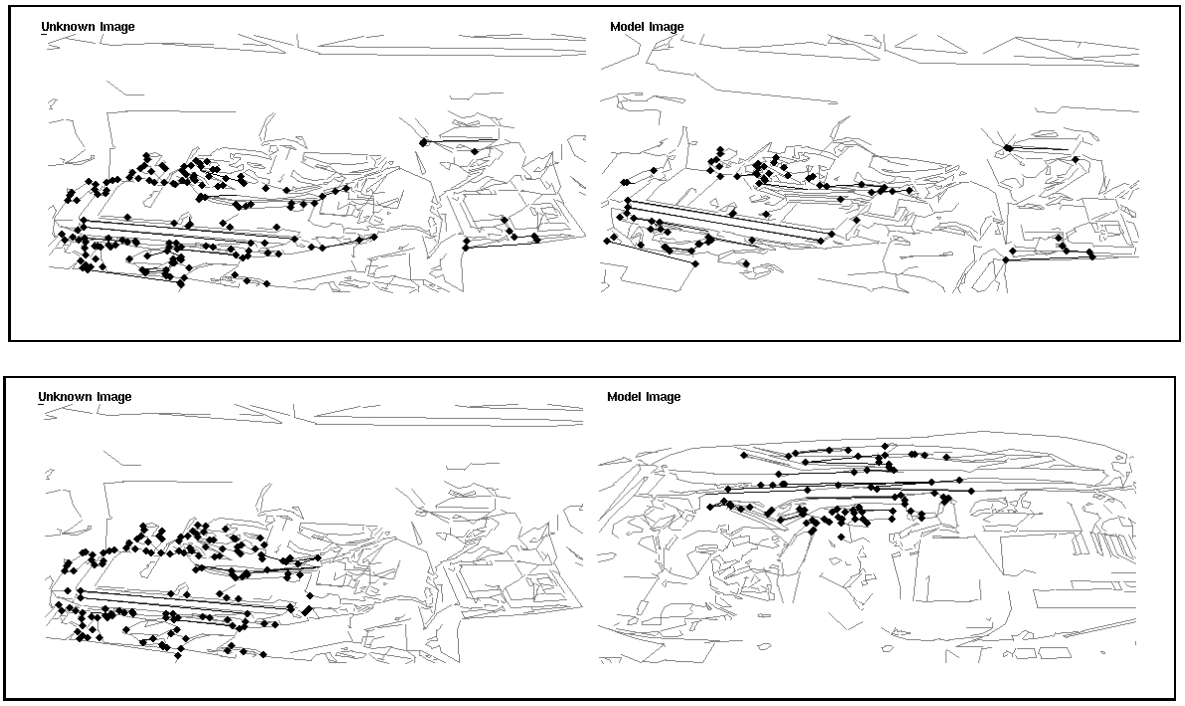


FIG. 4.21: *Requête 10 : mise en correspondance obtenue pour le premier et second choix.*

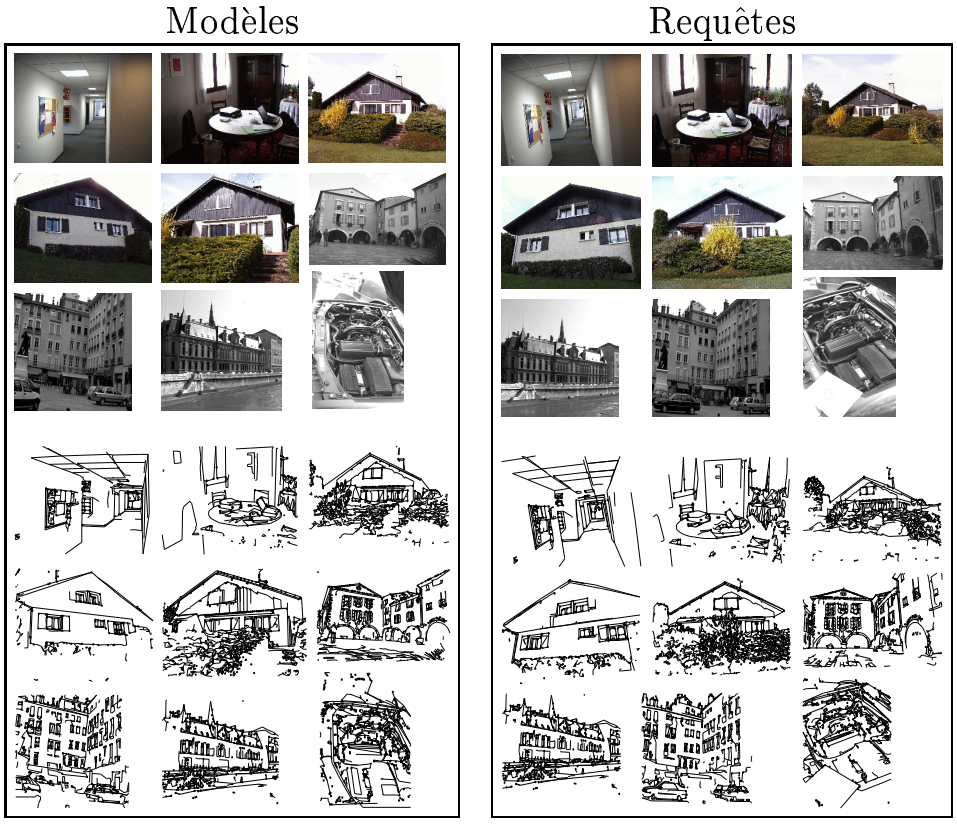


FIG. 4.22: *Base de modèles utilisée, et requêtes correspondantes pour une base hétérogène. Les mêmes figures, agrandies, se trouvent p. 104 et p. 105.*

4.2.4.3 Reconnaissance dans une base hétérogène

Dans cette section nous avons composé une base plus hétérogène (*cf.* FIG. 4.22) avec 9 objets réels complexes.

Un seul objet est présent deux fois dans la base, la maison vue de face, représentée par deux vues, celle en haut à droite, et celle du milieu. Tous les autres ne sont représentés qu'une seule fois. Nous arrivons a distinguer sans problème entre les images et les modèles qui leur correspondent, et nous avons une correspondance 1 à 1 entre les images requêtes, à droite, et leurs modèles correspondants, à gauche. La seule exception est la maison qui est deux fois présente dans la base. Les deux requêtes correspondantes s'identifient à l'une des deux vues-modèles, l'autre vue venant en second choix. Ce qui donne les résultats de mise en correspondance montrés dans la figure FIG. 4.23.

4.3 Conclusion du chapitre

Dans ce chapitre, nous avons introduit une extension générale de notre méthode de vérification globale, afin qu'elle puisse être appliquée à un grand nombre de méthodes

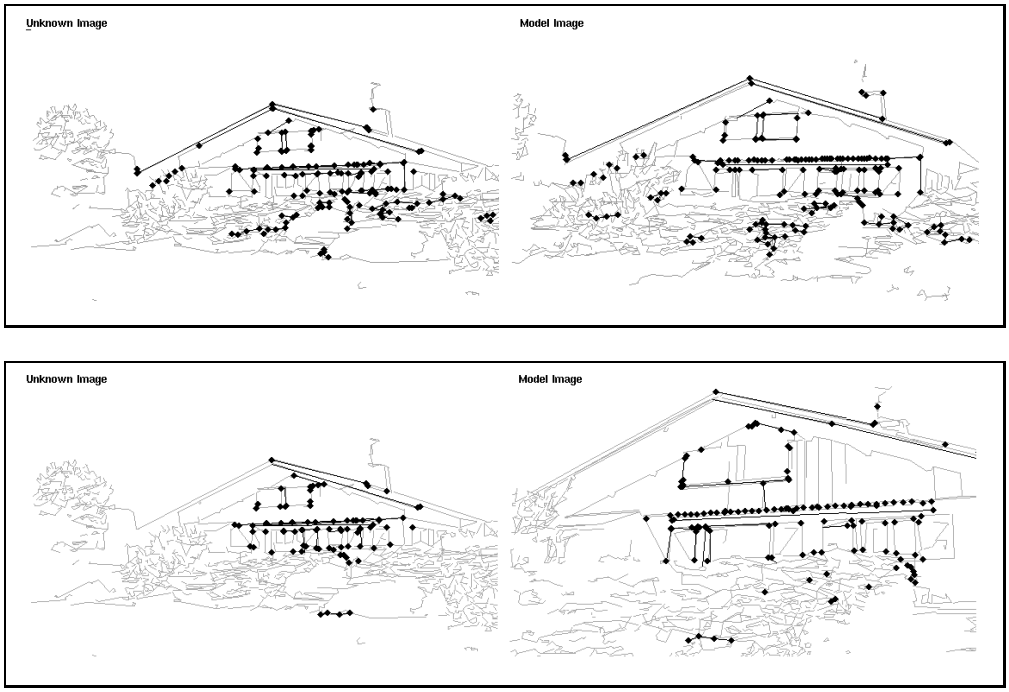


FIG. 4.23: *Mise en correspondance obtenue avec les images de la maison.*

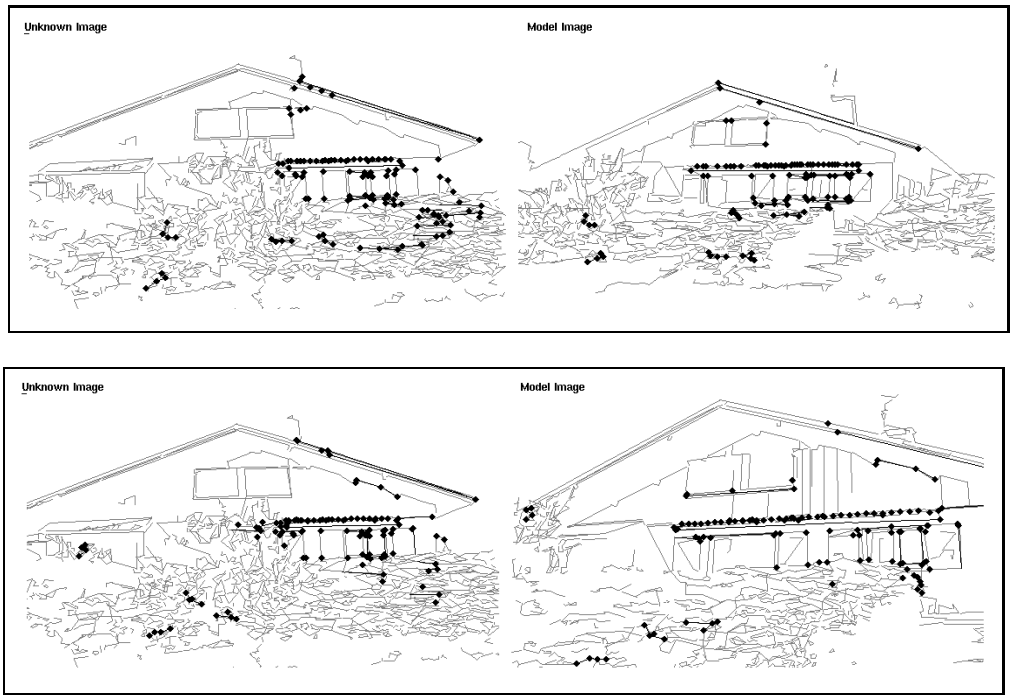


FIG. 4.24: *Mise en correspondance second choix avec les images de la maison.*

de reconnaissance utilisant des caractéristiques locales. Cette intégration de différentes approches permet également de faire coopérer différentes modélisations pour qu'elles élargissent leurs champs d'application respectifs. Nous avons validé cette méthode en montrant que notre programme de reconnaissance était capable, dans des situations complexes, et en absence de tout contrôle d'éclairage, d'étalonnage, etc. de procéder efficacement à des tâches d'identification.

Son principal inconvénient est son incapacité de gérer de très grands volumes de données. Bien qu'il soit raisonnable de penser que la méthode puisse être optimisée pour gérer une, voire quelques centaines de vues-modèles, il est irréaliste d'espérer aller au delà.

Par contre, la méthode peut se révéler très utile dans les domaines où l'on connaît, *a priori*, l'objet que l'on observe, comme dans des problèmes de robotique et d'asservissement visuel, par exemple. Elle constitue aussi une avancée notable, en termes de gain de performance, par rapport à des méthodes de mise en correspondance, telles que celle de Gros par exemple.

FIG. 4.25: *Base de modèles utilisée.*

FIG. 4.26: *Requêtes confrontées à la base de la figure* FIG. *4.25.*

Chapitre 5

Indexation dans des espaces de grande dimension

APRÈS avoir abordé les différentes approches de reconnaissance dans les chapitres précédents, nous avons vu que les résultats les plus prometteurs étaient obtenus par les méthodes se basant sur des caractéristiques locales. Ces caractéristiques, ou descripteurs, sont nombreux par image, et ne présentent, *a priori*, aucun ordre ou structure entre eux. De plus, les méthodes les plus récentes ont tendance à mettre en œuvre des descripteurs de dimension de plus en plus élevée. Se pose donc le problème de l'organisation des ces caractéristiques locales. Cette question est abordée dans ce chapitre.

5.1 Principes

Le principe de la reconnaissance par descripteurs locaux consiste à extraire de l'image inconnue des caractéristiques et à les identifier par rapport aux modèles connus. Il est donc nécessaire de structurer l'information contenue dans les modèles de façon à répondre au problème posé : « *Connaissant un descripteur local, quels sont les modèles connus auxquels il puisse appartenir ?* », tenant compte du fait que le nombre de descripteurs dans une image n'est pas constant, que le nombre de modèles n'est pas figé, et que les descripteurs sont des vecteurs de dimension n.

La structure de stockage doit donc permettre un accès rapide à un grand nombre de modèles, et éventuellement à leurs descripteurs, à partir d'une donnée n-dimensionnelle. Cette donnée est alors par définition un index multidimensionnel dans l'espace de stockage. On désignera par « *indexation* » l'organisation des descripteurs locaux pour en permettre l'accès à partir de ces index. La façon la plus directe de mettre en œuvre une telle

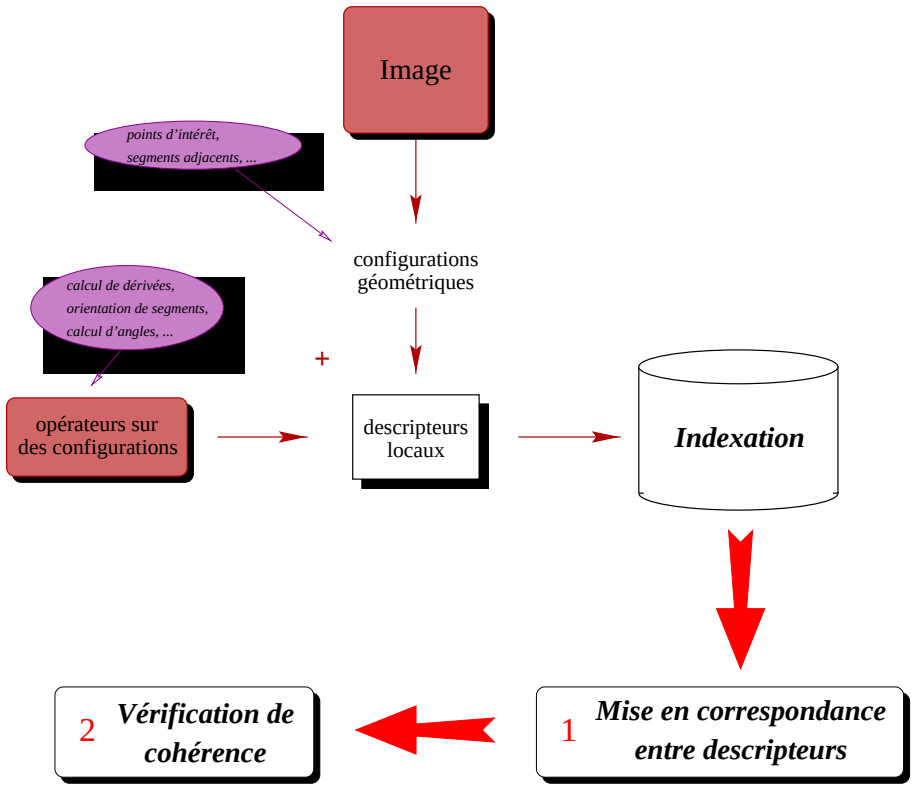


FIG. 5.1: *Principe général de la reconnaissance par indexation.*

structuration revient à créer une table contenant les n-uplets représentant les descripteurs. L'algorithme de reconnaissance n'aurait alors qu'à parcourir la liste, et à appliquer un critère de sélection pour la mise en correspondance entre ces descripteurs et ceux de l'image à reconnaître. Une telle organisation est optimale dans l'utilisation de l'espace, et ne demande pas de réarrangement après insertion de nouvelles données. Celle-ci peut donc s'effectuer de façon incrémentale. Par contre, le coût de la reconnaissance est linéaire en fonction du nombre de modèles dans la base. Pour de grandes quantités de modèles, il peut donc être intéressant de diminuer cette complexité en adaptant la structure d'indexation. Si l'on considère de plus que les descripteurs peuvent être sujets à des erreurs de mesure, la structure devra aussi permettre un accès aux voisins en fonction d'un seuil de tolérance à ces erreurs. On peut prendre la structuration en arbre de recherche décrite en § 3.3.3 comme exemple.

La suite de ce chapitre se décline en trois parties différentes. En premier lieu, nous étudierons les contraintes qu'impose le bruit. Nous regarderons ensuite les différentes techniques d'indexation, et nous finirons par une étude complète de la complexité algorithmique dans le cas d'une méthode de reconnaissance locale utilisant un système d'indexation.

5.2 Modélisation du bruit et indexation

Nous allons étudier dans cette partie l'influence du bruit attaché aux descripteurs sur la mise en correspondance. Nous nous restreindrons aux modèles de bruit gaussien et uniforme, ceux-ci nous paraissant les plus répandus. Nous supposerons également que les descripteurs couvrent l'espace dans lequel ils évoluent de façon uniforme. Des études d'autres types de distribution pour la population des descripteurs existent [67], mais n'ont, à notre connaissance, pas été appliqués à l'indexation de manière concluante avec de larges bases d'images réelles.

5.2.1 Contexte

L'identification des descripteurs est de première importance pour une méthode locale de reconnaissance. Le bruit qui leur est attaché introduit donc une difficulté pour la mise en correspondance. En modélisant le problème de mise en correspondance de façon statistique, on considère que les descripteurs sont des réalisations de variables aléatoires centrées sur la valeur théorique de ceux-ci et répondant à une certaine loi de probabilité. La mise en correspondance d'un descripteur inconnu avec un modèle peut alors être abordée de deux façons.

Variables aléatoires à support infini. La première part du principe que toutes les caractéristiques peuvent être appariées avec un taux de probabilité calculable non nulle. Dans ce cas, les fonctions de distribution des variables aléatoires ont un support infini (*cf.* FIG. 5.2 et FIG. 5.3 où la zone grisée représente intuitivement la probabilité que les deux variables se correspondent). Un schéma d'appariement, donnant le modèle correspondant le plus probablement à l'image inconnue, se base alors sur la probabilité générale de l'évidence accumulée descripteur par descripteur [89]. Cette approche requiert soit de mettre en correspondance tous les descripteurs des modèles avec tous ceux de l'image, soit de recourir à des méthodes sans mise en correspondance explicite.

Dans le premier cas, on considère que chaque descripteur de l'image à reconnaître est une réalisation d'une variable aléatoire à déterminer parmi ceux représentant les descripteurs locaux des modèles indexés. Des outils statistiques permettent d'attribuer une probabilité d'appartenance à chaque variable aléatoire. Ces résultats sont alors répercutés pour donner une probabilité générale d'appartenance à un modèle. Cette méthode a une complexité linéaire en fonction du nombre de descripteurs indexés. Dans le second cas, ce sont plutôt les modèles qui sont considérés comme des réalisations de variables aléatoires. La complexité de la reconnaissance devient alors linéaire dans le nombre de modèles, au détriment de la perte de l'information donnée par la mise en correspondance explicite entre descripteurs [88].

Variables aléatoires à support fini. Dans le second cas, on considère qu'à partir d'une certaine distance, la probabilité de correspondance entre deux descripteurs est nulle. Ceci revient donc à dire que les variables aléatoires entrant en jeu ont une densité de

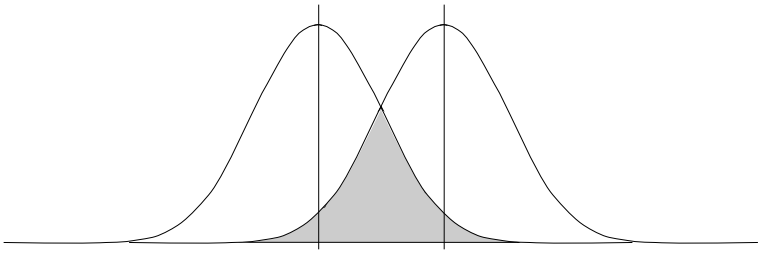


FIG. 5.2: *Deux variables aléatoires à support infini « proches ».*

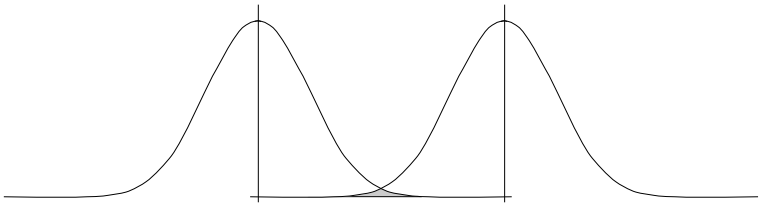


FIG. 5.3: *Deux variables aléatoires à support infini « éloignées ».*

probabilité à support fini (*cf.* FIG. 5.4 et FIG. 5.5 où la zone grisée représente intuitivement la probabilité que les deux variables se correspondent).

Dans le cas où la densité de probabilité ne se prête pas à cette approche, l'idée consiste à introduire un seuil de confiance en dessous duquel on rejette les mises en correspondance [90]. Pour le reste, elle est similaire à la précédente, à la différence près que le seuil introduit permet de réduire efficacement l'espace de recherche, donnant lieu à des organisations de données permettant d'être sub-linéaire en fonction du nombre de modèles. On peut considérer que dans ce cas, et du point de vue de l'indexation, l'utilisation du seuil réduit le problème à celui d'une probabilité uniforme sur un support dont la forme dépend de la fonction de départ. En effet, dans le cas d'un support fini, il est nécessaire d'accéder à tous les descripteurs à l'intérieur de la zone couverte par le support pour évaluer la mise en correspondance. Suivant la modélisation du bruit utilisée, ces appariements peuvent alors être pondérés, mais cette étape ne relève plus de l'indexation.

Dans cette section nous analyserons donc le problème suivant ; nous disposons d'une structure d'indexation, que nous appellerons base d'indexation, base des modèles ou encore base des descripteurs, dans laquelle sont stockés un certain nombre de descripteurs locaux $\mathcal{D}_i$. Chacun des descripteurs $\mathcal{D}_i$ appartient à un modèle $\mathcal{M}_k$. Notre objectif est de répondre à la question : « *Compte tenu que l'observation d'un descripteur inconnu x est sujet à du bruit, quels sont les descripteurs connus $\mathcal{D}_i$ qui peuvent lui correspondre ?* » La question peut éventuellement être reformulée pour prendre en compte une probabilité d'appartenance ou un classement des différents descripteurs sans fondamentalement changer notre approche. Nous ne considérerons que des modélisations à support fini, les autres défiant l'utilité d'une indexation basée sur les descripteurs, et nous décrirons une méthode permettant de contourner le support infini dans le cas d'un bruit gaussien.

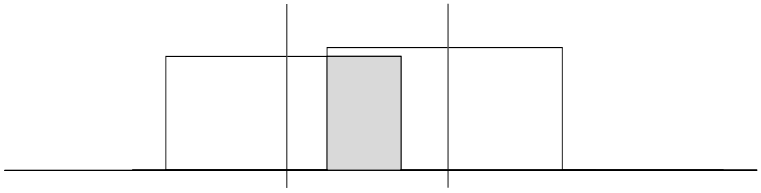


FIG. 5.4: *Deux variables aléatoires à support fini « proches ».*

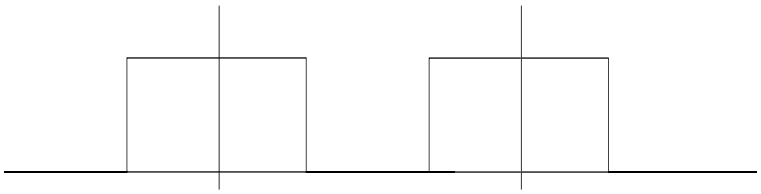


FIG. 5.5: *Deux variables aléatoires à support fini « éloignées ».*

5.2.2 Bruit gaussien

Nous supposons dans cette section que les descripteurs $\mathcal{D}_i$ sont liés à une fonction de distribution gaussienne, et que leur différentes composantes peuvent être corrélées entre elles. Dans le cas d'une comparaison seuillée, les descripteurs locaux devront alors être

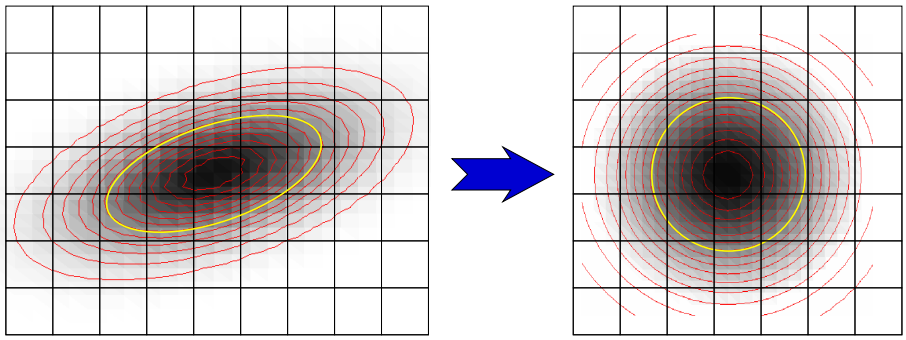


FIG. 5.6: *Distribution d'une variable aléatoire gaussienne à deux dimensions corrélées, superposée sur une grille d'indexation, suivie d'une décorrélation.*

comparés entre eux à une ellipsoïde n-dimensionnelle près, dont la taille varie avec la précision de comparaison souhaitée. La forme de l'ellipsoïde est déterminée par la matrice de covariance entre les différentes composantes du descripteur. Du point de vue de l'indexation, ceci revient à ne plus considérer l'emplacement déterminé par le descripteur dans l'espace d'indexation, mais à prendre aussi en compte les voisins de celui-ci dans le périmètre défini par l'ellipsoïde.

En termes probabilistes, la densité de probabilité de chaque descripteur dans la base

$\mathcal{D}_i$, suivant la loi $\mathcal{N}(\mu_i, \Sigma_i)$, s'exprime comme suit :

$$\mathrm{f}_i(x) = \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma_i|^{\frac{1}{2}}} e^{\left(-\frac{1}{2}(x-\mu_i)^T \Sigma_i^{-1}(x-\mu_i)\right)}$$

Cette expression donne uniquement une information relative à la probabilité qu'un événement se produise. Ou, dans notre terminologie, elle exprime la probabilité d'observer x parmi les instances de $\mathcal{D}_i$. Cette donnée, en tant que telle, ne nous intéresse guère, et elle nécessite des outils statistiques afin de répondre à la question qui nous préoccupe : « *Quelle est la probabilité que x soit une occurrence de $\mathcal{D}_i$?* »

Si l'on suppose que les descripteurs ont tous la même matrice de covariance Σ, la distance de MAHALANOBIS δ permet de réduire l'expression précédente à un calcul de distance entre les espérances μ_i et x :

$$\delta(x, \mu_i) = \sqrt{(x-\mu_i)^T \Sigma^{-1} (x-\mu_i)}$$

On sait que dans le cas où Σ est proportionnelle à la matrice identité, on peut remplacer δ par la distance euclidienne [83]. Étant donné que la matrice de covariance Σ est, par définition, définie positive, et par conséquent diagonalisable, il existe nécessairement un changement de base qui permet de réduire la distance de MAHALANOBIS à une simple distance euclidienne dans tous les cas où les descripteurs partagent la même matrice de covariance [90]. Ce changement de base est équivalent à une décorrélation des composantes des descripteurs (*cf.* FIG. 5.6). Nous supposerons donc dans la suite que cette décorrélation est faite.

Par ailleurs, le carré de la distance δ suit une loi de χ^2 ce qui nous permet d'exprimer la probabilité de l'appartenance d'une observation inconnue à chacune des classes i définies par les $\mathcal{D}_i$. À partir de cette constatation, les méthodes ébauchées dans la section précédente peuvent être mises en œuvre.

La distance $\delta(x, \mu_i)$ exprime la probabilité que le descripteur x corresponde à $\mathcal{D}_i$. L'information qu'elle véhicule suffit donc à calculer la probabilité d'appartenance à un modèle pour un descripteur (p. ex. en prenant $P(x \in \mathcal{M}_k) = \max_{\mathcal{D}_i \in \mathcal{M}_k} P(x = \mathcal{D}_i)$), et, *a posteriori*, à calculer le taux de ressemblance entre une image et un modèle, en intégrant ces probabilités pour chaque descripteur de l'image.

5.2.3 Approximation du support

Dans le cas d'une modélisation de bruit uniforme (ou de bruit à support fini en général) les descripteurs ne sont retenus pour mise en correspondance que s'ils sont suffisamment proches. Le principal problème relatif à l'indexation vient alors de la quantification du support à considérer pour la notion de « proche ». En général, il existe une expression analytique de ce support, mais cette expression n'est pas toujours exploitable dans le cas d'espaces d'indexation discrétisées comme nous verrons par la suite.

Un cas courant est la distribution gaussienne avec une distance de MAHALANOBIS dans laquelle on rejette toute mise en correspondance entre descripteurs dont la distance est supérieure à un seuil fixé. Dans ce cas, le support devient une ellipsoïde en n dimensions (ou

une boule dans le cas décorrélé). Dans le cadre d'une indexation, on approche souvent cette boule par le plus petit hypercube englobant, éventuellement complété par des vérifications point par point à proximité des frontières. Un problème particulier survient dans ce cas [1].

En effet, soit $\mathcal{B}$ l'hyper-volume d'une boule de rayon r en n dimensions :

$$\mathcal{B} = \frac{r^n.\pi^{\left(\frac{n}{2}\right)}}{\left(\frac{n}{2}!\right)}$$

En comparant $\mathcal{B}$ au volume de l'hypercube l'englobant $\mathcal{C} = (2r)^n$ on constate que :

$$\lim_{n\to\infty} \frac{\mathcal{B}}{\mathcal{C}} = 0$$

Ce qui revient à dire que l'erreur commise en approchant l'hyper-sphère par un hypercube est d'autant plus grande que n est grand, et tend vers l'infini quand n tend vers l'infini.

Les problèmes soulevés par ce constat sont multiples.

1° Les méthodes d'indexation qui l'utilisent doivent prendre en compte le fait que l'analyse de bruit est mise en défaut par la prise en compte d'un voisinage trop grand. Les résultats en annexe (FIG. B.1) montrent que cette approximation devient vite très mauvaise.

2° Les problèmes de dissymétrie qui sont introduits par la discrétisation sont accentués lorsque n est grand.

3° Le nombre d'accès à la base d'indexation augmente de façon sensible sans apport particulier de performance.

Les questions qui sont soulevées ici n'admettent pas de réponse absolue, valable dans toutes les situations. L'un des facteurs principaux entrant en jeu comme élément de réponse est le compromis classique entre occupation de la mémoire et le temps de calcul. HOWELL et FLYNN proposent une solution élégante dans [48], mais elle consiste à démultiplier le nombre d'index stockés, solution qui n'est pas envisageable dans de très grandes bases d'images. À défaut de réponse générale, nous nous contentons simplement de soulever le problème, et de mettre en garde contre une simplification trop hâtive.

5.3 Vitesse d'indexation et de consultation, espace mémoire

Nous avons énoncé ci-dessus des problèmes liés à l'occupation de l'espace mémoire et à l'accès aux données. Puisque les méthodes actuelles se basent sur des descripteurs de dimension de plus en plus élevée, et que les ambitions de reconnaissance vis-à-vis des capacités de stockage deviennent de plus en plus importantes [74, 76, 90, 88], l'organisation des index en mémoire primaire devient illusoire, et il devient incontournable de considérer des systèmes de stockage secondaires.

Il n'est pas dans le but de cette thèse de faire une étude approfondie des différentes méthodes de stockage (*cf.* remarque p. 59). Nous nous bornerons uniquement à soulever

1. Le développement des calculs est explicité en annexe B, p. 153.

les points principaux qui font déjà partie d'études dans le domaine des bases de données, ouvrant par la même occasion des perspectives de poursuite dans cette voie. D'autres travaux, notamment ceux utilisant une *Sparse Distributed Memory* [80], existent, mais bien que répondant aux préoccupations qui nous intéressent, ils ne semblent pas suffisamment flexibles pour permettre une gestion dynamique de l'espace de stockage.

Les tables d'indexation multidimensionnelles ont fait l'objet de différentes implémentations et améliorations dans le domaine des bases de données [84, 101, 38] depuis des années. Seulement depuis l'émergence des méthodes de reconnaissance d'objets à l'aide d'histogrammes [96] ou utilisant des décompositions par ACP [100], on s'est rendu compte de la nécessité d'indexation dans des espace de haute dimensionnalité. Les récents travaux de BERCHTOLD *et al.* et de BLOTT et WEBER montrent que les méthodes traditionnellement utilisées dégénèrent rapidement quand la dimension des descripteurs atteint des valeurs voisines de 16 [9, 15][2]. Ils montrent notamment que, dans ces cas, les méthodes arborescentes généralement utilisées deviennent plus pénalisantes qu'une simple comparaison séquentielle de tous les descripteurs stockés, et que les critères d'occupation uniforme de l'espace d'indexation, préconisés par la plupart des méthodes, ne sont pas utiles. De ce fait, ils montrent qu'il est plus important de faire un maillage uniforme de l'espace (approche par partitionnement de l'espace), plutôt que d'adapter le maillage à la population d'index pour que dans chaque panier il y ait un nombre équivalent d'index (approche par partitionnement des données). Ceci est principalement dû au fait que, pour de très grandes dimensions, les index sont tous équidistants en moyenne, quelle que soit leur position dans l'espace.

Il est néanmoins nécessaire de souligner que la problématique posée par les auteurs cités ci-dessus est quelque peu différente de celle abordée dans un contexte d'indexation par descripteurs locaux. En effet, toutes les analyses ont été faites dans une optique de recherche des k plus proches voisins, tandis que nos préoccupations sont plutôt la recherche de *tous* les voisins dans une n-boule autour d'un point, ce qui simplifie grandement la complexité du problème. À notre connaissance, aucune étude n'a été faite avec ce point de vue particulier. Une étude plus approfondie du domaine devra montrer s'il existe des méthodes de stockage plus adaptées et dira si les résultats énoncés plus haut se confirment ou s'infirment dans ces cas-ci.

Des travaux récents de NENE et NAYAR [72] montrent que des méthodes de recherche par reprojection sur les axes de l'espace d'indexation donnent des résultats très satisfaisants. L'avancement actuel de ces travaux ne permet pas de savoir si la méthode se révèle plus performante que celles proposées par BERCHTOLD *et al.* et par BLOTT et WEBER [9, 15]. Actuellement aucune comparaison n'est faite, et les méthodes n'ont pas été validées pour des dimensions au-delà de 25.

Pour les applications que nous avons évoquées dans les chapitres précédents, les dimensions restent toutefois modestes : de 2 à 6 pour les différentes configurations géométriques, et 9 pour l'indexation photométrique. Dans ces cas-là, nous utilisons la structure d'arbre de découpe de l'espace, telle qu'elle est décrite dans l'annexe A.

2. Cette dégénérescence est souvent appelée *dimensionnal curse* ou « *dérive dimensionnelle* ».

5.4 Étude et analyse de la complexité algorithmique

Dans cette partie nous développons l'étude de la complexité algorithmique associée au processus de reconnaissance par descripteurs locaux indexés. C'est une extension majeure du travail présenté dans [61].

Nous nous plaçons dans le cas où les descripteurs sont indexés dans une structure multidimensionnelle. On suppose que les descripteurs sont suffisamment génériques pour que l'appariement descripteur-modèle ne soit pas informatif en soi. Typiquement, un modèle possède de multiples descripteurs, et un descripteur peut appartenir à plusieurs modèles. De ce fait, la reconnaissance devient un processus à deux étapes (*cf.* FIG. 5.1, p. 108) : mise en correspondance de descripteurs, suivie d'une vérification de la cohérence globale de celle-ci, qui peut être un simple vote majoritaire [58], une vérification de cohérence locale [90] ou une opération plus globale [60] comme la transformée de HOUGH utilisée dans le chapitre 3.

Nous aborderons deux cas d'étude dans cette section. Dans le premier cas, nous supposons que les descripteurs ne sont pas sujets à du bruit (ce qui peut être le cas pour des données discrètes ou qualitatives plutôt que des valeurs continues). Il n'est, dans ce cas, pas nécessaire de chercher dans des cases voisines pour trouver des appariements. Un accès par descripteur suffit. Dans le deuxième cas, les descripteurs peuvent être bruités. Il est alors nécessaire de prendre en compte un voisinage de recherche dans l'espace d'indexation.

5.4.1 L'espace d'indexation

Considérons un espace d'indexation contenant les descripteurs locaux de M modèles. Nous prenons comme hypothèse simplificatrice qu'en moyenne un modèle possède $\tilde{D}$ descripteurs. Étant donné que le nombre de descripteurs est en général peu variable pour un type d'image, cela revient à supposer que les modèles indexés appartiennent tous à une même classe d'objets. Nous considérons donc que le nombre de descripteurs D d'un modèle vérifie $D \simeq \tilde{D}$, et que le nombre total de descripteurs dans l'espace d'indexation est $\tilde{D}.M$.

Soit τ maintenant la taille de l'espace, représentant le nombre d'emplacements distincts, dans lesquels peuvent être stockés les descripteurs. Dans ce cas de figure, le nombre moyen de descripteurs par emplacement de stockage est :

$$\bar{I} = \frac{\tilde{D}.M}{\tau} \tag{5.1}$$

Pour que l'indexation donne des performances optimales, la distribution théorique des index dans l'espace d'indexation doit être uniforme (ceci ne veut pas dire qu'ils doivent effectivement être dispersés de façon uniforme pour un ensemble de modèles donné). L'uniformité des index permet de faire des recherches dans des voisinages équivalents, indépendamment de l'endroit où se trouve le voisinage dans cet espace. Cette contrainte permet notamment de comparer facilement des rapports de distance décrits dans le chapitre 3,

p. 59. Pour des descripteurs n-dimensionnels, ceci implique que pour chaque dimension, l'index doive suivre une distribution uniforme[3].

De façon générale, l'espace d'indexation peut être considéré comme un tableau multidimensionnel de stockage, et l'on appellera k_i le nombre d'emplacements d'indexation selon la dimension $1 \leq i \leq n$. On peut alors réexprimer τ de l'équation (5.1) en fonction des k_i et on obtient la formule suivante :

$$\bar{I} = \frac{\tilde{D}.M}{\prod_{i=1}^{n} k_i} \tag{5.2}$$

qui détermine le nombre moyen de descripteurs par emplacement d'indexation.

Dans la suite de cette étude nous supposons que le calcul d'une clef d'indexation a un coût négligeable par rapport aux autres calculs qui interviennent dans l'estimation du modèle le plus semblable.

5.4.2 Mise en correspondance exacte

Afin de procéder à l'étape de reconnaissance, on confronte une image à l'ensemble des modèles indexés. Nous considérerons dans la suite que cette image possède également $\tilde{D}$ descripteurs locaux.

Dans un premier temps, nous allons supposer que la mise en correspondance s'effectue de façon exacte, sans bruit sur les descripteurs qui entrent en considération. La partie de l'algorithme qui consiste à récupérer les appariements plausibles entre l'image inconnue et les modèles se schématise alors comme montré dans l'algorithme 5.1.

La complexité de cet algorithme se mesure par la taille de la liste retournée. D'après les hypothèses, et l'équation (5.2) la boucle *(a)* est parcourue $\tilde{D}$ fois. À l'intérieur de celle-ci, la boucle *(b)* l'est en moyenne $\bar{I}$ fois par passage. Il en résulte que la taille de la liste retournée est en moyenne de $\tilde{D}.\bar{I}$ éléments, ou encore :

$$\frac{\tilde{D}^2.M}{\prod_{i=1}^{n} k_i} \tag{5.3}$$

5.4.3 Mise en correspondance bruitée

Nous considérons maintenant le cas plus général où les descripteurs sont sujets à des erreurs de mesure ou à d'autres formes de bruit. Nous supposerons que ce bruit est uniforme. Il n'y a, dans le cas général, aucune raison de faire cette supposition, et elle est d'ailleurs fausse. Nous la faisons uniquement à des fins d'illustration. Comme nous l'avons déjà indiqué partiellement dans § 5.2 et comme nous le montrerons plus loin, elle ne nuit en

3. Nous commettons ici volontairement un abus de langage afin de ne pas nuire à la clarté de l'exposé et du raisonnement. En effet, il n'y a nul besoin d'imposer des contraintes sur les descripteurs, ce qui importe est que leurs clefs d'indexation, elles, suivent une loi uniforme. Dans la partie relative à l'étude de la complexité, nous confondrons la notion de descripteur local et celle de son index de stockage.

Algorithme 5.1 Indexation avec mise en correspondance exacte

```
Paramètres d'entrée : IMAGE, ESPACE_INDEXATION
Paramètres de sortie : LISTE (I_i, D_j^m, M^m)

début
    LISTE = liste vide;

    pour les descripteurs I_i de IMAGE faire(a)
    début
        calculer INDEX de I_i dans ESPACE_INDEXATION;
        récupérer la liste L des descripteurs stockés à INDEX;
        pour les descripteurs D_j^m dans L faire(b)
        début
            ajouter (I_i, D_j^m, M^m) dans LISTE;
            /* M^m est le modèle dont D_j^m fait partie */
        fin
    fin

    retourner LISTE
fin
```

aucun cas à la généralité du résultat énoncé, et d'autres distributions du bruit donneraient des formules sensiblement comparables.

Les descripteurs locaux suivent donc une loi uniforme. Leur support est défini par l'hypercube $S_{D_k} = \left[D_k^i - \frac{\varepsilon^i}{2}, D_k^i + \frac{\varepsilon^i}{2}\right]^{i=1\ldots n}$, où D_k^i est la $i^{\text{ème}}$ composante du descripteur D_k. On formule la mise en correspondance entre deux descripteurs D_1 et D_2 comme le calcul d'une distance probabiliste :

$$P(D_1 = D_2) = \int\limits_{S_{D_1} \cap S_{D_2}} \frac{1}{\prod_{i=1}^{n} \varepsilon^i} d\sigma$$

Cette intégrale est non nulle dès lors que l'un des deux descripteurs, par exemple D_2, tombe dans l'hypercube $\mathcal{K} = \left[D_1^i - \varepsilon_i, D_1^i + \varepsilon_i\right]^{i=1\ldots n}$ défini par l'autre (dans ce cas-ci, D_1). Dans le cas où D_1 représente un descripteur d'une image inconnue, et D_2 celui d'un modèle indexé, ce résultat peut être directement implémenté dans un environnement d'indexation, à condition que l'espace d'indexation intègre la structure topologique de l'espace des descripteurs. Il suffit alors de trouver les sommets κ_{inf} et κ_{sup} définissant l'hypercube d'incertitude $\mathcal{K}$ d'en calculer leurs index $\vec{\imath}_{\kappa_{inf}}$ et $\vec{\imath}_{\kappa_{sup}}$ dans l'espace d'indexation et de considérer tous les descripteurs dans l'hypercube résultant $\mathcal{I}_{\mathcal{K}} = \left[\vec{\imath}_{\kappa_{inf}}, \vec{\imath}_{\kappa_{sup}}\right]$ dans cet espace. L'algorithme 5.1 est alors modifié comme indiqué dans l'algorithme 5.2, p. 118.

L'ajout de la boucle (c) multiplie donc la taille de la liste de sortie par le nombre de passages dans celle-ci, étant donné que le nombre d'itérations dans (c) correspond

Algorithme 5.2 Indexation avec mise en correspondance bruitée

```
Paramètres d'entrée : IMAGE, ESPACE_INDEXATION
Paramètres de sortie : LISTE (I_i, D_j^m, M^m)

début
    LISTE = liste vide;

    pour les descripteurs I_i de IMAGE faire(a)
    début
        déterminer ι_{κ_inf} et ι_{κ_sup} de I_i
        calculer INDEX_MIN de ι_{κ_inf} dans ESPACE_INDEXATION;
        calculer INDEX_MAX de ι_{κ_inf} dans ESPACE_INDEXATION;

        pour les INDEX dans l'hypercube I_K faire(c)
        début
              récupérer la liste L des descripteurs stockés à INDEX;
              pour les descripteurs D_j^m dans L faire(b)
              début
                  ajouter (I_i, D_j^m, M^m) dans LISTE;
              fin
        fin
    fin

    retourner LISTE
fin
```

au nombre d'emplacements d'indexation dans l'hypercube $\mathcal{I}_{\mathcal{K}}$. Soit donc η_i pour chaque dimension i tel que $\eta_i = \vec{\imath}^{\,i}_{\kappa_{sup}} - \vec{\imath}^{\,i}_{\kappa_{inf}}$. La taille de la liste retournée devient :

$$\left(\prod_{i=i}^{n} \eta_i\right) . \frac{\tilde{D}^2 . M}{\prod_{i=i}^{n} k_i}$$

ce qui peut être reformulé ainsi :

$$\frac{\tilde{D}^2 . M}{\prod_{i=i}^{n} \frac{k_i}{\eta_i}} \tag{5.4}$$

5.4.4 Complexité globale

Comme indiqué par le schéma général de la reconnaissance par indexation de descripteurs locaux FIG. 5.1, l'algorithme de reconnaissance proprement dit se divise en deux phases : la première étant l'accès à l'espace d'indexation pour récupérer des mises en correspondance plausibles entre les descripteurs de l'image « requête » et ceux des modèles indexés, la deuxième filtrant la sortie de la première pour vérifier la cohérence et classer les modèles. Si on appelle $\mathcal{M}$ la sortie de la première phase, on obtient comme complexité finale de l'algorithme, en fonction d'une image $\mathcal{I}$, la formule suivante :

$$\mathcal{C}_a(\mathcal{I}) + \mathcal{C}_v(\mathcal{M}(\mathcal{I}))$$

$\mathcal{C}_a(\mathcal{I})$	correspond à la complexité d'accès aux descripteurs stockés dans la base qui sont similaires à ceux de l'image $\mathcal{I}$.
$\mathcal{C}_v(x)$	représente la complexité de la vérification globale d'une liste x de mises en correspondance entre descripteurs inconnus d'une part, et des descripteurs de modèles connus d'autre part.
$\mathcal{M}(\mathcal{I})$	est la liste résultant de la première phase de l'algorithme pour une image $\mathcal{I}$ dont la taille est donnée par l'équation (5.4).

Puisqu'il n'y a, en général, aucune structuration particulière entre les différents descripteurs et qu'ils sont indépendants du contenu de la base, on peut considérer que :

$$\mathcal{C}_a(\mathcal{I}) = C_a . \tilde{D}$$

où C_a est la complexité pour accéder aux index dans la base d'indexation qui sont similaires à un descripteur, et $\tilde{D}$ le nombre de descripteurs dans $\mathcal{I}$.

5.4.4.1 Cas général « non bruité »

On considère que le cas « non bruité » représente les cas de figure où C_a est une constante, indépendante notamment de la dimension n[4], du nombre d'accès $\tilde{D}$ ou de la valeur du descripteur. Ceci est vérifié quand les descripteurs ne sont pas bruités, quand ils ont une sémantique qualificative ou si l'indexation est faite de façon à absorber le bruit des descripteurs.

De plus, dans les algorithmes de reconnaissance par modélisation locale, il n'y a, *a priori*, aucun ordonnancement entre les différentes mises en correspondances dans $\mathcal{M}(\mathcal{I})$. De la même façon, il n'y a aucun lien, *a priori*, entre les différents modèles stockés. Par conséquent, afin de pouvoir obtenir une liste triée des modèles à partir des mises en correspondance fournies, et n'ayant aucune information autre que celles-ci, et éventuellement une mesure de confiance pour chacune d'entre elles, la seule solution consiste à les faire intervenir toutes dans le processus de décision.

On en déduit donc que la complexité de la phase de vérification de cohérence est au moins linéaire en fonction du nombre d'entrées, et que par conséquent, la borne inférieure de la complexité de l'algorithme générique de reconnaissance par indexation s'écrit :

$$C_a.\tilde{D} + \frac{\tilde{D}^2.M}{\prod_{i=i}^{n} k_i} \tag{5.5}$$

Ce résultat est similaire à celui énoncé dans l'étude plus restreinte décrite dans [18].

Il est clair que l'on peut négliger la première partie de la formule, puisqu'elle ne dépend que de $\tilde{D}$ et de façon linéaire, tandis que la deuxième fait intervenir $\tilde{D}$ de façon quadratique. La fonction de complexité est monotone en chacun de ses paramètres, comme on constate aussi bien dans la formule que dans FIG. 5.7.

5.4.4.2 Cas général bruité

Contrairement au cas précédent, C_a n'est plus constante, mais varie selon la dimension n du descripteur. En effet, afin de prendre en compte le bruit, il est nécessaire de visiter un certain nombre de voisins de l'index n-dimensionnel de celui-ci. Ceci représente donc un hyper-volume, défini par l'incertitude autour du point. Par contre, on peut, comme au cas précédent, maintenir l'hypothèse que la partie de vérification globale est linéaire par rapport au nombre de mises en correspondance trouvées.

Par conséquent, la borne inférieure de la complexité de l'algorithme générique de reconnaissance par indexation s'écrit :

4. Dans des applications réelles l'indépendance par rapport à n n'est pas toujours vérifiée. En théorie on peut toujours supposer que l'espace d'indexation est un tableau n-dimensionnel à accès constant. Dans la pratique les limitations des ressources physiques empêchent généralement d'indexer des valeurs de cette manière, et on a recours à des structures de stockage plus complexes dont le temps d'accès peut alors dépendre de n.

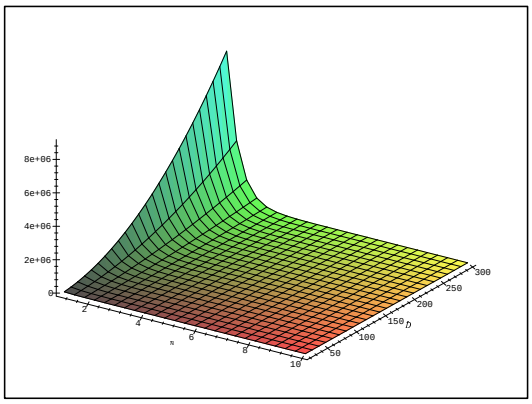


FIG. 5.7: *Aperçu de la complexité algorithmique de l'indexation locale non bruitée :* $\tilde{D} \in [20..300]$, $n \in [1..10]$, $k_i = 10$ *pour toutes les dimensions et* $M = 1000$.

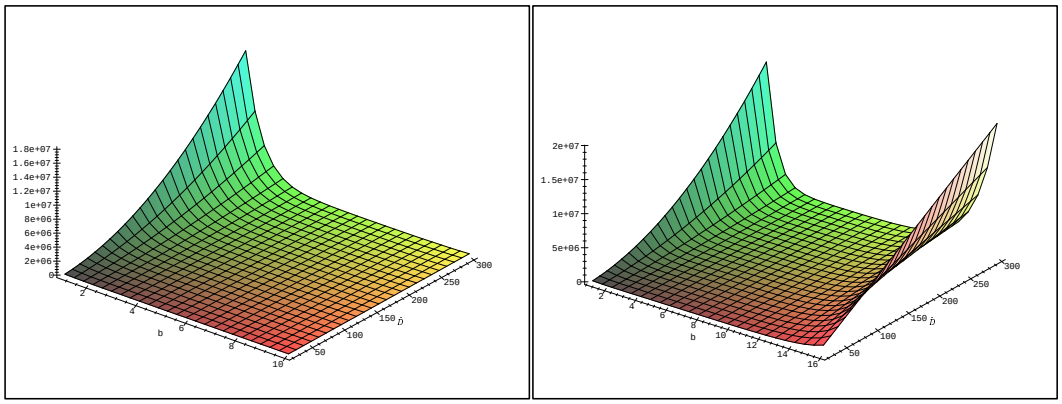


FIG. 5.8: *Aperçu de la complexité algorithmique de l'indexation locale bruitée :* $\tilde{D} \in [20..300]$, $n \in [1..10]$ *à gauche,* $n \in [1..16]$ *à droite,* $k_i = 10$ *et* $\eta_i = 2$ *pour toutes les dimensions et* $M = 1000$.

$$C_a^{'}\left(\prod_{i=1}^{n}\eta_i\right).\tilde{D}+\frac{\tilde{D}^2.M}{\prod_{i=1}^{n}\frac{k_i}{\eta_i}} \tag{5.6}$$

où $C_a^{'}$ représente le coût d'accès à une case d'indexation (équivalent au C_a du cas sans bruit). Nous prendrons sa valeur égale à 1 pour les représentations graphiques, et nous donnerons des ordres de grandeur de cette constante dans la section § 5.5.

On ne peut plus négliger la première partie de la fonction, comme on l'avait fait dans le cas non bruité, du fait qu'elle n'est plus monotone en n. Comme on le constate dans FIG. 5.8, pour une échelle de valeurs de n suffisamment petite, la fonction a un comportement similaire à celui du cas non bruité (figure de gauche, à comparer avec FIG. 5.7). Par contre, la complexité croît de façon exponentielle dès lors que n devient suffisamment grand (figure de droite).

5.4.5 Analyse du cas non bruité

Nous allons analyser maintenant les résultats énoncés dans les sections précédentes, en nous basant sur la formule finale (5.5) obtenue. Ceci nous permettra d'une part de mieux comprendre l'influence des différents paramètres entrant en jeu, et d'autre part de vérifier les résultats de façon expérimentale. Des expériences sur des images utilisées pour des applications réelles [62] nous ont d'ailleurs déjà indiqué qu'une augmentation des descripteurs dans les images peut rendre bon nombre de méthodes inutilisables. Nous montrerons ici comment éviter des problèmes de complexité dans ce contexte. Nous nous appuierons sur une implémentation de la reconnaissance par indexation décrite dans le chapitre 3 dans laquelle les descripteurs locaux sont calculés à partir de configurations de trois ou quatre points connectés par des segments de droites dans une image segmentée. La taille des descripteurs varie selon qu'ils sont calculés à partir de trois ou quatre points ou selon qu'ils font intervenir des segments orientés ou non. Leur dimension varie alors de deux à cinq.

5.4.5.1 Influence du nombre de modèles

Dans la formule (5.5), la taille de la base d'indexation est donnée par M, le nombre de modèles indexés. Il est trivial de voir que le nombre de modèles a une influence linéaire sur le comportement de l'algorithme. En guise de validation expérimentale, nous avons pris 200 images de complexité comparable, et nous les avons progressivement ajoutées dans la base d'indexation. À chaque étape nous avons interrogé la base avec une même image de test, de complexité comparable à celles indexées. Le schéma dans FIG. 5.9 montre clairement que le temps d'exécution évolue de façon linéaire avec la taille de la base.

5.4.5.2 Influence de la complexité des images

L'influence de la taille des images (au sens du nombre de descripteurs) entrant en jeu est beaucoup plus importante. Si on considère des applications pour des méthodes

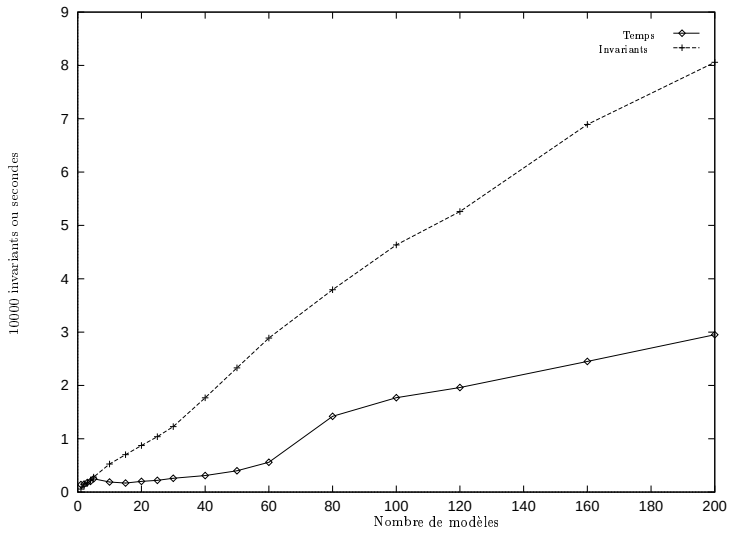


FIG. 5.9: *Évolution linéaire du temps d'exécution et de la population des emplacements d'indexation avec la taille de la base (en nombre de modèles).*

d'indexation existantes [60, 92, 89], on observe que leur limite est tracée par la complexité des images entrant en jeu. Étant donné que celle-ci a une influence quadratique sur le temps d'exécution, leur utilité s'en trouve remise en question.

L'expérience suivante montre que le temps d'exécution de l'algorithme augmente effectivement de façon quadratique. La figure FIG. 5.10 montre l'évolution du nombre de correspondants trouvés (indiqués en impulsions) et du temps d'exécution (indiqué en courbe continue) en fonction du nombre de descripteurs par image.

Dans cette expérience nous avons pris une base de 60 images que nous avons progressivement modélisées par de plus en plus de descripteurs (dans ce cas précis, nous avons utilisé les configurations de 4 points développées au chapitre précédent) allant de 100 à 1400 descripteurs par modèle. La courbe présente la moyenne sur 120 images pour lesquelles nous avons fait évoluer le nombre de descripteurs de façon similaire.

On note que, quand l'espace est « creux », c'est-à-dire, quand le taux de paniers occupés est très faible, l'évolution du temps d'exécution est linéaire. Cette situation se traduit par le fait que $\bar{I}$ des équations (5.1) et (5.1) est très petit par rapport à C_a, la complexité pour l'accès à une clef d'indexation de l'équation (5.5). Ceci est le cas lorsque n est élevé.

L'expérience suivante, représentée dans la figure FIG. 5.11, utilise les descripteurs de SCHMID sur une base de 12 images pour lesquels on fait varier le nombre de descripteurs de 10 à 140. On rappelle que l'espace d'indexation est de dimension 9. On y voit clairement que l'évolution du temps d'exécution est linéaire avec l'augmentation du nombre de descripteurs.

Il est intéressant, au vu de ces résultats, d'aborder l'optimisation que l'on serait tenté de faire afin de réduire le temps d'exécution en diminuant le nombre de descripteurs par image.

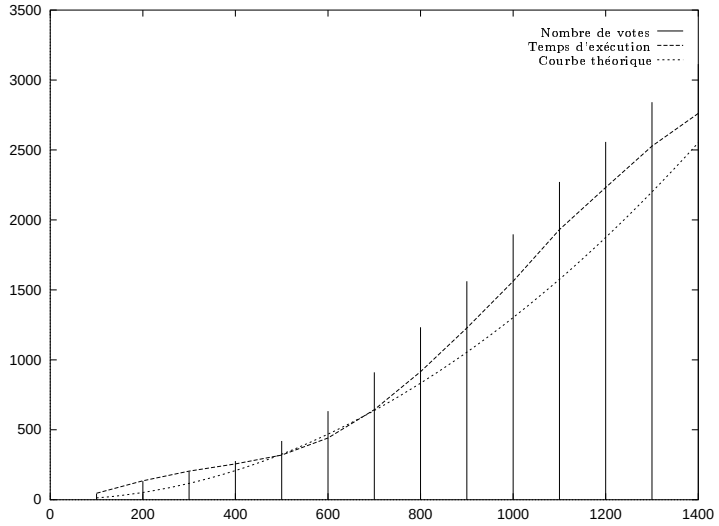


FIG. 5.10: *Évolution du temps d'exécution et du nombre d'appariements en fonction du nombre de descripteurs.*

Remarque D'après les définitions de la modélisation *globale* et *locale* données dans les sections 2.2 et 2.3, toute modification visant à réduire le nombre de descripteurs locaux peut être interprétée comme un éloignement du paradigme local pour devenir plus global. De ce fait la méthode deviendra moins robuste à des modifications de la scène comme des occultations, des rajouts d'autres objets dans la scène, etc. Plus elle deviendra globale, plus le risque d'avoir une dégradation de performances au niveau de la reconnaissance deviendra grand. Afin de pouvoir absorber ce manque de robustesse et de garder en même temps les performances de reconnaissance antérieures à la réduction du nombre de descripteurs, il devient donc nécessaire de modéliser plus d'aspects visuels pour un même objet. Il n'est pas sûr que l'augmentation de M ne soit pas plus importante que l'augmentation de $\tilde{D}^2$ étant donné que la relation entre les deux ne peut pas être exprimée de façon formelle. Le gain effectif en temps n'est donc pas assuré dans tous les cas.

5.4.5.3 Influence de la taille des descripteurs

Le dernier paramètre entrant en jeu dans la formule (5.5) est n, lié à la taille des descripteurs. C'est le seul qui sera capable de réduire de façon significative le temps de reconnaissance. On peut d'ailleurs voir sur FIG. 5.7 que la fonction qui donne le temps d'exécution décroît beaucoup plus vite selon l'axe de n qu'elle ne croît en fonction de $\tilde{D}$. De façon formelle, on constate par ailleurs que la fonction est strictement décroissante selon n.

Sur une même base de test de 20 images nous avons pris la moyenne sur 120 requêtes de reconnaissance. Nous avons, pour cette expérience, utilisé les descripteurs de SCHMID (*cf.* algorithme 4.1). La figure FIG. 5.12 montre en abscisse la taille du descripteur. Dans notre configuration, ceci correspond aux n premières composantes du vecteur du descripteur cité.

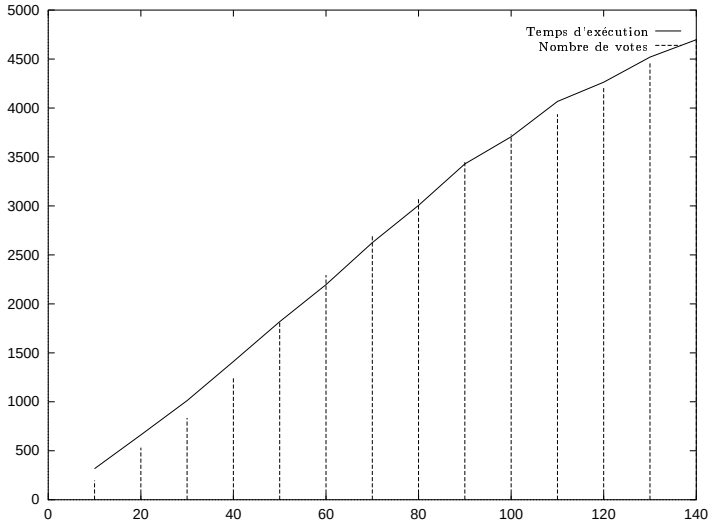


FIG. 5.11: *Évolution du temps d'exécution et du nombre d'appariements en fonction du nombre de descripteurs dans un espace « creux ».*

En ordonnée, la courbe montre le temps d'exécution (en centièmes de seconde), tandis que les barres indiquent le nombre de votes, en moyenne, reçus par le modèle gagnant[5].

On obtient un corollaire intéressant quand on combine l'information de la décroissance du temps de reconnaissance avec n avec l'observation des k_i. En effet, si l'on dispose d'une gamme suffisamment large de composantes pour nos descripteurs locaux, il n'est plus nécessaire de rester très précis sur la mise en correspondance entre eux (d'un point de vue du temps d'exécution). C'est-à-dire qu'il est plus intéressant d'avoir une subdivision en 10 emplacements par dimension en $n = 9$ dimensions (et donc d'avoir moins de précision sur la mise en correspondance initiale) que d'avoir une subdivision en 100 emplacements (et une plus grande précision) en $n = 4$ dimensions.

Bien que ceci puisse paraître trivial, les conséquences sont intéressantes à analyser. Étant donné que les descripteurs sont en général soit des invariants à part entière, et alors, dans la plupart des cas, très instables numériquement, soit des quasi-invariants, et alors non constants pour toutes les transformations du point d'observation, la possibilité d'être moins contraint lors de leur comparaison a pour résultat qu'un plus large spectre de transformations peut être couvert. Un effet de bord direct est alors qu'un nombre plus petit de points de vue, ou d'aspects, est nécessaire pour modéliser un objet, allégeant ainsi le poids des modèles sur la reconnaissance. On pourrait dire aussi que la réduction des contraintes revient à devenir moins quantitatif dans la reconnaissance, et plus qualitatif, ce qui est un point très intéressant au vu d'une reconnaissance de classes génériques. On trouve des résultats encourageants dans cette direction dans [19].

5. Cette expérience ne prend pas en compte la cohérence géométrique évoquée au chapitre précédent.

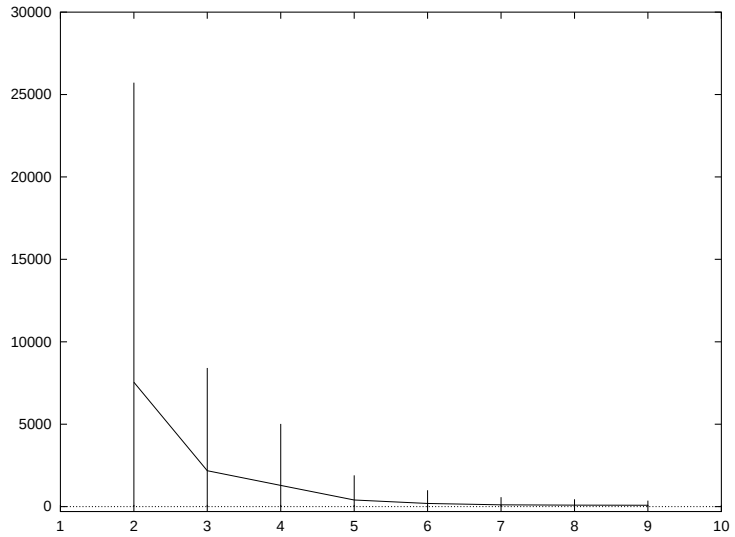


FIG. 5.12: *Évolution du temps d'exécution avec la taille du descripteur, pour une indexation exacte.*

5.4.5.4 Conclusion

En conclusion, il est évident que dans le cas d'une mise en correspondance sans bruit, l'augmentation de la dimension des index ne peut être que bénéfique. Ce résultat est une généralisation des travaux de CALIFANO et MOHAN [18] dans leur étude plus orientée vers le *Geometric hashing*.

On constate également qu'une diminution du nombre de paniers par réduction du pas d'échantillonnage couplée à une augmentation de la dimensionnalité permet de conserver le temps d'exécution, tout en absorbant des problèmes qui pourraient être dus à la discrétisation et qui mèneraient à des identifications de modèles erronées (*cf.* également [18]).

5.4.6 Analyse du cas bruité

Rappelons que dans l'équation (5.6), la complexité trouvée dans le cas bruité était :

$$C'_a \left(\prod_{i=1}^{n} \eta_i \right) .\tilde{D} + \frac{\tilde{D}^2 .M}{\prod_{i=1}^{n} \frac{k_i}{\eta_i}}$$

Comme dans le cas non bruité, il est clair que le nombre de modèles M a une influence linéaire. Nous n'y reviendrons donc pas. On pourrait, de la même façon, considérer que le comportement asymptotique en fonction du nombre de descripteurs $\tilde{D}$ reste aussi inchangé. Nous pensons tout de même qu'il est intéressant de l'étudier de façon plus détaillée, puisqu'en général, $\tilde{D}$ est une variable qui reste bornée. Nous n'étudierons pas non plus l'influence des k_i et η_i de façon exhaustive, car nous considérons que ce sont des données fixes, liées à la définition du type de descripteur utilisé, puisqu'elles représentent, en quelque sorte, la taille de l'intervalle sur lequel les valeurs des descripteurs évoluent

(k_i), et la précision nécessaire à leur comparaison (η_i). Nous évoquerons seulement leur influence dans le cas optimal.

5.4.6.1 Remarque concernant la précision

Dans le contexte où les k_i et η_i sont des données fortement liées au type de descripteurs utilisés, on peut se poser la question de savoir si la variation des η_i a une quelconque influence dans la pratique courante. En effet, si on suppose que le η_i est constant pour une dimension i donnée, et si on suppose que $\eta_i \geq 2$, cela veut dire que pour chaque requête il faudra systématiquement considérer une zone fixe dans l'espace d'indexation, indépendamment de cette requête. En d'autres termes, on peut obtenir un résultat de mise en correspondance similaire, en incluant dès le départ cette zone dans d'échantillonnage de la base d'indexation. On optimise ainsi le nombre d'accès, et on peut considérer qu'au mieux $\eta_i = 2$ dans ces cas-là.

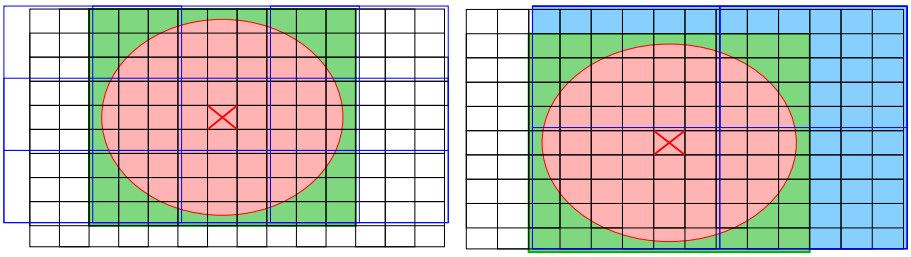


FIG. 5.13: *Influence du pas d'échantillonnage k et du pas d'erreur η sur la zone explorée.*

À titre d'exemple, considérons les échantillonnages proposés dans FIG. 5.13. La croix représente la position d'un descripteur, et l'ellipse rouge son incertitude. L'échantillonnage initial est représenté comme une grille noire, et la zone verte représente les voisins accédés lors d'une requête ($\eta = 9$). Deux autres échantillonnages ($k' = k/3$ et $k'' = k/6$) et les zones accédées lors d'une requête ($\eta' = 3$ et $\eta'' = 2$) sont représentés en bleu.

Dans le premier cas, on accède à la même zone pour un moindre coût. Dans le deuxième cas, on réduit le coût d'accès, mais on inclut plus de candidats potentiels. Du point de vue de l'étude de la complexité, on peut donc toujours considérer que $\eta = 2$. Il dépendra des applications réelles mises en œuvre, si le nombre de faux appariements introduits de cette façon peuvent être traités de manière robuste. Pour des applications moins tolérantes, η peut alors être supérieur à 2.

Dans [18], CALIFANO et MOHAN montrent que, pour des descripteurs invariants, une augmentation de la dimension liée à une réduction du pas d'échantillonnage préserve le pourcentage de votes corrects.

5.4.6.2 Influence de la complexité des images

Il faut noter tout d'abord que la fonction de complexité est strictement croissante en $\tilde{D}$. Son comportement à la limite, quand $\tilde{D}$ tend vers l'infini, est bien quadratique. Mais dans des cas réels, où $\tilde{D} < 1000$ et la valeur de n est suffisamment grande (c-à-d. $n > 3$),

le comportement linéaire est largement prédominant d'après les simulations (FIG. 5.16). On peut donc supposer que dans les cas pratiques, la reconnaissance se fait en un temps subquadratique, surtout dans les cas optimisés. Nous y reviendrons au moment d'aborder l'optimisation du problème dans § 5.4.6.4, p. 129.

5.4.6.3 Influence de la taille des descripteurs

Remarquons d'abord que pour n suffisamment petit, le comportement est identique pour les cas bruité et non bruité. Globalement, par contre, la fonction n'est plus monotone, et elle présente clairement une ligne minimale au milieu, bien qu'elle ne présente pas de minimum absolu. Vers l'infini, la fonction croît de façon exponentielle, la partie gauche de la formule (5.6) l'emportant sur la partie droite (*cf.* FIG. 5.8).

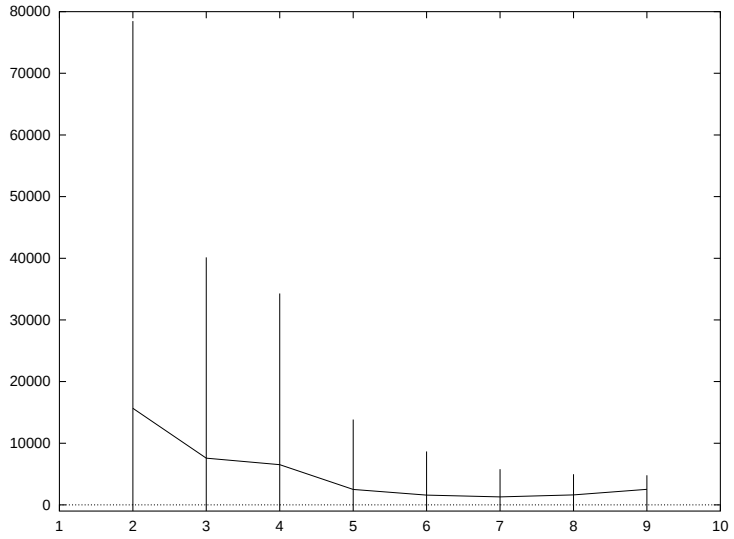


FIG. 5.14: *Évolution du temps d'exécution avec la taille du descripteur, pour une indexation bruitée.*

Il est clair que l'optimisation simple, proposée dans le cas de l'indexation sans bruit, p. 126 ne tient plus ici, et qu'un choix plus judicieux de n devient nécessaire. Nous étudierons cette optimisation dans la section qui suit. La figure FIG. 5.14 reprend la même expérimentation que celle décrite p. 124. Notez le temps d'exécution, qui commence à croître à partir d'une taille $n = 9$.

5.4.6.4 Optimisation

Nous chercherons dans cette partie les relations que doivent vérifier les paramètres de la fonction (5.6), que nous appellerons f par la suite, afin d'avoir un comportement optimal. Étant donné que la seule dimension dans laquelle elle n'est pas monotone est n, nous allons commencer notre optimisation par ce paramètre. Pour des raisons de simplification de la

présentation, nous supposerons, sans perte de généralité par ailleurs, que :

$$\forall i \quad \begin{cases} k_i = k \\ \eta_i = \eta \end{cases}$$

On exclut par ailleurs que $\eta = 1$ ou $\eta = k$, la première contrainte correspondant à une mise en correspondance sans bruit, la seconde correspondant à une mise en correspondance entre tous les descripteurs indexés, auquel cas un algorithme à parcours séquentiel est mieux adapté.

Tout ceci nous permet d'écrire que :

$$\mathrm{f} = C_a^{'}.\eta^n.\tilde{D} + \frac{\tilde{D}^2.M}{\left(\frac{k}{\eta}\right)^n}$$

$$\frac{\mathrm{df}}{\mathrm{d}n} = C_a^{'}.\eta^n \ln(\eta)\tilde{D} - \frac{\tilde{D}^2 M \ln\left(\frac{k}{\eta}\right)}{\left(\frac{k}{\eta}\right)^n} \tag{5.7}$$

En posant ensuite $\frac{\mathrm{df}}{\mathrm{d}n} = 0$, nous obtenons que

$$n_{min} = \log_k \left(\frac{\tilde{D}M}{C_a^{'}} \left(\frac{\ln(k)}{\ln(\eta)} - 1 \right) \right) \tag{5.8}$$

À titre d'exemple, avec les valeurs numériques utilisées dans les différentes figures précédentes ($k = 10$, $\eta = 2$, $D = 300$, $M = 1000$, $C_a^{'} = 1$), on obtient comme valeur théorique $n_{min} = 5.8$. Par continuité, on en déduit que le minimum, pour $n \in \mathbb{N}$ est obtenu pour $n_{min} = 6$. Cette valeur correspond visuellement au milieu de la zone creuse de la surface dans la FIG. 5.8.

On peut donc, faute d'avoir les mêmes moyens que dans le cas non bruité pour faire décroître le temps de reconnaissance, formuler (en fonction du nombre de modèles et de celui des descripteurs) la taille que doit avoir un descripteur pour fournir une performance optimale pour la reconnaissance.

Supposons maintenant que nous ayons la possibilité de choisir n, et donc qu'on l'ait pris comme n_{min}. Que peut-on dire dans ce cas-là du temps d'exécution en fonction de $\tilde{D}$ et de M ?[6]

En remplaçant n par n_{min} dans f, on obtient, après simplification :

$$\mathrm{f}_{min} = C_a^{'}.\eta^{\left(\frac{\ln\left(\frac{\tilde{D}M}{C_a^{'}} \left(\frac{\ln(k)}{\ln(\eta)} - 1 \right) \right)}{\ln(k)} \right)} \tilde{D} + \frac{\tilde{D}^2 M}{\left(\frac{k}{\eta}\right)^{\left(\frac{\ln\left(\frac{\tilde{D}M}{C_a^{'}} \left(\frac{\ln(k)}{\ln(\eta)} - 1 \right) \right)}{\ln(k)} \right)}}$$

6. Dans ce qui suit, nous nous contenterons de donner les étapes principales du raisonnement mathématique. Le lecteur est invité à lire l'annexe C pour les détails du raisonnement et des démonstrations.

On pose alors :

$$x = \frac{\ln(k)}{\ln(\eta)} \tag{5.9}$$

Puisque $1 < \eta \leq k < \infty$ on en conclut que $1 < x < \infty$.

En factorisant et en réorganisant les opérandes en $\tilde{D}$ et M, ceci permet de réécrire f_{min} comme :

$$\mathrm{f}_{min} = \left(e^{\frac{\ln(x-1)}{x}} \frac{x}{x-1} \right) \frac{\tilde{D}^{\frac{x+1}{x}} M^{\frac{1}{x}}}{C_a'^{\frac{1}{x}}} \tag{5.10}$$

Notons :

$$K = e^{\frac{\ln(x-1)}{x}} \frac{x}{x-1}$$

FIG. 5.15 donne un aperçu de K. On montre que le facteur est borné dans l'intervalle $x \in]1..\infty[$ et $K \in]1..2]$, avec un seul maximum $K(2) = 2$.

On peut donc conclure que, pour un n optimal, le problème est sub-quadratique en $\tilde{D}$ et sub-linéaire en M. Le degré des deux variables dépend de l'échantillonnage selon les différentes dimensions dans la base d'indexation. Le facteur multiplicatif K, de $\tilde{D}$ et M, n'est jamais très grand car il est borné par 2.

Remarque Notons que quand $x = \infty$, nous nous trouvons dans un cas d'indexation sans bruit. En effet, k est alors « infiniment plus grand » que η. Or dans ce cas nous avions trouvé une complexité en $\tilde{D}^2$ (*cf.* équation (5.3), p. 116), tandis que l'équation (5.10) donne une complexité linéaire. On trouve l'explication dans le fait qu'il n'existe pas de n_{min} pour le cas où $x = \infty$ (ou plutôt, $n = \infty$ dans ce cas-là). Si dans l'équation (5.3), on prend un n infini, on retrouve bien la complexité linéaire dans les deux cas.

5.4.6.5 Conclusion

Dans le cas d'une indexation de descripteurs bruités, on se heurte à un problème similaire à la dérive dimensionnelle évoquée dans la section 5.3.

Il est intéressant de comparer la complexité de l'équation (5.6) à celle d'une recherche séquentielle, exhaustive de tous les descripteurs indexés (*cf.* discussion p. 114). Celle-ci étant de la forme :

$$\tilde{D}.\tilde{D}M = \tilde{D}^2 M \tag{5.11}$$

elle devient plus petite que celle de (5.6) dès lors que :

$$\begin{aligned} & \left(\tfrac{A-1}{A}\right) \quad \tilde{D}^2 M \approx \tilde{D}^2 M && \leq \eta^n \\ \Leftrightarrow \quad & \log_\eta \left(\tilde{D}^2 M \right) && \leq n \end{aligned} \tag{5.12}$$

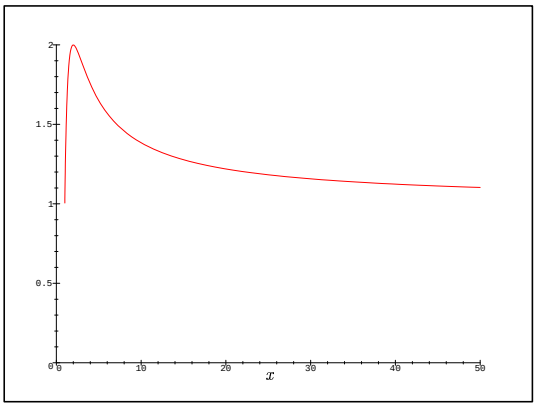


FIG. 5.15: *Tracé de* $e^{\frac{\ln(x-1)}{x}\frac{x}{x-1}}$.

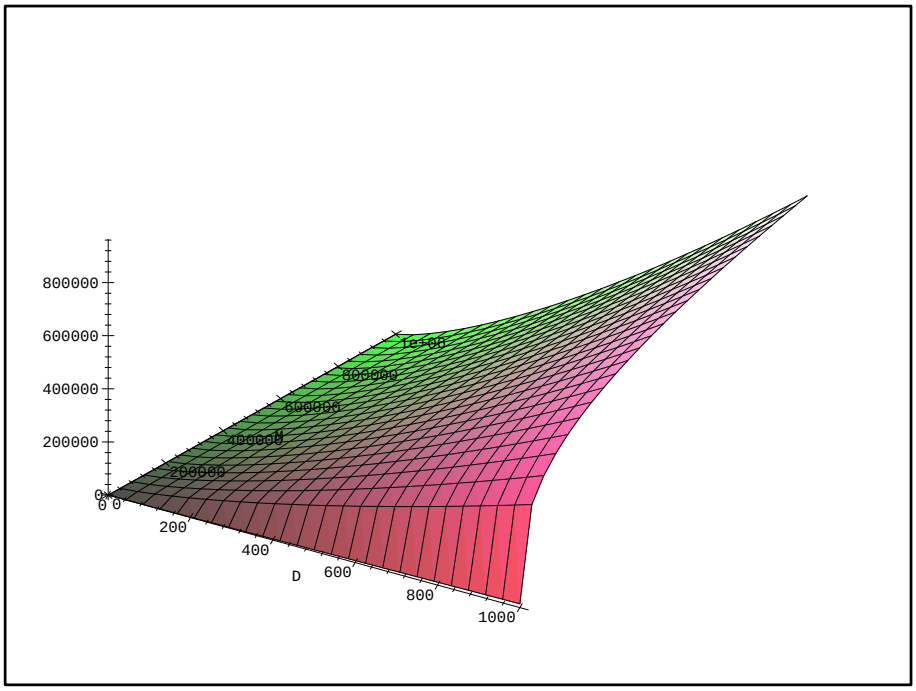


FIG. 5.16: *Évolution de* f *en fonction de* $\tilde{D}$ *et de* M *dans le cas optimisé, avec* $k = 10$, $\eta = 2$ *(équivalent à* $x = 6,64$*) et* $n = n_{min}$.

(sous l'hypothèse que les η_i sont identiques et en posant $A = \prod_{i=1}^{n} \frac{k_i}{\eta_i}$.)

Avec les ordres de grandeur déjà évoqués ($\eta = 2$, $M = 1000$, $\tilde{D} = 300$) la recherche séquentielle est moins coûteuse à partir de $n > 19$. Ce qui correspond aux résultats expérimentaux annoncés dans [9, 15].

S'il est possible de moduler la dimension des descripteurs, la complexité peut être réduite en prenant une valeur pour n en fonction du type d'image considéré. Il n'existe, par contre, pas de solution dans le cas général et le temps de reconnaissance croît de façon exponentielle avec n. Dans la section qui suit, nous donnerons les liens directs avec les méthodes de reconnaissance développées dans le chapitre 4 et nous proposerons quelques éléments qui permettront éventuellement de contourner le problème de cette croissance.

5.5 Application aux descripteurs du chapitre 4

Dans les sections précédentes nous avons fait une analyse détaillée de la complexité liée à l'indexation de descripteurs locaux dans une structure discrète. Elle s'applique de façon triviale aux méthodes de reconnaissance que nous avons abordées précédemment. Nous ferons ici une récapitulation en termes de complexité de ces approches, en concluant sur une piste de réduction de la complexité.

5.5.1 Rappel des classes d'approches principales

Nous distinguons, dans les méthodes que nous avons proposées, trois classes principales.

Premièrement, les méthodes *simples*, qui entrent directement dans le contexte de l'étude de complexité de ce chapitre. Elles énumèrent un ensemble de configurations dans une image, sur lesquelles elles calculent des vecteurs de descripteurs. Chaque élément de ce vecteur évolue dans un espace continu. Les méthodes appartenant à cette classe sont : la méthode initiale basée sur les configurations en « V » (§ 3.2.1), les extensions directes en « Z » et en « Y » (§ 4.1), les configurations hybrides avec un point et deux droites (« SSP ») ou une droite et deux points (« SPP ») ainsi que la méthode de SCHMID (§ 4.2.3).

Ensuite, il y a celles faisant intervenir des composantes discrètes parmi les descripteurs. Dans notre cas il s'agit typiquement de l'orientation des segments qui permet de distinguer les configurations (*cf.* § 4.1.2). Cette orientation a été introduite pour les 5 premières classes simples.

Finalement, nous distinguons les méthodes de *collaboration*, faisant intervenir plusieurs approches (de type quelconque) en les faisant voter dans le même espace de vote. Il s'agit dans notre cas de la méthode de simple collaboration « Z-Y », § 4.1.1, et la méthode de collaboration complète décrite dans la section § 4.2.3.

Ces trois classes sont confrontées à notre analyse de complexité dans les sections qui suivent. Nous n'aborderons pas la complexité dans le cas d'une identification entièrement sans bruit, puisqu'elle n'a généralement que peu d'intérêt dans des cas réels.

5.5.2 Méthodes simples

Dans le cas des méthodes simples, l'analyse faite section § 5.4.6 s'applique telle quelle. On rappelle que dans ce cas, la complexité est décrite dans l'équation (5.6), et vérifie :

$$C_a^{'}.\left(\prod_{i=1}^{n}\eta_i\right).\tilde{D}+\frac{\tilde{D}^2.M}{\prod_{i=1}^{n}\frac{k_i}{\eta_i}}$$

Dans notre cas $\forall i, \eta_i = \eta$. On la notera $f_1 + f_2$ pour mieux indiquer dans le tableau 5.1 que l'influence de la vérification globale (*i.e.* f_2) est, dans le cas d'une indexation bruitée, minime par rapport au coût d'accès aux descripteurs (*i.e.* f_1) pour des dimensions plus élevées. Le tableau 5.1 reprend pour toutes les méthodes simples citées le calcul de la complexité théorique qui y est associée.

	M	$\tilde{D}$	η	n	$C_a^{'}$	k_i	Complexité f_1+f_2	
V	12	526	2	2	41,81	$\begin{cases} k_\alpha = 24 \\ k_\rho = 20 \end{cases}$	$f_1 =$ $f_2 =$ $\rightarrow$	87968 27668 **115636**
Z	12	889	2	4	1,482	$\begin{cases} k_{\alpha_1} = k_{\alpha_2} = 24 \\ k_{\rho_1} = k_{\rho_2} = 20 \end{cases}$	$f_1 =$ $f_2 =$ $\rightarrow$	21080 659 **21739**
Y	12	233	2	4	1,078	$\begin{cases} k_{\alpha_1} = k_{\alpha_2} = 24 \\ k_{\rho_1} = k_{\rho_2} = 20 \end{cases}$	$f_1 =$ $f_2 =$ $\rightarrow$	4020 45 **4065**
SSP	12	841	2	3	19,40	$\begin{cases} k_{\alpha_1} = 24 \\ k_{\rho_1} = k_{\rho_2} = 20 \end{cases}$	$f_1 =$ $f_2 =$ $\rightarrow$	130523 7071 **137594**
SPP	12	2282	2	4	94,5	$\begin{cases} k_{\alpha_1} = k_{\alpha_2} = 24 \\ k_{\rho_1} = k_{\rho_2} = 20 \end{cases}$	$f_1 =$ $f_2 =$ $\rightarrow$	3441751 4340 **3454091**
SCHMID	12	125	3	9		$\begin{cases} k_1 = 1000 \\ k_2 = 360 \\ k_{3\ldots9} = 1500 \end{cases}$	$f_1 =$ $f_2 =$ $\rightarrow$	2460375 6.10^{-19} **2460375**

TAB. 5.1: *Récapitulatif des principaux paramètres d'indexation pour des méthodes simples.*

Dans cette partie, nous avons déterminé la valeur de $C_a^{'}$ expérimentalement à partir de données réelles pour chaque type de configuration. Les valeurs obtenues reflètent donc la relation entre la complexité d'un accès à un indice dans notre base d'indexation et l'expression d'un vote. Cette relation numérique n'est valable que pour l'implémentation algorithmique telle que nous l'avons faite et n'a, évidemment, pas de caractère général. Nous ne disposons pas de la valeur de $C_a^{'}$ pour la méthode de SCHMID donc nous l'avons supposé égal à 1, à titre indicatif[7]. L'annexe A contient quelques éléments permettant de

7. La valeur du paramètre se situe probablement entre 0,2 et 0,02, compte tenu du caractère très creux de l'espace.

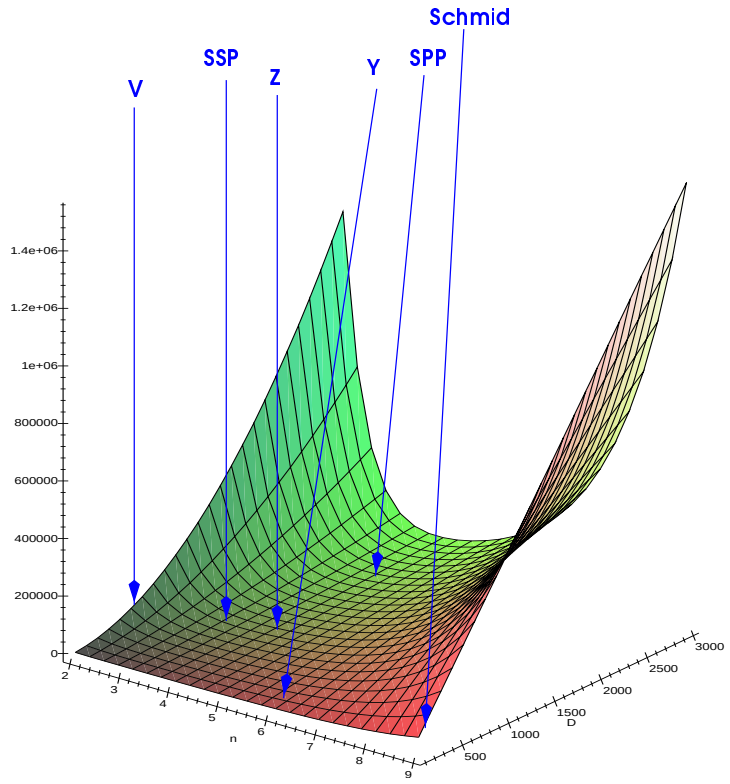


FIG. 5.17: *Situation de la complexité des différentes modélisations simples.*

comprendre la dépendance de ce paramètre par rapport aux choix de l'implémentation. Dans le cas d'un arbre de stockage tel que nous l'avons utilisé, les temps d'accès dépendent fortement de la densité de la population dans l'espace d'indexation, la profondeur de l'arbre, *etc.* Parallèlement, le choix d'implémentation de recherche des voisins a également une influence sur $C_a^{'}$.

La figure FIG. 5.17 situe les valeurs obtenues sur le graphe des valeurs théoriques FIG. 5.8. À titre indicatif, la méthode d'indexation de SCHMID a été ramenée à une situation de recherche avec $\eta = 2$ et des valeurs de k_i similaires à celles des autres approches. Afin d'être cohérent avec les autres graphes de même type que nous avons déjà présenté, $C_a^{'}$ a été supposé égal à 1 pour tous les types de configuration.

Si l'on compare les valeurs théoriques obtenues ici avec les résultats de la comparaison en performance entre les configurations « V », « Z » et « Y » FIG. 4.3 p. 79, on note bien que l'ordre de grandeur des gains observés se justifie par la complexité théorique.

5.5.3 Collaboration entre méthodes

Nous avons également, dans la section § 4.2, présenté une méthode de collaboration entre différentes méthodes de reconnaissance locale. En ce qui concerne de l'étude de la complexité, les résultats de ce chapitre s'y transposent directement. En effet, chaque

méthode travaille de façon autonome. Le fait qu'elles expriment leurs votes dans un espace unique ne diminue, ni n'augmente le nombre des votes. La complexité de deux méthodes travaillant ensemble est par conséquent la somme de leurs complexités respectives.

Ce résultat est vérifié par l'observation des temps de calcul pour la méthode « Z-Y » dans les sections § 4.1.1 et § 4.1.2, où l'on observe que le temps pour cette méthode est la somme des temps pour les méthodes « Z » et « Y ».

5.5.4 Introduction de données discrètes

Dans la section § 4.1.2 nous avons constaté que l'utilisation de l'orientation des segments pour les configurations géométriques réduisait de façon significative les temps d'exécution (et augmentait par la même occasion le pouvoir discriminant des descripteurs). En termes d'étude de complexité, cette approche correspond, en quelque sorte, à mélanger des descripteurs bruités avec des descripteurs non bruités.

Concrètement, nous appelons « augmentation discrète » de la dimension l'ajout d'une caractérisation aux descripteurs qui ne peut prendre que des valeurs discrètes, et qui n'est pas sujette à un éventuel bruit nécessitant l'examen des voisins dans cette dimension. De façon générique, on peut considérer que, si pour des descripteurs comportant déjà n dimensions, la nouvelle dimension introduite peut prendre v valeurs différentes, le processus de reconnaissance est reparti en v sous-processus indépendants avec $\tilde{D'} = \frac{\tilde{D}}{v}$ en ce qui concerne la phase d'accès aux données, avec la taille de $\tilde{D'} = n$. Ceci donne alors comme complexité finale dans le cas bruité :

$$C_a^{'}.\left(\prod_{i=1}^{n}\eta_i\right).\tilde{D} + \frac{\tilde{D}^2.M}{v.\prod_{i=1}^{n}\frac{k_i}{\eta_i}}$$

Dans le cas non bruité l'analyse reste valable, mais n'a aucun sens, puisque il s'agit du cas de figure d'une augmentation de dimension normale. On constate par contre que, dans le cas bruité, la complexité globale diminue, et que le problème reste fondamentalement de dimension n. Ceci permet de l'appliquer à un cas optimisé (*cf.* Annexe § 5.4.6.4) et de réduire le temps d'exécution en le divisant par $v^{(1/x)}$, x ayant la même signification que dans l'équation (5.10).

Cette diminution ne prend pas en compte une modification éventuelle du paramètre $C_a^{'}$. Ce paramètre dépend d'une part de la dimension n de l'espace d'indexation, mais peut également dépendre de la densité de la population dans cet espace (*cf.* A). La distribution des descripteurs dans des espaces discrets différents verra donc la valeur de $C_a^{'}$ diminuer. Cette diminution est fortement liée à la mise en œuvre de l'espace d'indexation.

On peut alors reprendre le tableau 5.1 pour les configurations géométriques et y reporter les modifications en termes de complexité par rapport à l'ajout de l'orientation des segments. Le tableau 5.2 indique le nombre de dimensions continues n ainsi que le nombre de subdivisions v pour la dimension discrète qui a été rajoutée.

On remarque clairement les gains (d'un facteur 3 pour les configurations en « V » et en « Z », de l'ordre de $1,5$ pour « Y ») obtenus dans la section § 4.1.2 si on compare les complexités calculées avec celles du tableau 5.1. On remarque également que les rapports entre les différentes configurations sont respectées par rapport à la figure FIG. 4.6, p. 83.

	M	$\tilde{D}$	η	n	v	$C_a^{'}$	k_i	Complexité $f_1 + f_2$	
V	12	526	2	2	4	12,23	$\begin{cases} k_\alpha = 24 \\ k_\rho = 20 \end{cases}$	$f_1 =$ $f_2 =$ $\rightarrow$	25725 6917 **32642**
Z	12	889	2	4	8	0,4164	$\begin{cases} k_{\alpha_1} = k_{\alpha_2} = 24 \\ k_{\rho_1} = k_{\rho_2} = 20 \end{cases}$	$f_1 =$ $f_2 =$ $\rightarrow$	5923 82 **6005**
Y	12	233	2	4	8	0,912	$\begin{cases} k_{\alpha_1} = k_{\alpha_2} = 24 \\ k_{\rho_1} = k_{\rho_2} = 20 \end{cases}$	$f_1 =$ $f_2 =$ $\rightarrow$	3397 6 **3403**
SSP	12	841	2	3	16	2,239	$\begin{cases} k_{\alpha_1} = 24 \\ k_{\rho_1} = k_{\rho_2} = 20 \end{cases}$	$f_1 =$ $f_2 =$ $\rightarrow$	15063 442 **15505**
SPP	12	2282	2	4	4	46,91	$\begin{cases} k_{\alpha_1} = k_{\alpha_2} = 24 \\ k_{\rho_1} = k_{\rho_2} = 20 \end{cases}$	$f_1 =$ $f_2 =$ $\rightarrow$	1712778 1085 **1713863**

TAB. 5.2: *Récapitulatif des principaux paramètres d'indexation pour des méthodes à dimensions discrètes.*

5.5.5 Conclusion

Les résultats expérimentaux en termes de temps d'exécution que nous avons obtenus dans les chapitres 3 et 4 se justifient parfaitement avec le développement fait dans ce chapitre. On observe notamment que l'interaction entre la dimension de l'espace d'indexation et le nombre de descripteurs sont les éléments principaux de la complexité totale d'une méthode de reconnaissance basée sur l'indexation.

5.6 Conclusion du chapitre

Nous avons, dans ce chapitre, fait l'étude algorithmique de la complexité sous-jacente des méthodes par indexation de descripteurs locaux. Nous avons vu que les modélisations qui entrent en jeu dans ces méthodes doivent mettre en œuvre une gestion du bruit qui ne nécessite pas de prendre en compte une région infinie afin de pouvoir bénéficier des avantages d'une indexation. D'un autre côté, nous avons montré que cette gestion du bruit peut, dans certains cas, introduire une explosion combinatoire du temps d'exécution des algorithmes lorsque les descripteurs évoluent dans un espace de grande dimension. Cette augmentation n'a pas lieu dans le cas où aucune gestion de bruit n'est nécessaire ou lorsque celle-ci n'augmente pas le nombre d'accès à la base d'indexation.

Globalement, la diminution du nombre de descripteurs dans une image réduit toujours le temps d'exécution, bien que ce ne soit pas d'une façon spectaculaire. L'augmentation systématique de la dimension des descripteurs n'a de sens que lorsqu'on indexe des valeurs discrètes disjointes mais peut être néfaste dans le cas où une gestion du bruit est

nécessaire. Dans ce cas, la meilleure solution consiste à utiliser des dimensions de taille raisonnable. Lorsque la dimension est trop élevée, les seules solutions envisageables sont soit une diminution de celle-ci (éventuellement par des outils statistiques tels que l'analyse en composantes principales), soit l'introduction d'une augmentation de la dimension de façon discrète.

Conclusion

Dans cette thèse nous avons développé une nouvelle méthode de reconnaissance par l'apparence, en modélisant des images segmentées par des configurations géométriques. En introduisant une caractérisation de ces configurations par le calcul de *quasi-invariants* nous avons mis en place un schéma d'indexation qui permet de émettre efficacement des hypothèses de mise en correspondance entre des configurations locales d'une image à reconnaître et celles de modèles connus dans une base indexée. Nous avons abordé, d'une part, la problématique liée à l'organisation et la structure d'un espace d'indexation, et d'autre part nous avons introduit une notion de vérification de cohérence globale entre les candidats plausibles parmi les modèles et l'image inconnue proposée.

L'introduction de cette vérification globale rend notre système robuste et permet également de généraliser notre schéma de reconnaissance à d'autres méthodes, dites locales. En effet, dès que ces méthodes s'appuient sur la notion de mise en correspondance de configurations « géométriques » entre les modèles et l'image, nous sommes en mesure de mettre en œuvre cette phase de vérification qui s'appuie sur le calcul d'une approximation du mouvement apparent entre images. Ce calcul est implémenté par un vote dans un espace de transformée de HOUGH. La pertinence de cette approximation sert alors comme mesure de confiance entre les modèles candidats. Le fait que les différentes approches locales puissent s'intégrer dans cette structure nous donne la possibilité d'introduire la notion de *coopération* entre méthodes, dès lors qu'elles sont en mesure de voter dans le même espace de transformation.

Nous avons analysé, dans un deuxième temps, la complexité algorithmique qui est sous-jacente aux méthodes qui s'appuient sur une indexation de caractéristiques locales. Dans cette analyse nous nous basons sur une méthode de reconnaissance abstraite qui s'appuie sur l'indexation de caractéristiques locales, comme le font la plupart des méthodes locales. Notre étude montre que dans le cas où il n'est pas nécessaire de prendre en compte une éventuelle incertitude autour des caractéristiques, l'indexation est un outil très performant, réduisant considérablement la complexité du problème de reconnaissance lorsque la taille des descripteurs est grande. Ce constat est d'ailleurs la justification de beaucoup d'auteurs pour l'utilisation d'une indexation. Par contre dès qu'il est nécessaire d'introduire la notion d'incertitude dans le processus d'indexation, la complexité du problème est modifiée, et une augmentation inconsidérée de la taille des descripteurs ne réduit pas le temps d'exécution dans tous les cas. Nous avons présenté deux compromis pour pallier ce problème. D'une part, le contrôle de la taille des descripteurs, en fonction de la complexité

des images considérées permet de limiter le temps d'exécution. D'autre part, si l'augmentation de la taille des descripteurs est inévitable, nous avons montré que l'introduction de dimensions dites « discrètes » permet de limiter l'augmentation de la complexité.

Contributions

Nous distinguons trois contributions principales dans cette thèse.

1° L'utilisation de *quasi-invariants* pour la reconnaissance est utilisée par différents auteurs. Pour notre part, nous avons introduit un nouveau schéma d'indexation et de vérification de la cohérence géométrique. Il nous permet de nous affranchir du faible pouvoir descriptif des *quasi-invariants* que nous utilisons et d'obtenir une méthode de reconnaissance basée sur l'apparence qui soit exploitable.

2° Guidés par le besoin de rendre la méthode développée plus robuste et utilisable dans le cas d'images réelles dans un environnement non contrôlé, nous nous sommes intéressés aux autres approches par apparence locale qui s'appuient sur une indexation. Nous en avons extrait un facteur commun, qui est la configuration géométrique, et nous l'avons intégré dans notre méthode de vérification globale, utilisée, jusque là, uniquement pour les quasi-invariants. Cette intégration nous a permis de proposer une méthode de reconnaissance et de mise en correspondance qui est utilisable dans un grand éventail de situations, et qui dépasse largement les modèles polyédriques que nous abordions avec la première méthode.

3° Dans un souci de découvrir les origines des limitations de notre méthode, notamment des besoins de ressources, nous avons montré formellement que l'indexation, telle qu'elle est actuellement utilisée dans un grand nombre d'approches ne constitue pas toujours la bonne solution. Dans le cas d'une gestion de bruit le complexité peut croître de façon exponentielle. Nous fournissons, néanmoins, deux méthodes permettant de partiellement contourner le problème et que nous avons appliquées à nos algorithmes.

Perspectives

Ce travail ouvre un grand champ de nouvelles investigations. Premièrement la notion de coopération peut être développée plus en profondeur, et la vérification globale peut être élargie à d'autres types de transformations : celles liées à la la luminosité, par exemple.

Une étude plus fondamentale du problème doit être entreprise. Il est clair que les évolutions actuelles en termes d'organisation des données se voient confrontées à l'augmentation des dimensions qui entrent en jeu. Ici le problème ne reste plus borné aux méthodes de reconnaissance locales, mais à toutes les approches par l'apparence.

Nous voyons trois grands axes de poursuite de ce travail :

Efficacité en termes de ressources : il est clair que le prototype de reconnaissance que nous avons développé pendant cette thèse avait un but d'observation et d'ana-

lyse des différents facteurs et processus entrant en jeu. De ce fait, l'implémentation, bien que rigoureuse, souffre d'un surpoids d'informations inutiles et redondantes à la seule tâche d'indexation et reconnaissance. Comme nous l'avons déjà indiqué, il doit être possible d'augmenter les performances d'un ordre de grandeur. Évidemment cette augmentation ne répondra pas à tous les besoins de stockage que l'on peut avoir, et n'atteint pas, et de loin, les quantités souhaitées dans des projets de grande envergure. Des poursuites de recherche dans la formulation des descripteurs, permettant d'être plus robuste, en intégrant des informations de couleur ou de texture par exemple, pourraient nous amener a considérer moins de descripteurs pour une même qualité de reconnaissance.

Collaboration de méthodes : notre ébauche de collaboration peut être étendue au delà des premiers tests concluants que nous avons effectués avec les invariants de SCHMID. Une étude approfondie des autres types d'indexation devra être effectuée pour intégrer d'autres types de descripteurs, et permettra éventuellement de répondre aux interrogations soulevées au point précédent.

Efficacité de stockage : avec les méthodes actuelles d'indexation, il devient incontournable d'avoir recours à des espaces de stockage secondaires pour enregistrer toutes les images et les données (index) qui leur sont associés. Outre les problèmes de temps d'accès élevés et d'éventuelles répartitions physiques des zones de stockage, il est nécessaire, au vu des résultats présentés dans le chapitre 5 d'étudier l'impact de ces structures sur les performances et de proposer des organisations de données adaptées aux besoins particuliers de ce type de problèmes.

Annexe A

Optimisations pour des arbres à profondeur finie

Cette annexe aborde la mise en œuvre d'un espace de stockage multidimensionnel utilisant des arbres à profondeur finie. Cette structuration cherche à fournir une structure discrète à accès en temps constant qui couvre potentiellement un vaste nombre de paniers de stockage, mais qui réduit les besoins en mémoire lorsque cette structure est creuse, tout en essayant de limiter le temps d'accès.

Motivation et idée générale de mise en œuvre

Le but que nous poursuivons est de simuler un tableau à n dimensions à des fins d'indexation. Il est clair que, pour de grandes dimensions, il est inefficace du point de vue de l'utilisation de la mémoire, d'utiliser le type `tableau` prédéfini[1]. Il existe beaucoup de structures à base d'arbres qui contournent ce problème, et qui intègrent une gestion dynamique de l'allocation de la mémoire. Nous avons malheureusement d'autres contraintes. Étant donné que les objets stockés dans notre structure sont accédés fréquemment et qu'ils comportent une notion d'incertitude, ce qui requiert un accès aux voisins, il faut que les temps d'accès soient raisonnables. De plus nous savons que des approches de partitionnement de l'espace sont préférables aux approches de partitionnement des données (*cf.* § 3.3.3).

Le concept de *quadtree* nous a servi comme point de départ de notre structure de stockage. Cette structuration des données présente un nombre d'avantages qui s'intègrent

1. Dans cette annexe, nous ne cherchons pas à introduire ou à utiliser une notation algorithmique abstraite. Nous nous contenterons de nous référer à des notations en `C` ou `C++` et les mises en œuvre que l'utilisation de ces langages impliquent.

bien avec nos besoins.

1° C'est une organisation « creuse » qui permet d'allouer la mémoire nécessaire au fur et à mesure des besoins. Il n'est donc pas nécessaire de disposer de grandes quantités de mémoire dès le départ.

2° Elle permet d'atteindre une granularité (ou précision) arbitraire, en fonction de la profondeur de l'arbre sous-jacent.

3° La structure se généralise facilement à n dimensions.

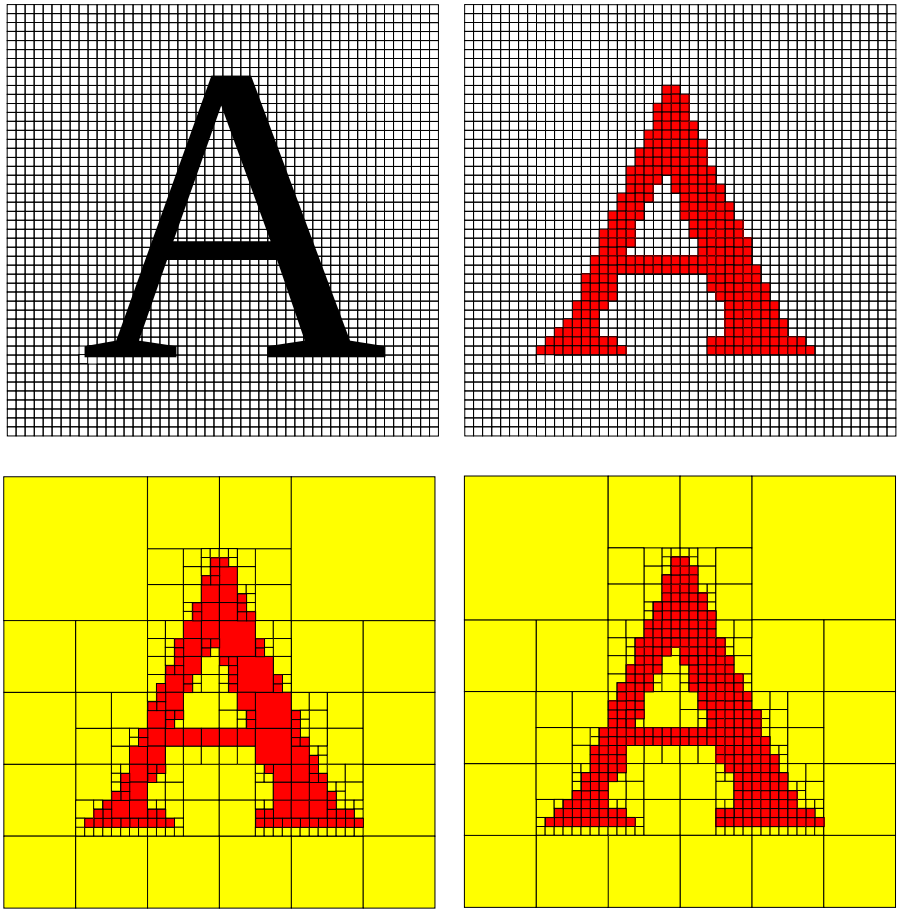


FIG. A.1: Quadtree *classique et* quadtree *à profondeur fixe.*

L'adaptation de cette structure pour l'indexation est immédiate : on considère les cases occupées comme des pixels noirs, et les vides comme des pixels blancs.

L'un des inconvénients de cette structure qui nous intéresse dans le contexte d'une indexation est le fait que l'accès aux voisins d'une feuille n'est pas trivial, et de plus, le temps d'accès n'est pas constant. Afin de pouvoir faciliter cet accès nous imposerons que l'arbre ait une profondeur fixe pour les feuilles occupées. Nous verrons dans la section traitant de l'accès aux données que cela facilite grandement l'implémentation et permet d'utiliser des structures binaires pour optimiser le calcul du chemin au voisin.

La figure FIG. A.1 montre la différence entre un *quadtree* à profondeur fixe et le *quadtree* classique. Afin de représenter un objet « continu » dans une structure discrète, on divise l'espace récursivement en zones vides et occupées, en s'arrêtant si une zone est entièrement homogène (vide ou pleine). C'est l'approche classique, et son résultat est montré dans la figure du bas à gauche. La figure de droite montre le résultat lorsqu'on impose que les feuilles occupées se trouvent toutes à la même profondeur. On préserve, dans ce cas, la structure creuse et dynamique, mais on détaille davantage le contenu de cases homogènes pleines.

Spécification de l'arbre de stockage

Nous donnerons ici nos besoins en termes d'indexation, et nous donnerons les grandes lignes d'implémentation par notre structure d'arbre.

Nous voulons indexer des objets de type `Objet` qui possèdent plusieurs dimensions `i` de 1 à `d`. Les valeurs des objets en chaque dimension sont bornées, les bornes sont connues, et nous disposons d'une fonction de calcul de clé d'indexation. Cette clé est un vecteur d'entiers non signés, à `d` dimensions. Suite aux remarques dans les sections § 3.3.3 et § 5.4.1, nous supposons que, quelle que soit la loi d'incertitude autour des descripteurs, l'incertitude autour des clés d'indexation possède un support uniforme, bien que ce ne soit pas une condition nécessaire. Les clés d'indexation sont également bornées, et on en connaît les bornes.

L'arbre d'indexation, de type `arbre`, est une structure récursive, ce qui veut dire que chaque nœud est un objet de type `arbre`. Les feuilles sont alors des `arbre` ne contenant plus de sous-arbre. Par rapport à la définition initiale des *quadtree*, notre type `arbre` présente deux différences majeures. Puisqu'il couvre un espace à `d` dimensions, un objet de type `arbre` partage l'espace en n suivant une seule des dimensions à chaque nœud, plutôt que de faire un partage en $2^{\mathtt{d}}$ sous-arbres comme le fait le *quadtree*. En plus, pour des raisons d'optimisation qui seront évoquées ultérieurement, n est une puissance entière de 2, et sa valeur peut varier suivant la dimension considérée, mais doit rester constante pour chaque subdivision selon cette dimension. Dans la figure FIG. A.2, cette différence est schématisée pour une figure simple. Pour subdiviser le petit damier, le *quadtree* utilise une étape ; les objets `arbre` le partagent d'abord selon une direction, puis ensuite dans l'autre.

Opérations d'accès

Les hypothèses que nous avons évoquées précédemment nous permettent d'avoir un accès très efficace au contenu de l'arbre. Notamment le fait que :

1. les index soient bornés,
2. les subdivisions par niveau soient des puissances de deux,
3. la profondeur soit fixe,

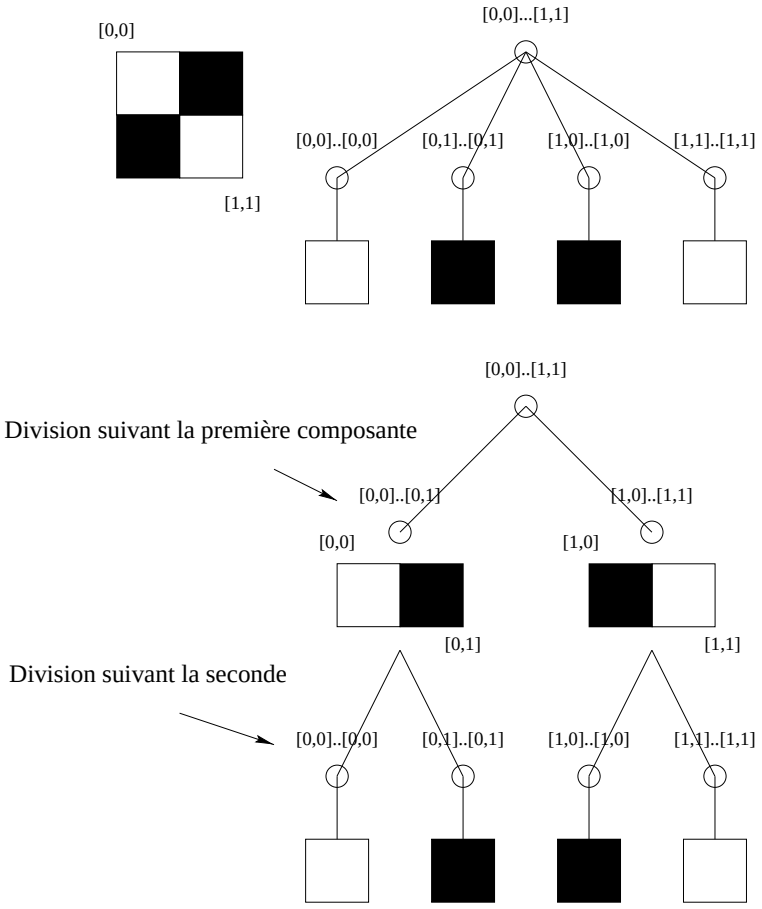


FIG. A.2: *Division de l'espace dimension par dimension.*

implique que les algorithmes suivants soient valables.

Accès à un élément donné par son index

Sachant que les `arbre` ont une profondeur fixe et que, à chaque subdivision on sépare l'espace des index en 2^n parties, on peut en conclure que les n premiers *bits* de l'index définissent le fils de la racine qui contiendra l'élément cherché. Grâce à la structure récursive, ceci reste valable pour les subdivisions suivantes. Il suffit alors de prendre à chaque niveau les n bits suivants pour savoir dans quelle branche l'élément est stocké. Il s'agit ici d'opérations binaires de très bas niveau, qui s'implémentent et s'exécutent de façon extrêmement efficace. L'algorithme A.1 détaille le schéma à suivre.

On peut de la même façon chercher le plus petit sous-arbre contant un intervalle de valeurs données, ce qui est utile lorsqu'on veut accéder à une zone indexée, plutôt qu'à une seule feuille. Dans ce cas l'algorithme A.1 s'adapte directement, et il suffit d'ajouter un test lors de chaque vérification de l'accessibilité de l'arbre courant pour qu'il couvre bien l'intervalle cherché.

Algorithme A.1 Accès à un élément dans un arbre à profondeur fixe

```
 /* Cette fonction retourne la feuille indexée par l'index fourni
   dans un arbre d'indexation, également fourni. Elle retourne 0 si la
   feuille n'existe pas */

Paramètres d'entrée : ARBRE, INDEX
Paramètres de sortie : FEUILLE

début

  shift = tableau contenant log2 du nombre de subdivisions pour
          chaque dimension;
  masque = tableau contenant, pour chaque dimension, les bits significatifs
           pour calculer le fils menant à la feuille cherchée;
  décalage = nombre de bits à décaler pour que (masque "et" INDEX) donne
             le numéro du fils menant à la feuille cherchée;

  arbre_courant = racine(ARBRE);

  tant que arbre_courant existe et n'est pas une feuille, faire
  début
      d = dimension que subdivise arbre_courant;
      n = (masque[d] & INDEX[d]) >> décalage[d]; /* opération de "et"
                                                    et décalage binaires */

      arbre_courant = n^ème fils de arbre_courant;

      masque[d] = masque[d] >> shift[d];
      décalage[d] = décalage[d]-shift[d];
  fin

  si arbre_courant existe alors
      retourner arbre_courant
  sinon
      retourner 0
fin
```

Coût moyen d'accès à une feuille

Le coût moyen d'accès à une feuille dans ce type de structure dépend évidemment du taux de remplissage de l'arbre. Lorsque l'arbre est plein, la profondeur moyenne d'une feuille est la profondeur maximale de l'arbre, et le coût de consultation est alors égal à l'accès à une feuille de cette profondeur. Inversement, lorsque l'arbre est vide, il est réduit à sa racine, et le temps d'accès est réduit à celui d'un simple test. On prendra donc comme coût moyen d'accès à une feuille, la profondeur moyenne des feuilles de l'arbre.

On se place dans le cas d'un k-arbre couvrant un espace de dimension n et de profondeur maximale p. On sait qu'à chaque niveau de l'arbre l'espace est récursivement subdivisé en k^n hypercubes. L'arbre atteint donc potentiellement $(k^n)^p$ feuilles. De même, un sous-arbre de profondeur r atteint potentiellement $(k^n)^{(p-r)}$ feuilles.

Soit I un indice de feuille atteignable dans l'arbre. On cherche à établir la probabilité qu'il faille descendre de r niveaux (exactement) dans l'arbre pour savoir si I correspond à une feuille pleine ou creuse. L'espérance qui y est associé donnera alors la profondeur moyenne pour accéder à une feuille.

On suppose que la probabilité qu'un indice I corresponde à une feuille pleine soit égale à τ.

La probabilité qu'une feuille f soit une feuille « vide », correspond à la probabilité que l'hypercube qu'elle définit soit vide. Par conséquent, elle correspond à la probabilité que tous les indices I couverts par f correspondent à des emplacements non occupés.

La probabilité qu'un sous-arbre de profondeur r soit vide est donc égale à :

$$P_r = (1-\tau)^{\left(k^{n(p-r)}\right)}$$

La probabilité que le chemin d'accès à un indice I soit de profondeur j est donc *la probabilité que le sous-arbre de profondeur j, menant à I, soit vide et que le sous-arbre de profondeur $j-1$, menant à I, ne soit pas vide.* On notera $P_I(j)$ cette probabilité. On peut d'ores et déjà écrire que

$$P_I(j) = P_j.P\left(j-1 \quad \text{non vide}|_{j \text{ vide}}\right)$$

La probabilité que le sous-arbre de profondeur $j-1$, menant à I, ne soit pas vide, sachant que l'un de ses fils est vide, est égale à *un moins la probabilité que tous les autres fils du nœud soient vides.* Ceci nous permet alors d'écrire :

$$P_I(j) = P_j.\left(1-\left((1-\tau)^{k^{n(p-j)}}\right)^{k^n-1}\right)$$

Cette formule n'est valable que dans le cas général. Pour les cas où $j=0$ ou $j=p$ la formule est modifiée. La probabilité que la racine soit également une feuille s'écrit :

$$P_I(0) = (1-\tau)^{k^{np}}$$

Cette probabilité correspond au fait que tous les indices correspondent à des feuilles vides.

Pour le cas où $j = p$, il faut également prendre en compte les feuilles pleines, et non seulement les feuilles vides comme dans le cas général. Un indice I correspond à une feuille de profondeur p dès lors que son parent n'est pas vide, et est indépendant de si la feuille en question est pleine ou vide. On obtient alors :

$$P_I(p) = 1 - (1 - \tau)^{k^n}$$

L'espérance $\overline{P_I}$ s'exprime donc comme suit :

$$\overline{P_I} = \left(\sum_{j=1}^{p-1} j \ (1-\tau)^{\left(k^{n(p-j)}\right)} \ \left(1 - \left((1-\tau)^{\left(k^{n(p-j)}\right)} \right)^{(k^n - 1)} \right) \right) + p \ \left(1 - (1-\tau)^{k^n} \right)$$

Ce polynôme en τ de degré dépendant de p et de k^n est strictement croissant pour $\tau \in [0..1]$ et croît de 0 à p, avec une tangente de 0 pour $\tau = 1$. La tangente en $\tau = 0$ est :

$$-\frac{k^n \left(k^n - 1 + k^{np} - k^{n(1+p)} \right)}{\left(k^n - 1 \right)^2}$$

et croît rapidement quand k^n ou p augmentent. Les graphes dans la figure FIG. A.3 montrent quelques exemples avec des valeurs numériques concrètes. On constate que la fonction croît d'autant plus vite vers p que les paramètres p et k^n sont élevés. Il reste à noter, tout de même, que dans ce genre de structure, plus p et k^n sont élevés, plus τ devient petit, donc la profondeur moyenne de l'arbre aussi. Si l'on fait un changement de variable en posant $\tau = \frac{D}{k^n p}$, et si on prend $p = 6 \quad , k = 8 \quad , n = 3$ on observe dans la figure FIG. A.4 que la profondeur moyenne se situe autour de 2.

Accès à un voisin

L'accès aux voisins directs est important pour une structure d'indexation telle que nous la concevons. La structure arborescente est malheureusement mal adaptée pour ce genre de requêtes (*cf.* § 5.3). Dans le cas où l'espace d'indexation est vaste et de haute dimension, les objets de type `arbre` peuvent avoir une profondeur non négligeable, rendant l'accès aux feuilles depuis la racine coûteux. Dans ce cas, il peut être intéressant d'accéder aux voisins d'une feuille à partir de cette même feuille.

Supposons que nous ayons un arbre k-aire en n dimensions (chaque nœud correspond à une division en k^n de l'espace). On se propose de calculer le coût d'accès à un voisin d'une feuille. On peut considérer, par symétrie du problème, que l'on accède à un voisin dans le sens « positif » pour chaque composante, et que l'éloignement de recherche est de 0 ou de 1 en chaque dimension.

La probabilité que ce voisin soit accessible directement par le père de la feuille considérée est égale à :

$$\left(1 - \frac{1}{k} \right)^n$$

puisque pour chaque dimension (que nous avons prises indépendantes), la probabilité qu'une feuille puisse accéder à son voisin dans celle-ci est égale à $\left(1 - \frac{1}{k}\right)$.

De façon équivalente, la probabilité qu'en remontant à p pères on puisse accéder au voisin est de :

$$\left(1 - \frac{1}{k^p}\right)^n \approx \left(1 - \frac{n}{k^p}\right)$$

avec, bien-sûr, p inférieur à la profondeur de l'arbre.

Le coût d'accès à un voisin, est égal à $2p$, où p représente le nombre de nœuds qu'il est nécessaire de remonter pour être en mesure de redescendre au voisin recherché. Nous rappelons que, dans le cas d'un accès par la racine, ce coût est constant, et est égal à la profondeur de l'arbre, tandis qu'ici, le coût varie entre 2 et $2\times$ la profondeur.

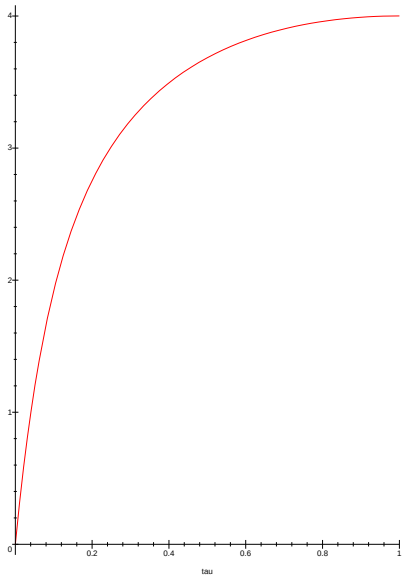
Or, si on prend un arbre 8-aire, en dimension 3 par exemple ($k = 8$ et $n = 3$), on observe que la probabilité de n'avoir un coût égal à 2 ($p = 1$), est égal à :

$$\left(1 - \frac{1}{8}\right)^3 = 0,67$$

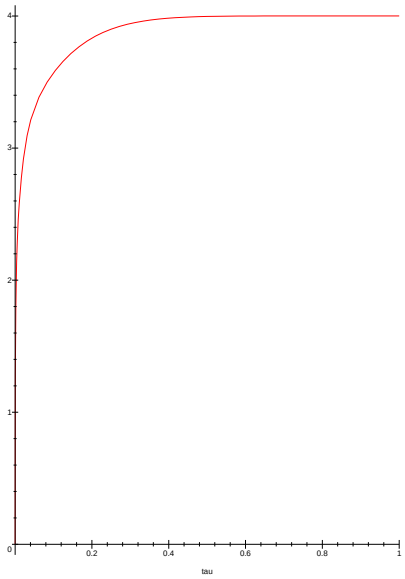
et que la probabilité d'avoir un coût égal à 4 ($p = 2$) est égal à

$$\left(1 - \frac{1}{64}\right)^3 = 0,95$$

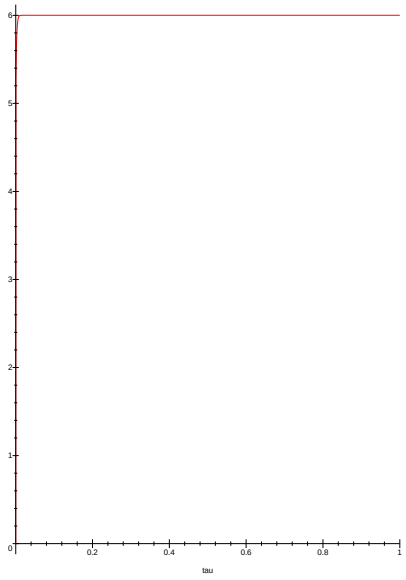
Dans un arbre de profondeur suffisante (dans le cas présent, il faut qu'il soit au moins de profondeur 4, et donc, qu'il atteint potentiellement 4096^3 cases) il est donc judicieux d'accéder aux voisins par les feuilles, et non par la racine.



$p = 4 \quad k = 2 \quad n = 1$



$p = 4 \quad k = 2 \quad n = 3$



$p = 6 \quad k = 8 \quad n = 3$

FIG. A.3: *Évolution du coût d'accès moyen à une feuille pour un* `arbre` *à plusieurs dimension et profondeurs.*

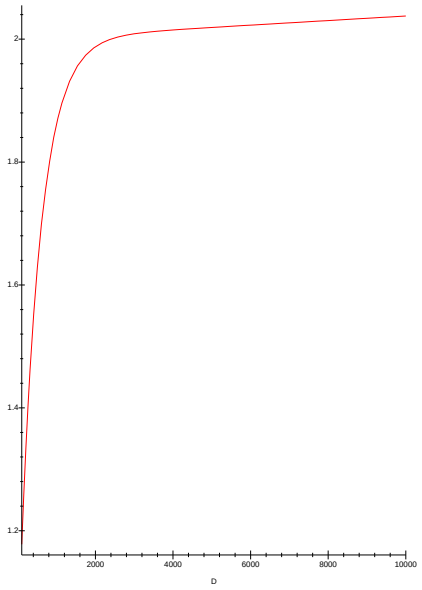


FIG. A.4: *Évolution du coût d'accès moyen à une feuille pour un* `arbre` *en fonction du nombre de feuilles pleines.*

Annexe B

Gestion du bruit pour l'indexation

Dans cette annexe nous complétons les calculs menant aux résultats énoncés dans la section 5.2.3.

B.1 Rapport du volume d'une boule et d'un hypercube

Soient $\mathcal{B}$ et $\mathcal{C}$ l'hypervolume de respectivement une boule de rayon r et un hypercube de côtés de longueur r dans un espace à n dimensions. Ils vérifient les équations suivantes :

$$\mathcal{B} = \frac{r^n.\pi^{\left(\frac{n}{2}\right)}}{\left(\frac{n}{2}!\right)} \tag{B.1}$$

$$\mathcal{C} = (2r)^n \tag{B.2}$$

Le rapport des deux volumes donne une mesure de l'erreur relative, commise en approchant une boule par un hypercube, comme il est courant de faire en indexation.

$$\begin{aligned} \frac{\mathcal{B}}{\mathcal{C}} &= \frac{\frac{r^n.\pi^{\left(\frac{n}{2}\right)}}{\left(\frac{n}{2}!\right)}}{(2r)^n} \\ &= \frac{r^n\sqrt{\pi}^n}{r^n\left(\frac{n}{2}!\right)2^n} \\ &= \frac{\left(\frac{\sqrt{\pi}}{2}\right)^n}{\left(\frac{n}{2}!\right)} \end{aligned} \tag{B.3}$$

On remarque facilement que

$$\begin{aligned} \lim_{n\to\infty} \left(\frac{\sqrt{\pi}}{2}\right)^n &= 0 \quad \text{puisque} \quad \frac{\sqrt{\pi}}{2} < 1 \\ \lim_{n\to\infty} \left(\frac{n}{2}!\right) &= \infty \end{aligned}$$

Par voie de conséquence,

$$\lim_{n\to\infty} \frac{\mathcal{B}}{\mathcal{C}} = 0 \tag{B.4}$$

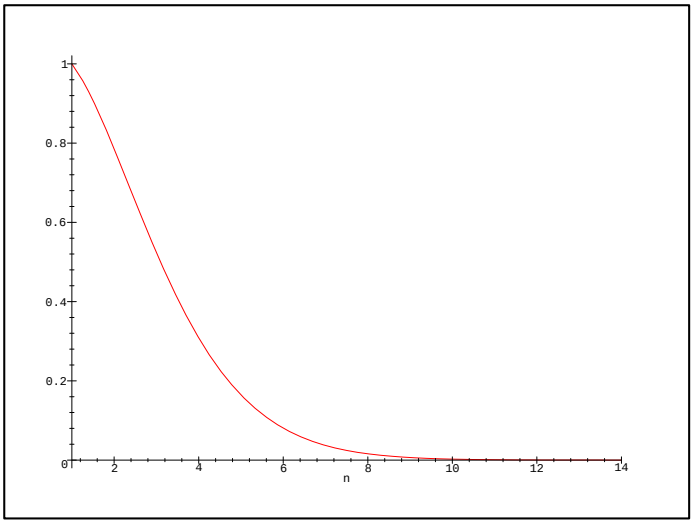


FIG. B.1: *Évolution avec n du rapport des volumes d'une boule et d'un hypercube.*

Annexe C

Optimisation dans le cas d'une indexation bruitée

C.1 Calcul de n_{min}

Cette partie décrit le développement complet des calculs menant au résultat énoncé dans l'équation (5.8), p. 129. On cherche à trouver n_{min} telle que $\frac{\mathrm{df}}{\mathrm{d}n}(n_{min}) = 0$

$$\frac{\mathrm{df}}{\mathrm{d}n} = \eta^n \ln(\eta)\tilde{D} - \frac{\tilde{D}^2 M \ln\left(\frac{k}{\eta}\right)}{\left(\frac{k}{\eta}\right)^n}$$

$$\eta^n \ln(\eta)\tilde{D} - \frac{\tilde{D}^2 M \ln\left(\frac{k}{\eta}\right)}{\left(\frac{k}{\eta}\right)^n} = 0$$

$$\eta^n \ln(\eta)\tilde{D} = \frac{\tilde{D}^2 M \ln\left(\frac{k}{\eta}\right)}{\left(\frac{k}{\eta}\right)^n}$$

$$\eta^n \ln(\eta) = \frac{\tilde{D} M \ln\left(\frac{k}{\eta}\right)}{\left(\frac{k}{\eta}\right)^n}$$

$$\frac{\ln(\eta)}{\tilde{D}M\ln\left(\frac{k}{\eta}\right)} = \left(\frac{\eta}{k}\right)^n \frac{1}{\eta^n}$$

1° $\eta > 1$. Nous sommes dans le cas de figure d'indexation en présence de bruit, et η est supérieur à 1.

$$\frac{\ln(\eta)}{\tilde{D}M\ln\left(\frac{k}{\eta}\right)} = \frac{1}{k^n}$$

$$k^n = \frac{\tilde{D}M\ln\left(\frac{k}{\eta}\right)}{\ln(\eta)}$$

$$n = \log_k\left(\frac{\tilde{D}M\ln\left(\frac{k}{\eta}\right)}{\ln(\eta)}\right)$$

2° $\eta = 1$. Dans le cas d'indexation sans bruit, la partie gauche de l'équation s'annule sous l'effet de $\ln(\eta) = 0$, et la condition sur n devient

$$0 = \frac{1}{k^n}$$

Ce qui imposerait $n = \infty$. On en déduit que dans le cas sans bruit il n'existe pas de n optimal.

C.2 Calcul de f_{min}

Cette partie décrit le développement complet des calculs menant au résultat énoncé dans l'équation (5.10), p. 130.

Nous rappelons d'abord les trois équations de départ, donnant la fonction f de complexité, la valeur de n minimisant f et l'expression de son minimum en n.

$$\mathrm{f} = \eta^n.\tilde{D} + \frac{\tilde{D}^2.M}{\left(\frac{k}{\eta}\right)^n}$$

$$n_{min} = \log_k\left(\tilde{D}M\left(\frac{\ln(k)}{\ln(\eta)} - 1\right)\right)$$

$$\mathrm{f}_{min} = \eta^{\left(\frac{\ln\left(\tilde{D}M\left(\frac{\ln(k)}{\ln(\eta)}-1\right)\right)}{\ln(k)}\right)}\tilde{D} + \frac{\tilde{D}^2M}{\left(\frac{k}{\eta}\right)^{\left(\frac{\ln\left(\tilde{D}M\left(\frac{\ln(k)}{\ln(\eta)}-1\right)\right)}{\ln(k)}\right)}}$$

Avec la notation donnée p. 130 telle que :

$$x = \frac{\ln(k)}{\ln(\eta)}$$

on réécrit f_{min} pour obtenir :

$$\begin{aligned}
\mathrm{f}_{min} &= \eta^{\left(\frac{\ln\left(\tilde{D}M(x-1)\right)}{\ln(k)}\right)}\tilde{D} + \tilde{D}^2 M \frac{\eta^{\left(\frac{\ln\left(\tilde{D}M(x-1)\right)}{\ln(k)}\right)}}{k^{\left(\frac{\ln\left(\tilde{D}M(x-1)\right)}{\ln(k)}\right)}} \\
&= e^{\left(\frac{\ln\left(\tilde{D}M(x-1)\right)}{x}\right)}\tilde{D} + \tilde{D}^2 M \frac{e^{\left(\frac{\ln\left(\tilde{D}M(x-1)\right)}{x}\right)}}{e^{\ln\left(\tilde{D}M(x-1)\right)}} \\
&= e^{\left(\frac{\ln\left(\tilde{D}M(x-1)\right)}{x}\right)}\tilde{D} + \frac{\tilde{D}^2 M}{\tilde{D}M(x-1)} e^{\left(\frac{\ln\left(\tilde{D}M(x-1)\right)}{x}\right)} \\
&= e^{\left(\frac{\ln\left(\tilde{D}M(x-1)\right)}{x}\right)}\tilde{D}\frac{x}{x-1} \\
&= \left(\tilde{D}M(x-1)\right)^{\frac{1}{x}}\tilde{D}\frac{x}{x-1} \\
&= \tilde{D}^{\frac{x+1}{x}} M^{\frac{1}{x}} \frac{x}{x-1}(x-1)^{\frac{1}{x}} \\
&= \left(\frac{x}{x-1} e^{\frac{\ln(x-1)}{x}}\right)\tilde{D}^{\frac{x+1}{x}} M^{\frac{1}{x}}
\end{aligned}$$

C.3 Étude de $K = \frac{x}{x-1} e^{\frac{\ln(x-1)}{x}}$

Afin de connaître le comportement de K sur l'intervalle $]1, \infty[$, on en calcule la dérivée que l'on cherche à annuler ensuite. Les résultats sont utilisés à la fin du paragraphe 5.4.6.4, p. 130.

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}x}\left(e^{\frac{\ln(x-1)}{x}}\frac{x}{x-1}\right) &= \frac{\mathrm{d}}{\mathrm{d}x}\left(e^{\frac{\ln(x-1)}{x}}\right)\frac{x}{x-1} + e^{\frac{\ln(x-1)}{x}}\frac{\mathrm{d}}{\mathrm{d}x}\left(\frac{x}{x-1}\right) \\
&= \frac{\mathrm{d}}{\mathrm{d}x}\left(\frac{\ln(x-1)}{x}\right)e^{\frac{\ln(x-1)}{x}}\frac{x}{x-1} + e^{\frac{\ln(x-1)}{x}}\left(\frac{1}{x-1} - \frac{x}{(x-1)^2}\right) \\
&= \left(\frac{1}{(x-1)x} - \frac{\ln(x-1)}{x^2}\right)e^{\frac{\ln(x-1)}{x}}\frac{x}{x-1} + e^{\frac{\ln(x-1)}{x}}\left(\frac{1}{x-1} - \frac{x}{(x-1)^2}\right)
\end{aligned}$$

Nous cherchons maintenant les racines de cette expression.

$$0 = \left(\frac{1}{(x-1)\,x} - \frac{\ln(x-1)}{x^2}\right) e^{\frac{\ln(x-1)}{x}} \frac{x}{x-1} + e^{\frac{\ln(x-1)}{x}} \left(\frac{1}{x-1} - \frac{x}{(x-1)^2}\right)$$

On peut simplifier par $e^{\frac{\ln(x-1)}{x}}$ car $x > 1$

$$\begin{aligned} 0 &= \left(\frac{1}{(x-1)\,x} - \frac{\ln(x-1)}{x^2}\right) \frac{x}{x-1} + \frac{1}{x-1} - \frac{x}{(x-1)^2} \\ 0 &= \frac{x}{(x-1)^2\,x} - \frac{\ln(x-1)\,x}{(x-1)\,x^2} + \frac{1}{x-1} - \frac{x}{(x-1)^2} \\ 0 &= \frac{x - (x-1)\ln(x-1) + (x-1)\,x - x^2}{(x-1)^2\,x} \end{aligned}$$

On peut simplifier par $(x-1)^2$ et x car $x > 1$

$$\begin{aligned} 0 &= x - (x-1)\ln(x-1) + (x-1)\,x - x^2 \\ 0 &= (x-1)\left(-\ln(x-1) + x - x\right) \\ 0 &= \ln(x-1) \\ x &= 2 \end{aligned}$$

K a donc un seul extremum local, en 2. En étudiant le signe de $\frac{\mathrm{d}K}{\mathrm{d}x}$ on s'aperçoit qu'il s'agit d'un maximum. Nous étudions ensuite les limites de la fonction pour savoir si elle reste bornée sur l'intervalle considéré.

$$\begin{aligned} \lim_{x\to 1} K &= \lim_{x\to 1} \left(\frac{x}{x-1} e^{\frac{\ln(x-1)}{x}}\right) \\ &= \lim_{x\to 1} \left(\frac{x}{x-1} (x-1)^{\frac{1}{x}}\right) \\ &= \lim_{x\to 1} \left(x\,(x-1)^{\frac{x-1}{x}}\right) \\ &= \lim_{y\to 1} (y^y) \\ &= 1 \quad \text{(par continuité)} \end{aligned}$$

$$\begin{aligned} \lim_{x\to\infty} K &= \lim_{x\to 1} \left(\frac{x}{x-1} e^{\frac{\ln(x-1)}{x}}\right) \\ &= e^{\left(\lim_{x\to\infty} \frac{\ln(x-1)}{x}\right)} \\ &= 1 \end{aligned}$$

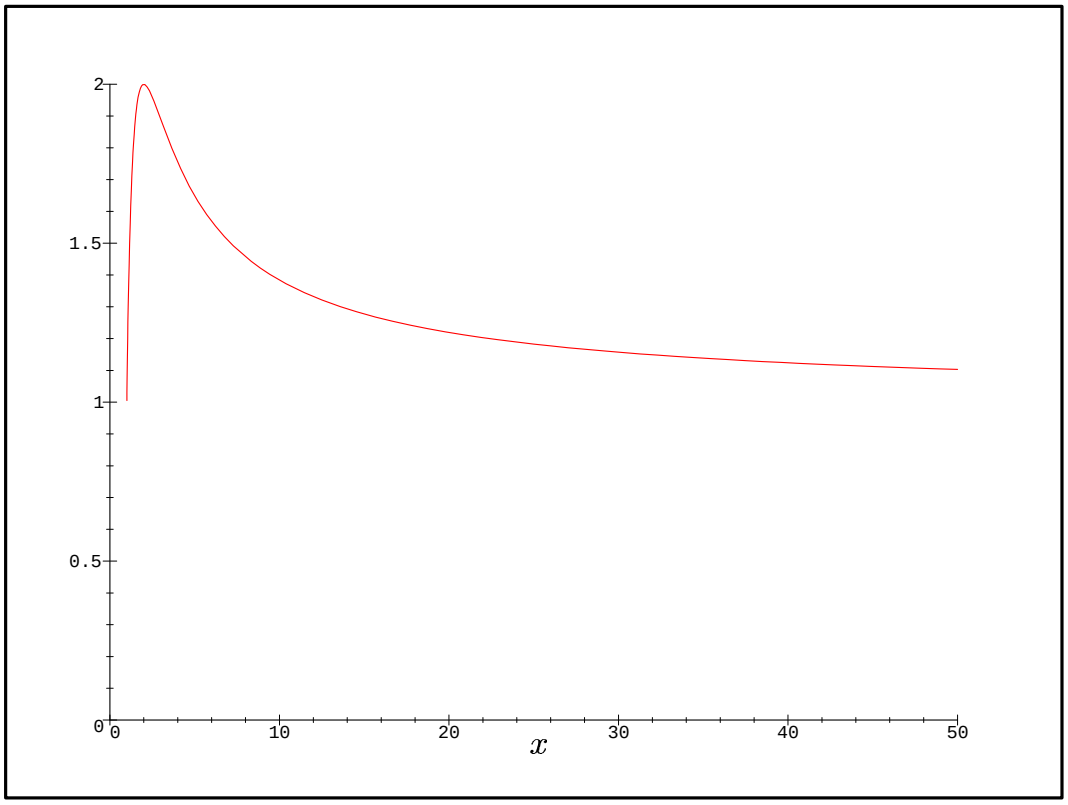


FIG. C.1: *Tracé de* $e^{\frac{\ln(x-1)}{x}} \frac{x}{x-1}$.

Bibliographie de l'auteur

Articles dans des revues

- « *Object Pose: The Link between Weak Perspective, Paraperspective and Full Perspective* », R. Horaud, F. Dornaika, B. Lamiroy et S. Christy (1997) dans « International Journal of Computer Vision », No. 2, pp. 173-189

Articles dans des conférences internationales avec comité de lecture

- « *Object Indexing is a Complex Matter* », B. Lamiroy et P. Gros (juin 1997) dans « Proceedings of the 10th Scandinavian Conference on Image Analysis », Lappeenranta, Finlande, Vol. I, pp. 277-283

- « *Rapid Object Indexing and Recognition Using Enhanced Geometric Hashing* », B. Lamiroy et P. Gros (avril 1996) dans « Proceedings of the 4th European Conference on Computer Vision », Cambridge, Angleterre, Vol. 1, pp. 59-70

- « *An Image Oriented CAD Approach* », C. Schmid, Ph. Bobet, B. Lamiroy et R. Mohr (1996) dans « Proceedings of the eccv workshop on Object Representation »

- « *Object Pose: Links Between Paraperspective and Perspective* », R. Horaud, S. Christy, F. Dornaika et B. Lamiroy (juin 1995) dans « Proceedings of the 5th International Conference on Computer Vision », Cambridge, Massachusetts, États-Unis, pp. 426-433

Articles lors de séminaires nationaux et internationaux

- « *Indexation et recherche d'images* », R. Mohr, P. Gros, B. Lamiroy, S. Picard et C. Schmid (septembre 1997) dans « Actes du 16^{e} colloque gretsi sur le traitement du signal et des images », Grenoble, France

- « *Computer Aided (dis)Assembly Using Visual Cues* », B. Lamiroy, C. Schmid, R. Mohr, M. Tonko, K. Schäfer et H.-H. Nagel (novembre 1996) dans « Proceedings of the iar Annual Meeting », Karlsruhe, Allemagne

- « *Reconnaissance d'objets par indexation géométrique étendue* », B. Lamiroy et P. Gros (mai 1996) aux Journées orasis 1996, Clermont-Ferrand, France, pp. 19-24

Rapports

- « *Reconnaissance d'objets polyédriques à l'aide d'invariants projectifs* », B. LAMIROY, Rapport de DEA, 1994
- « *Mise en correspondance dense de deux images par corrélation* », B. LAMIROY, Rapport de Magistère, 1993

Références bibliographiques

[1] Aristote. $\Pi\epsilon\rho\grave{\iota}\ \Psi\upsilon\chi\tilde{\eta}\varsigma$ - de Anima. 384 à 322 av. JC. (p 26)

[2] K. Åström. Affine and projective normalization of planar curves and regions. In Jan-Olof Eklundh, editor, *Proceedings of the 3rd European Conference on Computer Vision, Stockholm, Sweden*, pages 439–448. Springer-Verlag, May 1994. (pp 51, 52)

[3] N. Ayache and O.D. Faugeras. HYPER: a new approach for the recognition and positioning of 2D objects. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 8(1):44–54, 1986. (pp 28, 34, 35, 37)

[4] R. Bajcsy and A.K. Joshi. A partially ordered world model and natural outdoor scenes. In A.R. Hanson and E.M. Riseman, editors, *Computer Vision Systems*, pages 263–270. Academic Press, New York, USA, 1978. (p 34)

[5] D.H. Ballard. Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2):111–122, 1981. (p 53)

[6] G. Bebis, M. Georgiopoulos, and N. da Vitoria Lobo. Learning geometric hashing functions for model-based object recognition. In *Proceedings of the 5th International Conference on Computer Vision, Cambridge, Massachusetts, USA*, pages 543–548. IEEE, June 1995. (p 58)

[7] P.N. Belhumeur and D.J. Kriegman. What is the set of images of an object under all possible lighting conditions? In *Proceedings of the Conference on Computer Vision and Pattern Recognition, San Francisco, California, USA*, pages 270–277, June 1996. (p 39)

[8] J. Ben-Arie. The probabilistic peaking effect of viewed angles and distances with application to 3D object recognition. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 12(8):760–774, August 1990. (pp 51, 61, 62, 66)

[9] S. Berchtold, D.A. Keim, and H.P. Kriegel. The X-tree: An index structure for high-dimensional data. In *Proceedings of the 22nd VLDB Conference, Mumbai (Bombay), India*, pages 28–39. the Very Large Database Endowment, 1996. (pp 114, 132)

[10] I. Biederman. On the semantics of a glance at a scene. In M. Kubovy and J.K. Pomerantz, editors, *Perceptual Organization*, chapter 8, pages 213–255. Erlbaum, Hillsdale, N.J., 1981. (p 35)

[11] I. Biederman. Human image understanding: Recent research. *Computer Vision, Graphics and Image Processing*, 32:29–73, 1985. (p 34)

[12] I. Biederman. Recognition-by-components: a theory of human image understanding. *Psychological Review*, 94:115–147, 1987. (pp 27, 28)

[13] I. Biederman. Higher-level vision. In Hollerbach Osherson, Kosslyn, editor, *An Invitation to Cognitive Science*, chapter 2, pages 41–72. The MIT Press, Cambridge, MA, USA, 1990. (p 27)

[14] T.O. Binford and T.S. Levitt. Quasi-invariants: Theory and exploitation. In *Proceedings of* DARPA *Image Understanding Workshop*, pages 819–829, 1993. (pp 51, 52)

[15] S. Blott and R. Weber. A simple vector-approximation file for similarity in high-dimensional vector spaces. Technical Report 19, ESPRIT project HERMES (no. 9141), March 1997. Postscript version available by `ftp`[1]. (pp 114, 132)

[16] R.C. Bolles and R. Horaud. 3DPO : A three-dimensional Part Orientation system. *The International Journal of Robotics Research*, 5(3):3–26, 1986. (pp 28, 34, 35, 37)

[17] J.B. Burns, R. Weiss, and E.M. Riseman. View variation of point set and line segment features. In *Proceedings of* DARPA *Image Understanding Workshop, Pittsburgh, Pennsylvania, USA*, pages 650–659, 1990. (pp 51, 52)

[18] A. Califano and R. Mohan. Multidimensional indexing for recognizing visual shapes. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 16(4):373–392, April 1994. (pp 120, 126, 127)

[19] S. Carlsson. Combinatorial geometry for shape representation and indexing. In J. Ponce, A. Zisserman, and M. Hebert, editors, *Proceedings of the ECCV'96 International Workshop on Object Representation in Computer Vision II, Cambridge, England*, Lecture Notes in Computer Science, pages 53–78. Springer-Verlag, April 1996. (pp 81, 125)

[20] T.A. Cass. Polynomial-time geometric matching for object recognition. *International Journal of Computer Vision*, 1(21):37–61, 1997. (p 66)

[21] C.H. Chen and P.G. Mulgaonkar. CAD-based feature-utility measures for automatic vision programming. In *Direction in Automated CAD-Based Vision*, pages 106–114. IEEE Computer Society Press, 1991. (pp 33, 35, 37)

1. http://www-dbs.ethz.ch/~weber/paper/VAFILE.ps.gz

[22] D.J. Clemens and D.W. Jacobs. Model-group indexing for recognition. In *Proceedings of* DARPA *Image Understanding Workshop, Pittsburgh, Pennsylvania, USA*, pages 604–613, September 1990. (p 51)

[23] S. Covey. *Seven Habits of Highly Effective People.* Simon & Schuster, New York, 1989. (p 23)

[24] G. Csurka. *Modelisation projective des objets tridimensionnels en vision par ordinateur.* Thèse de doctorat, Université de Nice – Sophia Antipolis, April 1996. (p 52)

[25] S.J. Dickinson, R. Bergevin, I. Biederman, J.O. Eklund, R. Munck-Fairwood, and A. Pentland. The use of geons for generic 3D object recognition. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence, Chambery, France*, pages 1693–1699, 1993. (p 34)

[26] B.A. Draper, R.T. Collins, J. Brolio, A.R. Hanson, and E. Riseman. The schema system. *International Journal of Computer Vision*, 2:209–250, 1989. (pp 32, 33, 34, 35)

[27] B.A. Draper and E.M. Riseman. Learning 3D object strategies. In *Proceedings of the 3rd International Conference on Computer Vision, Osaka, Japan*, pages 320–324, 1990. (p 34)

[28] H. Dreyfus. *From Micro-Worlds to Knowledge Representation: AI at an Impasse.* The MIT Press, Cambridge, MA, USA, 1981. (p 18)

[29] O. Faugeras. *Three-Dimensional Computer Vision - A Geometric Viewpoint.* Artificial intelligence. The MIT Press, Cambridge, MA, USA, Cambridge, MA, 1993. (p 28)

[30] O. Faugeras, J. Mundy, N. Ahuja, C. Deyer, A. Pentland, R. Jain, and K. Ikeuchi. Why aspect graphs are not (yet) practical for computer vision. *Computer Vision, Graphics and Image Processing: Image Understanding*, 55(2):212–218, 1992. (p 43)

[31] M. R. Garey and D. S Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completness.* W. H. Freeman, San Francisco, 1979. (p 34)

[32] E. Gmur and H. Bunke. 3D object recognition based on subgraph matching in polynomial time. In R. Mohr and T. Pavlidis, editors, *Structural Pattern Analysis*, pages 131–148. World Scientific Pub., 1990. (p 34)

[33] L. Grewe and A.C. Kak. Interactive learning of a multiple-attribute hash table classifier for fast object recognition. *Computer Vision and Image Understanding*, 61(3):387–416, May 1995. (p 58)

[34] W.E.L. Grimson and D.P. Huttenlocher. On the sensitivity of the Hough transform for object recognition. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 12(3):225–274, March 1990. (pp 54, 62, 66)

[35] P. Gros. *Outils géométriques pour la modélisation et la reconnaissance d'objets polyédriques.* Thèse de doctorat, Institut National Polytechnique de Grenoble, July 1993. (pp 22, 40, 45, 52, 56, 67)

[36] P. Gros. Matching and clustering: Two steps towards object modelling in computer vision. *The International Journal of Robotics Research*, 14(6):633–642, December 1995. (pp 48, 55)

[37] P. Gros, O. Bournez, and E. Boyer. Using local planar geometric invariants to match and model images of line segments. *Computer Vision and Image Understanding*, 69(2):135–155, 1998. (pp 48, 55)

[38] A. Guttman. R-trees: a dynamic index structure for spatial searching. In M. Stonebraker, editor, *Readings in Database Systems*, chapter 2, pages 125–135. Morgan Kaufman, 1988. (pp 58, 114)

[39] J. Hafner, H.S. Sawhney, W. Equitz, M. Flickner, and W. Niblack. Efficient color histogram indexing for quadratic form distance functions. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 17(7):729–736, July 1995. (p 41)

[40] A.R. Hanson and E.M. Riseman. Visions: a computer system for interpreting scenes. In A.R. Hanson and E.M. Riseman, editors, *Computer Vision Systems*, pages 303–334. Academic Press, New York, USA, 1978. (pp 33, 34, 35)

[41] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey Vision Conference*, pages 147–151, 1988. (p 91)

[42] L. Hérault. *Réseaux de neurones récursifs pour l'optimisation combinatoire.* Thèse de doctorat, Institut National Polytechnique de Grenoble, France, 1991. (p 34)

[43] S. Herbin. *Éléments pour la formalisation d'une reconnaissance active. Application à la vision tri-dimensionnelle.* PhD thesis, École Normale Supérieure de Cachan, July 1997. (p 42)

[44] R. Horaud and O. Monga. *Vision par ordinateur : outils fondamentaux.* Éditions Hermès, Paris, 1993. (p 34)

[45] R. Horaud, F. Veillon, and Th. Skordas. Finding geometric and relational structures in an image. In O. Faugeras, editor, *Computer Vision – ECCV 90, Proceedings First European Conference on Computer Vision, Antibes, France*, pages 374–384. Springer Verlag, April 1990. (p 51)

[46] J. Hornegger and H. Niemann. Statistical learning, localization, and identification of objects. In *Proceedings of the 5th International Conference on Computer Vision, Cambridge, Massachusetts, USA*, pages 914–919, June 1995. (p 42)

[47] P.V.C. Hough. A method and means for recognition complex patterns. *U.S. Patent*, 1962. (p 53)

[48] M.P. Howell and P.J. Flynn. Guaranteed geometric hashing. In *Proceedings of the 12th International Conference on Pattern Recognition, Jerusalem, Israel*, pages 465–469, 1994. (p 113)

[49] D.P. Huttenlocher, G.A. Klanderman, and W.J. Rucklidge. Comparing images using the Hausdorff distance. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 15(9):850–863, September 1993. (p 41)

[50] D.P. Huttenlocher and W.J. Rucklidge. A multi-resolution technique for comparing images using the Hausdorff distance. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, New York, USA*, pages 705–706, 1993. (p 41)

[51] K. Ikeuchi. Generating an interpretation tree from a CAD model for 3D object recognition in binpicking tasks. *International Journal of Computer Vision*, pages 145–165, 1987. (pp 37, 56)

[52] K. Ikeuchi and T. Kanade. Automatic generation of object recognition programs. *Proceedings of the* IEEE, 76(8):1016–1035, August 1988. (pp 37, 56)

[53] A. Kalvin, E. Schomberg, J.T. Schwartz, and M. Sharir. Two-dimensional model-based boundary matching using footprints. *The International Journal of Robotics Research*, 5(4):38–54, 1986. (p 40)

[54] J. Koenderink and A.V. Doorn. The internal representation of solid shape with respect to vision. *Biological Cybernetics*, 32:211–216, 1979. (pp 25, 31)

[55] J.J. Koenderink. *Solid Shape*. The MIT Press, Cambridge, MA, USA, 1990. (p 43)

[56] J.J. Koenderink and A.J. van Doorn. Representation of local geometry in the visual system. *Biological Cybernetics*, 55:367–375, 1987. (p 44)

[57] Y. Lamdan, J.T. Schwartz, and H.J. Wolfson. Affine invariant model-based object recognition. IEEE *Journal of Robotics and Automation*, 6:578–589, 1990. (pp 54, 65)

[58] Y. Lamdan and H.J. Wolfson. Geometric hashing: a general and efficient model-based recognition scheme. In *Proceedings of the 2nd International Conference on Computer Vision, Tampa, Florida, USA*, pages 238–249, 1988. (pp 40, 41, 44, 47, 54, 65, 115)

[59] B. Lamiroy. Reconnaissance d'objets polyédriques à l'aide d'invariants projectifs. Mémoire de DEA informatique, Institut National Polytechnique de Grenoble, 1994. (p 52)

[60] B. Lamiroy and P. Gros. Rapid object indexing and recognition using enhanced geometric hashing. In *Proceedings of the 4th European Conference on Computer Vision, Cambridge, England*, volume 1, pages 59–70, April 1996. Postscript version available by `ftp`[2]. (pp 40, 41, 47, 48, 115, 123)

2. ftp://ftp.imag.fr/pub/MOVI/publications/Lamiroy_eccv96.ps.gz

[61] B. Lamiroy and P. Gros. Object indexing is a complex matter. In *Proceedings of the 10th Scandinavian Conference on Image Analysis, Lappeenranta, Finland*, volume I, pages 277–283, June 1997. Postscript version available by `ftp`[3]. (p 115)

[62] B. Lamiroy, C. Schmid, R. Mohr, M. Tonko, K. Schäfer, and H.H. Nagel. Computer aided (dis)assembly using visual cues. In *Annual IAR Conference, November 21-22, 1996, University Karlsruhe, Germany.* Institut franco-allemand pour les applications de la recherche - Deutch-Französisches Institut für Automation und Robotik, November 1996. Postscript version available by `ftp`[4]. (pp 45, 122)

[63] D. Marr. *Vision.* W.H. Freeman and Company, San Francisco, California, USA, 1982. (p 26)

[64] T. Matsuyama. Expert systems for image processing: Knowledge-based composition of image analysis processes. *Computer Vision, Graphics and Image Processing*, 48:22–49, 1989. (p 35)

[65] B. Moghaddam and A. Pentland. Probabilistic visual learning for object detection. In *Proceedings of the 5th International Conference on Computer Vision, Cambridge, Massachusetts, USA*, pages 786–793, 1995. (p 39)

[66] R. Mohr, P. Gros, B. Lamiroy, S. Picard, and C. Schmid. Indexation et recherche d'images. In *Actes du 16^e colloque* GRETSI *sur le traitement du signal et des images, Grenoble, France*, September 1997. (p 37)

[67] L. Morin. *Quelques contributions des invariants projectifs à la vision par ordinateur.* Thèse de doctorat, Institut National Polytechnique de Grenoble, January 1993. (pp 52, 59, 109)

[68] Y. Moses and S. Ullman. Limitations of non model–based recognition. In *Proceedings of the 2nd European Conference on Computer Vision, Santa Margherita Ligure, Italy*, pages 820–828, May 1992. (p 51)

[69] O. Munkelt. Aspect-trees: Generation and interpretation. *Computer Vision and Image Understanding*, 61(3):365–386, May 1995. (pp 37, 56)

[70] H. Murase and S.K. Nayar. Visual learning and recognition of 3D objects from appearance. *International Journal of Computer Vision*, 14:5–24, 1995. (p 39)

[71] R.C. Nelson. Memory-based recognition of parts for curved and polyedral objects. In *Proc. of the ARPA Image Understanding workshop, Palm Springs, CA*, 1996. (pp 40, 41, 47)

[72] S.A. Nene and S.K. Nayar. A simple algorithm for nearest neighbor search in high dimensions. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 19(9):989–1003, 1997. (p 114)

3. ftp://ftp.imag.fr/pub/MOVI/publications/Lamiroy_scia97.ps.gz
4. ftp://ftp.imag.fr/pub/MOVI/publications/Lamiroy_iar96.ps.gz

[73] R. Nevatia. Characterization and requirements of computer vision systems. In A.R. Hanson and E.M. Riseman, editors, *Computer Vision Systems*, pages 81–85. Academic Press, New York, USA, 1978. (p 34)

[74] W. Niblack, R. Barber, W. Equitz, M. Fickner, E. Glasman, D. Petkovic, and P. Yanker. The QBIC project: Querying images by content using color texture and shape. In *Proceedings of the* SPIE *Conference on Geometric Methods in Computer Vision II, San Diego, California, USA*, February 1993. (p 113)

[75] C.F. Olson. Probabilistic indexing for object recognition. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 17(5):518–522, May 1995. (pp 51, 66)

[76] A. Pentland, R.W. Picard, and S. Sclaroff. Photobook: Content-based manipulation of image databases. *International Journal of Computer Vision*, 18(3):233–254, 1996. (pp 41, 113)

[77] S. Petitjean. The enumerative geometry of projective algebraic surfaces and the complexity of aspect graphs. *International Journal of Computer Vision*, 19(3):261–287, 1996. (p 43)

[78] S. Petitjean, J. Ponce, and D.J. Kriegman. Computing exact aspect graphs of curved objects: Algebraic surfaces. *International Journal of Computer Vision*, 9(3):231–255, 1992. (p 43)

[79] A.R. Pope. Model-based object recognition: a survey of recent research. Technical Report 94–04, UBC Department of Computer Science, January 1994. (p 37)

[80] R.P.N. Rao and D.H. Ballard. Object indexing using an iconic sparse distributed memory. In *Proceedings of the 5th International Conference on Computer Vision, Cambridge, Massachusetts, USA*, pages 24–31, 1995. (p 114)

[81] I. Rigoutsos and R. Hummel. Robust similarity invariant matching in the presence of noise. In *8th Israeli Symposium on Artificial Intelligence and Computer Vision*, pages 27–43, December 1991. (p 65)

[82] I. Rigoutsos and R. Hummel. A bayesian approach to model matching with geometric hashing. *Computer Vision and Image Understanding*, 62(1):11–26, 1995. (p 65)

[83] B.D. Ripley. *Pattern Recongition and Neural Networks.* Cambridge University Press, 1996. (p 112)

[84] J.T. Robinson. The K-D-B-tree: A search structure for large multidimensional dynamic indexes. In SIGMOD *'81, Ann Arbor, MI*, pages 10–18. Association for Computing Machinery, 1981. (pp 58, 114)

[85] A. Rosenfeld. Image analysis: Problems, progress and prospects. *Pattern Recognition*, 17(1):3–11, 1984. (p 18)

[86] C.A. Rothwell. *Object Recognition Through Invariant Indexing.* Oxford Science Publication, 1995. (pp 40, 52, 53, 58)

[87] W.J. Rucklidge. Locating objects using the Hausdorff distance. In *Proceedings of the 5th International Conference on Computer Vision, Cambridge, Massachusetts, USA*, pages 457–464, 1995. (p 41)

[88] B. Schiele. *Reconnaissance d'objets utilisant des histogrammes multidimensionnels de champs réceptifs.* Thèse de doctorat, GRAVIR – IMAG – INRIA Rhône–Alpes, July 1997. (pp 41, 45, 47, 85, 109, 113)

[89] B. Schiele and J.L. Crowley. Object recognition using multidimensional receptive field histograms. In *Proceedings of the 4th European Conference on Computer Vision, Cambridge, England*, pages 610–619, 1996. (pp 47, 109, 123)

[90] C. Schmid. *Appariement d'images par invariants locaux de niveaux de gris.* Thèse de doctorat, Institut National Polytechnique de Grenoble, GRAVIR – IMAG – INRIA Rhône–Alpes, July 1996. (pp 40, 44, 45, 47, 85, 90, 91, 110, 112, 113, 115)

[91] C. Schmid, Ph. Bobet, B. Lamiroy, and R. Mohr. An image oriented CAD approach. In *Proceedings of the ECCV workshop on Object Representation*, 1996. Postscript version available by `ftp`[5]. (p 45)

[92] C. Schmid and R. Mohr. Combining greyvalue invariants with local constraints for object recognition. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, San Francisco, California, USA*, June 1996. [6] (p 123)

[93] I. Shimshoni and J. Ponce. Probabilistic 3D object recognition. In *Proceedings of the 5th International Conference on Computer Vision, Cambridge, Massachusetts, USA*, pages 488–493, 1995. (pp 51, 66)

[94] L. Sirovitch and M. Kirby. Low-dimensional procedure for the characterization of human faces. *Journal of the Optical Society of America*, 2:586–591, 1987. (p 39)

[95] H. Sossa. *Reconnaissance d'objets polyédriques dans une base de modèles.* Thèse de doctorat, Institut National Polytechnique de Grenoble, France, December 1992. (pp 43, 72)

[96] M.J. Swain and D.H. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991. (pp 41, 114)

[97] D.W. Thompson and J.L. Mundy. Three-dimensional model matching from an unconstraint viewpoint. In *Proceedings of* IEEE *International Conference on Robotics and Automation, Raleigh, North Carolina, USA*, pages 208–220, 1987. (p 53)

5. ftp://ftp.imag.fr/pub/MOVI/publications/Schmid_WSeccv96.ps.gz
6. ftp://ftp.imag.fr/pub/MOVI/publications/Schmid_cvpr96.ps.gz.

[98] J. Tricot. *De l'âme*. Bibliothèque des textes philosophiques. Librairie Philosophique J. Vrin, 6, Place de la Sorbonne, V^{e}, PARIS, 1985. trad. Aristote (-384 à -322) ΠΕΡΙ ΨΥΧΗΣ. (p 26)

[99] F.C.D. Tsai. A probabilistic approach to geometric hashing using line features. *Computer Vision and Image Understanding*, 63(1):182–195, January 1996. (p 65)

[100] M. Turk and A. Pentland. Face recognition using eigenfaces. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Maui, Hawaii, USA*, pages 586–591, 1991. (pp 39, 114)

[101] D.A. White and R. Jain. Similarity indexing with the SS-tree. In *12th International Conference on Data Engineering, New Orleans, LA*, pages 516–523. IEEE, February 1996. (pp 58, 114)

[102] H.J. Wolfson. Model-based object recognition by geometric hashing. In O. Faugeras, editor, *Proceedings of the 1st European Conference on Computer Vision, Antibes, - France*, pages 526–536. Springer-Verlag, April 1990. (pp 54, 65)

Index des auteurs cités

Index des mots clef

Reconnaissance et modélisation d'objets 3D à l'aide d'invariants projectifs et affines.

Résumé

Le travail de cette thèse s'inscrit dans le cadre de la modélisation et de la reconnaissance d'objets par leur apparence et par des descripteurs locaux. Nous partons, dans une première partie de cette thèse, d'images d'où sont extraits des contours puis des segments approchant ces derniers. À partir de ces segments, nous calculons des descripteurs locaux, appelés *quasi-invariants*, qui ont la particularité d'être très stables par rapport à des changements modérés de point de vue. En stockant ces quasi-invariants dans une structure adaptée, et en modélisant un objet 3D par un ensemble limité de vues 2D, nous montrons qu'il est possible de reconnaître des objets sous tout angle de vue. La reconnaissance est obtenue en deux étapes. D'abord les quasi-invariants locaux entre image et modèles sont mis en correspondance en utilisant une méthode d'indexation. Ensuite, une vérification globale exprimant une cohérence géométrique permet de filtrer des appariements erronés et de sélectionner le modèle le plus semblable à l'image. Constatant des faiblesses dans l'extraction et dans le pouvoir discriminant des descripteurs initiaux, nous étendons ensuite notre approche pour fournir une méthode d'intégration avec toute une classe de méthodes locales existantes. Les résultats expérimentaux fournis par cette extension forment une validation complète de notre travail.

Dans un deuxième temps, nous analysons le problème de la complexité algorithmique soulevé par le genre d'approches utilisées. En effet, nous montrons formellement que certaines méthodes d'indexation sont très mal adaptées à la reconnaissance par descripteurs locaux dès lors que ces descripteurs évoluent dans un espace de dimension élevée. La complexité est telle, que, dans certains cas, elle peut dépasser celle d'une comparaison séquentielle de tous les modèles et leurs descripteurs. Nous montrons quels sont ces cas, et ce qui peut être fait pour les éviter.

Mots clefs : reconnaissance d'objets, vision par ordinateur, indexation, vote, quasi-invariants, reconnaissance par apparence, coopération, modélisation locale, cohérence géométrique, vérification globale, complexité algorithmique.

Recognition and Modeling of 3D Objects Through Use of Projective and Affine Invariants

Abstract

This work belongs to the class of appearance based object modeling and recognition through local descriptors. In a first stage, we use a line approximation of contour-segmented images. The line approximation allows us to extract *quasi-invariants*. Quasi-invariants have the particularity of being robust to moderate viewpoint changes. By storing them in an appropriate structure on the one hand, and by modeling 3D objects by a series of 2D views on the other hand, we show that it is possible to recognize objects from any viewpoint. This recognition is obtained in two steps. First, local quasi-invariants are matched between the image and the models by using an indexing technique. Second, a global verification step, expressing a geometric coherence between the found matches, allows us to filter out incorrect ones and to select the closest model. Limitations in the extraction and the descriptive power of the considered local descriptors have pushed us towards a new approach, combining our method with a whole class of existing local appearance based solutions. The included experimental results provide a complete validation of our method.

In a second stage we analyze the computational complexity inherent to the indexing methods described in the first part. We give formal proof showing that indexing is not adapted to local recognition methods as soon as the descriptors are represented in too high a dimension. Complexity is such that, in certain cases, sequential comparison with the known models and their descriptors is faster than indexing. We show which cases are concerned and give indications on how to avoid them.

Keywords : object recognition, computer vision, indexing, voting, quasi-invariants, appearance based, collaboration, local, geometric coherence, global verification, computational complexity.

Thèse préparée dans le laboratoire GRAVIR–IMAG et INRIA Rhône-Alpes, au sein de l'équipe MOVI.
INRIA Rhône-Alpes, 655, Av. de l'Europe, ZIRST, 38330 Montbonnot Saint-Martin.