

HABILITATION A DIRIGER DES RECHERCHES

Présentée à
Université de Nice - Sophia Antipolis

Spécialité
Sciences pour l'ingénieur

**Vers une vision cognitive:
mise en oeuvre de connaissances et de raisonnements
pour l'analyse et l'interprétation d'images.**

par

Monique THONNAT

soutenue le 7 octobre 2003 devant le jury composé de :

Michel	Barlaud , Professeur à l'université de Nice Sophia Antipolis	Président
Mike	Brady , Professeur à Oxford,	Rapporteur
Jan-Olof	Eklundh , Professeur au KTH Stockholm,	Rapporteur
Malik	Ghallab , Directeur de recherches CNRS au LAAS à Toulouse,	Rapporteur
Marc	Berthod , Directeur de Recherches INRIA	Examineur
Albert	Bijaoui , Astronome à l'université de Nice Sophia Antipolis	Examineur

Remerciements

Je remercie chaleureusement les scientifiques qui m'ont fait l'honneur de faire partie de mon jury d'habilitation,

Michel Barlaud, Professeur à l'université de Nice Sophia Antipolis qui a présidé ce jury

Mike Brady, Professeur à Oxford, Jan-Olof Eklundh, Professeur au KTH à Stockholm et Malik Ghallab, Directeur de recherches CNRS au LAAS à Toulouse, qui ont consacré du temps à la lecture de ce manuscrit et à la rédaction de rapports riches en réflexion; c'est toujours avec plaisir que j'écoute leurs avis pertinents.

Marc Berthod, Directeur de Recherches INRIA qui a été celui qui m'a accueillie à l'INRIA dans son équipe, le projet Pastis et qui m'a donné les moyens de poursuivre mes recherches

Albert Bijaoui, Astronome à l'Observatoire de la Cote d'Azur qui a présidé ma thèse en traitement d'images astronomiques, il y a 21 ans et qui m'a ouvert les portes de l'observatoire de Nice pour démarrer une nouvelle activité en classification de galaxies.

Je tiens à remercier les chercheurs avec lesquels j'ai développé mes thèmes de recherches à l'INRIA dans le projet Pastis puis dans le projet Orion, en particulier par ordre chronologique Marie-Helene Gandelin, Shan Liu-Yu, Jean-Christophe Ossola, Alain Boucher et Nicolas Dey sur la reconnaissance d'objets, Véronique Clément, Sabine Moisan, Chandra Shekhar, John van den Elst, Régis Vincent, Monica Crubezy et Jean-Christophe Noel sur le pilotage de programmes, François Brémond, Nicolas Chleq et Nathanael Rota sur l'interprétation vidéo.

Ce travail continue grâce à l'énergie et l'enthousiasme des membres actuels du projet Orion, Alberto Avanzi, Bui Binh, François Brémond, Frédéric Cupillard, Benoît Géoris, Céline Hudelot, Nicolas Maillot, Magali Maziere, Sabine Moisan, Christophe Tornieri et Thinh van Vu.

Organisation du document

Ce document de synthèse résume mon activité de recherches depuis 1983, c'est à dire depuis mon entrée à l'INRIA après ma thèse. Ce document comprend deux parties, une partie factuelle contenant mon curriculum vitae et la liste de mes publications et une partie scientifique résumant mes trois axes de recherches sur le couplage entre des techniques de vision par ordinateur et d'intelligence artificielle en analyse et interprétation d'images, à savoir la classification sémantique d'objets, le pilotage de programmes et l'interprétation vidéo.

Plus précisément:

- Le chapitre 1 contient un curriculum vitae et un bref résumé de mon activité de recherche.
- Le chapitre 2 fournit la liste de mes publications.
- Le chapitre 3 est une présentation de mes thèmes de recherches. Ces thèmes sont détaillés dans les trois chapitres suivants.
- Le chapitre 4 décrit les travaux que j'ai effectués et dirigés en classification d'objets.
- Le chapitre 5 décrit les travaux que j'ai effectués et dirigés en pilotage de programmes.
- Le chapitre 6 décrit les travaux plus récents que j'effectue et que je dirige dans le cadre du projet Orion en interprétation vidéo.
- Le chapitre 7 présente mes futures directions de recherches en vision cognitive.
- Enfin une selection de 7 articles scientifiques est jointe en annexe dans un document séparé.

Chapitre 1

Curriculum vitae

1.1 Curriculum vitae

THONNAT Monique
Née le 31 Mai 1957 à ISSOIRE (63)
Célibataire
Adresse: 13 Jardins de Pomone
Chemin du Val de Pome
06410 BIOT
Nationalité : française
Directeur de recherche INRIA 2ième classe depuis 1991
Chef du projet ORION depuis 1995.

1.2 Activité Professionnelle

Chargé de Recherches à l'Inria en aout 1983.
Directeur de Recherches à l'Inria depuis 1991.
Directrice de 17 thèses, 13 soutenues et 4 en cours.
Chef de l'avant projet Orion de 1992 à 1995
Chef du projet Orion depuis 1995 comprenant actuellement 17 personnes dont 4 chercheurs permanents.
Membre du bureau executif du réseau d'excellence européen ECVision (2002 à 2005)
Membre du Steering Committee et workpackage leader de six projets européens de type R&D (3ième, 4ième et 5ième PCRD).
Membre de la commission d'évaluation INRIA de 1989 à 1996 et depuis septembre 2002.

1.3 Formation

1980: Diplôme d'ingénieur ENSPM
DEA Signaux et Systèmes Spatio Temporels (Mention Bien).
1982: Diplôme Docteur-Ingénieur en Optique et Traitement du Signal de l'Université de Marseille III, thèse préparée au Laboratoire d'Astronomie Spatiale du C.N.R.S. (L.A.S.).
(Sujet : Méthodes interactives de dépouillement de clichés astronomiques : restitution du fond et obtention de vitesses radiales).

1.4 Enseignement

DEA Vision et Robotique (université de Nice-Sophia Antipolis) cours de vision par ordinateur (1990 à 1994)
DEA Imagerie et Sciences de l'Univers (université de Nice-Sophia Antipolis) cours de vision par ordinateur (1991 à 1994)

Ecole Nationale des Telecommunications (**Telecom Paris**) cours de Vision et intelligence artificielle en 3ieme année (1991 à 1995)

DEA Astronomie, Imagerie and Haute résolution angulaire (université de Nice-Sophia Antipolis) cours de vision par ordinateur (1995 à 1996)

ENS Cachan, cours de Vision et intelligence artificielle en 3ieme année et en formation permanente (1996 et 1997).

Formation Permanente du CNRS, Ecole de la Grande-Motte du 15 au 20 mai 2000 cours systèmes à base de connaissances pour le traitement et l'interprétation d'images

DEA Astronomie haute rsolution angulaire, image et gravitation (université de Nice-Sophia Antipolis) cours de classification de 2000 à 2003

1.5 Encadrement d'activités de recherche

Direction de 17 thèses

- M-H. Gandelin: "Une approche système expert pour la reconnaissance automatique d'organismes marins à partir d'images monoculaire" soutenue en 1989.

- V. Clement: "Pilotage automatique d'algorithmes de vision: le système OCAP" soutenue en 1990.

Cette thèse a été celle qui a permis de définir les techniques de pilotage de programmes, ce qui a conduit à 6 autres thèses dans le projet Orion et au moins autant à l'extérieur.

- A. Meygret: "Interprétation de scènes en milieu routier pour la détection d'obstacles" soutenue en 1990.

- S. Yu: "Classement automatique à partir de plusieurs vues de microorganismes fossiles (foraminifères)" soutenue en 1992.

- R. Ma: "Fusion et interprétation de données appliquées à la détection d'obstacles en environnement routier" soutenue en 1994.

- J-C. Ossola: "Coopération de systèmes à base de connaissances pour l'analyse et la reconnaissance d'objets naturels complexes: application au classement de galaxies ou de zooplanktons" soutenue en 1996.

- J. Van Den Elst: "Modélisation de connaissances pour le pilotage de programmes de traitement d'images" soutenue en 1996.

- R. Vincent: "Etude des mécanismes de gestion des erreurs dans les systèmes à base de connaissances de pilotage de programmes" soutenue en 1997.

- F. Brémond: "Environnement de résolution de problèmes pour l'interprétation de séquences d'images" soutenue en 1997.

Cette thèse a été la première en interprétation vidéo, elle a conduit à de nombreux travaux de recherches aussi bien chez Orion que dans des laboratoires internationaux.

- M. Crubézy: "Pilotage de programmes pour le traitement d'images médicales" soutenue en 1999.

- J-C. Noel: "Système à base de connaissances pour le pilotage de codes de calcul en simulation numérique" soutenue en 1999.
- N. Rota: "Contribution à la reconnaissance de comportements humains à partir de séquences videos" soutenue en 2001.
- N. Dey: "Etude de la formation de l'image d'un objet microscopique 3D translucide" soutenue en novembre 2002.
Cette thèse est la première de l'équipe sur la modélisation optique du processus de formation d'image faisant intervenir des connaissances de physique et de synthèse d'images.
- T. Van Vu: "Couplage entre l'analyse et la synthèse d'images pour l'interprétation vidéo." soutenance prévue 2004.
- C. Hudelot: "Interprétation d'images de végétaux pour le diagnostic automatique de pathologies" soutenance prévue 2004.
- N. Maillot: "Système cognitif d'interprétation d'images pour la reconnaissance d'objet 3D" soutenance prévue 2005.
- B. Georis: "Knowledge-based reconfigurable tracker" en coencadrement avec le Prof. B. Macq de l'Université Catholique de Louvain (Belgique), soutenance prévue en 2005.

1.6 Valorisation et transfert technologique

-1 **brevet européen** (déposé 01/09/2000 et publié le 06/03/2002) sur la reconnaissance automatique de pollens avec images 3D.

Expertise scientifique:

-Conseil scientifique chez CGA-HBS (Groupe CGE) et membre du comité de review pour Advanced Research in Techniques for Intelligent Singulation of Manual Letters (United States Post Office proposal) en 1986.

-Expertise en vision et intelligence artificielle, reviewer du projet Eureka franco-espagnol ROSAL, contrat Inria-Meilland (1988 à 1992).

Responsable scientifique de contrats industriels:

-**Aérospatiale Cannes:** contrat avec Aérospatiale pour la réalisation d'un système expert de diagnostic de pannes d'antennes par classification (1986, 1987)

-**Ministère de la Mer:** contrat avec le Ministère de la Mer et le Museum d'Histoire Naturelle pour l'étude d'un système de tri de poisson (1989)

-**Noesis:** contrat avec Noesis pour le développement d'une base de connaissance OCAPI sur VI-SILOG (1991, 1992)

-**Michelin:** contrat avec la société Michelin pour l'étude de techniques de reconnaissances automatiques de formes complexes appliquées à des pneumatiques (1996, 227KF).

-**RATP:** contrat avec RATP et Timeat sur la classification automatique de voyageurs à partir de capteurs.(3 ans 2001-2004, 1.6 MF)

-**Cassiopée:** groupe Credit Agricole avec Eurotelis et Ciel (3 ans 2002-2005, 405 KEuros) conception d'un site pilote de vidéosurveillance bancaire.

-Responsable de l'action de transfert Dyade Telescope entre Inria et Bull (1998 à 2001) et de l'action de transfert Telescope2 entre Orion et Bull Multimedia (2001 à 2002) sur les systèmes de vidéosurveillance intelligente.

-Responsable scientifique de 6 contrats de projets européens R&D (membre du Steering committee et workpackage leader), les montants indiqués correspondent au financement Orion:

-**Eureka Prometheus:** stéréovision et pilotage de programmes temps-réel embarqués sur voiture, avec entre autres PSA et RNUR (8 ans, 1987 à 1994),

-**Esprit 3 Passwords:** vidéosurveillance avec VIGITEC, SEPA, DIBE, AUCHAN (3 ans, 1994 à 1997, 1.4 MF),

-**Esprit 4 AVS-RTPW:** vidéosurveillance temps-réel avec Timeat et SEPA (1997 à 1998),

-**Esprit 4 AVS-PV:** vidéosurveillance pour sécurité dans métro avec VIGITEC, DIBE, métros Nuremberg, Bruxelles et Charleroi (1997 à 1998)

-**IST ADVISOR:** vidéosurveillance multi-caméras et archivage avec THALES, BULL, VIGITEC, Univ. Kingston, Univ. Reading, KCL Londres (3 ans, 2000 à 2002, 2.4 MF)

-**Climat et Environnement Asthma:** reconnaissance automatique de pollens avec Zambon, ACRI, LASMEA, Univ. Barcelone, Univ. Cordoba, Univ. Lille (3 ans, 1999 à 2001, 1.2 MF)

1.7 Réalisation et diffusion de logiciels

Réalisation de 4 gros logiciels:

Classic moteur de classification produit commercialisé par ILOG

Ocapi et Pegase moteurs de pilotage de programmes (planification et contrôle d'exécution) prototypes distribués environ 10 exemplaires extérieurs .

VSIS plateforme de vidéosurveillance intelligente action de transfert auprès de Bull Multimedia, nouvelle version **actuellement intégrée chez Thales (UK)** et partiellement chez l'intégrateur Ciel.

1.8 Diffusion de l'information scientifique

Un **nombre important d'articles auprès des médias** professionnels ou grand public ont permis de diffuser les résultats de nos travaux en interprétation vidéo dont *L'Ordinateur individuel* (mars 1999), *Sciences et Avenir* (janvier 2000), *Industries et Techniques* (juillet 2000), *Transfert* (juin 2001), *Le Monde* (août 2001), *De Morgen* (Belgique août 2001), *La Meuse* (Belgique août 2001), *Les Echos* (septembre 2001), *l'Espresso* (Italie, septembre 2001), *Ercim News* (Europe octobre 2001).

1.9 Mobilité

Mobilité thématique:

1979: Traitement d'images en Astronomie au Laboratoire d'Astronomie Spatiale du CNRS à

Marseille

1983: Vision par ordinateur, systèmes à base de connaissances dans le projet Pastis à l'INRIA.

1992: Intelligence Artificielle, Pilotage de programmes, Interprétation d'images vidéos, Apprentissage, création du projet Orion à l'INRIA à Sophia Antipolis.

1.10 Responsabilités collectives

1.10.1 Tâches d'intérêt collectif à l'Inria

-Chef d'avant projet depuis 1992 puis chef du projet Orion depuis 1995. Après l'évaluation de 1997 les experts avaient conclu par: *le projet Orion s'intéresse d'une manière fructueuse et bien focalisée, à une problématique essentielle qu'il est important de pouvoir poursuivre, Il dispose d'un domaine riche en extensions thématiques, en développements et en applications futures. Après l'évaluation de 2001 la CE avait conclu: La CE se rjouit de l'avis des valueurs sur le positionnement scientifique du projet problme difficile auquel peu d'quipes se sont atteles , vrai besoin) et sur la qualitt des travaux accomplis (ensemble complet d'outils , meilleur niveau international , plusieurs applications menes avec succs , bonne reconnaissance , atout important pour la recherche franaise, nombreuses collaborations, futures directions de recherche trs cohrentes avec les travaux actuels.)*

-Membre de la commission d'évaluation pendant 8 ans (89 à 96) et depuis septembre 2002.

-Membre du bureau du comité des projets à Sophia de 1997 à 1999.

1.10.2 Responsabilités scientifiques

-Relecteur des revues Artificial Intelligence, IJCV (International Journal on Computer Vision), CVIU (Journal of Computer Vision and Image Understanding), IEEE Transactions - Pattern Analysis and Machine Intelligence PAMI, Videre, IEEE Transactions on CSTV, PATREC, Pattern Recognition Letters, Image and Vision Computing Journal Astr. Lettres&Com., IJPRAI, TS et de la revue RIA (Revue d'Intelligence Artificielle).

-Relecteur des conférences ICPR, IJCAI, RFIA, KBUP, EWSP, CARI, FG2000, TAIMA, PETS et VS2000

-Membre du comité d'organisation de la conférence internationale IEA/AIE 98, de la confrence internationale FG2000 Face and Gesture recognition et du comité scientifique des conférences RFIA98 et ICVS03, présidente de la conférence KBUP'95, responsable scientifique du workshop franco-taiwanais sur le Multimedia et les technologies web en 2002 et 2003. Scientific cochair avec Bruce Draper de la 4ieme conférence ICVS en 2005.

-Expert auprès de la commission européenne pour l'évaluation de projets européens IST en mai 2001.

-Membre du bureau exécutif (area leader) du réseau d'excellence européen ECVision en vision cognitive depuis mars 2002 pour 3 ans (50 laboratoires de 12 pays).

-Responsable d'une action intégrée PICASSO franco-espagnole sur le thème de la vérification et validation de base de connaissances avec l'université de Castellon (1995 et 1996).

-Responsable d'une action STIC franco-tunisienne sur le thème des systèmes intelligents distribués pour la vision (1999, 2001).

-Responsable de l'action incitative LIRE sur le thème de la résolution de problèmes avec limitation de ressources avec les projets SYCO (actuellement MAIA) et OMEGA de Nancy et le projet SHERPA (actuellement HELIX) de Grenoble (1998 et 1999)

-Responsable de l'action incitative Mediaspace sur le thème de l'interprétation de séquences d'images appliquée des environnements de travail collectif distants (ou mediaspaces) avec l'équipe PRIMA de Jim Crowley de l'ENSIMAG l'INRIA Monbonnot et l'équipe IIHM de Joelle Coutaz l'Université Joseph Fourier Grenoble (1999 et 2000).

-Coopération avec F. Van Harmelen de l'Université Libre d'Amsterdam sur le thème de la validation et de la vérification de systèmes à base de connaissances.

-Coopération avec C. Shekhar de l'équipe du professeur Chelappa de l'université de Maryland à College Park sur le thème du pilotage de programmes de traitement du signal et d'images ainsi qu'avec le professeur Larry Davis sur le thème de l'interprétation automatique de séquences vidéos.

-Coopération avec l'équipe des professeurs Ram Nevatia et Gérard Medioni de l'University of Southern California (USC) sur le thème de la reconnaissance de scénarios (post-doc de deux ans de F. Brémont).

1.11 Autres éléments

-Membre du conseil scientifique du département Astrophysique et Méthodes Observationnelles des Observatoires OCA.

-Membre du conseil d'administration de PERSAN (association des instituts d'enseignement et de recherche des Alpes Maritimes).

-Membre du conseil d'administration de l'UFR STAPS de l'Université de Nice Sophia Antipolis.

1.12 Résumé de l'activité de recherche

De 1979 à 1982: Recherches en traitement d'images pour l'astronomie. Dans le cadre de ma thèse au Laboratoire d'Astronomie Spatiale du CNRS à Marseille, j'ai étudié le développement de méthodes facilitant la mesure photométrique d'objets astronomiques. J'ai proposé une nouvelle technique d'analyse photométrique de sources étendues; le logiciel issu de ces recherches est toujours utilisé par les astronomes au LAS. J'ai également proposé une méthode de calcul de vitesse radiale de galaxies par interférogrammes de Fabry-Pérot.

De 1983 à 1991: Recherches en vision par ordinateur. En 1983, je suis entrée comme chargée de recherche à l'I.N.R.I.A. dans le projet d'analyse de scènes et de traitement d'images symboliques dirigé par M. Berthod à Sophia Antipolis. Le thème de recherche sur lequel j'ai travaillé est l'étude du problème de la **reconnaissance d'objets naturels complexes** à partir d'images 2D

de l'intensité lumineuse; je me suis attachée à étudier les différents aspects de ce problème qui sont d'une part la mesure de paramètres descriptifs significatifs de la forme perçue par le capteur, d'autre part la modélisation et la description des objets et enfin la conception d'un processus de reconnaissance à base de connaissances explicites. L'objectif de ce travail étant l'automatisation complète de l'interprétation d'images, j'ai choisi de sélectionner des applications correspondant à des cas réels difficiles (galaxies, zooplanctons, poissons, foraminifères). L'étude du problème de l'interprétation d'images 2D comportant un objet de forme complexe m'a conduite à devoir gérer des connaissances non seulement sur les modèles des objets recherchés mais aussi sur les programmes effectuant des mesures sur l'image; j'ai donc été amenée à aborder le problème de la modélisation du raisonnement et des connaissances en traitement d'images. L'aspect auquel je me suis intéressée est l'automatisation de l'utilisation d'une bibliothèque de programmes de traitement d'image. Ces travaux ont conduit à la nouvelle thématique de recherche sur le **pilotage de programmes**. D'autre part je me suis intéressée non plus seulement à l'interprétation d'images 2D de l'intensité lumineuse, mais aussi à la description et à l'interprétation de scènes 3D à partir de données 3D (stéréovision et analyse du mouvement).

Depuis 1992: Recherches en vision cognitive. Les travaux que j'avais effectués jusqu'alors ont montré l'intérêt des techniques d'intelligence artificielle mais aussi la grande difficulté qu'il y a à modéliser des connaissances complexes. Afin d'approfondir les recherches initialisées, j'ai créé en 1992 une **équipe pluridisciplinaire** à la frontière des domaines de l'**Intelligence Artificielle, de la Vision par Ordinateur et du Génie Logiciel**. ORION se focalise sur deux axes de recherches: le pilotage automatique de programmes et l'interprétation automatique d'images. Le **premier axe de recherches** est l'analyse de la tâche de **pilotage automatique de traitements numériques**. Plus précisément il s'agit de faciliter et même d'automatiser la planification et le contrôle d'exécution de programmes de traitements numériques contenus dans une bibliothèque. Ces travaux sont l'approfondissement des travaux initialisés avec les recherches sur le moteur OCAP. Le **deuxième axe de recherches** d'ORION est l'analyse de la tâche d'**interprétation automatique d'images**. L'originalité des travaux menés au sein d'Orion est l'étude de la problématique de l'**interprétation automatique de séquences temporelles d'images** ou **interprétation vidéo**, c'est à dire la conception de systèmes temps-réels complets effectuant une analyse du flux video (détection et suivi des objets mobiles) et une interprétation sémantique du comportement des acteurs (reconnaissance d'événements et de scénarios).

Chapitre 2

Publications

Publications

Chapitres de livres

- [1] N. Chleq, F. Bremond, and M. Thonnat. *Advanced Video-based Surveillance Systems*, chapter Image Understanding for Prevention of Vandalism in Metro Stations, pages 106–116. Kluwer A.P. , Hangham, MA, USA, C.S. Regazzoni, G. Fabri, and G. Vernazza edition, 1999.
- [2] V. Clément and M. Thonnat. Utilization of the expert system generator ocapl for the integration of a stereovisual processing. In E. Granum J.L. Crowley and R. Duller, editors, *Integration and Control in Real-Time Active Vision*, Basic Research series. Springer Verlag, 1991.
- [3] A. Meygret, C. Shekhar, and M. Thonnat. Stereo image analysis of traffic scenes. In *Intelligent automotive systems*, International series on road vehicle automation. Chapman and Hall, 1993.
- [4] A. Meygret and M. Thonnat. Segmentation of optical flow and 3d data for the interpretation of mobile objects. In R. Kasturi and R. Jain, editors, *Computer Vision: Principles and Applications*, tutorial series. IEEE Press, 1991.
- [5] M. Thonnat. *The World of Galaxies*, chapter Toward an automatic classification of galaxies, pages 53–74. Springer Verlag, Corwin, H. Jr. and Bottinelli, L. edition, 1989.
- [6] M. Thonnat. Knowledge-based techniques for image processing and for image understanding. In A. Heck and F. Murtagh, editors, *Voies nouvelles pour l'Analyse de Données en Sciences de l'Univers*, volume 12 of *Journal Phys. IV FRance*, pages 189–236. EDP Sciences, Les Ulis, 2002.
- [7] M. Thonnat and A. Bijaoui. Knowledge-based galaxy classification systems. In A. Heck and F. Murtagh, editors, *Knowledge-based systems in astronomy*, volume 329 of *Lecture Notes in Physics*. Springer Verlag, 1989.

Revue Internationale avec comité de lecture

- [8] P. Bonton, A. Boucher, M. Thonnat, R. Tomczak, P.J. Hidalgo, J. Belmonte, and C. Galan. Colour image in 2d and 3d microscopy for the automation of pollen rate measurement. *Image Analysis and Stereology*, 20(Suppl 1):527–532, 2001. also Proc. 8th European Congress for Stereology and Image Analysis (ECSIA).

- [9] A. Boucher, P.J. Hidalgo, M. Thonnat, J. Belmonte, C. Galan, P. Bonton, and R. Tomczak. Development of a Semi-Automatic System for Pollen rRecognition. *Aerobiologia, International Journal of Aerobiology*, 18(3-4):195–201, 2002.
- [10] F. Brémond and M. Thonnat. Issues of representing context illustrated by video-surveillance applications. *International Journal of Human-Computer Studies Special Issue on Context*, 48:375–391, 1998.
- [11] F. Brémond and M. Thonnat. Tracking multiple non-rigid objects in video sequences. *IEEE Transaction on Circuits and Systems for Video Technology Journal*, 8(5), September 1998.
- [12] V. Clément and M. Thonnat. Integration of image processing procedures, ocap: a knowledge-based approach. *Computer Vision Graphics and Image Processing: Image Understanding*, 57(2), March 1993.
- [13] S. Liu and M. Thonnat. Description of object shapes by apparent boundary and convex hull. 1993, 26(1), Pattern Recognition.
- [14] C. Shekhar, S. Moisan, and M. Thonnat. Towards an intelligent problem-solving environment for signal processing. *Mathematics and Computers in Simulation*, 36, 1994.
- [15] M. Thonnat. Semantic interpretation of 3d stereo data: finding the main structures. *International Journal on Pattern Recognition and Artificial Intelligence*, 2(3), September 1988.
- [16] M. Thonnat, V. Clément, and J. C. Ossola. Automatic galaxy description. *Astrophysical Letters and Communication*, 31(1-6):65–72, 1995.
- [17] M. Thonnat, V. Clément, and J. van den Elst. Supervision of perception tasks for autonomous systems: the ocap approach. *International Journal on Information Science and Technology IJIST*, 3(2):140–163, January 1994.
- [18] M. Thonnat and S. Moisan. What can Program Supervision do for Program Reuse? *IEE Proceedings - Software*, 147(5):179–185, October 2000.
- [19] M. Thonnat, B. Ruffini, and J. Caplan. A new method for determining nebular radial velocities from fabry-perot interferograms. *Astronomy and Astrophysics*, 1983.
- [20] M. Xie and M. Thonnat. An algorithm for finding closed curves. *Pattern Recognition Letters*, 13:73–81, 1992.
- [21] S. Yu, P. Saint-Marc, M. Thonnat, and M. Berthod. Feasibility study of automatic identification of planktic foraminifera by computer vision. *Journal of Foraminiferal Research*, 26(2), April 1996.

Revues Nationales avec comité de lecture

- [22] V. Clément and M. Thonnat. Pilotage de procédures de traitement d'images pour la description morphologique de galaxies. *TS*, 9(5), January 1993. numéro spécial Intelligence Artificielle.
- [23] M. Thonnat and M-H. Gandelin. Un système expert pour la description et le classement automatiques de classement de zooplanctons à partir d'images monoculaires. *TS*, 9(5), January 1993. Numéro spécial Intelligence Artificielle.

Conférences Internationales avec comité de lecture

- [24] A. Avanzi, F. Brmond, and M. Thonnat. Tracking multiple individuals for video communication. In *International Conference on Image Processing ICIP'01*, Thessaloniki, Greece, October 2001. IEEE Signal Processing Society.
- [25] M. Bogaert, N. Chleq, P. Cornez, C.S. Regazzoni, A. Teschioni, and M. Thonnat. The passwords project. In *IEEE International Conference on Image Processing*, September 1996.
- [26] A. Boucher and M. Thonnat. Object recognition from 3d blurred images. In *International Conference on Pattern Recognition ICPR2002*, Quebec, Canada, August 2002.
- [27] F. Brémond and M. Thonnat. A context representation for surveillance systems. In *ECCV Workshop on Conceptual Descriptions from Images*, April 1996.
- [28] F. Brémond and M. Thonnat. Analysis of human activities described by image sequences. In *Proc. of the International Florida Artificial Intelligence Research Symposium (FLAIRS'97)*, May 1997.
- [29] F. Brémond and M. Thonnat. Issues in representing context illustrated by scene interpretation applications. In *Proc. of the International and Interdisciplinary Conference on Modeling and Using Context (CONTEX'97)*, February 1997.
- [30] F. Brémond and M. Thonnat. Object tracking and scenario recognition for video-surveillance. In *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI'97)*, August 1997.
- [31] F. Brémond and M. Thonnat. Recognition of scenarios describing human activities. In *Proc. of the International Workshop on Dynamic Scene Recognition from Sensor Data*, June 1997.
- [32] F. Brémond and M. Thonnat. Tracking multiple non-rigid objects in a cluttered scene. In *Proc. of the Scandinavian Conference on Image Analysis (SCIA'97)*, June 1997.
- [33] S. Chandrashekhar, A. Meygret, and M. Thonnat. Temporal analysis of sequences of stereo images of traffic scenes. In *Vehicle Navigation and Information Systems Conference*, Dearborn, Michigan, USA, October 1991.
- [34] N. Chleq, C. Regazzoni, A. Teschioni, and M. Thonnat. A visual surveillance system for the prevention of vandalism in the metro stations. In *European, Multimedia, Microprocessor Systems and Electronic Commerce EMMSEC'97*, November 1997.
- [35] N. Chleq and M. Thonnat. Realtime image sequence interpretation for video-surveillance applications. In *International Conference on Image Processing*, September 1996.
- [36] V. Clément and M. Thonnat. Handling knowledge in image processing libraries to build automatic systems. In *International Workshop on Machine Intelligence and Vision*, Tokyo, April 1989.
- [37] V. Clément and M. Thonnat. Supervision of perception tasks for autonomous systems: the ocap1 approach. In *3rd Annual Conference on Artificial Intelligence, Simulation and Planning in High Autonomy Systems*, pages 202–212, Perth, Western Australia, July 1992.
- [38] M. Crubzy, F. Aubry, S. Moisan, V. Chameroy, M. Thonnat, and R. Di Paola. Managing Complex Processing of Medical Image Sequences by Program Supervision Techniques. In *SPIE International Symposium on Medical Imaging'97*, volume 3035, February 1997.

- [39] F. Cupillard, F. Brmond, and M. Thonnat. Tracking groups of people for videosurveillance. In *2nd European Workshop on Advanced Video-based surveillance Systems AVBS'01*, Kingston, UK, September 2001.
- [40] N. Dey, A. Boucher, and M. Thonnat. Image formation model of a 3d translucent object observed in light microscopy. In *International Conference on Image Processing ICIP2002*, Rochester, USA, September 2002.
- [41] D. Hutber, S. Moisan, C. Shekhar, and M. Thonnat. Perception-interpretation interfacing for the prolab2 road vehicle. In *7th Symposium on Transportation Systems: Theory and Application of Advanced Technology*, Tianjin, China, August 1994.
- [42] S. Liu, M. Thonnat, and M. Berthod. Automatic classification of planktonic foraminifera by a knowledge-based system. In *The Tenth Conference on Artificial Intelligence for Applications*, pages 358–364, San Antonio, Texas, March 1994. IEEE Computer Society Press.
- [43] R. Ma and M. Thonnat. Object detection in outdoor scenes by disparity map segmentation. In *11th International Conference on Pattern Recognition*, The Hague, Nederland, August 1992.
- [44] A. Meygret and M. Thonnat. Segmentation of optical flow and 3d data for the interpretation of mobile objects. In *3rd International Conference on Computer Vision*, Osaka, Japan, December 1990. long paper.
- [45] A. Meygret, M. Thonnat, and M. Berthod. A pyramidal stereovision algorithm based on contour chain points. In *European Conference on Computer Vision*, Antibes, April 1990.
- [46] S Moisan, M. Bor, C. Thiebaut, F. Tricoire, and M. Thonnat. A versatile scheduler for automatic telescopes. In *SPIE Conference on Astronomical Telescopes and Instrumentation*, Hawaii, USA, August 2002.
- [47] J.C. Ossola, F. Brémond, and M. Thonnat. A communication level in a distributed architecture for object recognition. In *8th International Conference on Systems Research Informatics and Cybernetics*, August 1996.
- [48] J.C. Ossola and M. Thonnat. Cooperation of knowledge based systems for galaxy classification. In *8th International Conference on Image Analysis and Processing (ICIAP'95)*, September 1995.
- [49] J.C. Ossola and M. Thonnat. A distributed architecture for automatic object recognition. In *Advances in Intelligent Data Analysis (IDA'95)*, pages 134–138, August 1995.
- [50] S. Pouget, M. Thonnat, V. Clément, and A. de Fombelle. Automatic diagnosis of a technical device by classification, the expert system dante : an application to antenna failures. In *6th International Conference on Reliability and Maintainability*, Strasbourg, October 1988.
- [51] N. Rota, R. Stahr, and M. Thonnat. Tracking for Visual Surveillance in VSIS. In *First IEEE International Workshop on Performance Evaluation of Tracking and Surveillance PETS2000.*, Grenoble, France, March 2000.
- [52] N. Rota and M. Thonnat. Activity Recognition from Video Sequences using Declarative Models. In *14th European Conference on Artificial Intelligence. ECAI2000.*, Berlin, Germany, August 2000.
- [53] N. Rota and M. Thonnat. Video Sequence Interpretation for Visual Surveillance. In *Workshop on Visual Surveillance. VS'00.*, pages 59–67, Dublin,Ireland , July 2000.

- [54] C. Shekhar, V. Gaudin, S. Moisan, and M. Thonnat. Real-time supervision of perception programs for prolab2. In T. Fukuda P. Borne and S.G. Tzafestas, editors, *IMACS International Symposium on Signal Processing, Robotics and Neural Networks*, pages 11–14, Villeneuve-d'Ascq, France, April 1994.
- [55] C. Shekhar, S. Moisan, and M. Thonnat. Use of a real-time perception program supervisor in a driving scenario. In *Intelligent Vehicle Symposium '94*, Paris, France, October 1994.
- [56] C. Shekhar, S. Moisan, and M. Thonnat. Real-time perception program supervision for vehicule driving assistance. In *International Conference on Recent Advances in Mechatronics ICRAM'95*, Istambul, Turkey, August 1995.
- [57] M. Thonnat. Semantic interpretation of 3d stereo data : finding the main structures. In *8th International Conference on Pattern Recognition*, Paris, October 1986.
- [58] M. Thonnat. Toward an automated classification of galaxies. In *International Conference Le Monde des Galaxies*, Paris, April 1988. Invited contribution.
- [59] M. Thonnat. Automatic recognition of complex objects by expert system. In *Second Conference of the International Federation of Classification Society*, Charlottesville, V. A., USA, June 1989. Invited paper.
- [60] M. Thonnat. Knowledge-based classification of galaxies. In *XIth Strasbourg Astronomical Day : Artificial Intelligence Techniques for Astronomy*, Strasbourg, April 1989. Invited paper.
- [61] M. Thonnat, A. Avanzi, and F. Brmond. Analyse de scnes de bureaux pour la communication video. In *2mes Ateliers Traitement et Analyse d'Images Mthodes et Applications, TAIMA01*, Hammamet, Tunisia, October 2001.
- [62] M. Thonnat and M. Berthod. Automatic classification of galaxies into morphological types. In *7th International Conference on Pattern Recognition*, Montreal, August 1984.
- [63] M. Thonnat and V. Clément. Ocapi: a monitoring tool for the automatic control of image processing procedures. In *IMACS World Congress*, Paris, July 1988.
- [64] M. Thonnat and V. Clément. Ocapi: an artificial intelligence tool for the automatic selection and control of execution of image processing procedures. In *3rd International Workshop on Data Analysis in Astronomy*, Errice, Italy, June 1988. Invited paper.
- [65] M. Thonnat and M-H. Gandelin. An expert system for the automatic classification and description of zooplanktons from monocular images. In *9th International Conference on Pattern Recognition*, Roma, Italy, November 1988.
- [66] M. Thonnat, C. Granger, and M. Berthod. Design of an expert system for object classification through an application to the classification of galaxies. In *Computer Vision and Pattern Recognition*, pages 206–208, San Francisco, June 1985.
- [67] M Thonnat and A. Llebaria. Background restitution in an astronomical image using histogram of local contrast. In *SPIE International Technical Conference/Europe*, Geneva, April 1983.
- [68] M. Thonnat and S. Moisan. Knowledge-based systems for program supervision. In *First international workshop on Knowledge-Based systems for the (re)Use of Programs libraries KBUP'95*, Sophia Antipolis, France, 1995. INRIA.
- [69] M. Thonnat and S. Moisan. What can program supervision do for program re-use? In Jose Mira Angel Pasqual del Pobil and Moonis Ali, editors, *Proceeding of the 11th Internatio-*

- nal Conference on Industrial Applications of Artificial Intelligence and Expert Systems*, Lecture Notes in Computer Science, Benicassim, Spain, June 1998. Springer-Verlag.
- [70] M. Thonnat, S. Moisan, and M. Crubézy. Experience in integrating image processing programs. In Henrik Christensen, editor, *Proceeding of the 1st International Conference on Vision Systems*, Lecture Notes in Computer Science, pages 200–215, Las Palmas, Gran Canaria, Spain, January 1999. Springer-Verlag.
 - [71] M. Thonnat and N. Rota. Image Understanding for Visual Surveillance Applications. In *Third International Workshop on Cooperative Distributed Vision*, pages 51–82, Kyoto, Japan, November 1999. Invited paper.
 - [72] M. Thonnat and M. Xie. A cooperative algorithm for the matching of contour points and contour chains. In *6th International Conference on Image Analysis and Processing*, Como, Italy, September 1991.
 - [73] L. Trassoudaine, D. Hutber, P. Checchin, J. Alizon, J. Gallice, and M. Thonnat. Building an environment map around the prolab2 vehicle using a controllable range sensor. In *Intelligent Vehicle Symposium'94*, Paris, France, October 1994.
 - [74] J. van den Elst, F. van Harmelen, G. Schreiber, and M. Thonnat. A functional specification of reusing software components. In *Sixth International Conference on Software Engineering and Knowledge Engineering*, pages 374–381. Knowledge Systems Institute, June 1994.
 - [75] J. van den Elst, F. van Harmelen, and M. Thonnat. Modelling Software Components for Reuse. In *Seventh International Conference on Software Engineering and Knowledge Engineering*, pages 350–357. Knowledge Systems Institute, June 1995.
 - [76] R. Vincent, S. Moisan, and M. Thonnat. Learning has a means to refine a knowledge-based system. In *Third Japanese Knowledge Acquisition for Knowledge-Based Systems Workshop*, Hatoyama, Japan, November 1994.
 - [77] R. Vincent and M. Thonnat. Planning, executing, controlling and replanning for ip program library. In *Proc. of 8th Artificial intelligence and Soft computing ASC'97*, July 1997.
 - [78] R. Vincent, M. Thonnat, and J.C. Ossola. Program supervision for automatic galaxy classification. In *Proc. of the International Conference on Imaging Science, Systems, and Technology CISST'97*, June 1997.
 - [79] V.T. Vu, F. Brmond, and M. Thonnat. Human behaviour visualisation and simulation for automatic video understanding. In *10-th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision'2002 WSCG 2002*, Plzen, Czech Republic, February 2002.
 - [80] V.T. Vu, F. Brmond, and M. Thonnat. Temporal constraints for video interpretation. In *15th European Conference on Artificial Intelligence*, Lyon, FRANCE, July 2002.
 - [81] V.T. Vu, F. Brmond, and M. Thonnat. Video surveillance: human behaviour representation and on-line recognition. In *Sixth International Conference on Knowledge-Based Intelligent Information and Engineering Systems KES2002*, Crema, Italy, September 2002.
 - [82] M. Xie and M. Thonnat. A method of 3d motion estimation from combined 2d-3d line segment data. In *2nd Singapore International Conference on Image Processing*, Singapore, NUS, September 1992.

- [83] M. Xie, L. Trassoudaine, J. Alizon, M. Thonnat, and J. Gallice. Active and intelligent sensing of road obstacles: application to the european eureka prometheus project. In *4th International Conference on Computer Vision*, Berlin, May 1993.
- [84] S. Yu and M. Thonnat. Determination of the apparent boundary of an object. In *International conference on Computer Vision and Pattern Recognition*, Urbana, Champaign, June 1992.
- [85] S. Yu and M. Thonnat. Using apparent boundary and convex hull for the shape characterization of foraminifera images. In *11th International Conference on Pattern Recognition*, The Hague, Nederland, August 1992.

Thèse

- [86] M. Thonnat. *Méthodes interactives de dépouillement de clichés astronomiques : restitution du fond et obtention de vitesses radiales*. PhD thesis, Thèse Université Marseille III, November 1982.

Rapports de Recherche

- [87] V. Clément and M. Thonnat. Integration of image processing procedures, ocapi: a knowledge-based approach. Rapport de Recherche 1307, INRIA, October 1990.
- [88] R. Ma and M. Thonnat. A robust and efficient stereo matching algorithm. Rapport de Recherche 1860, I.N.R.I.A., February 1993.
- [89] S. Moisan, R. Vincent, V. Thonnat, M. Clément, and J. van den Elst. Manuel de référence du Logiciel OCAPI, Version 2.0. Technical Report 183, I.N.R.I.A., 1995.
- [90] N. Rota, R. Stahr, and M. Thonnat. Ai techniques for visis human tracker. Rapport de Recherches 4138, INRIA, February 2001.
- [91] M. Thonnat. Automatic morphological description of galaxies and classification by an expert system. Rapport de Recherche 387, INRIA, March 1985.
- [92] M. Thonnat, V. Clément, and J. van den Elst. Supervision of perception tasks for autonomous systems: the ocapi approach. Rapport de Recherche 2000, I.N.R.I.A., June 1993.
- [93] R. Vincent, S. Moisan, and M. Thonnat. Une bibliothèque pour des moteurs de pilotage de programmes. Rapport de Recherche 3011, I.N.R.I.A., 1996.

Conférences Nationales avec comité de lecture

- [94] F. Brmond and M. Thonnat. Interprétation de séquences d'images et incertitude. In *Rencontres Francophones sur la Logique Floue et ses Applications*, décembre 1996.
- [95] V. Clément and M. Thonnat. Progal: un système expert en traitement d'images de galaxies. In *8ième Congrès Reconnaissance des Formes et Intelligence Artificielle*, Lyon, November 1991.

- [96] N. Dey, A. Boucher, and M. Thonnat. Modélisation et étude de la formation de l'image d'un objet 3d transparent. In *2^{me} colloque francophone méthodes et techniques optiques pour l'industrie.*, Tricastel, France, November 2001. Société Française d'Optique.
- [97] M-H. Gaudin and M. Thonnat. Étude pour l'automatisation de la reconnaissance d'organismes zooplaniques à l'aide d'un système expert. In *Sixième Congrès Reconnaissance des Formes et Intelligence Artificielle*, Antibes, November 1987.
- [98] C. Granger and M. Thonnat. Un système de vision fondé sur une méthode structurale de classification. In *Cognitive*, pages 137–142, Paris, June 1985.
- [99] C. Granger, M. Thonnat, and P. Vignard. Étude d'un système expert pour la classification de galaxies : Sygal. In *Les systèmes experts et leurs applications, Journées d'étude et exposition*, Avignon, May 1984.
- [100] A. Meygret, M. Thonnat, and M. Berthod. Un algorithme de stéréovision adapté aux scènes comportant à la fois des objets naturels et des objets manufacturés. In *Deuxièmes Journées Nationales du GRECO-PRC Communication-Homme-Machine*, Toulouse, January 1991.
- [101] S. Yu and M. Thonnat. Détermination du contour apparent d'un objet pour la description de sa forme. In *8ième Congrès Reconnaissance des Formes et Intelligence Artificielle*, Lyon, November 1991.

Chapitre 3

Thèmes de recherches

Ce document de synthèse décrit les travaux de recherche que j'ai effectués sur le couplage entre des techniques de vision par ordinateur et d'intelligence artificielle pour automatiser l'analyse et l'interprétation d'images. On parle aujourd'hui de Vision cognitive (voir le site du réseau européen d'excellence en cognitive vision ECVision <http://www.ecvision.info>). Ces travaux ont commencé en 1983 à l'Inria Sophia Antipolis, dans le projet de vision par ordinateur PASTIS, dirigé par Marc Berthod. Afin d'approfondir ces recherches, j'ai créé en 1995 le projet ORION, une équipe pluridisciplinaire à la frontière des domaines de l'**Intelligence Artificielle**, de la **Vision par Ordinateur** et du **Génie Logiciel**.

3.1 Objectif

Mon objectif est de comprendre la problématique de l'**interprétation sémantique d'images** et celle de la mise au point de systèmes **d'analyse automatique d'images**. Nous verrons comment ces deux problématiques sont liées et quelles solutions il est possible d'apporter.

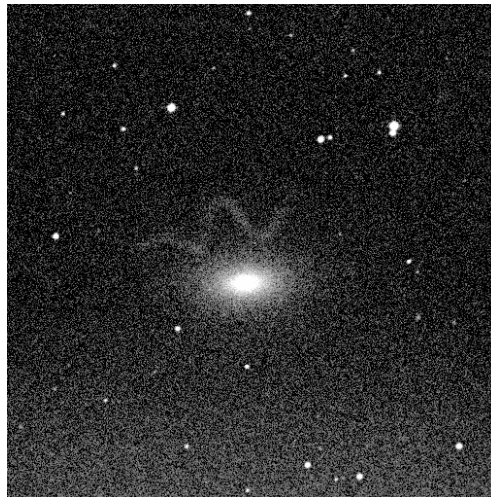


FIG. 3.1 – Exemple d'image à interpréter

Tout d'abord posons le *problème de l'interprétation d'image* avec un exemple très simple. Si l'on regarde une image telle que celle de la figure 3.1, il s'agit de répondre à la question que contient cette image, ou dit autrement, que représente-t-elle? Les réponses peuvent être l'image contient une tache brillante sur un fond sombre, ou il s'agit d'un objet astronomique nébuleux ou c'est une galaxie elliptique et, pour les plus initiés, il s'agit de la galaxie NGC4473 de l'amas de Virgo. Chacune de ces réponses est correcte, mais ces réponses sont différentes car elles ne correspondent pas au même niveau de connaissances a priori de l'interpréteur. Sur le plan philosophique la relation entre la perception (le monde sensoriel) et la représentation (le monde des idées) est un problème abordé depuis l'antiquité. Ici nous aborderons ce problème sous l'angle pragmatique de la construction de systèmes d'interprétation automatique. Mais est-ce possible puisque nous avons vu qu'il n'y a pas de réponse unique?

Le problème de la mise au point de systèmes d'analyse automatique d'images peut se formuler de la façon suivante: étant donné une image et un ensemble de programmes de vision par ordinateur, quelle suite de programmes faut-il appliquer à cette image? Pour reprendre notre exemple de la figure 3.1, les traitements à effectuer sur l'image peuvent consister à extraire et décrire le gros objet lumineux central ou bien à détecter et localiser tous les objets présents sur cette image y compris les objets qui apparaissent ici sous la forme de points lumineux. Ce problème amène à se poser plusieurs questions, dont la suivante, est-il possible d'automatiser complètement la mise au point de systèmes d'analyse d'images? La suite de programmes à appliquer va bien sûr dépendre des programmes disponibles et de leur fonction. Que faut-il savoir sur ces programmes pour automatiser leur sélection? Y-a-t-il une suite unique de traitements à effectuer sur l'image pour optimiser sa description? Si ce n'est pas le cas, le choix entre plusieurs possibilités de traitements, dépend-il de l'objectif recherché (par exemple on cherche à classer une galaxie ou à cartographier les étoiles et les galaxies)? Dans quelle mesure les choix effectués pour extraire des informations pertinentes sur l'image influencent-ils la qualité de l'interprétation?

3.2 Résumé des travaux présentés et plan du manuscrit

3.2.1 Résumé des travaux présentés

L'approche que je propose consiste en une **explicitation des connaissances** et en une **formalisation des raisonnements** permettant l'analyse et l'interprétation d'images. Si l'interprétation d'une image dépend de la connaissance a priori de l'interpréteur alors un système automatique doit lui aussi pouvoir disposer d'une connaissance plus ou moins sophistiquée. La sémantique de l'image n'est pas dans l'image elle-même mais dans la connaissance a priori du système d'interprétation (humain ou informatique).

Je propose donc de représenter explicitement cette connaissance a priori dans des bases de connaissances. Une même image pourra donc avoir différentes interprétations si elle est traitée par des systèmes d'interprétation ayant des bases de connaissances différentes; par exemple, une base de connaissance générale sur des objets astronomiques ou bien une base de connaissance spécialisée sur un modèle de classification de galaxies.

Si un système d'interprétation contient une base de connaissances, il est nécessaire d'étudier plusieurs aspects: tout d'abord, quel contenu mettre dans la base? c'est un problème bien connu en intelligence artificielle de modélisation de connaissances. Ensuite, comment rereprésenter cette connaissance? là aussi, les techniques d'intelligence artificielle nous apportent des solutions sous la forme de formalismes et de langages pour représenter le contenu des connaissances modélisées. J'ai choisi dans un premier temps (voir chapitres 4 et 5) d'utiliser des réseaux de frames et de règles de production mais d'autres outils sont également intéressants comme les chroniques (voir chapitre 6).

Les techniques d'intelligence artificielle et plus précisément les systèmes à base de connaissances apportent donc des éléments utiles. Cependant, pour construire des systèmes d'interprétation d'images efficaces, il est important de concevoir des systèmes dédiés avec d'une part des langages de représentation des connaissances adaptés et d'autre part un raisonnement

focalisé sur l'interprétation recherchée. En effet, le raisonnement à mettre en oeuvre, c'est à dire l'algorithme de gestion de la base de connaissances (le terme moteur est couramment employé), n'est pas le même selon que l'on cherche à classer un objet (voir chapitre 4) ou à reconnaître une séquence temporelle d'actions (voir chapitre 6). Je me suis posé le problème de discerner entre ce qui est commun à différents problèmes d'interprétation (le raisonnement et la formalisation des connaissances) de ce qui est spécifique au domaine d'application (le contenu d'une base de connaissances et l'analyse des images).

Dans un premier temps, j'ai étudié un sous-problème d'interprétation d'images, celui du **classement d'un objet complexe**. Il s'agit de retrouver la classe d'appartenance d'un objet connaissant les classes possibles, à ne pas confondre avec le problème de classification qui consiste à déterminer les classes. Je me suis attachée à étudier les différents aspects de ce problème qui sont la mesure de paramètres descriptifs significatifs de la forme perçue par le capteur, la modélisation et la représentation des classes d'objets et enfin la conception d'un processus de reconnaissance à base de connaissances explicites. Afin de prouver l'intérêt de l'approche proposée j'ai choisi de l'appliquer à des domaines très différents impliquant des connaissances réellement expertes et bien formalisées. J'ai donc sélectionné des applications scientifiques correspondant à des cas réels difficiles (galaxies, zooplanctons, poissons, foraminifères) pour lesquelles cette formalisation peut-être conduite de façon rigoureuse et me permet d'automatiser des processus complexes.

L'étude du problème de l'interprétation d'images comportant un objet de forme complexe m'a conduite à devoir gérer des connaissances non seulement sur les modèles des objets recherchés mais aussi sur les programmes effectuant des mesures sur l'image; j'ai donc été amenée à aborder le problème de la modélisation du raisonnement et des connaissances en traitement d'images. L'aspect auquel je me suis intéressée est l'automatisation de l'utilisation d'une bibliothèque de programmes de traitement d'image. Ces travaux ont conduit à la nouvelle thématique de recherche sur le **pilotage de programmes**. Plus précisément il s'agit de faciliter et même d'automatiser la planification et le contrôle d'exécution de programmes de traitements numériques contenus dans une bibliothèque.

Plus récemment, la production massive de vidéos a fait émerger un nouveau problème d'interprétation sémantique. Celui de l'étude de la problématique de l'interprétation automatique de séquences temporelles d'images ou **interprétation vidéo**. C'est à dire la conception de systèmes temps-réels complets effectuant une analyse du flux video (détection et suivi des objets mobiles) et une interprétation sémantique du comportement des acteurs (reconnaissance d'événements et de scénarios). L'approche que je propose consiste en un raisonnement spatio-temporel basé sur une connaissance a priori de l'espace 3D observé et des comportements à reconnaître. La représentation de ces connaissances nécessite une formalisation de ces connaissances. Dans le cas d'images scientifiques (en astronomie, ichtyologie, palynologie, etc...) cette formalisation peut s'appuyer sur des modèles théoriques. Dans le cas d'images vidéos un effort de modélisation plus important est nécessaire car les domaines applicatifs concernés, qui présentent un réel intérêt industriel, sont encore peu formalisés. Hormis le problème de la représentation des connaissances et de la conception d'algorithmes (ou moteurs) de gestion de ces connaissances la difficulté majeure réside dans le passage entre les données numériques issues des traitements d'images et ces connaissances symboliques.

3.2.2 Plan du manuscript

La suite du manuscript décrit successivement les travaux portant sur la reconnaissance d'objets complexes, le pilotage de programmes et enfin les travaux récents sur l'interprétation vidéo.

Plus précisément le chapitre 4 intitulé Classification d'objets décrit les travaux sur la reconnaissance de la classe d'objets complexes; les connaissances sur les hiérarchies de classes d'objets à reconnaître sont explicitées dans des bases de connaissances ainsi que les inférences pour interpréter les résultats des algorithmes d'analyses d'images. Le raisonnement est un raisonnement de classification ou catégorisation.

Le chapitre 5 intitulé Pilotage de programmes décrit les travaux sur l'utilisation d'une bibliothèque de programmes; les connaissances sur les programmes d'analyse d'images et sur leur utilisation sont explicitées dans des bases de connaissances. Le raisonnement est un raisonnement de planification et de contrôle d'exécution.

Le chapitre 6 intitulé Interprétation vidéo décrit les travaux sur l'interprétation sémantique de séquences vidéo; deux types de connaissances sont explicités, les modèles 3D des lieux à observer et les comportements à reconnaître. Le raisonnement est un raisonnement d'interprétation.

Les perspectives de recherches dans le prolongement de ces travaux sont décrits dans le chapitre 7.

Chapitre 4

Classification d'objet

Ce chapitre décrit les travaux que j'ai effectués sur l'interprétation sémantique d'une image contenant un objet à classer. L'approche suivie consiste dans le couplage de techniques de vision par ordinateur pour l'analyse de l'image et de techniques d'intelligence artificielle pour la modélisation et la mise en oeuvre des connaissances a priori sur les objets à reconnaître.

Ce chapitre est structuré de la manière suivante, après une introduction présentant le problème de la reconnaissance d'objets complexes, le principe général de l'approche est décrit, puis sont successivement présentés les travaux sur SYGAL pour la classification de galaxies à partir d'une image 2D, les travaux sur la classification de foraminifères à partir de plusieurs vues 2D et les travaux sur la reconnaissance de zooplanktons avec retour sur des traitements bas-niveaux. Enfin la dernière partie décrit les besoins de communication entre interprétation d'image et analyse d'image.

4.1 Introduction

Le problème auquel on s'intéresse ici est de trouver la classe d'appartenance d'un objet naturel en utilisant une taxonomie prédéfinie.

Les travaux sur la reconnaissance d'objet en vision par ordinateur se sont beaucoup intéressés à la reconnaissance d'objets manufacturés rigides en utilisant, ce qui est logique, des contraintes géométriques. On peut se référer au livre de Grimson pour un bon état de l'art sur ces questions [34]. La reconnaissance d'objets naturels est intéressante car elle pose de nouveaux problèmes. En effet, alors que pour les objets manufacturés rigides, les individus d'une même classe sont semblables, il n'en est pas de même pour les objets naturels. Par exemple les individus de la classe personne (i.e. les humains) sont très différents en taille, corpulence ou morphologie. La biologie et la botanique nous offrent également de nombreux exemples d'une telle diversité interclasse, c'est à dire entre les individus d'une même classe. Par contre, dans ces domaines, le plus souvent il est possible de s'appuyer sur l'existence d'une taxonomie décrivant les classes et leur caractéristiques.

La démarche proposée va donc consister en l'utilisation d'une connaissance a priori, la taxonomie des classes pour compenser la difficulté de reconnaissance inhérente aux objets naturels due à leur variabilité d'apparence. Le problème auquel on s'intéresse est un problème de catégorisation, c'est à dire retrouver la classe d'un objet, et non d'identification, c'est à dire retrouver une instance de classe ou un individu particulier. Il est à noter que les travaux récents en vision par apparence qui utilisent des résultats classiques d'analyse statistique [30, 31] telle l'analyse en composantes principales ne sont pas adaptés à la reconnaissance de classes d'objets naturels. Par contre ces techniques sont pertinentes pour la reconnaissance d'individus particuliers ou l'identification d'objets [17]. Il est donc normal de retrouver cette approche par apparence pour des problèmes d'identification comme par exemple l'indexation de bases de données de visages.

Dans la suite de ce chapitre il sera donc question de déterminer la classe d'un objet naturel. La description numérique des objets est réalisée par des algorithmes de vision et la classification par un système à base de connaissances dédié à la classification d'objets. Les hypothèses de travail sont les suivantes: tout d'abord on suppose que les objets sont complexes et struc-

turés et qu'ils sont isolés sans recouvrement dans l'image. Une classe a généralement plusieurs apparences dans l'image. Par exemple, l'objet est un objet 3D mais il est observé avec une seule image 2D. Un autre cas fréquent est celui d'un objet non rigide (par exemple un animal ou un être humain) avec différentes tailles, postures ou positions possibles. La seconde hypothèse que l'on fait est que les classes sont connues et que la description symbolique et numérique d'objets prototypes appartenant à chaque classe est possible.

Cette méthode de reconnaissance d'objet a été appliquée à différents types d'objets naturels. J'ai conçu cette méthode pour un problème d'astronomie, la classification de galaxies selon leur type morphologique [Tho89]. Les objets sont des objets lumineux 3D sans frontières marquées, plus ou moins distants et de taille variable. Les classes sont fonction de modèles photométriques et morphologiques. J'ai ensuite cherché à appliquer cette méthode à d'autres problèmes de reconnaissance d'objets naturels dans le cadre de thèses que j'ai encadrées sur ce thème: la thèse de Shan-Liu-Yu [42] pour la reconnaissance de foraminifères à partir de plusieurs vues et la thèse de Marie-Hélène Gandelin [32] pour la reconnaissance de zooplanctons marins. Les foraminifères sont des organismes naturels fossilisés rigides. Les zooplanctons sont des organismes 3D vivants plus ou moins transparents et non-rigides. Dans les deux cas, foraminifères et zooplanctons, les classes correspondent à la taxonomie biologique.

Dans la suite de ce chapitre je présente le principe de la reconnaissance de classe avec le moteur de classification CLASSIC puis successivement les systèmes de reconnaissance de galaxies, de foraminifères et de zooplanctons. Un chapitre de livre [TB89] et deux articles de revues [78, 64] joints en annexe B décrivent plus en détail ces travaux.

4.2 L'approche CLASSIC

Les objets du monde réel peuvent être modélisés à l'aide de primitives géométriques [5] ou de modèles CAO [36]; dans l'approche proposée on cherche à utiliser des modèles tels que les expriment les experts (biologistes, astronomes,...), c'est-à-dire des modèles symboliques organisés hiérarchiquement en classes et sous-classes. L'approche méthodologique choisie, présentée figure 4.1, est donc la suivante : un ensemble d'algorithmes de vision détecte et décrit les objets contenus dans les images, puis un système à base de connaissances, construit avec le moteur de classification CLASSIC, les classe en utilisant des modèles symboliques hiérarchiques explicites.

Le moteur de classification CLASSIC a été conçu en 1985 [Gra85a], [Gra85b]. Sa philosophie repose sur les concepts suivants:

- l'utilisation simultanée d'objets structurés, les classes, contenant la connaissance descriptive des objets à classer, et de règles de production décrivant une connaissance heuristique,
- l'utilisation d'un mécanisme de comparaison floue dérivant de la théorie des possibilités et des ensembles flous, permettant la manipulation de données bruitées, incomplètes ou imprécises.

Il a été mis au point pour le développement d'un système expert de classification de galaxies à partir d'images monoculaires, SYGAL [Tho89, TB89]. Au fil des années, sans remise en cause

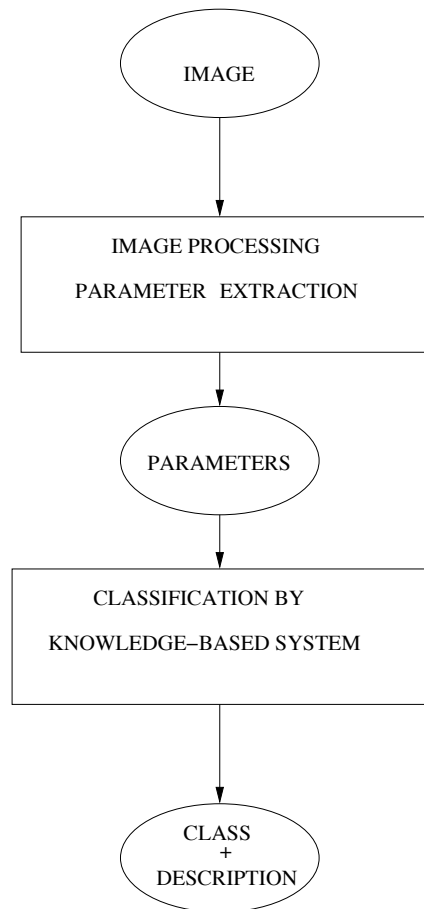


FIG. 4.1 – Schéma de l'approche

de sa philosophie initiale, CLASSIC a évolué dans un esprit d'élargissement des concepts manipulés, notamment l'introduction des possibilités d'appel à des procédures externes de calcul en cours de raisonnement [Cle90] et le développement d'une interface graphique multifenêtres [Ilo88].

4.2.1 L'organisation de la connaissance

On présente ici les différents éléments permettant d'exprimer la connaissance dans CLASSIC : les classes, l'arbre des classes et les règles d'inférence.

Les classes

Les modèles des différentes classes d'objets connus sont décrits sous la forme de classes, objets structurés (ou frames) comprenant des attributs et organisés selon un réseau arborescent

pour modéliser les relations existant entre les différents objets. Lorsqu'une classe a des sous-parties, chacune est décrite par une arborescence.

Exemple de classe :

```
Class Tiger
  superclass : Carnivor
  teeth : pointed
  pattern : striped
  color : (yellow black)
```

L'arbre des classes

Afin de structurer la connaissance, les classes peuvent être reliées pour former un arbre de spécialisation partant de la classe la plus générale vers les plus précises. Dans l'exemple précédent, la classe Tiger est une sous-classe de la classe plus générale Carnivor. La figure 4.2 montre un exemple d'arbre de classes utilisé pour décrire les types morphologiques de galaxies.

Une telle organisation remplit deux objectifs:

- La structuration de la connaissance : par l'héritage des descriptions des classes de mère en filles la redondance est éliminée et la modularité est augmentée.
- Le contrôle du raisonnement : l'organisation hiérarchique de la connaissance définit des étapes naturelles de raisonnement et permet un contrôle efficace.

Les règles d'inférences

Les règles d'inférence sont utilisées pour compléter la description d'un objet en cours de classement en déduisant la valeur de certains attributs à partir de celle(s) d'autre(s) attributs). Elles ont pour rôle d'augmenter le niveau d'abstraction de la description de l'objet à classer. Elles permettent donc d'effectuer le passage d'une description numérique obtenue par les algorithmes de vision à une description symbolique.

- Exemple de transformation d'information numérique en information symbolique:
 if temperature > 38 then fever = high
 if length > width × 10 then shape = elongated

Les règles permettent également d'augmenter le niveau d'abstraction des valeurs symboliques.

- Exemple d'augmentation du niveau d'abstraction symbolique:
 if shape of contour1 member of (average, spiral)
 and shape of contour2 = average
 then arms of galaxy = visible

Les règles d'inférence sont automatiquement regroupées en de petites bases de règles et attachées aux différentes classes de l'arbre des classes.

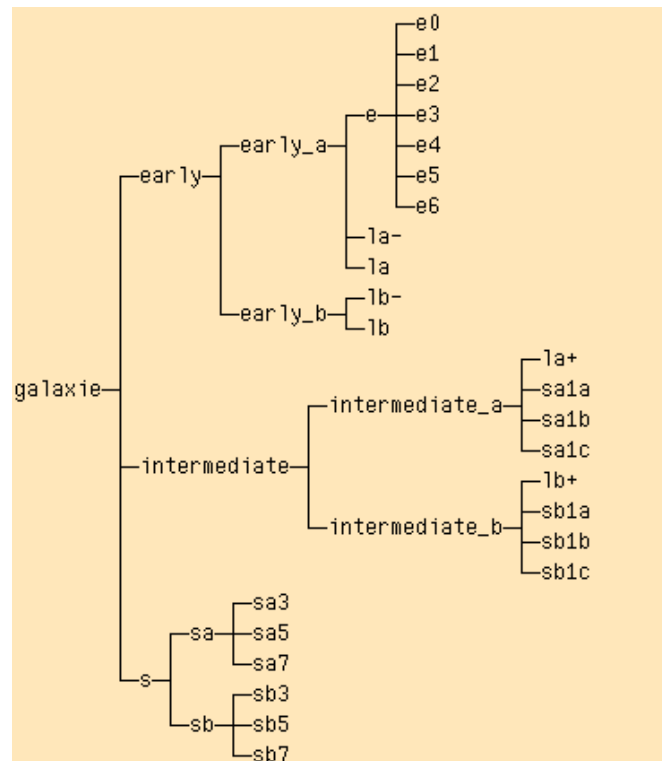


FIG. 4.2 – Un arbre de classes de galaxies.

CLASSIC étant lui-même écrit en Le Lisp, il est possible de faire appel à des fonctions Lisp dans les règles et dans les facettes *valeur à calculer* des attributs de classes. Néanmoins Lisp apparaissant inadapté dans de nombreux cas de calcul, et plus particulièrement en traitement d'images, une interface a été réalisée [Cle90] [Ilo88], pour appeler des procédures C ou Fortran.

4.2.2 La base de faits

La base de faits donnée en entrée au système à base de connaissances correspond à l'objet à classer.

Un objet est une instance particulière de classe décrite par son nom, sa classe, la totalité des attributs de la classe principale.

L'objet à classer correspond dans cette approche à l'ensemble des paramètres issus de la phase d'analyse d'image, et à des paramètres reflétant le contexte local de l'objet. Par exemple, les conditions d'observation, le lieu de pêche, la profondeur, la maille du filet sont des paramètres utiles pour restreindre la liste des classes possibles dans le cas des zooplanctons.

4.2.3 Le fonctionnement du moteur

Classer un objet, c'est ici rechercher le(s) type(s) qui lui correspond(ent) le mieux dans l'arbre des classes, donner une mesure de cette correspondance et compléter la description de l'objet.

Pendant la classification l'objet à classer est comparé à chaque noeud de l'arbre des classes, de la racine vers les feuilles, afin de trouver la classe qui correspond le mieux à l'objet. Chaque classe dans la hierarchie est associée à une base de règles. La base de règles est activée quand l'objet est compatible avec la classe pour compléter sa description. Quand plus aucune règle ne fournit de l'information la comparaison va vers le noeud suivant, et ainsi de suite.

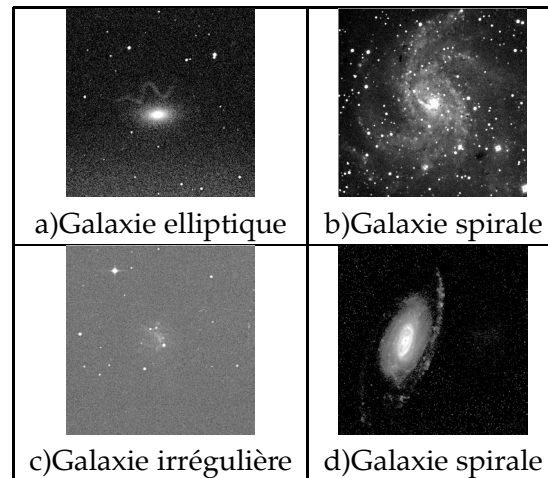
Une étape très importante est celle de la mise en correspondance des données d'entrée avec les valeurs définies comme admissibles pour un type d'objet donné au niveau de la racine de l'arbre principal. Le parcours de l'arborescence ne sera en effet déclenché que dans la mesure où les caractéristiques de l'objet inconnu seront compatibles avec les valeurs descriptives admissibles pour un type d'objet donné.

Le raisonnement en CLASSIC conduit récursivement trois phases: abstraction des données, mise en correspondance objet-classe et raffinement de classe:

- La phase d'abstraction des données permet le passage des mesures obtenues par les algorithmes d'analyse d'image vers les descriptions symboliques utilisées par les experts. Cette phase est réalisée en activant les règles d'inférence.
- La phase de mise en correspondance permet de comparer l'objet à classer à une classe particulière. La comparaison est effectuée en deux étapes:
 - (1) Calcul des coefficients de compatibilité et d'incompatibilité avec la classe sur la base d'une combinaison des coefficients de compatibilité et d'incompatibilité de chaque attribut de la classe considérée.
 - (2) Validation ou rejet de la classe par comparaison de ces coefficients avec des seuils de compatibilité et d'incompatibilité.
- La phase de raffinement a pour rôle de classer plus précisément l'objet sachant sa description courante. De façon standard, si une classe est validée, le système sélectionne ses filles (on descend le plus loin possible dans l'arbre), sinon il y a retour vers une classe précédemment sélectionnée et laissée en suspens et ce jusqu'à ce que l'ensemble des branches ait été parcouru; le contrôle peut être redéfini de façon à ce que le raisonnement s'arrête dès qu'une solution est trouvée (validation d'une feuille) ou dès qu'une branche a été explorée.

4.3 Le système SYGAL

Cette partie de'crit le système SYGAL dont l'objectif est l'automatisation complète de la classification de galaxies à partir d'une image monoculaire 2D. Ces travaux sont décrits plus en détail dans [Tho85], [Tho89] et [TB89].

FIG. 4.3 – *Différentes classes de galaxies*

Le système prend en entrée la description numérique d'une galaxie extraite à partir d'une image 2D contenant une galaxie plus des étoiles. La figure 4.3 montre des images typiques. Les classes correspondent au système de classification de G. de Vaucouleurs et sont fonction de modèles morphologiques et photométriques.

La figure 4.4 montre le synoptique de ce système.

Il a été développé pour la classification automatique d'images de galaxies brillantes prises au télescope de Schmidt au *CERGA*.¹ Le pas d'échantillonnage des images était de (20×20) microns et la taille des images était de (512×512) pixels.

Parmi les caractéristiques de ces images on notera que:

- ces images ont un très faible rapport signal sur bruit, qui devient critique pour les objets faibles;
- il n'y a pas de limite claire discriminant la galaxie du fond;
- d'autres objets peuvent être proches de la galaxie ou même partiellement superposés à la galaxie.

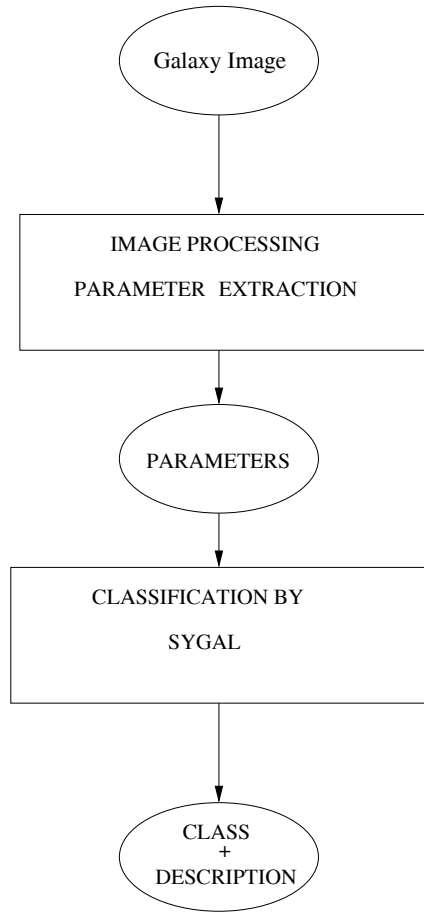
4.3.1 Prétraitement

La phase de prétraitement était une phase entièrement automatique qui était réalisée de façon purement procédurale; c'est à dire de manière déterministe et sans utiliser de système à base de connaissances. Le chapitre suivant 5 décrit comment incorporer des connaissances sur les programmes de traitement d'images.

Cette phase réalise les fonctions suivantes:

- localisation de la position exacte de la galaxie;
- détermination des limites de la galaxie;

1. CERGA: Centre d'Etudes et de Recherches Geodynamiques et Astronomiques, Grasse FRANCE

FIG. 4.4 – *Synoptic of SYGAL*

- calcul et soustraction d'une carte du fond de ciel;
- suppression des objets du voisinage;
- elimination du bruit.

4.3.2 Extraction de paramètres

Cette partie décrit les différents paramètres extraits des galaxies isolées pour décrire leur forme.

Axes principaux

Le premier paramètre calculé est l'orientation de l'axe principal de la galaxie. La valeur de sa direction est donnée par l'angle θ qui minimise le moment d'inertie J :

$$J = \sum_{x,y} d(x,y)(y \cos \theta - x \sin \theta)^2$$

Soit θ_1 et θ_2 deux solutions de

$$\tan(2\theta) = 2 \frac{\sum_x \sum_y d(x,y)xy}{\sum_x \sum_y d(x,y)(x^2-y^2)} \text{ with } J(\theta_1) < J(\theta_2)$$

- θ_1 est l'orientation de l'axe principal;
- θ_2 est l'orientation de l'axe secondaire (axe orthogonal à l'axe principal).

Un ensemble de courbes représentant l'intensité de la galaxie à partir du centre dans différentes directions sont construites. Ces directions sont les deux demi-axes principaux et les deux demi-axes secondaires.

Ellipticité

La forme de la galaxie est très dépendante du point de vue sous lequel l'objet est observé. Par conséquent une mesure importante est l'ellipticité apparente de la galaxie:

$e = 1 - \frac{b}{a}$ a et b étant respectivement la largeur de l'axe principal et la largeur de l'axe orthogonal. Ces valeurs sont mesurées sur les courbes précédentes représentant la largeur de la galaxie dans la direction de l'axe principal et de l'axe secondaire. Comme ces courbes sont bruitées et qu'elles ont une pente douce en raison de la diffusion dans l'émulsion, on utilise une largeur intégrée. Soit $F(x)$ la distribution moyenne de l'intensité le long de l'axe principal, et $f(x)$ la distribution moyenne de l'intensité le long de l'axe secondaire;

soit w_1 et w_2 respectivement la largeur de la galaxie dans les deux directions:

Les largeurs sont déduites de:

$$\int_0^{w_1} F(x)dx = \frac{3}{4} \int_0^\infty F(x)dx$$

et

$$\int_0^{w_2} f(x)dx = \frac{3}{4} \int_0^\infty f(x)dx$$

Finalement l'ellipticité apparente est donnée par:

$$e = 1 - \frac{w_2}{w_1}$$

Taille

Afin d'estimer la fiabilité des paramètres mesurés on calcule la taille S de la galaxie sous l'hypothèse d'une forme elliptique:

$$S = \pi w_1 w_2$$

Profil moyen

En utilisant la mesure de l'ellipticité e on construit la courbe de distribution moyenne de l'intensité à partir du centre $g(x)$. La courbe $g(x)$ est la distribution radiale de l'intensité dans le nouveau référentiel

$$(X, Y = \frac{w_1}{w_2 y})$$

où X est la coordonnée de l'axe principal et Y la coordonnée de l'axe secondaire.

Profil projeté

Des études théoriques [?] [?] ont montrées que la variation radiale de la luminosité à partir du centre des galaxies est la somme de deux fonctions;

La première fonction est la composante sphérique caractérisant les galaxies elliptiques,

- l'intensité I est donnée en fonction du rayon r par:

$$I_1(r) = a_1 e^{-b_1 r^{\frac{1}{4}}}$$

- la densité $d = \log(\frac{I}{I_0})$ est donnée en fonction du rayon r par:

$$d_1(r) = \alpha_1 - \beta_1 r^{\frac{1}{4}}$$

La seconde fonction est la composante plate caractérisant le disque des galaxies lenticulaires et spirales.

- l'intensité I est donnée en fonction du rayon r par:

$$I_2(r) = a_2 e^{-b_2 r}$$

- la densité $d = \log(\frac{I}{I_0})$ est donnée en fonction du rayon r par:

$$d_2(r) = \alpha_2 - \beta_2 r$$

Afin d'éviter l'influence du point de vue sous lequel la galaxie est observée on construit une nouvelle courbe: le profil projeté d_p . Cette courbe est obtenue par projection orthogonale des densités d de la galaxie le long de l'axe principal.

$$d_p(u) = \sum_{x,y, u=x \cos \theta + y \sin \theta} d(x,y)$$

avec θ la direction de l'axe principal, x et y les coordonnées des pixels appartenant à la galaxie. A partir de cette courbe on extrait deux paramètres:

- le paramètre (profile) qui est le rapport des erreurs au sens des moindres carrés faites en approximant le profil projeté respectivement avec d_1 et d_2 pour r plus grans que le rayon du bulbe. L'estimation du rayon du bulbe est obtenu à partir du profil projeté construit précédemment.

$$profile = \frac{\sum_{r=r_{bulge}}^{r_{limit}} (d_p(r) - d_2(r))^2}{\sum_{r=r_{bulge}}^{r_{limit}} (d_p(r) - d_1(r))^2}$$

- le paramètre (*linear_err*) mesurant directement l'erreur au sens des moindres carrés faite en approximant le profil projeté complet de $r=0$ à $r = r_{max}$ avec une fonction linéaire.

$$linear_err = \sum_{r=0}^{r_{limit}} (d_p(r) - d_1(r))^2$$

Contours

Construction des contours

Afin de décrire la variation de la forme dans les différentes régions de la galaxie, on doit extraire plusieurs isophotes ou courbes d'égales intensité. Les isophotes doivent être complètement représentatifs de chaque région à l'intérieur de la galaxie et ils doivent être construits sans processus interactifs. Tout d'abord on calcule cinq seuils sur la distribution de la densité le long de l'axe principal $F(x)$, puis ces seuils sont utilisés pour obtenir cinq images binaires sur lesquelles on applique un algorithme de détection de contour.

Les cinq seuils $t_i, i \in [1, \dots, 5]$ sont donnés:

$$\int_0^{t_i} F(x) dx = \frac{1}{n_i} \int_0^\infty F(x) dx$$

avec n_i respectivement égal à 0.20, 0.50, 0.75, 0.85 et 0.90

Pour chaque image seuillée on utilise un détecteur de contour de Sobel, puis on recherche la plus grande chaîne de contour grâce à l'algorithme de chaînage de la bibliothèque INRIMAGE [?].

Paramètres extraits de ces contours

Pour chacun de ces contours on extrait cinq paramètres caractérisant sa forme: l'ellipticité, l'angle de l'axe principal, la position relative du centre, la compacité et la distance à la plus proche ellipse.

l'angle de l'axe principal:

De la même manière que pour l'image, on calcule l'angle θ_i qui minimise le moment d'inertie J_i et l'angle orthogonal ψ_i .

l'ellipticité :

Pour chaque contour on calcule l'ellipticité (la description a été faite plus-haut).

la position relative du centre:

Pour chaque contour on calcule la distance euclidienne (*center_err*) entre le centre du bulbe et le centre de gravité du contour g_i .

la compacité:

La compacité C mesure la rotondité de la forme et est minimale pour un cercle ($c \leq 4\pi$).

$$c = \frac{Perimeter^2}{area}$$

En fait on tient compte de la valeur estimée de l'ellipticité $e = 1 - \frac{b}{a}$ et on mesure la quantité:

$$C = \frac{b}{a} \frac{c}{e\pi}$$

La distance à la plus proche ellipse:

Le dernier paramètre (*ellipse_err*) est la distance entre chaque contour et l'ellipse qui a son axe principal dans la direction θ_i , une ellipticité et qui est centrée au centre de gravité g_i .

Une normalisation est réalisée de la façon suivante:

Soit S_E la surface de l'ellipse, S_C la surface du contour et S_{dif} la somme des surfaces entre les deux courbes fermées

$$ellipse_err = \frac{S_{dif}}{S_E + S_C}$$

S_d est approximé en divisant la surface en petits triangles, dont les sommets sont respectivement les points de contour C_i , les projections de ces points sur l'ellipse p_i et les intersections du contour et de l'ellipse P_j . Soit x_{p_i} et y_{p_i} les coordonnées des projections p_i et x_i et y_i les coordonnées des points C_i appartenant au contour.

$$x_{p_i} = a \cos(\arctan(\frac{ay_i}{bx_i}))$$

et

$$y_{p_i} = b \sin(\arctan(\frac{ay_i}{bx_i}))$$

4.3.3 Exemple de paramètres mesurés

On montre ci-dessous les paramètres extraits de la galaxie NGC4569.

orientation : 79.33
ellipticity : 0.50
linear_err : 0.039
profile : 2.77
area : 38013.3

<i>Contours :</i>	<i>center_err</i>	<i>ellipse_err</i>	<i>compactity</i>	<i>angle</i>	<i>ellipticity</i>
<i>c1 :</i>	1	0.06	1.6	-6.3	0.09
<i>c2 :</i>	2	0.21	5.4	70.8	0.36
<i>c3 :</i>	11	0.16	7.1	83.6	0.58
<i>c4 :</i>	9	0.21	11.8	76.8	0.47
<i>c5 :</i>	7	0.14	8.6	76.1	0.50

4.3.4 Classification automatique

Cette phase est réalisée par un système à base connaissances construit avec le moteur CLAS-SIC d'écrit précédemment et conçu pour résoudre ce problème.

La base de connaissances

Les paramètres symboliques

La description d'une classe de galaxies est faite par un astronome avec des descripteurs symboliques qui représentent les différents éléments structurels des galaxies. J'ai choisi d'expliquer ces descripteurs dans la base de connaissance sous la forme de paramètres symboliques en complément des paramètres numériques décrits précédemment.

- *Le paramètre symbolique H :*
Ce paramètre symbolique correspond à l'étiquette de codage hiérarchique; il prend les valeurs des classes possibles:
H: E, L, S, E0, E1, ..., E6, LA, LB, LA-, LB-, LA+, LB+, SA, SB, SA0, ...SA7, SB0, ..SB7
Comme les galaxies irrégulières ne sont pas encore classées les valeurs correspondantes ne sont pas disponibles.
- *Le paramètre symbolique T :*
Ce paramètre symbolique correspond à l'étiquette de codage continue; bien qu'il prenne des valeurs numériques, ce n'est pas un paramètre mesuré, chaque valeur représente une étiquette:
T: -5, -3, -2, -1, 0, 1, ..., 7
Les valeurs supérieures à 7 ne sont pas considérées car elles correspondent à des galaxies irrégulières.
- *Le paramètre symbolique barre (bar):*
Ce paramètre spécifie la connaissance sur la présence éventuelle d'une barre transversale; les valeurs possibles sont donc:
bar: present, absent, unknown
- *Le paramètre symbolique forme (shape) :*
Ce paramètre qualifie la structure générale de la galaxie; il peut prendre les valeurs:
shape: elliptical, average, spiral
- *Le paramètre symbolique isophotes :*
Ce paramètre montre le degré de perturbation des contours; les valeurs possibles sont:
isophotes: smooth, normal, distorted
- *Le paramètre symbolique bras (arms) :*
Ce paramètre qualifie les bras de la galaxie; ces valeurs sont:
arms: absent, incipient, evident, branched
- *Le paramètre symbolique bulbe (bulge) :*
Ce paramètre mentionne si le bulbe central peut-être détecté sur le profil projeté; les valeurs sont donc:
bulge: visible, invisible
- *Le paramètre symbolique aplatissement (flatness) :*
Ce paramètre qualifie l'ellipticité apparente de la galaxie.
flatness: null, negligible, very_faint, faint, light, average

- *Le paramètre symbolique centrage (centring) :*
Le paramètre centrage mesure la différence entre le centre de la galaxie et le centre des contours.
centring: good, average, mediocre, indifferent
- *Le paramètre symbolique concavité_profil (profile_concavity) :*
Ce paramètre précise si la concavité de la courbe du profil projeté est grande ou non.
profile_concavity: great, average, null
- *Le paramètre symbolique validité (validity) :*
Ce paramètre important spécifie la qualité de l'observation et donc la qualité des paramètres mesurés; il dépend de la taille de l'image de la galaxie; ces valeurs sont:
validity: good, bad
On a vu dans le paragraphe précédent que pour chaque galaxie traitée cinq contours sont construits; ces contours sont décrits à la fois avec des paramètres symboliques et des paramètres numériques. Les paramètres numériques correspondent aux paramètres qui ont été extraits sur chaque contour: center_err, ellipse_err, compacity, angle, ellipticity. Trois paramètres symboliques décrivent également les contours dans les différentes régions de la galaxie.
- *Le paramètre symbolique contour c_i : shape :*
Ce paramètre qualifie la forme de chaque contour de la même façon que le paramètre symbolique shape; il prend les mêmes valeurs que le paramètre global:
contour c_i : shape: elliptical, average, spiral
- *Le paramètre symbolique contour c_i : isophotes :*
Ce paramètre indique le degré de perturbation de chaque contour; comme le paramètre global associé à la galaxie il prend les valeurs suivantes:
contour c_i : isophotes: smooth, normal, distorted
Le paramètre symbolique contour c_i : flatness :
Ce paramètre indique si l'ellipticité de chaque contour n'est pas trop grande
contour c_i : flatness: valid

L'arbre des classes

La base de connaissance contient 37 objets structurés ou classes pour définir les différentes classes de galaxie. Ces classes sont organisés en une hiérarchie qui reflète la hiérarchie des classes astronomiques d'après le modèle de G. de Vaucouleurs.

On montre tout d'abord la hiérarchie des 37 classes sous forme iconique figure 4.5

L'arbre des classes est montré figure 4.6 avec chaque classe représentée par son étiquette symbolique H.

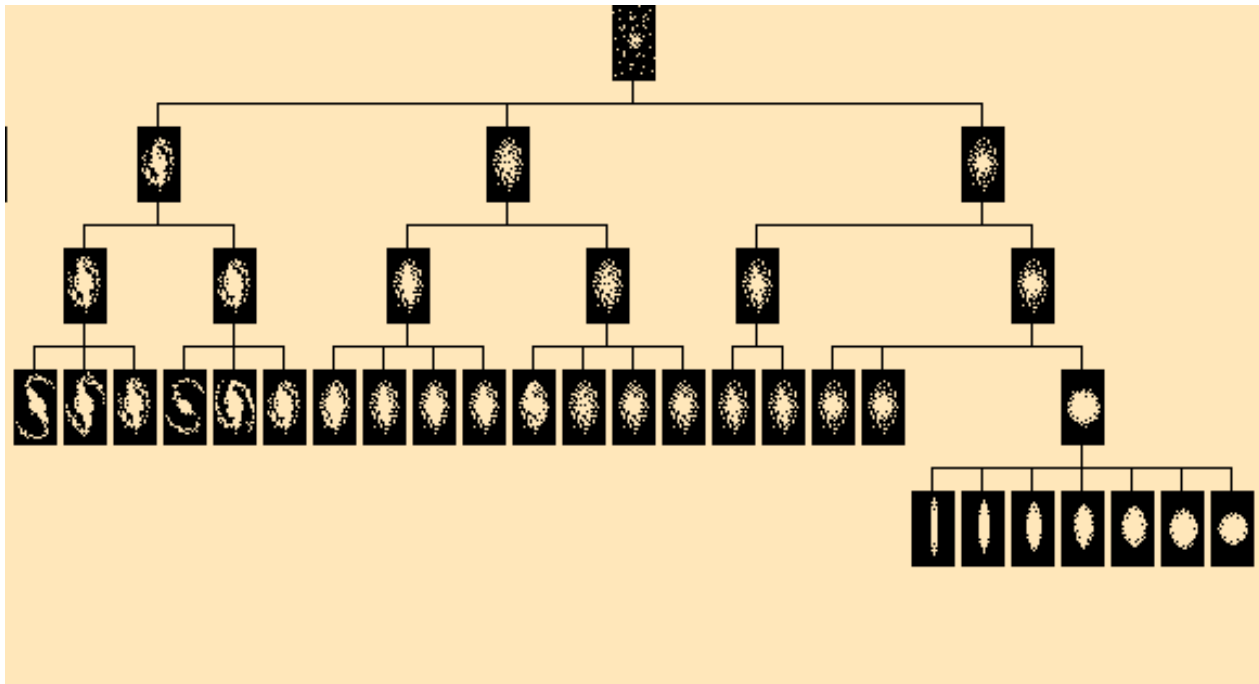


FIG. 4.5 – L'arbre des classes (représentation iconique).

Les classes

Les classes expriment de façon déclarative la connaissance sur les types de galaxie à l'aide d'attributs. Ces attributs peuvent être à valeur symbolique, numérique ou être eux-mêmes des objets structurés comme dans l'exemple suivant:

Class contour

compacity: [0, 10000];
 ellipticity: [-0.9, 1];
 angle: [-90, 90];
 ellipse_err: [0, 10000];
 center_err: [0, 10000];
 shape: elliptical, average, spiral;
 isophotes: unknown;
 flatness: unknown.

On montre ci-dessous des exemples de classes décrivant la classe générale Galaxy et les

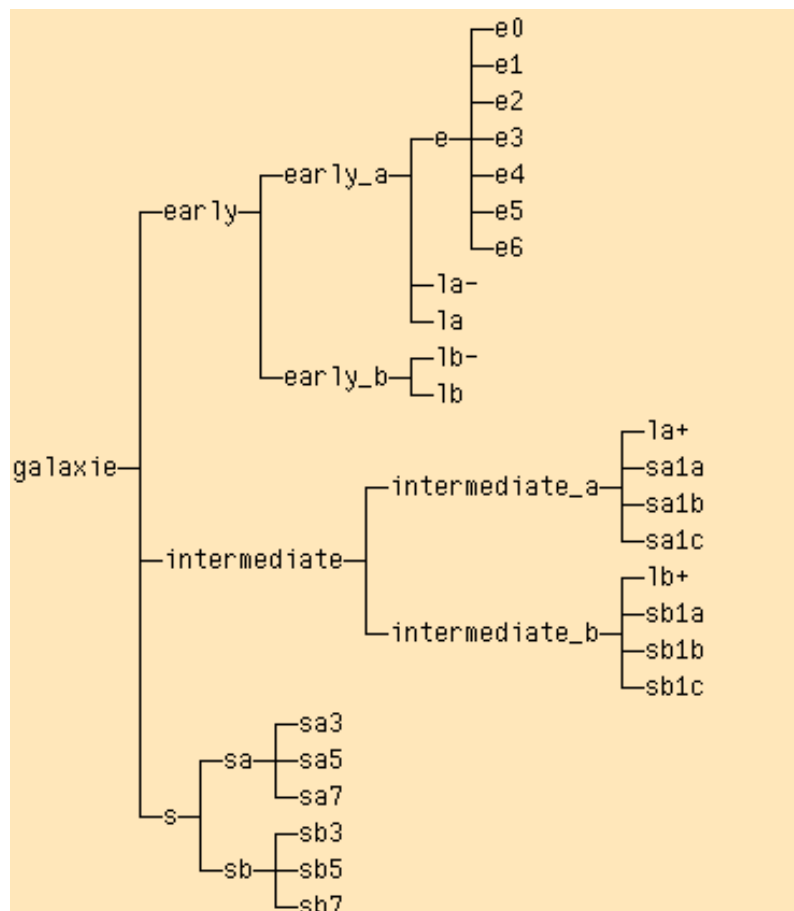


FIG. 4.6 – L'arbre des classes.

sous-classes SB7 et E0.

Class Galaxy

T: $[-5, 10]$
 profile: $[0, 10000]$
 ellipticity: $[-0.9, 1]$
 orientation: $[-90, 90]$
 linear_err: $[0, 100000]$
 area: $[0, 900000]$
 c1: Class contour
 c2: Class contour
 c3: Class contour
 c4: Class contour
 c5: Class contour

Class E0		Class SB7	
upperclass	<i>E</i>	upperclass	<i>SB</i>
H:	<i>E0</i>	H:	<i>SB7</i>
T:	<i>[-5,-5]</i>	T:	<i>[7,7]</i>
shape:	<i>elliptical</i>	shape:	<i>spiral</i>
bulge:	<i>visible</i>	validity:	<i>good</i>
validity:	<i>good</i>	bar:	<i>present</i>
profile_concavity:	<i>great</i>	centring:	<i>indifferent</i>
flatness:	<i>null</i>	arms:	<i>late</i>
bar:	<i>absent</i>		
isophotes:	<i>smooth</i>		
centring:	<i>good</i>		
arms:	<i>absent</i>		

La base de règles

Il y a 106 règles d'inférence. Elles représentent la connaissance déductive de l'expert. Les règles sont utilisées pour affecter une valeur symbolique à un attribut de l'objet.

Exemple de règle:

Rule 8:

If ellipticity of contour3 $\gg$ ellipticity 0.1
 ellipticity of contour3 $>$ 0.4
then bar be present

"As the ellipticity of the third contour is greater than the global ellipticity of the galaxy and is important, the bar is present."

La base de règles est constituée de 5 types de règles:

- Les règles utilisées pour interpréter les paramètres photométriques (comme *linear_err* ou *profile*) et pour affecter une valeur symbolique au paramètre *bulge* ou *profile_concavity*;
- Les règles utilisées pour interpréter la forme de chaque contour en fonction des valeurs des paramètres *ellipticity* et *compacity*;
- Les règles utilisées pour interpréter la forme globale de la galaxie en fonction de la forme de chaque contour et du paramètre *center_err* de chaque contour;
- Les règles utilisées pour détecter la présence d'une barre en fonction des paramètres *orientation* et *ellipticity* de la galaxie et, *angle* et *ellipticity* des contours;
- Les règles utilisées pour préciser la qualité des paramètres d'entrée et donc la qualité de la classification en fonction du paramètre *area*.

La base de faits

Les faits correspondent aux données du cas courant et sont temporairement ajoutées à la base; pour cette application, les faits représentent l'information associée à une galaxie. La base

de faits contient uniquement l'objet à classer. Cet objet a la même structure que la classe racine Galaxy. Au début toutes les valeurs de ses attributs sont inconnues, sauf ceux correspondant aux paramètres mesurés. A la fin de la classification les paramètres symboliques de l'objet ont la même valeur que l'objet structuré représentant la classe de l'objet.

4.3.5 Exemple de session

Description initiale de la galaxie NGC4473 : (paramètres mesurés)

```
orientation :  -3.18
ellipticity :    0.4
linear_err :   0.061
profile :      0.35
area :        10935
```

<i>contours :</i>	<i>center_err</i>	<i>ellipse_err</i>	<i>compactity</i>	<i>angle</i>	<i>ellipticity</i>
<i>c1 :</i>	0	0.07	1.6	-1.8	0.41
<i>c2 :</i>	1	0.04	1.9	-1.5	0.35
<i>c3 :</i>	0	0.05	3.2	-2.3	0.41
<i>c4 :</i>	0	0.05	3.8	-1.4	0.43
<i>c5 :</i>	5	0.09	6	-3	0.42

24 règles d'inférence (parmi 106 règles) ont été activées afin de compléter la description de la galaxie analysée; la description symbolique finale est:

```
GalaxyNGC4473 :
class :          E4(compatibility : total,incompatibility : null)
validity :       good(certain)
arms :          absent(certain)
shape :         elliptical(certain)
bulge :         visible(certain)
flatness :      faint(certain)
profile_concavity : important(certain)
centring :      good(certain)
bar :           absent(certain)
isophotes :     smooth(certain)
```

4.3.6 Résultats

Une base de connaissances de 37 classes et 106 règles a été construite. Le système SYGAL a été testé sur 21 galaxies de l'amas de Virgo numérisées avec la même plaque de Schmidt. Les résultats de la classification automatique de ces galaxies est présenté dans la tableau suivant:

Name	Real class	Result	Comment
NGC4459	LAR+	LA+	correct
NGC4569	SXT2	SX3	correct
NGC4474	L...P*	LA+/LB+	correct
NGC4473	E.5..	E4	correct
NGC4477	LBS.*\$	L	correct
NGC4571	SAR7	SA7	correct
NGC4468	LA...*	bad validity	small area
NGC4531	S..1	LA+	slight difference
NGC4419	SBS0	SB1/SA1	almost correct
IC3392	SA.3	SB1	almost correct (dust lanes)
M91	SBT3	SB1/SB3	correct
NGC4461	LBS+*	LA/LB	almost correct
NGC4438	SAS0	LB/SB1	correct (dust lanes)
NGC4421	SBS0	LB/SB1	correct
IC800	SBT4P\$	SB5	correct
NGC4523	SBS8	SB	irregular
NGC4595	SXT3\$	SA1/SA3	correct
NGC4639	SXT4	SB3	correct
NGC4540	SXT6	SB7	correct
M86	.E.3..	E3	correct
M84	.E.1..	E0	correct

Bien que ces résultats soient très bons, il reste un problème de mauvaise détection de la barre transversale en présence de bande de poussière (dust lane). Les tests ont montrés que la méthode nécessite une taille minimale pour l'image de galaxie (environ 50×50). la règle:

```

if area < 2500
then validity bad

```

est donc activée pour prévenir une mauvaise classification; Les résultats obtenus sur un ensemble de galaxies observées avec un autre télescope (Schmidt telescope of Observatoire de Haute-Provence) et avec différents temps de pose montrent que la base de connaissance doit contenir des informations sur ces paramètres symboliques (télescope et temps de pose).

4.3.7 Limitations

SYGAL est une méthode complètement automatique pour classer des objets complexes (galaxies) avec une approche utilisant des connaissances a priori. Le système contient une base de connaissances constituée de 37 classes implémentées sous formes de frames et de 106 règles d'inférence implémentées sous forme de règles de production. n peut noter deux types de limitations dans SYGAL, d'une part au niveau des algorithmes de vision et d'autre part dans la base de connaissances.

Algorithmes

Il n'y a pas de réel détecteur de bras mais plutôt une analyse de la forme des contours; il n'y a donc pas de mesure bras/interbras. Il n'y a pas de détection et d'extraction d'anneaux (ring pattern). De même il n'a pas de mesure du rapport bulbe/disque mais une mesure du degré de visibilité du bulbe sur le contour projeté et une analyse de la forme du contour.

Base de connaissances

En fait, la base de connaissances ne contient pas encore d'information sur les galaxies irrégulières (irregular) et particulières (peculiar) et sur les anneaux. Cette absence d'information est essentiellement due à l'absence de telles galaxies dans l'ensemble d'images numérisées. Des études ultérieures sont donc nécessaires dans ces domaines. En raison de la résolution des images utilisées pour le développement de SYGAL, il n'y a pas de connaissances sur le rapport bras/interbras et sur la structure des bras.

4.3.8 Conclusion

On a vu à travers la description du système SYGAL que la classification de galaxies est une tâche très complexe, mais qu'il n'est pas impossible de l'automatiser. Cela peut être utile pour traiter objectivement de larges ensembles de données obtenus par des instruments automatisés ou embarqués. De plus l'explicitation de la connaissance sur la classification d'images de galaxies permet de préserver un savoir-faire sur une expertise pointue.

Cette méthode peut être étendue à l'étude d'autres objets que des galaxies. C'est dans le but de montrer la généralité de l'approche que j'ai étudié d'autres problèmes de classification d'objets naturels dont la classification de foraminifères avec Shan Liu-Yu et la classification de zooplanctons marins et de poissons avec Marie-Hélène Gandelin.

4.4 Classification de foraminifères planctoniques

4.4.1 Introduction

Les foraminifères sont des organismes monocellulaires contenus dans une carapace minérale (voir un exemple figure 4.7). Ils vivent au fond de la mer (foraminifères benthiques) ou parmi le zooplancton marin. (foraminifères planctoniques) [11]. Leur taille est le plus souvent inférieure à 1 mm. Ils constituent le groupe le plus important des microfossiles (environ 2000 genres et 30000 espèces). Les foraminifères sont actuellement les mieux connus et les mieux étudiés des microfossiles calcaires. Ils sont des indicateurs précieux de l'âge des couches sédimentaires et sont utilisés pour la prospection pétrolière. Le type de foraminifère peut aider à préciser la probabilité de présence et la qualité du pétrole présent. Cependant il est impossible pour des raisons de coût d'avoir des experts en microfossiles disponibles partout où leur expertise est nécessaire [58].

Il est donc souhaitable que des systèmes "intelligents" puissent accélérer la reconnaissance de microfossiles. Plusieurs essais ont été développés [13, 24, 52, 58]. En particulier le système Vides [58] développé par British Petroleum a pour objectif d'accélérer la classification de microfossiles pour les compagnies pétrolières. Cependant tous ces systèmes sont in-

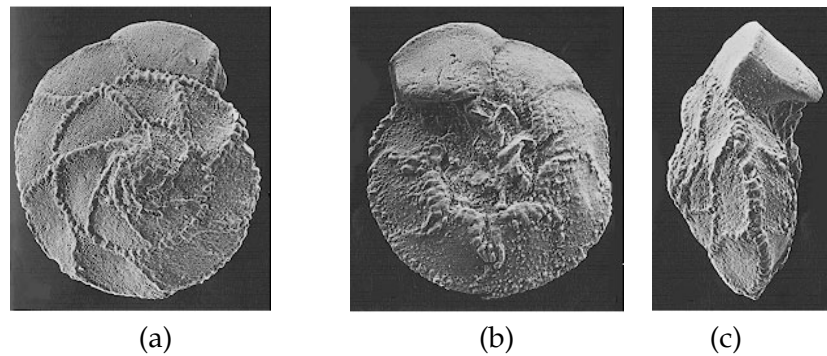


FIG. 4.7 – Un specimen à reconnaître ($\times 100$). (a) vue spirale. (b) vue ombilicale. (c) vue de profil.

teractifs et l'intervention humaine est nécessaire pendant le processus de reconnaissance. C'est spécialement le cas pour la description des spécimens qui doit être effectuée par l'utilisateur.

Dans les travaux présentés on se focalise sur certaines espèces du groupe des Globotruncanides qui sont représentatifs des foraminifères planctoniques du Crétacé (cf. [15, 53]). Ces espèces ont été choisies pour leur forte diversité en terme de variations morphologiques.

La figure 4.8 montre la structure de ce système. Il est composé de deux sous-parties comme dans le cas de la classification de galaxies, la description de la forme par analyse d'images et la classification par un système à base de connaissance construit avec le moteur CLASSIC. La ligne pointillée indique un guidage possible de l'analyse d'image par le système à base de connaissances.

La reconnaissance est basée sur la description de la forme de trois vues caractéristiques: les vues spirale, ombilicale et de profil (cf. Figure 4.7). Les images de ces trois vues sont traitées automatiquement pour mesurer des paramètres numériques décrivant le spécimen. Le système à base de connaissances utilise ensuite ces mesures pour en déduire la classe du spécimen.

L'analyse d'image permet d'obtenir la plupart des paramètres descriptifs qui sont utilisés par les micropaleontologues pour la reconnaissance de foraminifères. Il reste cependant quelques éléments, comme la structure ombilicale, qui sont très difficiles à extraire à partir des trois types d'images, même pour des experts. Le processus de classification permet de raisonner même si certaines valeurs de paramètres sont inconnues, par contre la précision de la classification peut-être réduite. En effet les méthodes traditionnelles de classification utilisent des arbres de décisions fixes, alors que cette approche effectue la mise en correspondance (matching) de la description de l'objet à classer avec l'arbre de description des classes (voir paragraphe suivant). De plus, en fonction des besoins de la classification le système permet aussi à l'utilisateur d'entrer de manière interactive les valeurs inconnues en mode interactif (voir ligne pointillée sur le schéma de la figure 4.8). Pour plus de détails sur le système d'analyse d'image voir [42].

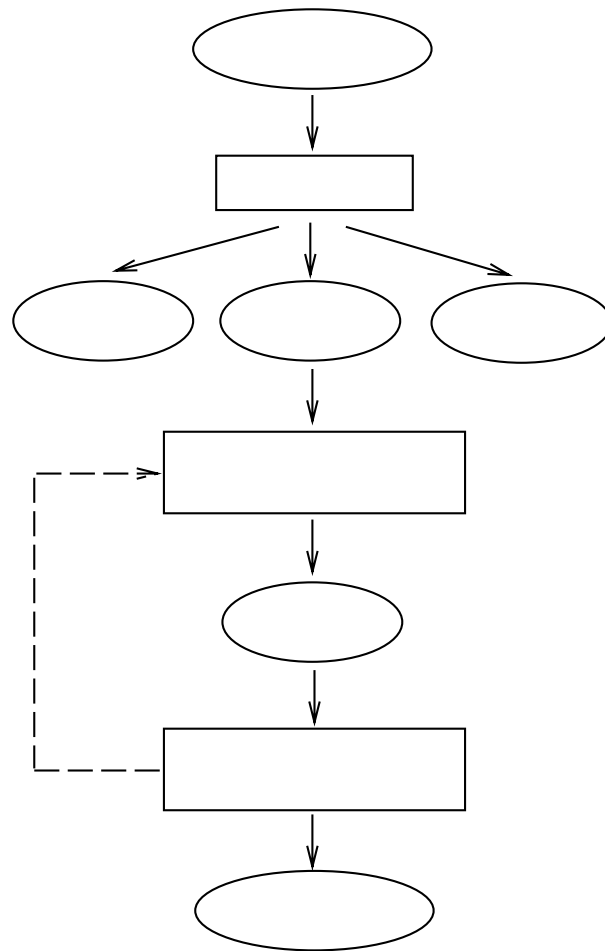


FIG. 4.8 – *Schema de la classification automatique de foraminifères planctoniques*

4.4.2 La base de connaissances

La base de connaissances (arbre des classes, description des classes) est construite en utilisant les termes et les critères de classification des experts du domaine. Comme les paramètres calculés sur les images sont souvent différents des termes utilisés par les experts on a besoin de règles d'inférence pour établir des liens entre ces paramètres et les termes utilisés dans le domaine de la micropaléontologie.

Les classes

La figure 4.9 montre l'arbre des classes du groupe des Globotruncanides contenu dans la base de connaissances. Cet arbre reflète directement la hiérarchie des classes du domaine. Les noeuds de niveau intermédiaire et les feuilles correspondent respectivement au genres et aux espèces du groupe.

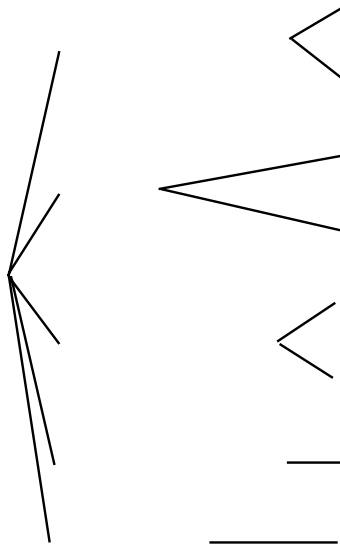


FIG. 4.9 – L'arbre hiérarchique des classes du groupe des Globotruncanites.

L'objet structuré (frame) décrivant la classe Globotruncanita est montré ci-dessous:

Prototype	Globotruncanita:
superclass:	<i>globotruncanites</i>
contour:	{ <i>circular spinal</i> }
chamber-nb:	[5, 9]
spi-chamber-form:	{ <i>trapezoidal</i> }
spi-chamber-ornement:	{ <i>perforated, rugose</i> }
spi-suture-insert:	{ <i>oblique slightly-oblique perpendicular</i> }
umb-chamber-ornement:	{ <i>perforated pustulous</i> }
* primary-aperture:	{ <i>umbilical</i> }
* accessory:	{ <i>portici tegilla</i> }
p-spi-form:	{ <i>convex slightly-convex flat</i> }
p-umb-form:	{ <i>very-convex convex</i> }
* keel-nb:	[1, 1]
* keel-band:	{ <i>absent</i> }

L'objet structuré (frame) décrivant la classe Stuarti est montré ci-dessous:

Prototype	stuarti
superclass:	globotruncanita
contour:	circular
chamber-nb:	[7, 9]
spi-suture-width:	thick
spi-suture-depression:	non-depressed
spi-suture-insert:	slightly-oblique perpendicular
umb-chamber-ornement:	perforated
accessory-aperture:	present
p-spi-form:	convex
p-umb-form:	convex
p-symmetry:	symmetric

La base de règles

La base de connaissances comporte 25 règles d'inférence. Les règles complètent la description de l'objet à classer. Elles sont utilisées de manière dynamique pour transformer les données numériques en valeurs symboliques, comme:

```

rule      "spi-chamber-form-trapezoidal "
if        spi-chamber-compa  $\geq$  thresh.spi-chamber-compa
          spi-chamber-a-c  $\geq$  thresh.spi-chamber-a-c
then      spi-chamber-form = trapezoidal
comment   For a chamber on the spiral side, if both its compactness
          and the ratio of its arc to its chord are small,
          then its form is trapezoidal.
```

ou pour augmenter le niveau d'abstraction des termes symboliques, comme:

```

rule      "p-symmetry"
if        p-spi-form is known
          p-umb-form is known
          p-spi-form = p-v-form
then      p-symmetry = symmetric
comment   Viewed from the profile of a specimen, if the form
          of its spiral side is the same as that of its umbilical side,
          then the object has a symmetric profile.
```

Dans cette base de règles les prédicat flous tels que $\sim=$, $\sim<$, *close to* et *far from* sont utilisés pour manipuler les données imprécises. Les règles sont automatiquement attachées au noeud adéquat de l'arbre des classes.

4.4.3 La base de faits

La base de faits contient l'information descriptive sur l'objet à classer. Au début de la classification la base de faits contient les paramètres mesurés sur les trois images, de nouvelles données déduites sont rajoutées pendant le raisonnement.

La base de faits contient 37 paramètres dont 21 sont extraits par analyse des images les autres étant déduits pendant la classification.

surface:	<i>area of the spiral/umbilical sides</i>
diameter:	<i>diameter of the spiral/umbilical sides</i>
perimeter:	<i>perimeter of the spiral/umbilical sides</i>
peak-nb:	<i>number of peaks on the distance curve</i>
peak-form:	<i>shape of the peaks on the distance curve</i>
ch-vertices:	<i>number of vertices on the convex hull</i>
ch-points:	<i>total number of points on the convex hull</i>
chamber-nb:	<i>number of chambers on the last whorl</i>
spi-chamber-surface:	<i>surface of a chamber (spiral side)</i>
spi-chamber-perimeter:	<i>perimeter of a chamber (spiral side)</i>
spi-chamber-chord:	<i>length of a chord of a chamber (spiral side)</i>
spi-chamber-arc:	<i>length of the external arc of a chamber (spiral side)</i>
spi-chamber-height:	<i>height of a chamber (spiral side)</i>
spi-chamber-ornement:	<i>ornement of a chamber (spiral side)</i>
spi-suture-width:	<i>width of the suture (spiral side)</i>
spi-suture-depression:	<i>depression of the suture (spiral side)</i>
spi-suture-angle:	<i>junction angle of the suture and the external contour</i>
umb-chamber-ornement:	<i>ornement of a chamber (umbilical side)</i>
maj-axis:	<i>length of the major axis of the lateral side</i>
min-axis:	<i>length of the minor axis of the lateral side</i>
trocho-spi:	<i>width of the spiral side</i>

4.4.4 Exemple de session

La classification d'un foraminifère est effectuée comme suit: Etant donné trois vues du specimen (voir Figure 4.7), les algorithmes d'analyse d'image sont appliqués pour obtenir une description initiale de l'objet. Le système de classification recherche dans l'arbre des classes la (les) classe(s) la (les) plus compatible(s). Les règles d'inférence attachées à cette classe sont activées pour compléter la description de l'objet. Si tous les paramètres requis n'ont pas de valeurs le système permet à l'utilisateur de fournir les valeurs manquantes. Si l'utilisateur ne peut pas donner ces valeurs le système les considère comme étant "inconnues".

Pour illustrer le principe une session est montrée ci-dessous:

Initial description of	p31-3-2
surface:	31369
diameter:	216
perimeter:	598
peak-nb:	1
peak-form:	concave
ch-vertices:	60
ch-point:	642
chamber-nb:	7.5
spi-chamber-surface:	2828
spi-chamber-perimeter:	252
spi-chamber-chord:	76
spi-chamber-arc:	80
spi-chamber-height:	5
spi-chamber-ornement:	perforated
spi-sutu-width:	thick
spi-sutu-depression:	non-depressed
spi-sutu-angle:	72
umb-chamber-ornement:	perforated
maj-axis:	208
min-axis:	102
trocho-spi:	64

Le système vérifie que les données d'entrée sont compatibles avec la description de la classe Globotruncanids et valide cette classe.

Le système parcourt la base de règles attachées à la classe Globotruncanids et active celles pour lesquelles les prémisses sont vérifiées:

Rule spi-sutu-r-angle

(determining that sutures on the spiral side join the spiral suture almost at right angle)

Rule p-umb-form-convex

(determining that the profile has a convex umbilical side)

Rule p-spi-form-convex

(determining that the profile has a convex spiral side)

Rule spi-chamber-form-trapezoidal

(determining that chambers on the spiral side are trapezoidal)

Rule cont-circu

(determining that the outline is circular)

Rule p-symmetry

(determining that the profile is symmetrical)

En fonction des nouvelles valeurs rajoutées dans la base de faits par les règles le système choisit parmi les sous-classes de Globotruncanids celles qui sont compatibles avec le specimen p31-3-2. La classe est Globotruncanita donc validée.

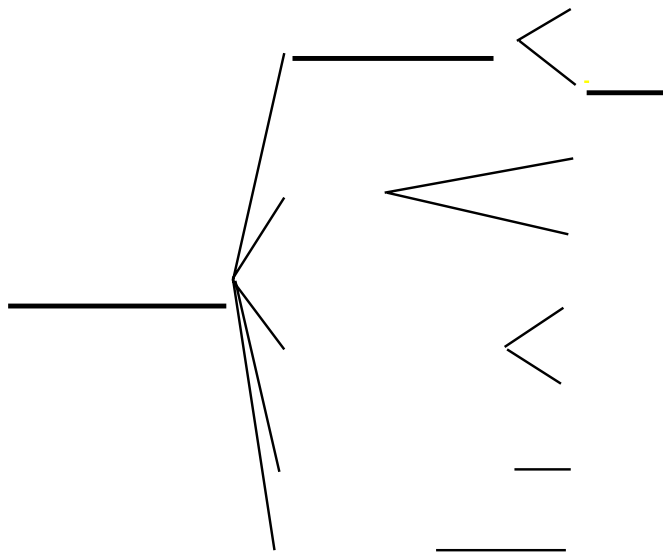


FIG. 4.10 – La résultat de la classification d'un specimen. Les noeuds soulignés correspondent aux noeuds validés. Les noeuds grisés correpsondent aux noeuds incompatibles. Les autres noeuds ne sont pas exploités.

Puis le système valide la classe Stuarti.

Final description of	p31-3-2
class:	stuarti
compactness:	11.39992
ch-v-over-p:	.09345794
spi-chamber-compa:	22.45544
spi-chamber-a-c:	1.052632
spi-chamber-h-over-c:	.06578948
half-maj-axis:	104
trocho-umb:	38
trocho-spi-over-hm:	.6153846
trocho-umb-over-hm:	.3653846
thresholds:	thresholds
contour:	circular
spi-chamber-form:	trapezoidal
spi-suture-insert:	slightly-oblique
p-spi-form:	convex
p-umb-form:	convex
p-symmetry:	symmetric

La figure 4.10 montre le résultat de la classification du specimen montré sur la figure 4.7. Les noeuds soulignés correspondent aux noeuds validés. Les noeuds grisés correspondent aux noeuds incompatibles. Les autres noeuds ne sont pas exploités. Le specimen a été classé comme appartenant aux classes du groupe Globotruncanids du genre Globotruncanita et de l'espèce Stuarti. Pendant le raisonnement 6 règles d'inférence ont été activées.

4.4.5 Conclusion

Pour plus de détails sur ces travaux on peut consulter un article de revue [78] en annexe B de ce manuscrit ainsi que [41] et la thèse de Shan Liu-Yu [42]. L'objectif de cette étude était de montrer que l'approche de classification par un système à base de connaissances construit avec le moteur CLASSIC était indépendante d'une application particulière mais pouvait très facilement être utilisée pour d'autres problèmes de reconnaissance de la classe d'un objet naturel. Le paragraphe suivant a pour objectif de décrire le problème de l'appel de procédure en cours de raisonnement et son intérêt pour la reconnaissance d'organismes biologiques vivants ayant des formes très variées.

4.5 Classification et retour bas-niveaux

L'approche proposée pour la reconnaissance d'objets naturels doit permettre de classer des organismes dont l'apparence est très diverse. Un cas intéressant est, en biologie, le problème de la reconnaissance de zooplanctons marins [63]. Les objets à classer sont des organismes vivants 3D plus ou moins transparents avec des orientations et des positions variées, plusieurs états de maturation et des articulations mobiles. La figure 4.11 montre des images typiques de zooplanctons marins. .

L'entrée est une unique image 2D d'un zooplancton vivant dans son environnement aqueux. Les classes correspondent à la taxonomie biologique. La figure 4.12 montre le synoptique de ce système.

Un ensemble d'algorithmes de vision détecte et décrit les zooplanctons contenus dans les images, puis un système à base de connaissances, construit avec le noyau CLASSIC, les classe en utilisant des modèles symboliques organisés hiérarchiquement. Pendant le classement, le système à base de connaissances lance l'exécution de nouveaux programmes de traitement d'image pour compléter la description des formes de zooplanctons afin de raffiner leur classement.

Le rôle des programmes lancés en cours de raisonnement est d'extraire des informations qui ne sont pas valides pour tout type de zooplancton. En effet un critère de classification fine de copépodes est la taille, la position et la forme de leur antennes. Il n'est pas possible de lancer des programmes d'extraction d'antennes et de description de leurs caractéristiques pour la plupart des zooplanctons car ceux-ci n'ont pas en général d'antennes. Par exemple les méduses sont des organismes gélatineux transparents sans antennes. Par contre pour ces organismes il est important de décrire l'intérieur observé par transparence car la forme et la position des organes internes est un élément important de classification fine. On voit que pour bien classer

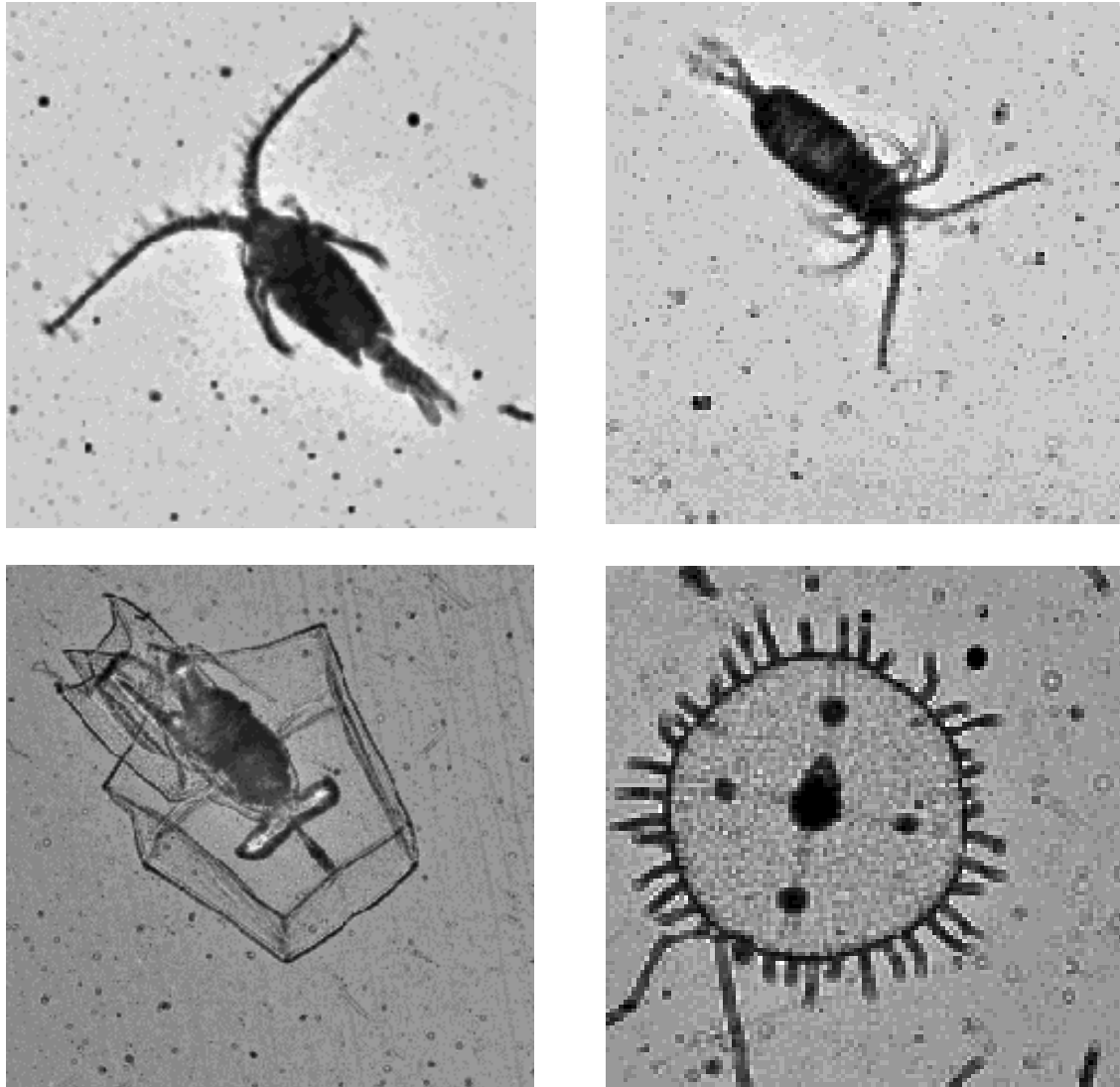


FIG. 4.11 – *Exemples d'images de zooplanctons marins.*

des organismes très variés il faut pouvoir dynamiquement en fonction de la classification courante dans l'arbre des classes pouvoir lancer des programmes pertinents qui fournissent des mesures complémentaires. Dans le cas des travaux sur les zooplanctons on a modifié le fonctionnement du moteur CLASSIC pour qu'il puisse lancer des procédures spécialisées. Pour plus de détails sur ces travaux on peut consulter un article de revue [64] en annexe B de ce manuscrit ainsi que [63] et la thèse de Marie-Hélène Gandelin [32]. Ces travaux présentent tou-

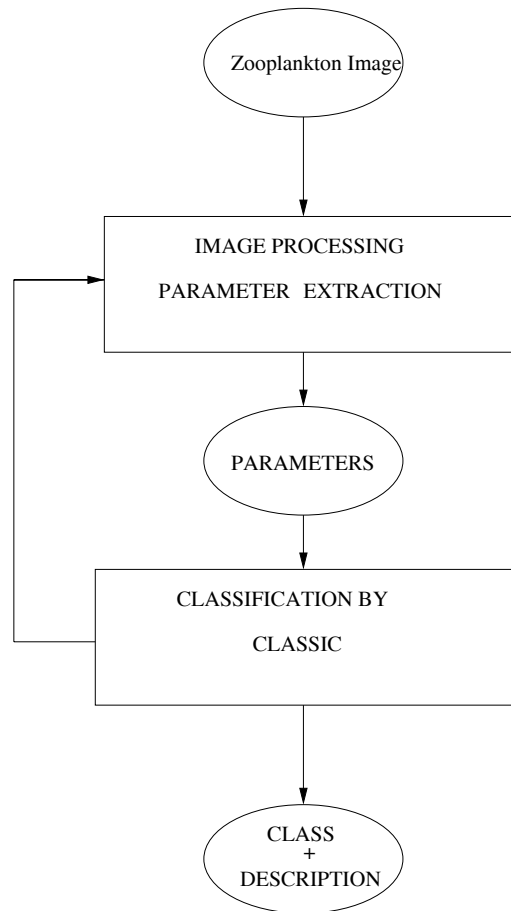


FIG. 4.12 – *Classification avec retour bas-niveau vers l'analyse d'image.*

tefois l'inconvénient de devoir prendre des décisions de bas niveau sur l'appel de procédure en spécifiant le noms des programmes de traitement d'image et leurs paramètres de réglage. J'ai donc cherché à étudier deux autres problèmes d'une part la formalisation de la connaissance en traitement d'image ou pilotage de programmes et d'autre part le couplage entre pilotage de programmes et classification.

4.6 Classification et pilotage de programmes

Dans le cadre de la thèse de Jean-Christophe Ossola [48] j'ai proposé de coupler deux systèmes à base de connaissances pour résoudre la reconnaissance d'objets naturels. L'interprétation haut niveau de l'image est toujours effectuée avec un système gérant la connaissance sur les classes d'objets à reconnaître et est construit avec le moteur CLASSIC. Par contre l'analyse d'image n'est plus un enchaînement systématique de programmes mais un système à base de connaissances gérant la connaissance sur les programmes de traitement d'image (fonction,

alternatives, paramètres de réglage, etc...) construit avec un moteur dédié au pilotage de programmes [49] et [61].

De plus en cours de raisonnement CLASSIC émet des requêtes à des buts abstraits de traitement d'image au lieu des appels de programmes spécialisés.

Un schéma de l'architecture globale est montré figure 4.13 dans le cas de la classification de galaxies.

Cette approche tient à respecter d'une part, la séparation entre les expertises des deux domaines (traitement d'image et interprétation) et d'autre part, à respecter le processus de raisonnement utilisé par un expert du domaine d'application lors de l'interprétation.

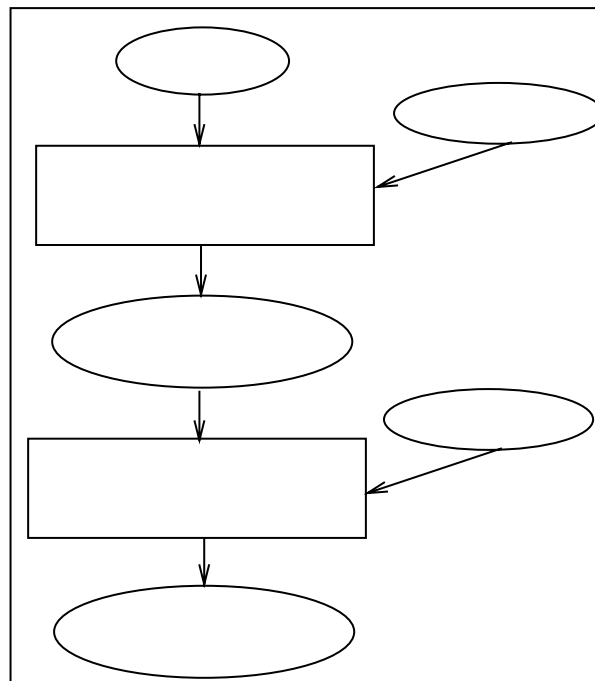


FIG. 4.13 – Schéma du couplage de deux systèmes à base de connaissances construits avec un moteur spécialisé en pilotage de programmes OCAPI et un moteur spécialisé en classification CLASSIC dans le cas de la reconnaissance de galaxies.

Nous avons donc conçu un système distribué entre un système de traitement d'image et un système d'interprétation en nous appuyant sur les générateurs de systèmes à base de connaissances : OCAPI et CLASSIC. Par ailleurs, l'utilisation du gestionnaire d'environnements distribués CHOOE permet de réaliser une communication entre les processus contenant ces deux systèmes bien qu'ils aient été conçus, à l'origine sans cette perspective.

Une première application sur des images de zoo-plancton a nécessité le développement d'une base de connaissances dédiée au traitement d'images de zoo-plancton. Nous avons étendu les possibilités de traitement de notre système en développant un algorithme de traitement d'image destiné à caractériser certaines sous-parties de l'objet d'origine. Ainsi, les classes de

zooplanctons possédant des antennes (sous-partie de zoo-plancton) sont désormais classées avec davantage de précision.

Une deuxième application a été réalisée dans le domaine astrophysique sur la reconnaissance automatique de galaxies. Un travail important de mise à jour et de vérification de cohérence des systèmes existants en traitement d'images (PROGAL) et en interprétation (SYGAL) a du être accompli. De plus, nous avons développé un algorithme de traitement d'images qui permet de mettre en évidence des zones de l'images dont l'intensité lumineuse est discontinue et d'en extraire des paramètres numériques. Les différentes étapes de cet algorithme ont ensuite été structurées dans PROGAL. Les paramètres numériques obtenus sont particulièrement utiles pour caractériser les galaxies possédant des structures appelées "bras spiraux".

4.7 Conclusion

Nous avons présenté une approche à base de connaissances explicites pour la classification d'objets naturels. Après la description du problème et la présentation du moteur CLASSIC spécialisé en classification on a montré à travers trois applications complexes la reconnaissance de galaxies, de foraminifères et de zooplanctons la généralité de l'approche proposée.

En ce qui concerne le problème général de la reconnaissance d'objets à partir d'informations fournies par des capteurs, on constate que cette méthode s'appuie sur la disponibilité d'une expertise consciente sur la description des formes recherchées et sur la notion de reconnaissance progressive de classes organisées hiérarchiquement. Les domaines pour lesquels les objets à reconnaître sont complexes (car possédant de nombreuses caractéristiques et pour lesquels il existe une variabilité importante entre les apparences des individus) sont particulièrement bien adaptés à ce type de méthode; en effet pour de tels domaines (par exemple en agriculture, en biologie, en astronomie, où l'on s'intéresse à des objets naturels ou à des organismes vivants), il n'est pas possible de générer systématiquement toutes les images de référence correspondant aux classes recherchées. Par contre, on dispose d'une connaissance stable sur la description symbolique (abstraite) des objets. Le problème de la description numérique (concrète) des objets est un problème de traitement du signal ou de traitement d'images. Dans la méthode proposée la connaissance sur la manière de traiter ce problème est implicite (enchaînement prédéterminé de programmes précis avant la classification puis en cours de classement appel éventuel à certains programmes spécialisés); les travaux décrits dans le chapitre 5 suivant intitulé pilotage de programmes ont pour but de gérer de manière explicite la connaissance sur les traitements d'image à l'aide d'un moteur spécialisé en planification de programmes de traitement d'images [21].

Chapitre 5

Pilotage de programmes

Ce chapitre décrit les travaux que j'ai effectués sur l'utilisation et la réutilisation de bibliothèques de programmes ou pilotage de programmes. L'approche suivie consiste dans la conception de techniques d'intelligence artificielle pour la gestion de connaissances sur des programmes d'analyse d'images.

Ce chapitre est structuré de la manière suivante, après une introduction présentant le problème de la réutilisation de bibliothèques de programmes, je décris les techniques de pilotage de programmes proposées pour résoudre ce problème, puis je présente rapidement plusieurs applications en analyse d'images avant de détailler le pilotage de programmes pour l'analyse d'images de galaxies.

5.1 Introduction

Le domaine du traitement d'images a produit un grand nombre de programmes efficaces et de nombreuses bibliothèques ont été développées. Dans ces bibliothèques les programmes ont été intégrés d'un point de vue concret bas-niveau. Par contre aucune aide n'est fournie aux utilisateurs qui cherchent à résoudre des problèmes particuliers d'analyse d'image. Chaque utilisateur final ne peut pas avoir une compréhension profonde de la sémantique et de la syntaxe des programmes. Des utilisateurs inexpérimentés peuvent avoir seulement une compréhension de base du domaine du traitement d'images et de sa terminologie. D'autre part les programmes implémentent des fonctionnalités de plus en plus complexes et leur utilisation est délicate. Si l'effort demandé à un utilisateur est trop important pour maîtriser la complexité de nouveaux programmes ceux-ci ne seront jamais largement utilisés concrètement.

Dans ce chapitre je présente de nouvelles techniques que j'ai appelées techniques de pilotage de programmes (ou Program Supervision en anglais) pour aider des utilisateurs à gérer des techniques de traitement d'images nécessaires pour leurs applications.

A partir de l'analyse de la tâche d'un expert en traitement d'images qui traite des données en utilisant un ensemble de programmes on a dérivé des modèles de connaissances. En s'appuyant sur ces modèles des systèmes à base de connaissances pour le pilotage de programmes prennent en charge la gestion de la bibliothèque de programmes et libèrent l'utilisateur de la tâche de l'effectuer manuellement. Cette aide va d'un guide intelligent jusqu'au contrôle automatique des programmes. L'idée de base est d'automatiser le choix et l'exécution de programmes contenus dans une bibliothèque pour atteindre un objectif de traitement de l'utilisateur. Ceci est réalisé en encapsulant la connaissance sur l'utilisation des programmes dans une base de connaissances et en émulant la stratégie d'un expert sur l'utilisation de ces programmes.

Le pilotage de programmes (Program Supervision) est un domaine avec un nombre croissant de travaux provenant de domaines applicatifs et techniques très différents [67]. Ces recherches sont souvent motivées par un domaine d'application particulier (comme le traitement d'images, le traitement du signal ou le calcul scientifique). Contrairement aux systèmes à base de connaissances pour l'interprétation d'images qui ont été étudiés depuis plus de vingt ans, les systèmes de pilotage de programmes d'analyse d'images sont plus récents. Au Japon de nombreuses équipes appartenant aussi bien aux secteurs scientifiques qu'industriels ont

fourni un effort important sur ce problème (voir [70], [56] et [43]). En Europe les travaux ont été développés soit pour l'étude d'outils généraux (comme OCAPI [21]) ou pour le développement d'outils industriels pour une application particulière (comme le projet Esprit VIDIMUS [12]). Aux Etats-Unis des travaux préliminaires ont été effectués par Johnston [40] et par Bailey [3]. Plus récemment [33] et [18] ont utilisés des techniques de planification pour la composition de processus d'analyse d'images.

L'approche que je présente par la suite a été conçue pour gérer l'automatisation de l'analyse d'images contenant une galaxie à partir d'une base de connaissances par SYGAL (voir le chapitre 4 précédent). Un premier moteur OCAPI a été développé dans le cadre de la thèse de Véronique Clément [Cle90]. Ces travaux ont ensuite été approfondis notamment dans le cadre des thèses de John van den Elst [72] pour la modélisation abstraite du processus de pilotage, de Régis Vincent [74] pour l'amélioration de la réparation et de Monica Crubézy [25] pour l'amélioration de la planification pour l'imagerie médicale et en coopération avec Sabine Moisan [75] pour la conception d'une plate-forme de développement de moteurs de pilotage de programmes. Dans la suite de ce chapitre je présente le principe du pilotage de programmes avec les moteurs OCAPI/PEGASE, puis successivement un résumé de trois applications en vision robotique et en imagerie médicale, je termine en décrivant la base de connaissances PROGAL pour le pilotage de programmes d'analyse d'images de galaxies. Trois articles de revues [21], [57] et [68] joints en annexe B décrivent plus en détail ces travaux.

5.2 Pilotage de programmes

L'objectif du pilotage de programmes est de faciliter la (re)configuration de programmes de traitement numérique de données. Cette mise au point de traitement peut impliquer l'enchaînement de plusieurs programmes. Il est à noter que les programmes existent déjà (par exemple dans une bibliothèque) et que le but n'est pas d'optimiser les programmes eux-même mais leur utilisation. Un système à base de connaissances de pilotage de programmes aide un utilisateur non spécialiste à appliquer ces programmes dans différentes situations comme montré sur la figure 5.1.

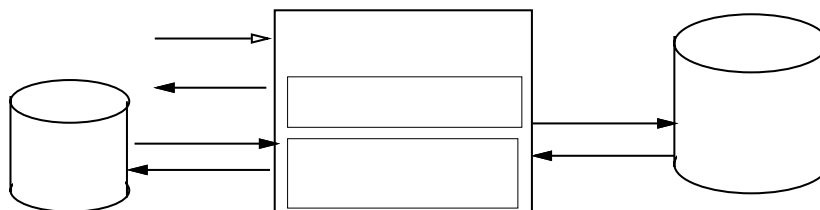


FIG. 5.1 – Un système à base de connaissances de pilotage de programmes aide un utilisateur à (ré)utiliser un ensemble de programmes pour résoudre une requête sur des données d'entrée $\mathcal{I}$ pour obtenir des données de sortie $\mathcal{O}$ résultant de l'exécution d'un plan de programmes. Il est composé d'un moteur de pilotage de programmes, d'une base de connaissances et d'un accès à un ensemble de programmes.

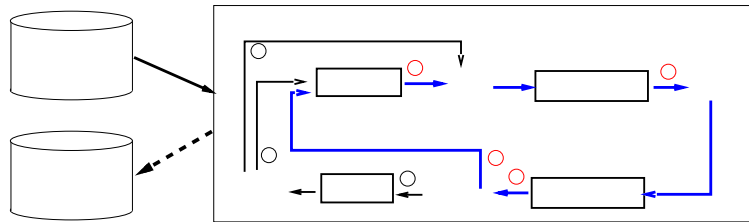


FIG. 5.2 – Les différentes phases d'un système de pilotage

Un système à base de connaissances de pilotage de programmes est composé de:

1. un ensemble de programmes pré-existants,
2. une base de connaissances décrivant comment utiliser ces programmes,
3. un moteur de pilotage de programmes.

Le rôle du *moteur* de pilotage de programmes est d'exploiter la connaissance sur les programmes afin de construire un plan de programmes qui atteigne le but de l'utilisateur. Il émule la stratégie d'un expert sur l'utilisation des programmes. Le plan final qui produit des sorties satisfaisantes n'est habituellement pas immédiat, il résulte en effet souvent de plusieurs essais-erreurs. Le raisonnement effectué par le moteur explore les différentes possibilités et calcule la meilleure, en fonction de critères experts disponibles dans la base de connaissances. Le raisonnement d'un système de pilotage de programmes consiste en différentes phases qui peuvent être complètement ou partiellement automatisées. Dans le modèle que je propose il est décomposé en quatre phases: *planification*, *exécution* des programmes, *évaluation* des résultats et *réparation*. La figure 5.2 montre les rôles de ces phases et leurs interreactions. La figure 5.3 montre l'algorithme simplifié du moteur OCAPI [21]).

- 1- mise en correspondance globale
tant que toutes les requêtes n'ont pas été traitées
- 2- sélection de la requête à traiter
- 3- classification des opérateurs (en utilisant les **règles de choix**)
tant que la requête n'est pas satisfaite
- 4- sélection du meilleur opérateur
- 5- exécution de l'opérateur (en utilisant les **règles d'initialisation ou d'ajustement**)
- 6- jugement des résultats (en utilisant les **règles d'évaluation**)

FIG. 5.3 – L'algorithme du moteur de pilotage de programmes (OCAPI)

La *base de connaissances* contient des *opérateurs* qui sont des représentations des programmes (incluant la description de leurs données et paramètres), des combinaisons typiques de programmes ainsi que des critères pour guider le raisonnement. Le contenu de la base de connaissances doit être suffisant pour que le moteur sélectionne les programmes, initialise leurs pa-

ramètres, gère des flots de données sophistiqués et enchaîne les programmes pour produire un plan de programmes satisfaisant en fonction de la requête de l'utilisateur et de ses données d'entrée. Un exemple montrant comment un moteur de pilotage utilise ce type de connaissances est montré sur la figure 5.3.

Dans la suite sont décrits les principaux concepts d'une base de connaissances de pilotage de programmes: les **Données**, les **Buts**, les **Requêtes**, les **Opérateurs** et les **Critères**.

- Les **Données** contiennent toute l'information nécessaire sur le problème de l'utilisateur;
- Les **Buts** expriment les contraintes sur l'état final souhaité;
- Les **Requêtes** sont des instances de buts sur des données particulières et sous des contraintes particulières.
- Les **Opérateurs** sont de deux types: les opérateurs élémentaires et les opérateurs complexes. Un opérateur *élémentaire* représente un programme particulier et un opérateur *complexe* représente une combinaison de programmes. Les combinaisons de programmes correspondent à des décompositions en programmes plus concrets à différents niveaux d'abstraction, soit par spécialisation (alternatives), soit par composition (séquences, parallélisme, boucles, etc...).

1. Les **opérateurs élémentaires** sont des composants de base qui manipulent des données. Leur représentation est composée de:

- Une fonctionnalité (information sur "que fait l'opérateur?");
- Une méthode ou une action (i.e. syntaxe d'appel, méthode de simulation);
- Une information sur les arguments (i.e. nom et types des arguments d'entrée et de sortie, signature) "sur quoi l'opérateur agit-il?";
- Des informations sémantiques: caractéristiques connues par l'expert, contraintes, pré et post conditions (sur les données);
- Des critères d'évaluation, d'initialisation et d'ajustement des arguments.

Un exemple d'opérateur élémentaire *thres-hyst* est montré sur la figure 5.4. La fonctionnalité est un seuillage, le programme correspondant *hyster* travaille avec une méthode par hystérésis et deux seuils; il a comme arguments une image d'entrée, deux paramètres ajustables (les deux seuils) et une image de sortie; plusieurs pré-conditions sur le format et sur le bruit de l'image d'entrée sont spécifiées ainsi que l'effet sur la sortie.

2. Les **Opérateurs complexes** sont des squelettes de plans fournis par l'expert qui construit la base de connaissances. Ils décrivent les réseaux connus de connexions possibles entre les opérateurs (choix, séquences, répétitions, etc...) afin d'atteindre un but donné. Chaque opérateur complexe peut être décomposé en sous-composants. Un opérateur complexe peut être représenté par un graphe dont les feuilles sont reliées à des opérateurs élémentaires. En plus de l'information propre à l'opérateur les opérateurs complexes contiennent des informations sur le flot de données entre leurs sous-composants ainsi qu'une connaissance tactique (i.e. des critères pour décider comment parcourir le réseau).

Name	<i>thres-hyst</i>
Functionality	<i>thresholding</i>
Characteristics	<i>>-1-threshold, hysteresis-method</i>
Input arguments	<i>in : image</i>
Output arguments	<i>out : image</i>
Parameters	<i>upper-threshold : thres-hyst-threshold</i> <i>lower-threshold : thres-hyst-threshold</i>
Preconditions	<i>in.coding.format == inimage</i> <i>in.presentation.noise.kind == gaussian</i>
Effects	<i>out.contents.segmented = value-based</i>
Calling syntax	<i>hyster in out -sh upper-threshold</i> <i>-sb lower-threshold</i>

FIG. 5.4 – Un exemple d'opérateur élémentaire

Name	<i>pyr-level-match</i>
Functionality	<i>pyramidal-stereo-matching</i>
Input arguments	<i>right-image: promethee-image</i> <i>left-image: promethee-image</i>
Output arguments	<i>3D-image: promethee-image</i>
Subcomponents	<i>images-reduction, primitives-extraction,</i> <i>stereo-matching</i>
Control	<i>DO images-reduction</i> <i>THEN primitives-extraction</i> <i>THEN stereo-matching</i>
Data flow	<i>stereo-matching.out-image → 3D-image</i> <i>...</i>

FIG. 5.5 – Un exemple d'opérateur complexe

Un exemple d'opérateur complexe *pyr-level-match* est montré sur la figure 5.5. Sa fonctionnalité est une mise en correspondance pyramidale; il a une décomposition séquentielle en trois sous-étapes, en premier une réduction d'images, puis l'extraction de primitives et enfin une mise en correspondance des primitives; les arguments d'entrée et de sortie sont spécifiés ainsi que le flot de données.

- **Les Critères** Différents critères implémentés par des règles de production jouent un rôle important au cours du raisonnement, i.e. pour choisir entre différentes alternatives (*critères de choix*), pour adapter l'exécution des programmes (*critères d'initialisation*), pour diagnostiquer la qualité des résultats (*critères d'évaluation*) et pour réparer une mauvaise exécution (*critères de réparation* et *critères d'ajustement*).

1. **Les Règles de choix** Le rôle des règles de choix est de choisir entre différents opérateurs alternatifs ayant la même fonctionnalité. Un exemple de deux opérateurs alternatifs

Functionality :	stereo-matching
Operator-1:	o-ma-stereo-match
Characteristics:	high-quality low-adaptability
Operator-2:	o-meygret-stereo-match
Characteristics:	average-quality high-adaptability

FIG. 5.6 – Un exemple d'opérateurs alternatifs

qui ont la même fonctionnalité de mise en correspondance stéréo (stereo-matching) mais qui ont des caractéristiques différentes est montré sur la figure 5.6. Un exemple de règle de choix décrivant comment choisir entre ces deux opérateurs de mise en correspondance stéréo est montré sur la figure 5.7.

If	<i>context.user-constraints.quality-matching == high</i>
Then	<i>use-operator-with-characteristic high-quality</i>

FIG. 5.7 – Un exemple de règle de choix

2. **Les Règles d'évaluation** Le rôle d'une règle d'évaluation est de diagnostiquer la qualité des résultats. Deux exemples de telles règles sont montrés sur la figure 5.8. La première règle calcule une évaluation dans le cas d'une taille de filtre trop grosse et décide qu'il y a échec. La seconde règle visualise les résultats pour un utilisateur dans le cas d'un mode interactif et propose trois qualificatifs possibles (correct, too-low, too-high).
3. **Les Règles d'initialisation** Le rôle d'une règle d'initialisation est de calculer la valeur des paramètres ajustables d'un opérateur avant son exécution. Deux exemples de telles règles sont montrés sur la figure 5.9.
4. **Les Règles de réparation**

[h]	If	<i>xmax <= 255</i>
	and	<i>rayleighmax > 50</i>
	Then	<i>global-assess filter-size = too-big</i>
	and	<i>failure</i>
	If	<i>context.user-constraints.mode == interactive</i>
	Then	<i>assess-by-user thresholding (correct, too-low, too-high)</i>

FIG. 5.8 – Un exemple de deux règles d'évaluation

If	<i>context.details == too-few</i>
Then	<i>initialise thresh to 90</i>
If	<i>context.noise == high</i>
Then	<i>initialise window-size to 7</i>

FIG. 5.9 – Une exemple de deux règles d’initialisation *thresh* et *window-size* sont des paramètres ajustables)

If	<i>assessed size == too-low</i>
Then	<i>re-execute</i>

FIG. 5.10 – Un exemple de règle de réparation

Le rôle des règles de réparation est de définir une stratégie de réparation après la détection d’un échec par les règles d’évaluation. Par exemple la règle de réparation montrée sur la figure 5.10 implémente la stratégie suivante: si sur la taille a été jugée comme trop faible il faut réexécuter le même opérateur.

5. **Les Règles d’ajustement** Le rôle d’une règle d’ajustement est de proposer une méthode pour régler la valeur d’un paramètre pour améliorer la qualité des résultats d’un opérateur. Deux règles d’ajustement sont montrées sur la figure 5.11. La première règle exprime que le paramètre (*window-size*) doit être réglé après un certain type d’évaluation (*noise == too-high*) et comment ce paramètre doit être modifié (*increase*), la seconde règle exprime qu’après une détection ambiguë le paramètre *thresh* doit être diminué et qu’une option doit être changée.

If	<i>assessed? noise == too-high</i>
Then	<i>increase window-size</i>
If	<i>global-assess? detection == ambiguous</i>
Then	<i>decrease thresh</i>
and	<i>option := with</i>

FIG. 5.11 – Un exemple de deux règles d’ajustement (*window-size* et *thresh* sont des paramètres ajustables)

5.3 Application des techniques de pilotage de programmes

Cette partie montre brièvement des exemples de systèmes de pilotage de programmes dans des domaines d’application très différents. Deux exemples concernent la détection d’obstacles

routiers et un exemple traite d'imagerie médicale. De plus un exemple en astronomie pour la description morphologique de galaxies est détaillé dans la partie suivante.

5.3.1 Détection d'obstacles routiers par stéréovision

L'objectif dans cet exemple est d'utiliser des techniques de pilotage de programmes pour la détection d'objets tels que des voitures dans des scènes urbaines en s'appuyant sur des données stéréo (voir figures 5.12 et 5.13). La bibliothèque de programmes est composée de 24 modules (principalement un algorithme de stéréovision pyramidal basé sur des chaînes de points de contours (algorithmes développés durant la thèse d'Aimé Meygret et publiés dans [45])). Le rôle du pilotage de programmes est de permettre la réutilisation de ces programmes pour des images très différentes. Pour cela 54 paramètres numériques doivent être initialisés. Dans cet exemple deux sortes de problèmes sont détectés et réparés automatiquement: une mauvaise extraction de primitives et une mauvaise mise en correspondance. Deux autres types de problèmes de seuillage sont détectés interactivement et réparés automatiquement.



FIG. 5.12 – Une paire d'images stéréo dans une scène urbaine.

5.3.2 Détection temps-réel d'obstacle routiers

Le deuxième exemple a été développé dans le cadre du projet européen Eureka Prometheus pour l'assistance à la conduite. L'objectif était d'utiliser des techniques de pilotage de programmes pour reconfigurer en temps-réel les programmes de vision [46]. Les programmes sont organisés en trois modules de perception:

- un **détecteur d'obstacle 3D par télémétrie** (3 modes: *full-scanning*, *unique-measure*, *tracking*, et des contraintes hardware)
- un **détecteur d'obstacles par morphologie mathématique** (2 modes: *detection* ou *focusing*),
- une **segmentation de mouvement** (input data: switching entre 2 caméras, et contraintes hardware).

Chaque module a été développé par un laboratoire de recherches en vision français différent (LASMEA à Clermont, IRISA à Rennes et Mines Paris). Le rôle du pilotage de programmes est



FIG. 5.13 – Voitures détectées dans une scène urbaine.

de décider quelles données doivent être traitées et comment elles doivent être traitées (lancement ou arrêt d'un module, réglage de différents modes, prise en compte de contraintes matérielles, etc...). Une démonstration en octobre 1994 a permis de démontrer le fonctionnement temps-réel avec un temps de cycle de 200ms de ce concept sur une voiture dans des conditions de circulation variées.

5.3.3 Imagerie médicale

Dans cet exemple l'objectif est de fournir à un clinicien un meilleur accès à de nouvelles techniques de traitement d'images. Cet exemple a été conçu et développé dans le cadre de la thèse de Monica Crubézy [26]. Les images sont des images IRM 3D du cerveau (voir figure 5.14). Les programmes comprennent 21 modules (développés par le projet Epidaure de l'INRIA Sophia). Le rôle du pilotage de programmes est de travailler en interaction avec l'utilisateur pour lui laisser la tâche d'interprétation des résultats. Les décisions de traitement d'images qui ont été automatisées sont: *séquencement* de 10 à 15 programmes en fonction des caractéristiques des images, *initialisation* de 27 paramètres numériques et 7 *choix* entre des opérateurs alternatifs.

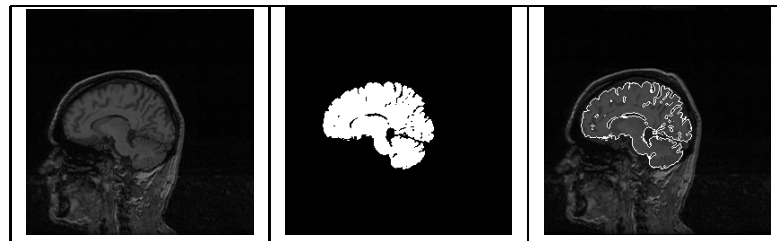


FIG. 5.14 – Segmentation du cerveau avec des images 3D IRM

5.4 Pilotage de programmes de traitement d'image de galaxies

On s'intéresse ici à l'automatisation du traitement d'images contenant une galaxie. Un des objectifs est de capitaliser la connaissance sur le traitement d'images de galaxies. Une autre motivation importante de ce travail est de fournir un outil entièrement automatique de classification de galaxies. Il est bien connu que l'analyse d'image est une tâche difficile, c'est particulièrement vrai lorsque les images à traiter contiennent des objets naturels. Ce type d'images possède plusieurs caractéristiques: les objets naturels ont des formes très complexes et il n'existe pas de modèles géométriques simples pour les décrire. L'automatisation de l'analyse de telles images requiert donc une grande quantité de connaissances en traitement d'image. Les galaxies sont des objets naturels typiques. Elles présentent une grande variabilité aussi bien dans leurs formes que dans leurs luminosités en fonction de plusieurs facteurs (temps d'exposition, luminosité intrinsèque de la galaxie, distance entre la galaxie et le télescope,...). Comme mon objectif est de classer les galaxies de la même manière que les astronomes experts sur ce sujet ([73]), on doit extraire plusieurs données numériques décrivant aussi précisément que possible les galaxies. On a donc besoin, mais ce n'est pas suffisant, d'un système capable de traiter ces images. Ce système doit être plus intelligent que cela; il doit être capable de s'adapter aux différentes situations de traitement. Une solution est d'utiliser un programme spécialisé pour traiter les images. Mais un tel programme est généralement trop rigide et n'est pas capable de s'adapter à des situations aussi différentes que des qualités variables des images d'entrées (i.e. présence de bruit), des natures différentes d'images (en intensité ou en densité), différentes tailles, différents systèmes d'acquisition (caméra CCD, photographies, etc....). De plus un programme spécialisé est difficilement lisible par quelqu'un d'autre que son auteur. En conséquence un tel programme devient de plus en plus difficile à étendre et mettre à jour.

On propose donc d'utiliser des techniques de pilotage de programmes dédiées au traitement d'image de galaxies ([61, Tho89]) pour adapter automatiquement le traitement aux variations dans les images (voir figure 5.15) et pour fournir une entrée optimale à un système de classification automatique de galaxies (voir SYGAL présenté au chapitre 4 précédant). On a développé PROGAL un système à base de connaissances dédié au pilotage d'une bibliothèque de programmes en astronomie. La bibliothèque est un ensemble de 37 programmes modulaires de traitement d'images. L'objectif est d'automatiser la détection d'une galaxie dans une image astronomique (voir figure refprogalphases). Le résultat final fourni par ces programmes de traitement d'images est une description morphologique de galaxies en terme d'un ensemble de mesures numériques (voir figure 5.17).

Une première version de ce tel système [49] a été développée avec le moteur de pilotage de programmes OCAP [21]. Ici, je présente la base de connaissance développée avec le moteur de pilotage de programmes PEGASE, successeur d'OCAP. C'est un planificateur hiérarchique à base de squelettes ayant un mécanisme sophistiqué de réparation. PEGASE est implémenté avec la plate-forme LAMA [75]. Pour plus de détails sur le moteur PEGASE voir [76].

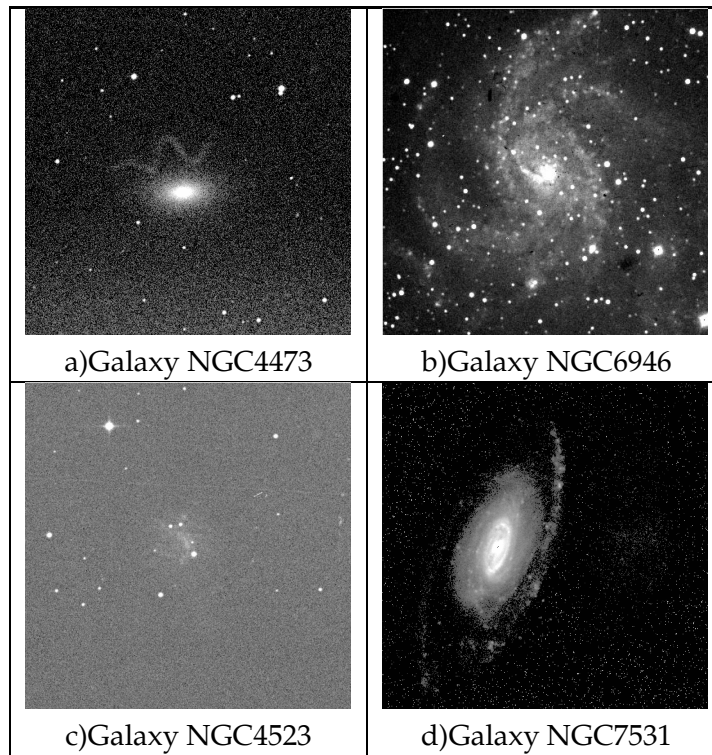


FIG. 5.15 – Exemples de variations possibles dans des images de galaxies

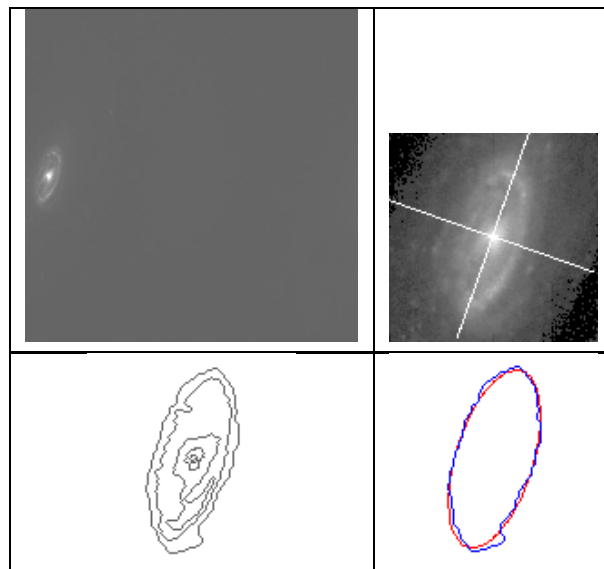


FIG. 5.16 – Différentes étapes du traitement d'images de galaxies

galaxy: <i>ngc6946</i>			
area	: 67886.7;	contour c3: {	centre_err : 3;
ellipticity	: 0.35;		ellipse_err : 0.30;
linear_err	: 0.034;		compactness : 14.1;
profile	: 1.96;		angle : 50.4;
orientation	: 86.46;		excentricity : 0.40 };
contour c1: {	centre_err : 0;	contour c4: {	centre_err : 12;
	ellipse_err : 0.19;		ellipse_err : 0.25;
	compactness : 1.8;		compactness : 25.7;
	angle : -9.5;		angle : 55.4;
	excentricity : 0.22 };		excentricity : 0.31 };
contour c2: {	centre_err : 7;	contour c5: {	centre_err : 20;
	ellipse_err : 0.37;		ellipse_err : 0.48;
	compactness : 8.1;		compactness : 38.6;
	angle : -4.4;		angle : 84.2;
	excentricity : 0.34 };		excentricity : 0.54 }.

FIG. 5.17 – La sortie du traitement

5.4.1 Contenu de la base de connaissances

La dernière version de PROGAL est une grosse base de connaissances. PROGAL contient 88 opérateurs: 49 opérateurs élémentaires correspondant à des programmes et 39 opérateurs complexes représentant de nombreux niveaux d'abstraction. Il y a 30 opérateurs complexes avec des décompositions séquentielles et 9 opérateurs complexes avec des décompositions alternatives (ou choix). PROGAL contient aussi 114 règles: 36 règles de choix, 39 règles d'initialisation, 11 règles d'évaluation, 21 règles de réparation et 7 règles d'ajustement.

Tous ces éléments de connaissances partagent une entité commune: la Contexte. Cet objet structuré implémenté sous forme de frame (voir figure 5.18) est utilisé pour exprimer les variations possibles de l'information contextuelle de différentes images de galaxies. Ce Contexte décrit si la galaxies est a priori centrée ou non, si l'image est particulièrement bruitée, si l'image a été calibrée ou pas (densité ou intensité), quelle est la taille minimale des objets dans l'image et enfin si d'autres objets comme des étoiles sont a priori connus dans cette région du ciel.

Dans la suite je décris la base de connaissances sous deux aspects principaux: la connaissance de planification et la connaissance de réparation.

5.4.2 Connaissance de planification

La connaissance de planification es principalement décrite avec les opérateurs (élémentaires et complexes) ainsi que les règles de choix et d'initialisation. Le traitement global pour la description de galaxies comporte huit sous-étapes:

1. Création et initialisation du fichier contenant les paramètres numériques.

Context		
Attributes	Values	Comments
galaxy-position	<i>centered</i> <i>uncentered</i>	position of the object in the image
min-size	[1,1000000]	minimal size of the object
noise	<i>with</i> <i>without</i>	noisy image or not
image-type	<i>intensity</i> <i>density</i>	image type
stars	<i>in-front-of-the-galaxy</i> <i>absent everywhere</i> <i>outside-the-galaxy</i>	eventual presence of other objects

FIG. 5.18 – Description de l'information contextuelle

2. Isolation de l'objet d'intérêt, c'est à dire la galaxie. Cette étape est la plus difficile.
3. Détermination d'un espace de référence centré-objet.
4. Calcul de paramètres globaux. Une fois la galaxie correctement isolée, extraction de paramètres concernant des propriétés globales de la galaxie tels que *orientation*, *barycentre*, *surface*, *longueur le long de la direction principale* ... (voir la figure 5.17).
5. Construction de contours. Afin de décrire la galaxie de façon multi-échelles, le traitement construit cinq contours d'iso-intensité. La construction elle-même est une tâche complexe qui est décomposée en sous-tâches:
 - Extraction d'un *profil moyen* de la galaxie le long de son axe principal
 - Sur ce *profil moyen*, calcul de cinq seuils permettant de construire cinq contours différents, Cette étape dépend du contexte" si l'image d'entrée est en **intensité** les seuils sont corrigés.
 - seuillage de l'image d'entrée avec les cinq seuils calculés précédemment.
 - Construction de cinq chaînes de contour.
 - une **étape optionnelle** permet à l'utilisateur de construire des fichiers visualisables des contours. Cette étape peut être importante pour vérifier que le traitement a été correctement effectué ou pas.
6. Calcul de paramètres à partir des contours. Pour chacun des cinq contours calculés, plusieurs paramètres sont extraits: orientation, surface, périmètre, longueur le long des axes principaux et secondaire, barycentre, distance entre le contour et son approximation par une ellipse.
7. Visualisation. Puis, de façon optionnelle la visualisation des contours superposés avec les meilleurs ellipses est possible grâce à la construction préalable des fichiers visualisables (voir figure 5.16).
8. Stockage des paramètres numériques dans un fichier.

Cette décomposition est exprimée dans la base de connaissances par un opérateur complexe **c-galaxy-description** ayant une décomposition séquentielle en huit sous-étapes comme montré

Complex	Operator
	Name: c-galaxy-description
	Comment: "image processing before classification"
	Functionality: galaxy-description
	Input Data: ie
	Output Data: gparam
	Body:
	p-initialisation -
	c-object-isolation -
	p-determination-of-axes -
	c-global-parameters -
	c-contours-construction -
	c-5contours-parameters -
	c-5ellipses-visualization -
	p-parameter-storing;

FIG. 5.19 – L'opérateur complexe *c-galaxy-description* ayant une décomposition en 8 operators exprimée dans l'attribut *Body*. Le signe "-" entre les opérateurs signifie qu'il s'agit du séparateur séquentiel "then".

sur la figure 5.19. Parmi des sous-étapes, trois sous-étapes (la première, la troisième et la dernière) sont simples et sont réalisées par un programme unique; ces trois étapes sont donc représentées respectivement dans la base de connaissances par des opérateurs élémentaires **p-initialisation**, **p-determination-of-axes** et **p-parameter-storing**. Les cinq autres sous-étapes ne sont pas directement réalisées par un programme, elles sont donc représentées respectivement par cinq opérateurs complexes avec des décompositions séquentielles:

- **c-object-isolation**,
- **c-global-parameters**,
- **c-contours-construction**,
- **c-5contours-parameters**
- **c-5ellipses-visualization**

On détaille maintenant la connaissance de planification pour la partie la plus difficile: *l'isolation de l'objet d'intérêt*. C'est une étape très importante de laquelle dépend la qualité du résultat final. Le système doit être capable de prendre en compte plusieurs conditions et doit être capable d'évaluer les résultats intermédiaires du traitement et si nécessaire d'ajuster le traitement. Cette étape est représentée par un opérateur complexe *c-object-isolation* comme montré sur la figure 5.20. Elle consiste en deux étapes principales qui sont la localisation du centre de la galaxie et l'isolation effective de la galaxie. La seconde étape d'isolation effective de la galaxie est réalisée en calculant les limites de la galaxie puis en soustrayant le fond de l'image. La première étape (localisation du centre de la galaxie) est la partie la plus difficile. Cette étape est décrite par un opérateur complexe **c-object-location** ayant 5 sous-étapes (voir figure 5.21). Premièrement afin de réduire le temps de calcul un opérateur **p-preprocessing** est appliqué en

Complex	Operator
	Name: c-object-isolation
	Comment: "detection and extraction of the object"
	Functionality: object-isolation
	Input Data: ie
	Output Data: iisolee
	Body:
	c-object-location - c-effective-isol ;

FIG. 5.20 – L'opérateur complexe *c-object-isolation*

Complex	Operator
	Name: c-object-location
	Comment: " research of the object in an image "
	Functionality: object-location
	Input Data: ie
	Output Data: is
	Body
	p-preprocessing -
	p-thresh-his-
	c-galaxy-area-location -
	c-max-conv -
	p-center ;

FIG. 5.21 – l'opérateur complexe *c-object-location* avec une décomposition séquentielle en 5 opérateurs

fonction de la taille de l'image d'entrée. Un *critère optionnel* est activé pour décider si l'image doit être échantillonnée ou pas. Si la taille de l'image est supérieures à un certain seuil contenu dans l'(information contextuelle l'image est échantillonnée.

La troisième sous-étape est **c-galaxy-area-location** comme montré sur la figure 5.22.

En fonction du contexte d'utilisation, des *critères de choix* (voir règles r-gc et r-guc sur la figure 5.23) permettent au système de traiter l'image de deux façons possibles.

- Si la **galaxie est centrée** dans l'image, une sous-image est extraite autour du centre de l'image; un opérateur élémentaire **p-centered-image** représente cette étape dans la base de connaissances.
- Si **on n'a pas d'information a priori information sur la position** de la galaxie, le système recherche l'objet le plus important présent dans l'image (la plus grosse source étendue). C'est la **partie la plus difficile**, un opérateur complexe **c-uncentered-image** représente cette étape dans la base de connaissances (voir figure 5.24). Cet opérateur complexe a 3 sous-étapes dont **c-detection** (voir figure 5.25).

Complex	Operator
	Name: c-galaxy-area-location
	Comment: " galaxy location with two alternatives "
	Functionality: galaxy-area-location
	Input Data: ie
	Output Data: is
	Body
	c-uncentered-image — p-centered-image ;

FIG. 5.22 – L'opérateur complexe *c-galaxy-area-location* avec une décomposition en deux opérateurs alternatifs. Le signe "—" signifie qu'il s'agit du séparateur alternatif "or".

Choice	Rule
	Name : r-gc
	Let?c a Context
	If ?c.galaxy-position == 'centered
	Then use-operator p-centered-image
Choice	Rule
	Name : r-guc
	Let?c a Context
	If ?c.galaxy-position == 'uncentered
	Then use-operator c-uncentered-image

FIG. 5.23 – Les règles de cohix pour sélectionner entre les opérateurs de localisation de la zone de la galaxie

On va voir dans la partie suivante sur la connaissance de réparation 5.4.3 comment on peut vérifier la qualité de la détection de la galaxie.

Complex	Operator
	Name: c-uncentered-image
	Comment: " research of the biggest object in an image "
	Functionality: galaxy-area-location
	Input Data: ie
	Output Data: is
	Body
	c-detection - p-extraction - p-mu ;

FIG. 5.24 – L'opérateur Complexe *c-uncentered-image* avec une décomposition séquentielle en 3 opérateurs

Complex	Operator
	Name: c-detection
	Comment: " detection of the brightest region"
	Functionality: galaxy-detection
	Input Data: ie
	sm
	Output Data: is
	taille
	...
	Body:
	p-thresh-muls - p-muls - c-morphlis -
	c-contour-chain - p-bbox;
	Repair rules: rule-5

FIG. 5.25 – L'opérateur complexe *c-detection*

5.4.3 Connaissance de réparation

La connaissance de réparation est essentiellement décrite par les règles d'évaluation, de réparation et d'ajustement. Dans cette partie on présente comment la connaissance pour détecter des erreurs et les réparer est exprimée.

Il y a principalement trois types d'erreur qui sont réparés dans le traitement complet. Comme c'est souvent le cas en traitement d'image, la détection d'une erreur n'est possible immédiatement après l'exécution de l'opérateur "fautif". Sur la figure 5.26 on montre une vue synthétique du traitement complet. Le premier cas de réparation d'erreur a lieu lorsque la taille de l'objet détecté est similaire à la taille d'une étoile, Le deuxième cas a lieu lorsque la galaxie n'a pas été correctement isolée. Ce cas est facile à réparer (agrandissement de la taille estimée de la galaxie) mais l'erreur peut seulement être détectée à la fin du traitement. Enfin le dernier cas d'erreur a lieu lorsque les cinq iso-contours se superposent. On va détailler comment la connaissance est exprimée pour le premier cas (mauvaise détection de la galaxie).

L'opérateur p-bbox L'opérateur **p-bbox** (voir figure 5.27) calcule la boîte englobante de l'objet détecté. Plusieurs critères d'évaluation: les règles rule-1 (voir figure 5.28), rule-2 (voir figure 5.29) et rule-3 (voir figure 5.30) attachées à cet opérateur comparent la taille de cette boîte englobante (une sortie de cet opérateur) avec la taille normale des étoiles du champ (une valeur fournie dans le contexte). Si l'évaluation est ambiguë, la connaissance de réparation envoie cette information à l'opérateur complexe **c-detection** en utilisant la fonction de transmission **send-up** (voir la règle rule-4 sur la figure 5.31).

L'opérateur c-detection Si une erreur est envoyée à l'opérateur **c-detection** (déjà montré figure 5.25 dans la partie Connaissances de Planification, PEGASE utilise la base de règles de réparation attachée à l'opérateur **c-detection** pour le réparer. Dans ce cas, si l'erreur est *size-*

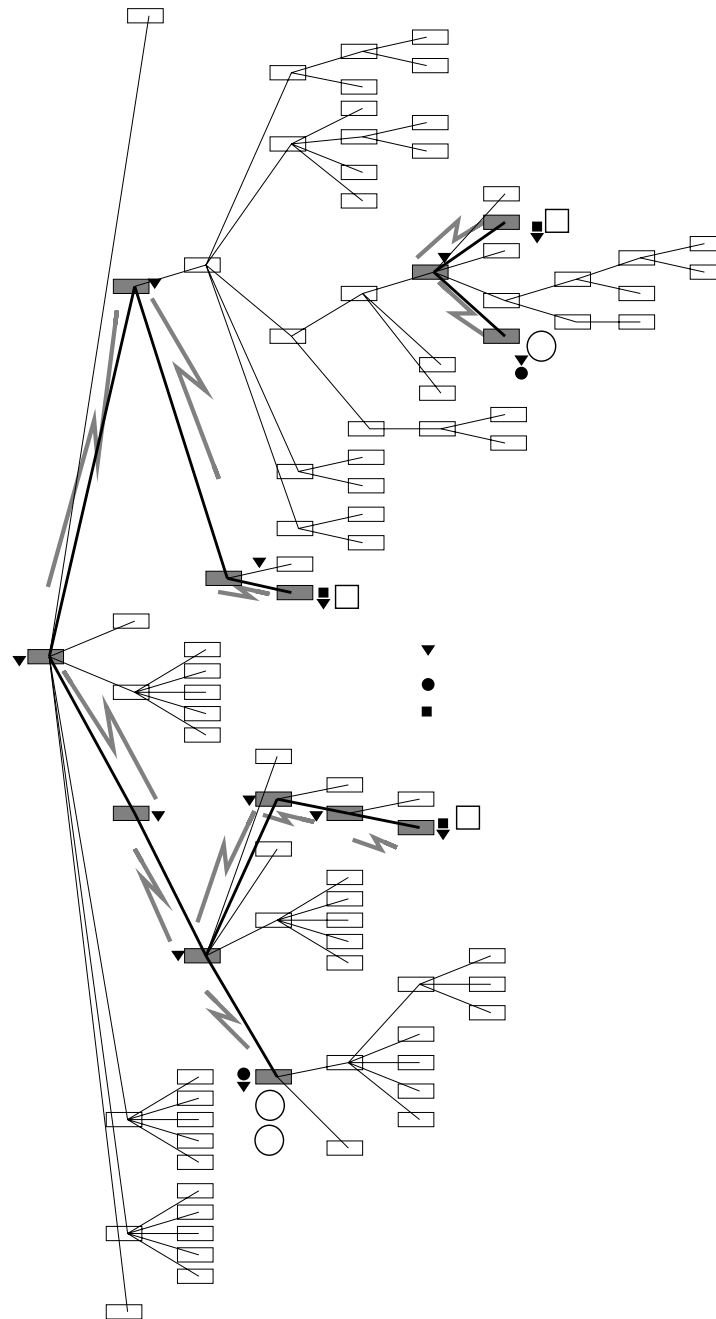


FIG. 5.26 – Un arbre montrant la décomposition de l'opérateur complexe chaîne-astro en 7 sous-étapes, Ces étapes étant décomposées en sous-étapes. Les feuilles de l'arbre sont des programmes concrets (ou opérateurs élémentaires). Les règles d'évaluation, de réparation et d'ajustement sont localisées dans l'arbre.

Primitive	Operator
	Name: p-bbox
	Comment: "computing the bounding box of a region"
	Functionality: bounding-box
	Input Data: ie
	sm
	Output Data: ix
	iy
	x
	y
	size
	...
	Evaluation rules: rule-1, rule-2, rule-3
	Repair rules: rule-4

FIG. 5.27 – L'opérateur élémentaire *p-bbox*

Evaluation	Rule
	Name : rule-1
	Comment : "If the object size is really larger than the size of a star, the detection is good"
	Let ?context a Context
	If size $\geq$?context.star-size + (?context.star-size / 2)
	Then assess-operator good continue

FIG. 5.28 – La règle d'évaluation attachée à l'opérateur *p-bbox* décidant d'un jugement positif

ambiguous (envoyée par **p-bbox**), la réparation consiste simplement à l'envoyer à un de ses sous-opérateurs **p-muls** en utilisant la fonction **send-down** (voir règle rule-5 sur la figure 5.32).

Evaluation	Rule
	Name : rule-2
	Comment : "If the object size is slightly larger than the size of a star, the detection is limit"
	Let ?context a Context
	If size $\geq$?context.star-size + (?context.star-size / 2), size $\geq$?context.star-size
	Then assess-operator limit continue

FIG. 5.29 – La règle d'évaluation attachée à l'opérateur *p-bbox* décidant que la détection est limite

Evaluation	Rule
	Name : rule-3
	Comment : "If the object size is smaller or equal to the size of a star, the detection is ambiguous "
	Let ?context a Context
	If size ; ?context.star.size
	Then assess-operator ambiguous repair

FIG. 5.30 – La règle d'évaluation attachée à l'opérateur *p-bbox* décidant que la détection est ambiguë.

Repair	Rule
	Name : rule-4
	Comment : "If the detection is ambiguous, send-up"
	If assess-operator? p-bbox ambiguous
	Then send-up size-ambiguous

FIG. 5.31 – La règle de réparation attachée à l'opérateur *p-bbox* décidant d'envoyer le jugement négatif au dessus.

L'opérateur p-muls L'opérateur **p-muls** (voir figure 5.33) contient une règle de réparation et une règle d'ajustement, Quand l'erreur *size-ambiguous*, résultat du jugement effectué après évaluation des résultats de **p-bbox**, est reçue par **p-muls**, la stratégie exprimée dans la règle de réparation rule-6 (voir figure 5.34) est de ré-exécuter cet opérateur. La fonction **re-execute** active la base de règle d'ajustement attachée à cet opérateur pour calculer une nouvelle valeur pour le seuil *smuls*. La méthode d'ajustement (voir règle rule-7 sur la figure 5.35) est un ajustement par pourcentage avec un pas de 0.05 %. Comme le but est d'augmenter la taille de l'objet détecté dans l'image binaire, le seuil a été diminué. Le traitement complet peut être vu comme un arbre de traitement hiérarchique (voir figure 5.26).

Repair	Rule
	Name : rule-5
	Comment : "If the detection is ambiguous, send-down p-muls"
	If assess-operator? c-detection size-ambiguous
	Then send-down p-muls size-ambiguous

FIG. 5.32 – La règle de réparation attachée à l'opérateur *c-detection* décidant d'envoyer le jugement négatif en dessous à l'opérateur *p-muls*

Primitive	Operator
Name:	p-muls
Comment:	"binarization of an image"
Functionality:	thresholding
Input Data:	ie
Parameters:	smuls
Output Data:	is
	...
Repair rules:	rule-6
Adjustment rules:	rule-7

FIG. 5.33 – L'opérateur élémentaire p-muls

Repair	Rule
Name :	rule-6
Comment :	"If the error is size-ambiguous, re-execute"
If	assess-operator? p-muls size-ambiguous
Then	re-execute

FIG. 5.34 – La règle de réparation attachée à l'opérateur p-muls décidant de ré-exécuter l'opérateur p-muls

5.4.4 Conclusion

Le système PROGAL qui vient d'être présenté est un système de pilotage de programmes qui permet à un expert d'exprimer sa connaissance de manière naturelle. Un avantage important d'un système à base de connaissances est qu'il permet de capitaliser la connaissance: de nouvelles méthodes de traitement peuvent être ajoutées ou la connaissance peut être facilement mise à jour. De plus, cette technique rend le traitement plus dynamique dans le sens où il s'adapte de lui-même à différentes situations qui peuvent arriver pendant le traitement d'images de galaxies. PROGAL est donc un outil flexible capable de traiter différents types

Adjustment	Rule
Name :	rule-7
Comment :	"Decrease threshold smuls of five per cent"
If	assess-operator? p-muls size-ambiguous
Then	adjustment-method smuls percent-float , adjustment-step smuls .05, decrease smuls

FIG. 5.35 – La règle d'ajustement attachée à l'opérateur p-muls décidant de modifier la valeur du paramètre smuls

d'images prises dans différentes conditions.

5.5 Conclusion

J'ai présenté dans ce chapitre les travaux que j'ai effectués sur les techniques à base de connaissances pour construire des systèmes intelligents d'analyse d'images, c'est à dire les techniques de pilotage de programmes. Ces techniques rendent possible la capitalisation de la connaissance sur l'utilisation d'un ensemble de programmes grâce à une base de connaissances formalisée et rendent également possible la construction de systèmes flexibles et automatiques grâce à des mécanismes de raisonnement dédié. Des systèmes de pilotage de programmes peuvent être développés si plusieurs conditions sont vérifiées: premièrement l'existence d'un ensemble modulaire de programmes; deuxièmement, le besoin d'une utilisation complexe et flexible de ces programmes (différentes valeurs de paramètres d'entrée, différentes sélections de programmes, ...); troisièmement, l'existence d'un ensemble explicite de critères pour décider quoi faire (connaissance sur des décompositions typiques, critères d'évaluation, stratégies de réparation,...).

Chapitre 6

Interprétation vidéo

Ce chapitre décrit les travaux que j’ai effectués sur l’analyse et l’interprétation de séquences temporelles d’images ou interprétation vidéo. L’approche suivie consiste dans le couplage de techniques de vision par ordinateur pour la détection d’objets mobiles et des techniques d’intelligence artificielle pour la gestion de connaissances a priori sur le lieu observé et sur les comportements à reconnaître.

Ce chapitre est structuré de la manière suivante, après une introduction présentant le problème de l’interprétation vidéo, puis une présentation de travaux proches, je décris l’approche proposée pour la modélisation 3D du décor puis je détaille l’interprétation de séquences vidéo et enfin je montre des résultats de cette technique pour des scènes de métro en vidéosurveillance.

6.1 Introduction

L’objectif de ces travaux est de détecter des objets mobiles (plus particulièrement des personnes) et de d’analyser leurs comportements. Ce travail est basé sur trois hypothèses: premièrement on considère la caméra comme étant statique, deuxièmement on utilise une unique caméra monoculaire et troisièmement on tient compte de contraintes temps-réel. La première hypothèse (caméra statique) est souvent vérifiée dans les réseau actuels de vidéosurveillance et nous permet de simplifier la détection bas-niveau des objets mobiles par rapport à un environnement fixe. La deuxième hypothèse (caméra monoculaire unique) est vérifiée dans la plupart des réseau actuels de vidéosurveillance. La troisième hypothèse (contraintes temps-réel) est très intéressante car elle implique que les solutions doivent être obtenus avec un temps de calcul très court, c’est à dire plus court que la fréquence entre deux images consécutives de la séquence. Mais elle implique aussi l’existence d’un système entièrement automatique.

Dans la suite de ce chapitre après une présentation des travaux proches, je présente brièvement les techniques bas-niveau utilisées pour la détection et le suivi d’objets mobiles. Puis deux sortes de connaissances a priori que l’on utilise sont décrites. Premièrement je décris le rôle du modèle 3D de la scène vide et différentes manières de représenter cette information. Deuxièmement je traite le problème de la description haut-niveau du comportement d’objets mobiles en utilisant des événements observables génériques et des modèles de scénarios dépendant de l’application. Enfin des résultats obtenus sur différentes applications de vidéosurveillance dans le cadre du projet européen AVS-PV sont montrés et discutés. Ces travaux ont été conduits dans le cadre des thèses de François Brémond et de Nathanael Rota et avec la collaboration de Nicolas Chleq. Ils continuent dans le cadre de plusieurs projets en vidéosurveillance (dont le projet européen ADVISOR). Un article de revue [4] et un chapitre de livre [60] joints en annexe B décrivent plus en détail ces travaux.

6.2 Etat de l’art

Cohen, Bremond Medioni and Nevatia (University of South California), dans le projet VSAM de la DARPA se focalisent sur la reconnaissance d’événements comportant des véhicules et des personnes, ([44] et [23]). La particularité de leurs travaux est que les vidéos sont filmées par des caméras mobiles. Ils utilisent des modèles de cartes de l’environnement pour recalibrer des images

aériennes dans une carte connue *a priori* et un réseau de propriétés pour calculer les événements et les états contenus dans un automate prédéfini décrivant les situations recherchées. Herzog (VITRA) propose un système capable de décrire dynamiquement des scènes comportant des personnes. L'originalité de leur travail est le domaine d'application: un stade de football ([1] et [35]) ainsi que la méthode d'inférence basée sur une logique d'intervalle de temps pour décrire une séquence temporelle d'événements qui sont calculés et typés séparément. Intille and Bobick (MIT Media lab) dans une application similaire se focalisent sur l'analyse de scènes de football américain. Leur but est de reconnaître des stratégies particulières dans des interactions complexes de joueurs ([37] et [38]). Le point principal réside dans le fait que ces activités ne sont seulement des comportements de personnes mais des comportement de groupes de personnes. Shah (University of Central Florida) est intéressé par la description dynamique de comportements de personnes dans des environnements de bureau ([2] et [27]). Même si le problème est la reconnaissance d'activités de longue durée les auteurs insistent sur l'importance de la reconnaissance d'"instants clefs" qui sont les conditions de changement d'état dans un automate représentant le comportement global. Les "instants clefs" sont générés si certaines conditions sont réalisées. Tessier (ONERA), dans le projet PERCEPTION propose une méthode originale pour décrire des comportement, en effet des réseaux de Petri sont utilisés pour représenter les évolutions dynamiques d'une scène de parking comportant des voitures et des personnes ([59] et [16]). Buxton and Gong (University of Sussex) ont fait une contribution importante au domaine avec le projet européen VIEWS([14]). Le système était capable de gérer des personnes et des véhicules dans des rues et des parkings. Une représentation haut niveau basée sur des réseaux bayésiens était calculé. Ce travail souligne la nécessité de gérer l'incertitude et d'utiliser une information contextuelle pour améliorer les résultats de détection et de suivi. Dans le même esprit, Ivanov et de Grimson (MIT) étudient la détection de comportement de personnes et de véhicules dans un parking. L'intérêt de cette recherche réside dans la méthode de combinaison d'événements ([39]). Un comportement est représenté par un ensemble de règles basé sur une grammaire context-free qui permet certaines combinaisons de prédicats constants simples.

Le schéma général de l'approche que je présente dans ce chapitre est basé sur l'utilisation de scénarios prédéfinis [20] et de modèle de la scène vide (ou décor) [19]. Je décris dans la partie interprétation une méthode basée sur des arbres pour déclarer les événements [69] et sur une logique temporelle pour déclarer les scénarios dépendant de l'application.

6.3 Perception

Dans cette partie je décris très rapidement le module de perception que l'on utilise en entrée de l'interprétation de vidéos. Une description plus détaillée peut-être trouvée dans [54]. Les méthodes de perception sont des méthodes standards qui vérifient des contraintes de temps-réel. Leur rôle est de fournir incrémentalement un historique de personnes qui ont été détectées dans la scène. Ce module est composé de quatre sous-parties: une détection de mouvement, une détection de personnes, un suivi de personnes et un lissage. Chaque sous-partie contient des méthodes alternatives qui sont sélectionnées et paramétrées manuellement dans une phase de configuration. On notera que pour l'instant il n'y a pas de pilotage de programmes de ces

méthodes de bas niveau.

Détection de mouvement

la détection de mouvement est essentiellement un seuillage de la différence entre l'image courante et une image de référence. Avant de calculer cette différence on filtre l'image courante avec un filtre gaussien 3×3 pour réduire le bruit. Puis pour chaque pixel on calcule la différence absolue entre son intensité (gris ou couleur) et l'intensité du pixel correspondant de l'image de référence. Si cette différence est supérieure à un certain seuil α , le pixel est étiqueté comme mobile sinon il est étiqueté comme stationnaire. Puis on met à jour l'image de référence I_r avec l'information provenant de l'image courante I_c selon l'équation suivante:

$$I_r = (1 - \beta) I_r + \beta I_c$$

On voit que pour $\beta = 1$ on détecte le mouvement comme la différence entre des images consécutives de la séquence et que pour $\beta = 0$ on détecte le mouvement par rapport à une image de fond fixe. α et β sont des paramètres du détecteur de mouvement.

Détection de personnes

Le but est de détecter des régions mobiles (blobs) correspondant à des personnes. On utilise un modèle de personne ayant 8 paramètres: la position du centre de gravité (px_{img}, py_{img}) , la hauteur h_{img} et la largeur l_{img} dans l'image 2D, la position 3D (px_{3D}, py_{3D}) sur le plan du sol de la scène, la largeur 3D l_{3D} et la taille 3D h_{3D} . La boîte englobante d'une personne est définie par les mesures sur l'image (px_{img}, py_{img}) , h_{img} et l_{img} . L'algorithme de détection analyse l'ensemble de blobs détectés sur l'image par la détection de mouvement. Des critères sur l'image 2D ainsi que des critères sur la scène 3D sont utilisés. Les premiers sont basés sur la distance 2D entre les blobs dans l'image. Les seconds sont basés sur des contraintes de hauteur et de largeur 3D. Les mesures 3D sont obtenues par projection linéaire du plan image et grâce à une calibration de la caméra.

Suivi de personnes

Le but de cette étape est de mettre à jour l'ensemble des trajectoires. Pour cela les personnes qui ont été détectées dans l'image courante doivent être mise en correspondance avec celles détectées précédemment. cette mise en correspondance peut être définie comme une fonction d'un ensemble P_{t-1} de personnes détectées au temps $t - 1$ vers un ensemble P_t de personnes détectées au temps t . On utilise 3 méthodes alternatives: une méthode basée sur une quantité de superposition dans l'image 2D, une méthode basée sur la proximité des personnes dans la scène 3D et une méthode plus contraignante basée sur la proximité des personnes dans la scène 3D. La première méthode (basée sur une quantité de superposition dans l'image 2D) dit que deux personnes détectées à deux instants consécutifs sont la même personne réelle si le pourcentage de superposition de leur boîte englobante est supérieur à un seuil. La deuxième méthode mis en correspondance une personne au temps t avec une personne au temps $t - 1$ si leur distance 3D est inférieure à un seuil. La troisième méthode est similaire à la deuxième mais la fonction doit être soit une injection soit une surjection.

Lissage

Le but de cette étape est corriger les erreurs faites dans les étapes précédentes de perception sur les différentes mesures 3D d'une personne: (px_{3D}, py_{3D}) la position sur le plan du sol, h_{img}

la hauteur et l_{img} la largeur. Le deuxième but est d'estimer la vitesse instantanée des personnes. Trois méthodes de lissage sont utilisées, La première méthode est un filtre de Kalman standard. Le vecteur d'état est défini par $(px_{3D}, py_{3D}, vx_{3D}, vy_{3D})$. Le modèle dynamique est basé sur l'hypothèse d'une vitesse constante. Les deuxième et troisième méthodes sont respectivement un filtrage médian et moyenne avec une fenêtre temporelle de taille 3, 5 ou 7. (vx_{3D}, vy_{3D}) est initialisée en calculant $v(t) = \frac{p(t) - p(t-1)}{\delta t}$ puis chacune des valeurs $px_{3D}, py_{3D}, vx_{3D}$ et vy_{3D} sont filtrées.

6.4 Modèle 3D du décor

Comme le but est de fournir un cadre qui peut être adapté à des conditions spécifiques on propose de définir deux sortes d'informations a priori: un modèle 3D (décrit dans cette partie) et des scénarios prédéfinis (voir partie 6.5 suivante). Le modèle 3D est une connaissance a priori qui contient une description de l'environnement statique observé par une caméra. Pour chaque caméra regardant un environnement particulier un opérateur de sécurité doit fournir, dans une phase de configuration, des informations pertinentes selon le formalisme que l'on propose. Pour plus de détails sur le rôle du modèle du décor en interprétation d'image voir [19].

Le modèle 3D du décor contient en plus d'une information géométrique une information sémantique, Sa structure est constituée d'un ensemble d'objets physiques, d'un ensemble de zones d'intérêt et d'une matrice de calibration pour la transformation de coordonnées sur le plan image 2D en coordonnées 3D. La géométrie des zones d'intérêt est décrite par une liste de polygones définis sur des plans qui peuvent n'importe quelle orientation. La géométrie de chaque objet physique ou élément de l'équipement est un cylindre généralisé défini par sa hauteur et sa base polygonale.

L'information sémantique sur chaque objet et chaque zone d'intérêt est constituée de six attributs: quatre à valeur symbolique et deux à valeur numérique. Les quatre attributs à valeur symbolique sont: le type (objet ou zone), la fonction (par exemple table, siège, couloir, etc...) le nom (par exemple siège3, couloir2, etc...) et des caractéristiques (par exemple jaune, fragile, etc...). Les deux attributs à valeur numérique qui sont utilisés pour la reconnaissance de scénario sont: la distance normale et le temps normal d'usage d'un objet.

Les figures 6.1 et 6.2 montrent deux exemples d'environnements que l'on a modélisés; pour chaque exemple l'image de gauche montre la vue observée par la caméra et l'image de droite montre une vue 3D du même environnement basé sur l'information géométrique contenue dans le modèle.

Dans le premier exemple, un coin café, le modèle 3D contient: une matrice de calibration, la description de neuf objets (trois sièges, une table, une machine à café, un ascenseur, une poubelle, une porte et un radiateur) et la description de trois zones (une zone café, une entrée et un couloir). Le deuxième exemple est une scène réelle du métro de Nuremberg qui a été sélectionnée au cours du projet européen AVS-PV. C'est une entrée d'une station de métro avec huit objets et deux zones. Les objets sont six composteurs et deux machines de distribution de tickets. Les zones sont une entrée et un couloir.

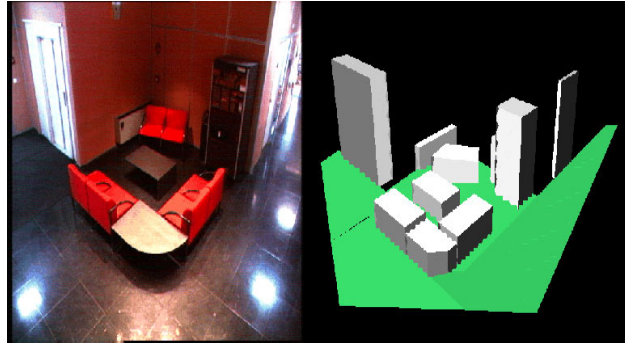


FIG. 6.1 – A gauche: image d'un coin café. A droite: modèle 3D du décor.

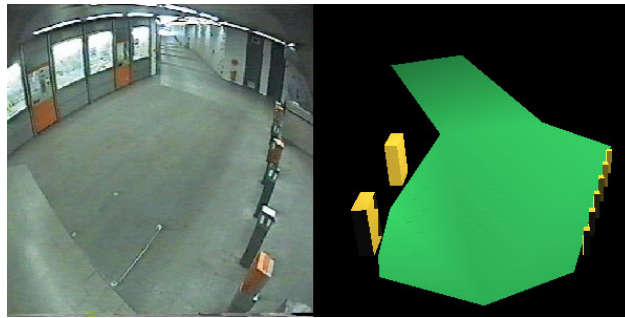


FIG. 6.2 – A gauche: image d'une station de métro. A droite: décor.

La figure 6.3 montre un exemple de description complète d'une machine de distribution de tickets dans une station de métro à Nuremberg en Allemagne.

Le contenu du modèle 3D du décor n'a pas pour objectif d'être exhaustif et très détaillé. Il doit contenir toute l'information nécessaire pour aider l'interprétation de séquences vidéo. Comme cette information est remplie par un opérateur humain pour chaque caméra son contenu

name =	ticket vending machine 1
type =	equipment
function =	ticket vending machine
characteritics =	fragile
proximity =	100 cm
normal time =	30 s
polygon =	([0, 415, 0],[0, 520, 0], [-50, 520, 0], [-50, 415, 0])
height =	180 cm

FIG. 6.3 – Un exemple de description d'une machine de distribution de tickets.

doit être limité à l'information utile. Cependant le modèle géométrique et le modèle sémantique peuvent être enrichis ultérieurement.

6.5 Interprétation

Dans cette partie je m'intéresse au problème de la description haut-niveau du comportement d'objets mobiles en utilisant des événements génériques observables et des scénarios dépendant du domaine d'application. Le processus de reconnaissance de concepts temporels peut être réduit à la reconnaissance de concepts atemporels: les états des objets. Un événement est une propriété spatio-temporelle qui représente un changement significatif dans l'état des objets de la scène. Des événements typiques sont habituellement "to enter", "to start running", "to stand up" ou "to leave". L'algorithme pour la reconnaissance d'événements est le suivant: un événement est reconnu si pour un état donné d'un objet, la valeur de cet état est significativement différente entre l'image (correspondant au temps t_0) et une autre image (correspondant au temps t_n , avec $t_n = t_0 + d_{rec}$). L'intervalle de temps entre t_0 et t_n , d_{rec} est appelé délai de reconnaissance.

Par exemple, si à l'instant t_0 , une personne est loin de la machine à café et en est proche à l'instant t_n , alors l'événement "la personne s'approche de la machine à café" est reconnu.

Le problème de la reconnaissance d'événement peut être réduit au problème de rechercher un ensemble d'états décrivant la scène avec suffisamment de précision. En d'autres termes résoudre le problème de la reconnaissance d'événement est ramené à celui de la description symbolique de la scène. Donc, si pour chaque image on a une description symbolique de la scène, il est suffisant de comparer ces descriptions pour connaître ce qui s'est passé, c'est à dire les événements qui se sont déroulés, Cette description de la scène doit inclure une traduction de valeurs numériques en valeurs symboliques et soit être suffisamment générique pour être appliquée à des environnements et des domaines d'application très différents.

Modèle d'état

L'objectif est de définir des modèles d'état qui puissent être étendus et paramétrisés. Un état d'objets de la scène est défini par un arbre n-aire qui représente la manière de calculer cet état (voir figure 6.4 pour un exemple abstrait d'un tel arbre). Quatre types de noeuds sont distingués: les noeuds objets, les noeuds descripteurs, les noeuds opérateurs et les noeuds classificateurs (voir ci-dessous pour leur définition). Les feuilles de cet arbre sont les objets concernés par cet état. Les noeuds pères des feuilles sont des descripteurs numériques de ces objets. Tous les noeuds intermédiaires sont des noeuds opérateurs. Le noeud racine est toujours un noeud classificateur qui calcule la valeur symbolique de l'état de l'objet. La structure minimale d'arbre est réduite à trois noeuds, un noeud objet feuille, un noeud descripteur intermédiaire et un noeud racine classificateur. Le nombre de branches de l'arbre et la longueur des branches sont libres.

Objets. Les objets sont les objets de la scène au temps t , c'est à dire un élément de O , l'ensemble des objets $o_{i,j}$ où i est la classe de l'objet et j son étiquette. Par exemple l'objet $o_{person,1}$ est un objet mobile qui a été reconnu comme étant une personne et dont l'étiquette est 1. $o_{equipment,door2}$ est un objet appartenant à la classe équipement étiqueté door2.

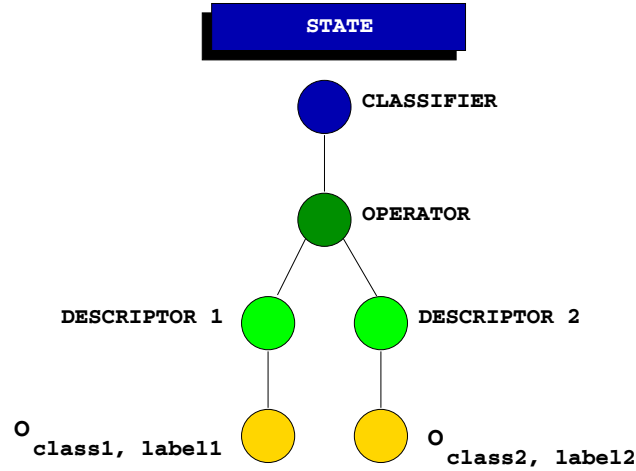


FIG. 6.4 – Un exemple de modélisation d'état. Les objets sont en gris clair, les descripteurs en gris, l'opérateur en gris foncé et le classificateur en noir.

Descripteurs. Les descripteurs sont des fonctions définies de O vers R^p pour accéder à une mesure de l'objet. Par exemple la taille, la position, la forme, la trajectoire, l'orientation ou le volume sont des descripteurs possibles. Cette notion assure l'ancrage du modèle dans les résultats numériques du module de perception.

Opérateurs. Les opérateurs sont des fonctions définies de $(R^{p_1} \times \dots \times R^{p_n})$ vers R^q afin d'opérer sur les mesures. Des exemples d'opérateurs sont la distance, la norme et les opérateurs arithmétiques et logiques classiques.

Classificateurs. Les classificateurs sont des fonctions de R^p vers S , l'ensemble des symboles autorisés pour les états. Par exemple *close* et *far* peuvent être des symboles possibles pour un état. Ces classificateurs assurent la transformation de nombres en symboles en définissant un domaine numérique de définition pour chaque valeur symbolique.

On a utilisé ce modèle d'état pour définir un premier ensemble d'états (voir deux exemples sur la figure 6.5). Pour cela on a défini trois classes d'objets, quatre descripteurs, quatre opérateurs et huit classificateurs. Les trois classes d'objets sont *person*, *area*, et *equipment*.

Les personnes sont des objets mobiles qui ont été détectés par le module de perception. Les personnes sont décrites par un vecteur (px_{3D}, py_{3D}) représentant la localisation de la personne sur le sol, un vecteur (vx_{3D}, vy_{3D}) représentant le vecteur vitesse de la personne et la taille h_{3D} de cette personne. Les zones et les équipements sont ceux qui sont définis dans le modèle 3D du décor (voir partie précédente). Une zone est un objet statique représentant une sous-partie du sol de la scène avec un polygone. Un équipement représente n'importe quel objet volumique de l'environnement pour lequel on connaît la base polygonale et la hauteur h .

Les quatre descripteurs sont: *position*, *size*, *speed* et *shape*. Plus précisément: $position(o_{i,j})$ appliqué à une personne donne accès à (px_{3D}, py_{3D}) , $size(o_{i,j})$ appliqué à une personne ou à un équipement permet de retrouver sa taille, $speed(o_{i,j})$ appliqué à une personne renvoie le vecteur vitesse (vx_{3D}, vy_{3D}) et $shape(o_{i,j})$ appliqué à un équipement ou à une zone renvoie son

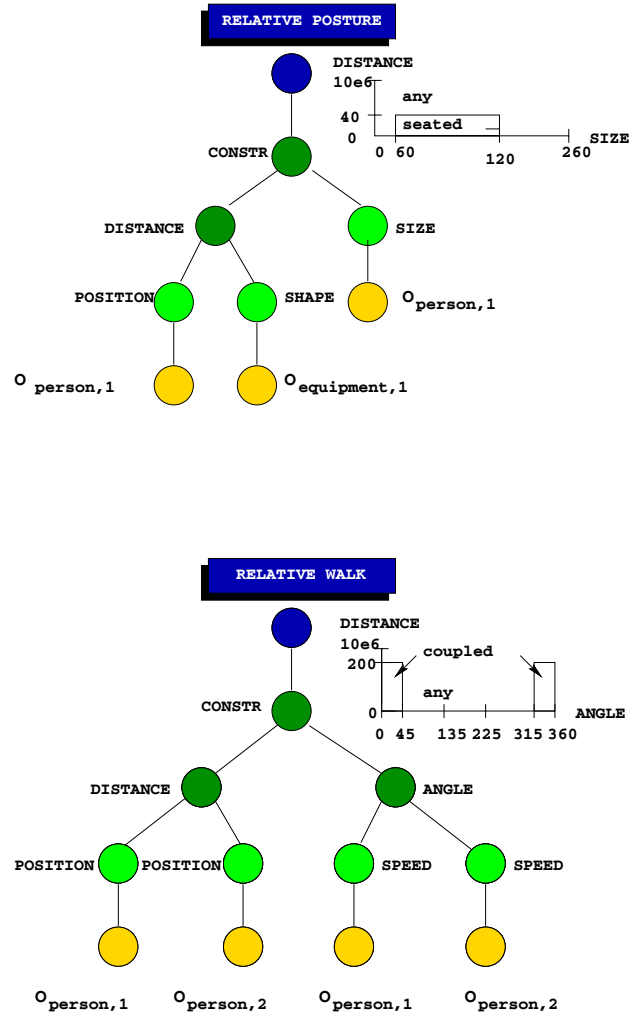


FIG. 6.5 – Deux instances de modèles d'état. Les objets sont en gris clair, les descripteurs en gris, l'opérateur en gris foncé et le classificateur en noir.

polygone associé.

Les quatre opérateurs sont: *distance* la distance euclidienne, *norm* la norme d'un vecteur, *angle* l'angle entre deux vecteurs en degrés et *constr* un opérateur qui construit un vecteur 2D avec ses composantes scalaires.

On a défini huit classificateurs qui calculent huit états: *posture*, *direction*, *velocity*, *location*, *proximity*, *relative location*, *relative posture* et *relative walk*. Par exemple on a défini (voir figure 6.5) l'état *relative walk*($o_{person,i}$, $o_{person,j}$) en mesurant l'angle entre les vecteurs vitesse de $o_{person,i}$ et $o_{person,j}$ et la distance entre ces personnes. Si les vecteurs vitesse ont une orientation similaire (un angle inférieur à 45 degrés ou supérieur à 315 degrés) et si la distance est petite (inférieure à 200cm) alors ces personnes sont considérées

comme ayant une marche relative coulée (*coupled*).

Reconnaissance d'événement

La reconnaissance d'événement est maintenant simple: pour chaque image les états des objets sont calculés pour les objets détectés dans la scène à cet instant. Si pour un objet détecté et pour un modèle d'état il y a un changement dans sa valeur symbolique un nouvel événement est créé à cet instant.

Les huit états prédéfinis permettent de définir dix-huit types d'événements.

Des changements de posture créent les événements $o_{person,i}$ **falls down** ou **crouches down** ou **stands up**.

Des changements de direction créent les événements $o_{person,i}$ **goes right side** or **goes left side**, ou **goes away** ou **arrives**. Des changements de vitesse créent les événements **stops** ou **walks** ou **starts running**.

Des changements de localisation créent les événements $o_{person,i}$ **leaves** ou **enters** $o_{area,j}$.

Des changements de proximité créent les événements $o_{person,i}$ **moves close to** ou **moves away from** $o_{equipment,j}$.

Des changements de localisation relative créent les événements $o_{person,i}$ **moves close to** ou **moves away from** $o_{person,j}$.

Des changements de posture relative créent les événements $o_{person,i}$ **sits on** $o_{equipment,j}$.

Des changements de marche relative créent les événements $o_{person,i}$ et $o_{person,j}$ **walk together**.

Reconnaissance de scénario

Le dernier problème est de reconnaître incrémentalement des scénarios prédéfinis représentant des comportements. Un scénario est un ensemble interdépendant d'événements. Reconnaître un scénario implique de reconnaître tous les événements qui le composent et de vérifier leur dépendances. Les contraintes peuvent être temporelles, spatiales, logiques ou algébriques. On va maintenant donner des détails sur le modèle que l'on utilise. Un scénario $s_{i,t}$, où i est l'identificateur du scénario et t l'instant courant de reconnaissance est composé de quatre parties: Des Events, Constraints, Conditions, et Success. Des exemples de scénarios sont montrés dans la partie suivante (voir figures 6.6, 6.7, 6.11 et 6.12).

Events. Ce sont les événements $\{e_1, \dots, e_i, \dots, e_n\}$ requis par le scénario. Chaque événement e_i est associé à la variable t_i qui représente l'instant où e_i s'est passé. Il y a deux catégories d'événements dans cette partie: les événements positifs et les événements négatifs. Les événements positifs doivent arriver pour la reconnaissance totale du scénario et les événements négatifs ne doivent pas arriver pendant la reconnaissance du scénario.

Constraints. Ce sont des contraintes temporelles $\{c_1, \dots, c_i, \dots, c_m\}$. Ces contraintes sont décrites comme des inéquations du premier degré en $t_1, \dots, t_i, \dots, t_n$.

Conditions. Ce sont des contraintes non temporelles $\{k_1, \dots, k_i, \dots, k_p\}$ sur les objets impliqués dans les événements. Cela oblige un attribut d'un objet d'un événement à avoir une valeur prédéfinie. Cet attribut peut être symbolique (nom, fonction, etc....) ou numérique (hauteur, taille, vitesse, etc...).

Success. Ce sont des mots clefs $\{f_1, \dots, f_i, \dots, f_q\}$, qui indiquent le type de feedback associé au scénario. Cette partie est utilisée lorsque le scénario est totalement instancié. Il ya deux sortes de feedback: externe et interne. UN feedback externe est utilisé pour déclencher une alarme

vers un opérateur de sécurité et un feedback interne est utilisé pour générer un événement qui indique que le scénario a été reconnu.

Un scénario peut être totalement reconnu, si tous les événements et toutes les contraintes sont vérifiés; il peut être partiellement reconnu si un sous-ensemble S des événements sont reconnus et si les contraintes sur ces événements de S sont vérifiés; lorsque aucun événement d'un scénario est reconnu ce scénario est appelé scénario blanc.

Le principe de l'algorithme de reconnaissance de scénarios [20] consiste en deux points: comme décrit précédemment, on génère image après image les événements intéressants qui se déroulent dans la scène, puis avec ces événements on instancie en parallèle des modèles de scénarios prédéfinis. Cela signifie que la reconnaissance de scénario correspond à la mise à jour d'un ensemble prédéfini de scénarios partiellement reconnus. Cette méthode reconnaissance de scénario est une extension du travail sur les chroniques expliqués dans [28].

En résumé, soit un ensemble de scénarios $\{s_{1,t-1}, \dots, s_{i,t-1}, s_{i+1,0}, \dots, s_{k,0}\}$ composé de scénarios partiellement reconnus à $t-1$ et de scénarios blancs et un ensemble d'événements $\{e_{1,t}, \dots, e_{n,t}\}$ reconnus à l'instant t , le principe de l'algorithme de reconnaissance de scénarios est basé sur deux points.

Pour chaque $s_j \in \{s_{1,t-1}, \dots, s_{i,t-1}, s_{i+1,0}, \dots, s_{k,0}\}$, pour chaque événement de s_j si l'événement correspond à un événement $e_{1,t}, \dots, e_{n,t}$ et vérifie les contraintes temporelles $c_1, \dots, c_i, \dots, c_m$ et les contraintes atemporelles $k_1, \dots, k_i, \dots, k_p$, on crée $s_{j,t}$. Il en résulte un nouvel ensemble $\{s_{1,t}, \dots, s_{l,t}\}$ de scénarios. Dans ce contexte un événement de s_j correspondant à $e_{1,t}$ signifie que $e_{1,t}$ est une instance de cet événement. On supprime les scénarios non valides $s_{i,t}$ de $\{s_{1,t}, \dots, s_{l,t}\}$, si:

- un événement négatif e_k de $s_{i,t}$ a été instancié,
- un des $c_1, \dots, c_i, \dots, c_m$ implique que $s_{i,t}$ ne sera pas instancié.

6.6 Résultats d'applications de surveillance de métro

Dans cette partie je décris les résultats obtenus sur des applications réelles de vidéosurveillance dans des stations de métro. Les vidéos proviennent des réseaux CCTV des opérateurs de stations de métro partenaires du projet européen AVS-PV. On a formalisé l'expertise de trois ingénieurs de sécurité dans une base de connaissances contenant 15 scénarios.

Station de métro à Bruxelles. Dans la suite je détaille comment deux scénarios décrits dans les figures 6.6 et 6.7 sont reconnus. Ces scénarios appartiennent à la base de connaissances construite dans le cadre du projet AVS-PV et les vidéos ont été enregistrées dans une station de métro de STIB à Bruxelles. Les caméras observent le quai d'une station de métro. Les buts de ces deux scénarios sont: de prévenir du vandalisme contre les équipements et d'assurer la sécurité des passagers.

A l'instant t_1 , une personne détectée (nommée Person 1 par la suite) pénètre à l'intérieur (*inside*) la zone des rails. L'événement "person 1 enters tracks area" est déclenché. La zone est étiquetée comme une zone interdite, aussi le premier événement du scénario "forbidden access to area" est reconnu. Plusieurs images plus tard, à l'instant t_2 (voir figure 6.8), la personne Person 1 est encore à l'intérieur de la zone des rails aussi l'événement "Person 1 exits tracks area" n'a pas été déclenché. La non occurrence de l'événement correspond au deuxième

Scenario

Name = "forbidden access to area",
Events = $(t_1, \text{enters}(p_1 : \text{Person}, a_1 : \text{Area})),$
 $\text{not}(t_2, \text{leaves}(p_1 : \text{Person}, a_1 : \text{Area})),$
Constraints = $t_1 \leq t_2, t_2 \leq t_1 + 1.0,$
Conditions = $\text{function}(a_1, \text{"forbidden_access"}),$
Success = $\text{alarm}(p_1, \text{"has entered area"}, a_1)$

FIG. 6.6 – Modèle de scénario AVS-PV: "forbidden access to area"

Scenario

Name = "graffiti on wall",
Events =
 $(t_1, \text{moves close to}(p_1 : \text{Person}, e_1 : \text{Equipment})),$
 $\text{not}(t_2, \text{moves away from}(p_1 : \text{Person}, e_1 : \text{Equipment})),$
Constraints = $t_1 \leq t_2,$
 $t_2 \leq t_1 + \text{normal_presence_time}(e_1),$
Conditions = $\text{function}(e_1, \text{"wall"}),$
Success = $\text{alarm}(p_1, \text{"doing graffiti onto"}, e_1)$

FIG. 6.7 – Modèle de scénario AVS-PV: "graffiti on wall"

événement (négatif) du scénario "forbidden access to area". Une alarme est envoyée à l'opérateur de sécurité.

Plus tard à l'instant t_3 (voir figure 6.9), la personne Person 1 est proche de l'équipement mur, aussi l'événement "Person 1 moves close to equipment wall" est déclenché. Cet événement instancie le premier événement du scénario "graffiti on wall".

Plus tard à l'instant t_4 (voir figure 6.10), la personne Person 1 est encore proche du mur aussi l'événement "person 1 moves off equipment wall" n'a pas été déclenché. La non occurrence de l'événement correspond au deuxième événement (négatif) du scénario "graffiti on wall". Une

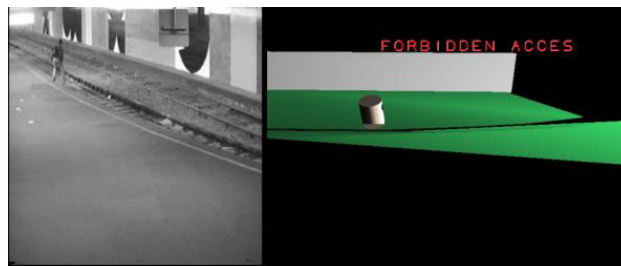


FIG. 6.8 – A gauche: un quai de métro à Bruxelles à t_2 . A droite: position 3D de la personne détectée (représentée par un cylindre) replacée dans le modèle 3D du décor. Une alarme "forbidden access to area" est envoyée à l'opérateur de sécurité.



FIG. 6.9 – A gauche: un quai de métro à Bruxelles à t_3 . A droite: un événement “person 1 moves close to equipment wall” est détecté.



FIG. 6.10 – A gauche: un quai de métro à Bruxelles à t_4 . A droite: une alarme “graffiti on wall” est envoyée à l’opérateur de sécurité.

alarme est envoyée à l’opérateur de sécurité.

Station de métro à Nuremberg.

Dans la suite je détaille comment deux autres scénarios décrits sur les figures 6.11 et 6.12 sont reconnus. Ces scénarios appartiennent également à la base de connaissances construire pour le projet AVS-PV et les vidéos ont été prises dans la station du métro de VAG à Nuremberg en Allemagne. Dans cet exemple, les caméras observent l’entrée d’une station de métro. Le but de ces scénarios est de prévenir le vandalisme contre des machines de distribution de tickets. Ces machines ont été définies dans le modèle 3D du décor comme un équipement fragile (voir partie 6.3).

A l’instant t_1 , une personne détectée (nommée Person 1 par la suite) est loin (*far*) d’un équipement étiqueté comme “fragile”. Plus tard, à l’instant t_2 (voir figure 6.13), la personne Person 1 est proche (*close*) de l’équipement étiqueté comme “fragile”, aussi l’événement “Person 1 moves close to an equipment” est déclenché. Le premier événement du scénario “Period near fragile equipment” est instancié.

Plus tard, à l’instant t_3 la personne Person 1 est encore proche de la machine, aussi l’événement négatif du scénario “Period near fragile equipment” est instancié. Puis le fait que la personne Person 1 s’arrête déclenche l’événement “Person 1 stops”. Les trois événements du scénario “Period near fragile equipment” sont reconnus et une alarme est envoyée vers l’opérateur de sécurité. La reconnaissance complète de ce scénario déclenche aussi un événement spécifique:

Scenario

Name = "Presence period near fragile equipment",

Events =

$(t_1, \text{moves close to } (p_1 : \text{Person}, e_1 : \text{Equipment})),$

$\text{not}(t_2, \text{moves away from } (p_1 : \text{Person}, e_1 : \text{Equipment})),$

$(t_3, \text{stops}(p_1 : \text{Person})),$

Constraints =

$t_1 \leq t_2, t_1 \leq t_3,$

$t_2 \leq t_1 + \text{normal_presence_time}(e_1),$

Conditions = function(e_1 , "fragile"),

Success =

alarm("Presence period near equipment", e_1),

loopback(t_2 , presence_period_near_fragile, e_1, p_1)

FIG. 6.11 – Modèle de scénario AVS-PV: "Presence period near fragile equipment"

Scenario

Name = "Repeated period near fragile equipment",

Events =

$(t_1, \text{presence_period_near_fragile}, e_1, p_1),$

$(t_2, \text{moves close to } (p_1 : \text{Person}, e_1 : \text{Equipment})),$

$\text{not}(t_3, \text{moves away from } (p_1 : \text{Person}, e_1 : \text{Equipment})),$

$(t_4, \text{stop}(p_1 : \text{Person}))$

Constraints =

$t_1 \leq t_2, t_2 \leq t_3, t_2 \leq t_4,$

$t_3 \leq t_2 + \text{normal_presence_time}(e_1),$

Conditions = function(e_1 , "fragile"),

Success = alarm("Vandalism on ", e_1)

FIG. 6.12 – Modèle de scénario AVS-PV: "Repeated Presence period near fragile equipment"

presence_near_fragile" equipment. Cet événement spécifique correspond au premier événement du scénario "Repeated Presence period near fragile equipment". Puis à l'instant t_4 (voir figure 6.14) un autre événement est détecté parce que la personne Person 1 s'éloigne dans la direction du couloir pour vérifier que personne n'arrive.

A l'instant t_5 (voir figure 6.15), l'événement "Person 1 *moves close to* equipment" est déclenché. Cet équipement est le même que l'équipement de l'instant t_2 . Le scénario "Repeated Presence period near fragile equipment" est maintenant totalement reconnu. Une alarme est envoyée vers l'opérateur de sécurité.

Les résultats de ces applications ont été considérés comme très satisfaisants pour les opérateurs de métro. La richesse du formalisme pour la description de scénarios permet de spécifier un ensemble de contraintes (temporelles aussi bien qu'atemporelles) qui réduit les fausses alarmes. Le formalisme proposé pour la description de scénario a permis de représenter l'expertise pour ces applications. La modélisation des connaissances est encore délicate. La raison principale est

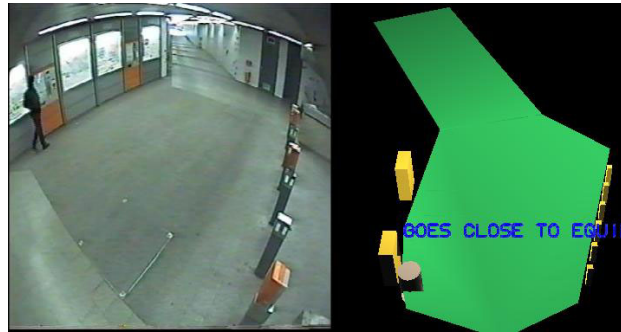


FIG. 6.13 – A gauche: une entrée du métro à Nuremberg à t_2 . A droite: un événement “person 1 moves close to an equipment” est détecté.



FIG. 6.14 – A gauche: une entrée du métro à Nuremberg à t_4 . A droite: un événement “person 1 moves away from an equipment” est détecté.



FIG. 6.15 – A gauche: une entrée du métro à Nuremberg à t_5 . A droite: une alarme “Vandalism” est envoyée à l’opérateur de sécurité.

que l'on a besoin de gérer le fossé entre des concepts vagues de sécurité (comme "comportement anormal") et des modèles rigoureux de scénarios.

6.7 Conclusion

Dans ce chapitre j'ai montré comment on peut effectuer une interprétation vidéo de haut niveau sur des images prises à partir d'une caméra statique et avec des méthodes simples de perception fonctionnant en temps-réel. Cela a été possible en utilisant deux ensembles de connaissances a priori: premièrement un modèle 3D décrivant la géométrie du décor observé et une information sémantique sur les objets statiques et les zones d'intérêt, deuxièmement une connaissance générale sur des scénarios prédéfinis valides pour un domaine d'application. On a proposé un formalisme pour représenter ces deux types de connaissance a priori et expliqué comment les utiliser pour l'interprétation vidéo. On a également proposé un formalisme pour la reconnaissance d'événements basé sur des modèles d'états des objets. Le formalisme est indépendant d'un domaine d'application particulier et il inclut une transformation des données de perception vers les modèles de scénarios. Le système actuel d'interprétation vidéo que l'on propose a quelques limitations. Un type de problèmes est l'imprécision et l'incertitude dans la détection et la localisation des objets mobiles; la plupart des erreurs de détection bas-niveau sont dues à des reflets, des ombres ou des occultations. Une solution pour résoudre ces problèmes est de relâcher la deuxième hypothèse sur l'utilisation d'une seule caméra. Un autre problème général est que comme pour tout système de vision cette approche nécessite pour chaque méthode de perception et pour chaque méthode d'interprétation de régler les valeurs de paramètres numériques dans une phase de configuration. Une solution pour résoudre ce problème est d'utiliser des techniques d'apprentissage pour trouver les meilleures valeurs de paramètre pour une application si elles existent. Une autre direction de recherche intéressante que je cherche à développer est d'utiliser des techniques de pilotage de programmes pour améliorer la flexibilité du système aussi bien en terme d'adaptation des méthodes que de paramétrisation (voir chapitre 5 précédant).

Chapitre 7

Perspectives

Les travaux présentés dans les chapitres précédants ont permis d'étudier comment formaliser et mettre en oeuvre des connaissances a priori pour résoudre deux problèmes différents **l'interprétation sémantique d'images et l'analyse automatique d'images**.

Les chapitres 4 et 6 ont décrit deux types de travaux que j'ai effectués pour l'interprétation d'images, à savoir la classification sémantique d'objets naturels et l'interprétation sémantique de séquences video. Le chapitre 5 a décrit les techniques de pilotage de programmes que j'ai proposées pour l'analyse automatique d'images.

Dans chaque chapitre j'ai montré l'approche générale proposée avec la modélisation des connaissances et le type de raisonnement pour la mise en oeuvre de cette connaissance. Afin de montrer l'intérêt et la faisabilité de ces approches plusieurs applications ont été présentées plus ou moins en détail.

On peut en conclure qu'il est possible de construire des systèmes automatiques "intelligents" au sens où ils incorporent effectivement une connaissance experte. Cependant il reste un certain nombre de problèmes ouverts et je vais décrire comment je compte approfondir ces travaux.

Directions de recherches futures

Tout d'abord on peut faire une remarque générale qui est vraie pour les trois problèmes traités dans les chapitres précédants: si l'on dispose maintenant de moteurs dédiés efficaces, voire de plate-formes de développement, il n'est toujours pas très aisé et rapide de construire des bases de connaissances. Il y a plusieurs raisons à cela, la première est qu'il est bien sur nécessaire de posséder une expertise la deuxième est que souvent une certaine pluri-disciplinarité est demandée, enfin ces bases de connaissances demandent aussi détablir la valeur précise de paramètres numériques tels des seuils qui ne sont pas toujours connus.

Plusieurs points nouveaux doivent être développés dans les années à venir dont:

– Classification d'objets

Un premier objectif est de développer un langage de description de concepts visuels destiné à un expert non informaticien pour la définition d'une catégorie d'objet à reconnaître. En effet, S'il semble logique de demander à un réel expert d'un domaine technique (biologiste, astronome, expert en traitement d'images, expert en vidéosurveillance, etc...) de construire une base de connaissances il faut alors lui permettre de s'affranchir au maximum des points qui ne font pas partie de son expertise. Pour cela je propose notamment de regarder du côté des travaux en acquisition des connaissances et d'utiliser des ontologies. Plus précisément, l'idée est de définir une ontologie de concepts visuels (par exemple contenant les concepts de forme, de texture et de couleur) et de proposer cette ontologie pour faciliter les échanges de connaissances entre experts d'un domaine applicatif en classification d'objet et experts en techniques de traitement d'images. Une thèse est en cours sur ce thème (Nicolas Maillot).

Un deuxième objectif très ambitieux est d'automatiser la construction de méthodes de reconnaissance de forme correspondant aux concepts visuels choisis par l'expert. Je souhaite pour cela étudier la coopération de techniques d'apprentissage et de techniques de

pilotage de programmes. Bien sur, tout le contenu d'une base de connaissances n'est pas forcément détenu sous forme consciente par un expert. Il est donc important de regarder dans quelle mesure il est possible et utile de compléter avec des techniques d'apprentissage automatique le contenu d'une base de connaissances. Plusieurs expériences préliminaires ont déjà abordé cette question sous la forme de stages de DEA, mais je compte développer cet axe de recherches prometteur.

– **Pilotage de programmes:**

Un premier objectif est l'extension du modèle de pilotage pour la prise en compte des aspects temporels et des flots de données. Une thèse est en cours en coopération avec l'Université UCL en Belgique (Benoit Georis).

– **Pilotage de programmes et Interprétation**

La conception et la construction de moteurs de systèmes à base de connaissances étant facilitée par l'existence d'une plate-forme telle que LAMA, un de mes objectifs à moyen terme est de pouvoir construire avec LAMA de nouveaux moteurs d'interprétation d'images. J'attends notamment comme résultat de ces travaux de pouvoir coupler des raisonnements de pilotage de programmes et d'interprétation d'images. Une thèse est en cours sur une nouvelle architecture d'analyse de scènes (Céline Hudelot).

– **Interprétation vidéo**

Un premier objectif est d'améliorer les modèles de représentation de scénarios dans deux directions, d'une part en proposant un langage convivial pour les experts évitant des formules de logique temporelle trop déroutantes et d'autre part en proposant des algorithmes efficaces non seulement de reconnaissance de scénarios mais aussi de compilation des bases de connaissances de scénarios. Une thèse est en cours sur ce sujet (Thinh van Vu).

Un deuxième objectif est d'étendre les modèles d'interprétation vidéo pour des séquences observées simultanément par plusieurs capteurs. En effet cela pose des problèmes intéressants d'indépendance entre la représentation des modèles d'interprétation et les données issues de la perception. Par exemple, je prévois d'étudier le cas de plusieurs caméras observant la même scène avec recouvrement partiel et celui de caméras observant des parties différentes d'un même site.

Un troisième objectif est de faciliter la construction de systèmes d'interprétation vidéo à l'aide de techniques d'apprentissage. Par exemple, il s'agira d'adapter les valeurs de paramètres à certaines applications: paramètres numériques de segmentation du mouvement, de suivi de personnes ou paramètres intervenant dans les modèles de personnes, d'événements ou de scénarios.

Enfin, j'envisage d'étudier le problème de la représentation des résultats de l'interprétation. Il s'agira de concevoir des représentations visuelles et textuelles du comportement de personnes en mouvement et de scénarios reconnus. Un des intérêts de cet objectif est l'annotation automatique de vidéos. Cela pose des problèmes difficiles de granularité temporelle (résumer des éléments importants d'une vidéo d'une journée ou d'une minute) et de représentation graphique de concepts abstraits (événement ou scénario).

A plus long terme je souhaite concevoir une plateforme intelligente de vision cognitive. Ces travaux seront effectués dans le cadre du réseau d'excellence ECVision en vision cognitive.

Définition: Une plateforme intelligente de vision cognitive est un système d'interprétation de scène ayant un degré complet d'autonomie pour s'adapter à de nouvelles situations, ayant des capacités d'interaction élevées grâce à un langage convivial lui permettant de comprendre les besoins de l'utilisateur et ayant de nombreux moyens pour communiquer son interprétation des scènes observées.

Autonomie:

La plateforme intelligente de vision cognitive aura des capacités d'autonomie selon deux modes différents: configuration statique et reconfiguration dynamique. Dans une phase de configuration la plateforme peut sélectionner automatiquement les algorithmes pertinents de vision à partir d'une bibliothèque de programmes et peut traiter des données provenant d'un ensemble hétérogène de capteurs (plug and play). Cette caractéristique est nécessaire pour faciliter le passage d'une application à l'autre et d'un site à un autre. Dans une phase de reconfiguration dynamique la plateforme doit s'adapter elle-même en temps-réel. Cette caractéristique est nécessaire pour une robustesse de fonctionnement 7 jours sur 7 et 24 heures sur 24.

Interactivité:

La plateforme intelligente de vision cognitive doit adapter son comportement directement à partir de spécifications de l'utilisateur final. En particulier un langage de haut niveau basé sur une ontologie de concepts visuels permettra de décrire de nouvelles classes, de nouveaux événements ou des activités complexes.

Communication:

La plateforme intelligente de vision cognitive doit disposer d'outils de visualisation configurables ("scalable") allant de simple écran de palm pilot à des environnements immersifs, avec la transmission temps-réel des données pertinentes et l'annotation automatique des images et des vidéos pour l'archivage.

Comment atteindre cet objectif?

Cet objectif est très ambitieux, cependant l'état de l'art des techniques en vision cognitive a montré qu'elles peuvent proposer des solutions partielles. Il y a un besoin de rassembler les meilleures techniques dans un cadre unique et de définir des standards. Les activités de recherches suivantes doivent être effectuées pour atteindre cet objectif: la création d'une bibliothèque standard de l'état de l'art en vision, le développement de techniques de pilotage de programmes, le développement de raisonnement spatio-temporel temps-réel, la conception d'une ontologie de concepts visuels, la création d'un langage haut-niveau pour le dialogue avec l'utilisateur final, le développement de techniques d'apprentissage numériques et symboliques et l'utilisation de techniques de simulation.

Applications cibles

Cette plateforme sera une solution pour plusieurs applications difficiles. Parmi celles-ci on peut citer les smart environments (environnement médiatisés), la vidéosurveillance et les virtual teachers.

Conclusion

Pour atteindre cet objectif ambitieux je propose de fédérer la recherche en vision cognitive et de concrétiser les résultats obtenus sous la forme d'une plateforme commune, afin de montrer que les techniques de vision cognitive fonctionnent effectivement et permettent de démarrer une nouvelle génération de systèmes de vision.

Références

- [1] E. Andr, G. Herzog, and T. Rist. On the simultaneous interpretation of real world image sequences and their natural language description: The system soccer. In *8th European Conference of Artificial Intelligence*, pages 449 – 454, Munich, 1988.
- [2] D. Ayers and M. Shah. Monitoring human behavior in an office environment. In *Computer Society Workshop on Interpretation of Visual Motion*, 1998.
- [3] D. G. Bailey. Research on Computer-assisted Generation of Image Processing Algorithms. In *Workshop on Computer Vision - Special Hardware and Industrial Applications*, Tokyo, October 1988. IAPR.
- [4] F. Brémond and M. Thonnat. Issues of representing context illustrated by video-surveillance applications. *International Journal of Human-Computer Studies Special Issue on Context*, 48:375–391, 1998.
- [5] R.A.Brooks, Symbolic Reasoning Among 3-D Models and 2-D Images, *Artificial Intelligence Journal* 17, 1981, 285-348.
- [6] A. Barr, P.R. Cohen and E.A. Feigenbaum editors. *The Handbook of Artificial Intelligence*, volume 4 Addison-Wesley, Reading, Massachusetts, 1989.
- [7] A. Barr and E.A. Feigenbaum editors. *The Handbook of Artificial Intelligence*, volume 1 HeurisTech Press and William Kaufman, Standford., 1981.
- [8] A. Barr and E.A. Feigenbaum editors. *The Handbook of Artificial Intelligence*, volume 2 HeurisTech Press and William Kaufman, Standford., 1982.
- [9] A. Bellon, J-P. Dérutin, F. Heitz, and Y. Ricquebourg. Real-Time Collision Avoidance at Road Crossing On Board the Prometheus Prolab2 Vehicle. In *Intelligent Vehicles'94 Symposium*, pages 56–61, Paris, (France), October 1994.
- [10] S. Beucher and M. Bilodeau. Road Segmentation and Obstacle Detection by a Fast Watershed Transformation. In *Intelligent Vehicles'94 Symposium*, pages 296–301, Paris, (France), October 1994.
- [11] A. Boersma, "Introduction to marine micropaleontology", in *Foraminifera*, B. U. Haq and A. Boersma, ed., Elsevier, New-York, pp. 19–78, 1978.
- [12] British-Aerospace. VIDIMUS Esprit Project Annual Report. Technical report, Sowerby Research Centre, Bristol, England, 1991.
- [13] D. R. Brough, I. F. Alexander, "The fossil expert system", *Expert Systems*, Vo. 3, No. 2, pp.76–83, 1986.
- [14] H. Buxton and S. Gong. Visual surveillance in dynamic and uncertain world. *Artificial Intelligence Journal*, 78:431 – 459, 1995.

- [15] M. Caron, *Cretaceous planktonic foraminifera*, Cambridge University Press, Eds. H. M. Bolli et al., pp. 17–85, 1985.
- [16] C. Castel, L. Chaudron, and C. Tessier. What is going on? a high level interpretation of sequences of images. In *4th European Conference on Computer Vision, Workshop on Conceptual Descriptions from Images*, Cambridge UK, April 1996.
- [17] R. Chelappa, C. Wilson and S. Sirohey, “Human and machine recognition of faces: a survey”, *Proceedings IEEE* **83**(5), 705-740, 1995.
- [18] S. A. Chien and H. B. Mortensen. Automating image processing for scientific data analysis of a large image database. *IEEE Transactions on Pattern Analysis and Machine Intelligence* , 18(8):854–859, August 1996.
- [19] N. Chleq, F. Bremond, and M. Thonnat. *Advanced Video-Based Surveillance Systems*, chapter Image Understanding for Prevention of Vandalism in Metro Station, pages 106 –117. Kluwer Academic Publishers, 1999.
- [20] N. Chleq and M. Thonnat. Realtime image sequence interpretation for videosurveillance. In IEEE, editor, *International Conference on Image Processing*, pages 801 – 804, Lausanne, Switzerland, 1996.
- [Cle90] V. Clément, ‘Raisonnements cognitifs appliqués au pilotage d’algorithmes de traitement d’image’, Thèse de doctorat, Université de Nice, 1990.
- [21] V. Clément and M. Thonnat. Integration of image processing procedures: OCAP, a knowledge based approach. *Computer Vision Graphics and Image Processing: Image Understanding*, 57(2), 1993.
- [22] P.R. Cohen and E.A. Feigenbaum editors. *The Handbook of Artificial Intelligence*, volume 3 HeurisTech Press and William Kaufman, Standford., 1982
- [23] I. Cohen and G. Medioni. Detecting and tracking moving objects for video surveillance. In *Computer Vision and Pattern Recognition*, Fort Collins, Colorado, June 1999.
- [24] M. A. Conrad, D. S. Beightol, “Expert systems identify fossils and manage large paleontological databases”, *Geobyte*, Vol. 3, No. 1, pp. 42-46, 1988.
- [25] M. Crubézy, ‘Pilotage de programmes pour le traitement d’images médicales’, Thèse de doctorat, Université de Nice, 1999.
- [26] M. Crubézy, F. Aubry, S. Moisan, V. Chamero, M. Thonnat, and R. Di Paola. Managing Complex Processing of Medical Image Sequences by Program Supervision Techniques. In *SPIE Medical Imaging’97*, volume 3035, pages 614–625, February 1997.
- [27] S. Dettmer, A. Seetharamaiah, L. Wang, and M. Shah. Model-based approach for recognizing human activities from video sequences. In *Workshop on Motion of Non-Rigid and Articulated Objects*, June 1998.
- [28] C. Dousson, P. Gaborit and M. Ghallab. Situation recognition:representation and algorithms. In *Proc. 13th IJCAI*, Chambéry, p.166-172, 1993.
- [29] B. Draper, A. Hanson, and E. Riseman. Knowledge-directed vision: Control, learning, and integration. *IEEE Signals and Symbols*, 84(11):1625–1637, 1996.
- [30] R. Duda and P. Hart, *Pattern Classification and Scene Analysis*, Wiley, 1973.
- [31] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, Academic Press. 2nd Edition, 1990.

- [32] M.-H. Gandelin, *Une approche système expert pour la reconnaissance automatique d'organismes marins à partir d'images monoculaires*, Thèse de doctorat, Université de Nice - Sophia Antipolis, France, Décembre 1989.
- [33] L. Gong and C. A. Kulikowski. Composition of Image Analysis Processes Through Object-Centered Hierarchical Planning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(10), October 1995.
- [Gra85a] C. Granger, 'Reconnaissance d'objets par mise en correspondance en vision par ordinateur', Thèse de doctorat, Univ. de Nice, Juin 1985.
- [Gra85b] C. Granger, M. Thonnat, 'Un système de vision fondé sur une méthode structurée de classification', *Cognitive*, 137-142, Paris, Juin 1985.
- [34] Eric L. Grimson. *Object Recognition by Computer Vision, The Role of Geometric Constraints*, The MIT Press, Cambridge Massachusetts, 1990
- [35] G. Herzog. Utilizing interval-based event representation for incremental high-level scene analysis. In *4th International Workshop on Semantics of Time, Space, and Movement and Spatio-Temporal Reasoning*, Chateau de Bonas, France, 1992.
- [36] K. Ikeuchi and T. Kanade, Automatic Generation of Object Recognition Programs, in *Proceedings of the IEEE* 78.8, 1988, pp.1016-1035.
- [Ilo88] 'CLASSIC générateur de systèmes experts de classification et de diagnostic. Manuel de l'utilisateur', ILOG, Paris 1988.
- [37] S. Intille and A. Bobick. Closed world tracking. In *5th International Conference on Computer Vision*, Cambridge, 1995.
- [38] S. Intille and A. F. Bobick. Visual recognition of multi-agent action using binary temporal relations. In *Computer Vision and Pattern Recognition*, Fort Collins, Colorado, June 1999.
- [39] Y. Ivanov, C. Stauffer, A. Bobick, and W. Grimson. Video surveillance of interactions. In *2nd International Workshop on Visual Surveillance*, pages 82 – 89, Fort Collins, Colorado, June 1999.
- [40] M.D. Johnston. An Expert System Approach to Astronomical Data Analysis. In *Proceedings, Goddard Conf. on Space Applications of Artificial Intelligence and Robotics*, pages 1–17, 1987.
- [41] S. Liu, M. Thonnat and M. Berthod. Automatic Classification of Planktonic Foraminifera by a Knowledge-based System. In *Proceedings of The Tenth Conference on Artificial Intelligence for Applications, CAIA*, IEEE Computer Society Press San Antonio, Texas, pp358-364, March, 1994.
- [42] S. Liu-Yu, *Reconnaissance de formes par vision par ordinateur: application à l'identification des foraminifères planctoniques*, Thèse de doctorat, Université de Nice - Sophia Antipolis, France, June 1992.
- [43] T. Matsuyama. Expert systems for image processing: Knowledge based composition of image analysis processes. *Computer Vision Graphics Image Processes*, 48:22–49, 1989.
- [44] G. Medioni, I. Cohen, F. Brmond, S. Hongeng, and R. Nevatia. Event detection and analysis from video streams. In *DARPA Image Understanding Workshop*, Monterey, November 1998.

- [45] A. Meygret, M. Thonnat, and M. Berthod. A pyramidal stereovision algorithm based on contour chain points. In *European Conference on Computer Vision*, Antibes, April 1990.
- [46] S. Moisan, C. Shekhar, and M. Thonnat. Real-Time Perception Program supervision for Vehicle Driving Assistance. In Okyay Kaynak, Mehmed Ozkan, Nurdan Bekiroglu, and Ilker Tunay, editors, *ICRAM'95 International Conference on recent Advances in Mechatronics*, pages 173–179, Istanbul, Turkey, August 1995.
- [47] S.H.Nawab and V.Lesser. Integrated processing and understanding of signals. In A.V.Oppenheim and S.H.Nawab, editors, *Symbolic and Knowledge-based Signal Processing*, pages 251–285. Prentice Hall, 1992.
- [48] J.-C. Ossola, *Coopération de systèmes à base de connaissances pour l'analyse et la reconnaissance d'objets naturels complexes: application au classement de galaxies ou de zooplanctons*, Thèse de doctorat, Université de Nice - Sophia Antipolis, France, Octobre 1996.
- [49] J.C. Ossola and M. Thonnat. Cooperation of knowledge based systems for galaxy classification. In *8th International Conference on Image Analysis and Processing (ICIAP'95)*, September 1995.
- [50] S. Pouget, M. Thonnat, V. Clement, A. de Fombelle, "Automatic diagnosis of a technical device by classification, the expert system DANTE: an application to antenna failures", *6th International conference on reliability and maintainability*, Strasbourg, October 1988.
- [51] Groupe PROART. Rapport de fin de contrat de recherche PROMETHEUS-PROART. Technical report, Projet PROMETHEUS, December 1994.
- [52] W. R. Riedel, "IDENTIFY: A Prolog program to help identify fossils", *Computer & Geosciences*, Vol. 15, No. 5, pp. 809–823, 1989.
- [53] F. Robaszynski, M. Caron, J. M. Gonzalez Donoso, A. H. Wonders, "Atlas of Late Cretaceous Globotruncanids", *Revue de Micropaléontologie*, Vol. 26, No. 3–4, Maison de la Géologie, Paris, 1984.
- [54] N. Rota, R. Stahr and M. Thonnat. Tracking for Visual Surveillance in VSIS. In *PETS2000*, Grenoble, March 2000.
- [55] S. Russel and P. Norvig. *Artificial Intelligence: a modern approach*, Prentice Hall International Editions, 1995
- [56] H. Sato, Y. Kitamura, and H. Tamura. A Knowledge-based Approach to Vision Algorithm Design for Industrial Parts Feeder. In *Proceedings, IAPR Workshop on Computer Vision, Special hardware and industrial applications*, pages 413–416, Tokyo, 1988.
- [57] C. Shekhar, S. Moisan, and M. Thonnat. Towards an intelligent problem-solving environment for signal processing. *Mathematics and Computers in Simulation*, 36, 1994.
- [58] P. A. Swaby, "Integrating Artificial Intelligence and Graphics in a Tool for Microfossil Identification for Use in the Petroleum Industry", *Second Annual Conference on Innovative Applications of Artificial Intelligence*, pp. 203 – 218, Washington, 1st – 3rd May, 1990.
- [59] C. Tessier. Reconnaissance de scènes dynamiques partir de données issues de capteurs: le projet perception. Technical report, Onera-Cert, 2 avenue Edouard-Belin BP 4025 31055 Toulouse Cedex France, Aot 1997.
- [Tho85] M. Thonnat. Automatic morphological description of galaxies and classification by an expert system. Rapport de Recherche 387, INRIA, March 1985. XS

- [Tho89] M. Thonnat. *The world of galaxies*, chapter Toward an automatic classification of galaxies, pages 53–74. Springer-Verlag, 1989.
- [60] M. Thonnat Knowledge-based techniques for image processing and for image understanding, pages 189–236. In *Journal Phys.IV France*, 12 (2002) EDP Sciences, Les Ulis.
- [TB89] M. Thonnat and A. Bijaoui. Knowledge-based galaxy classification systems. In A. Heck and F. Murtagh, editors, *Knowledge-based systems in astronomy*, volume 329 of *Lecture Notes in Physics*. Springer Verlag, 1989.
- [61] M. Thonnat, V. Clément, and J. C. Ossola. Automatic galaxy description. *Astrophysical Letters and Communication*, 31(1-6):65–72, 1995.
- [62] M. Thonnat, V. Clement, and J. van den Elst. Supervision of perception tasks for autonomous systems: the OCAP approach. *Journal of Information Science and Technology*, 3(2):140–163, Jan 1994. Also in Rapport de Recherche 2000, 1993, INRIA Sophia Antipolis.
- [63] M. Thonnat and M-H. Gandelin, “An expert system for the automatic classification and description of zooplanktons from monocular images”, ICPR9, pp. 114–118, Roma, Italy, November 1988.
- [64] M. Thonnat et M.-H. Gandelin. Un Système expert pour la description et le classement automatiques de zooplanctons à partir d’images monoculaires. *Traitement du Signal*, volume 9 no 5, janvier 1993.
- [65] M. Thonnat, C. Granger, M. Berthod, “Design of an expert system for object classification through an application to the classification of galaxies”, *Proc. of CVPR’85*, pp. 206–208, 1985.
- [66] M. Thonnat and S. Moisan. Knowledge-based systems for program supervision. In *First international workshop on Knowledge-Based systems for the (re)Use of Programs libraries KBUP’95*, Sophia Antipolis, France, 1995. INRIA.
- [67] M. Thonnat and S. Moisan. Knowledge-Based Systems for Program Supervision. In *Proceedings of KBUP’95*, pages 3–8, Sophia Antipolis, France, November 1995. INRIA.
- [68] M. Thonnat and S. Moisan. What can Program Supervision do for Program Reuse? *IEE Proceedings - Software.*, 147(5), October 2000.
- [69] M. Thonnat and N. Rota. Image understanding for visual surveillance applications. In *Third International Workshop on Cooperative Distributed Vision*, Kyoto, Japan, November 1999.
- [70] T. Toriu, H. Iwase, and M Yoshida. An Expert System for Image Processing. *FUJITSU Sci.Tech.Journal*, 23.2:111–118, 1987.
- [71] L. Trassoudaine, J. Alizon, F. Collange, and J. Gallice. Visual Tracking by a Multisensorial Approach. In *First IFAC International Workshop on Intelligent Autonomous Vehicles*, pages 111–116, Southampton, (UK), April 1993.
- [72] J. Van Den Elst, ‘Modélisation de connaissances pour le pilotage de programmes de traitement d’images’, Thèse de doctorat, Université de Nice, 1997.
- [73] G. de Vaucouleurs. Classification and morphology of external galaxies. *Handbuch der Physik*, 53:275, March 1957.
- [74] R. Vincent, ‘Etude des mécanismes de gestion des erreurs dans les systèmes à base’, Thèse de doctorat, Université de Nice, 1997.

- [75] R. Vincent, S. Moisan and M. Thonnat. Une bibliothèque pour des moteurs de pilotage de programmes. Research Report 3011, I.N.R.I.A., 1996. <http://www.inria.fr/orion/>.
- [76] R. Vincent and M. Thonnat. Planning, Executing, Controlling and Replanning for IP program library. In *8th Artificial intelligence and Soft computing (ASC'97)*, Banff, July 1997.
- [77] R. Vincent, M. Thonnat, and J.C. Ossola. Program supervision for automatic galaxy classification. In *Proc. of the International Conference on Imaging Science, Systems, and Technology CISST'97*, Las Vegas, USA, June 1997.
- [78] S. Yu, P. Saint-Marc, M. Thonnat and M. Berthod. Feasibility study of automatic identification of planktic foraminifera by computer vision. In *Journal of Foraminiferal Research*, 26, (2), April 1996.

Annexe A

Sélection d'articles (document annexe)

"Knowledge-based galaxy classification systems." M. Thonnat and A. Bijaoui. In A. Heck and F. Murtagh, editors, *Knowledge-based systems in astronomy*, volume 329 of *Lecture Notes in Physics*. Springer Verlag, 1989.

"Un Système expert pour la description et le classement automatiques de zooplanctons à partir d'images monoculaires." M. Thonnat et M.-H. Gandelin. *Traitement du Signal*, volume 9 no 5, janvier 1993.

"Feasibility study of automatic identification of planktonic foraminifera by computer vision." S. Yu, P. Saint-Marc, M. Thonnat, and M. Berthod. *Journal of Foramineferal Research*, 26(2), April 1996.

"A Knowledge-Based Approach to Integration of Image Processing." V. Clément and M. Thonnat. *Computer Vision Graphics and Image Processing: Image Understanding*, 57(2), March 1993.

"What can Program Supervision do for Program Reuse?" M. Thonnat and S. Moisan. *IEE Proceedings - Software.*, 147(5), October 2000.

"Issues of representing context illustrated by video-surveillance applications." F. Brémond and M. Thonnat. *International Journal of Human-Computer Studies Special Issue on Context*, 48:375–391, 1998.

"Knowledge-based techniques for image processing and for image understanding" M. Thonnat, In *Journal Phys.IV France*, 12 (2002), EDP Sciences, Les Ulis

Table des matières

1	Curriculum vitae	5
1.1	Curriculum vitae	7
1.2	Activité Professionnelle	7
1.3	Formation	7
1.4	Enseignement	7
1.5	Encadrement d'activités de recherche	8
1.6	Valorisation et transfert technologique	9
1.7	Réalisation et diffusion de logiciels	10
1.8	Diffusion de l'information scientifique	10
1.9	Mobilité	10
1.10	Responsabilités collectives	11
1.10.1	Tâches d'intérêt collectif à l'Inria	11
1.10.2	Responsabilités scientifiques	11
1.11	Autres éléments	12
1.12	Résumé de l'activité de recherche	12
2	Publications	15
3	Thèmes de recherches	25
3.1	Objectif	27
3.2	Résumé des travaux présentés et plan du manuscrit	28
3.2.1	Résumé des travaux présentés	28
3.2.2	Plan du manuscrit	30
4	Classification d'objet	31
4.1	Introduction	33
4.2	L'approche CLASSIC	34
4.2.1	L'organisation de la connaissance	35

4.2.2	La base de faits	37
4.2.3	Le fonctionnement du moteur	38
4.3	Le système SYGAL	38
4.3.1	Prétraitement	39
4.3.2	Extraction de paramètres	40
4.3.3	Exemple de paramètres mesurés	44
4.3.4	Classification automatique	44
4.3.5	Exemple de session	50
4.3.6	Résultats	50
4.3.7	Limitations	51
4.3.8	Conclusion	52
4.4	Classification de foraminifères planctoniques	52
4.4.1	Introduction	52
4.4.2	La base de connaissances	54
4.4.3	La base de faits	57
4.4.4	Exemple de session	57
4.4.5	Conclusion	60
4.5	Classification et retour bas-niveaux	60
4.6	Classification et pilotage de programmes	62
4.7	Conclusion	64
5	Pilotage de programmes	65
5.1	Introduction	67
5.2	Pilotage de programmes	68
5.3	Application des techniques de pilotage de programmes	73
5.3.1	Détection d'obstacles routiers par stéréovision	74
5.3.2	Détection temps-réel d'obstacle routiers	74
5.3.3	Imagerie médicale	75
5.4	Pilotage de programmes de traitement d'image de galaxies	76
5.4.1	Contenu de la base de connaissances	78
5.4.2	Connaissance de planification	78
5.4.3	Connaissance de réparation	83
5.4.4	Conclusion	87
5.5	Conclusion	88

6	Interprétation vidéo	89
6.1	Introduction	91
6.2	Etat de l'art	91
6.3	Perception	92
6.4	Modèle 3D du décor	94
6.5	Interprétation	96
6.6	Résultats d'applications de surveillance de métro	100
6.7	Conclusion	105
7	Perspectives	107
A	Sélection d'articles (document annexe)	119