

**UNIVERSITE JOSEPH FOURIER - GRENOBLE 1
SCIENCES & GEOGRAPHIE**

THESE

pour obtenir le grade de
DOCTEUR DE L'UNIVERSITE JOSEPH FOURIER

Discipline : INFORMATIQUE

Présentée et soutenue publiquement
par

Alain BOUCHER

Le 18 janvier 1999

**UNE APPROCHE DECENTRALISEE ET ADAPTATIVE
DE LA GESTION D'INFORMATIONS EN VISION
Application à l'interprétation d'images
de cellules en mouvement**

Directeur de thèse : Catherine Garbay

Composition du jury :

M. Yves CHIARAMELLA	Président
Mme Marinette REVENU	Rapporteur
Mme Monique THONNAT	Rapporteur
M. Patrice DALLE	Examineur
Mme Catherine GARBAY	Examineur
M. Xavier RNOT	Invité

Thèse préparée au sein du laboratoire TIMC-IMAG

Remerciements

Je remercie Mme Catherine GARBAY, Chargée de recherches CNRS et responsable de l'équipe SIC, qui m'a accueilli dans son équipe et qui a encadré mes travaux durant ces quatre années passées à Grenoble. Par ses conseils et son dévouement constant, elle m'a aidé à développer mes aptitudes et mon intérêt pour la recherche. Je lui en suis très reconnaissant.

Je remercie M. Xavier RONOT, Maître de conférence à l'École Pratique des Hautes Études, avec qui j'ai eu le plaisir de collaborer tout au long de mes travaux et qui a participé au jury de cette thèse.

Je remercie M. Yves CHIARAMELLA, Professeur à l'Université Joseph Fourier de Grenoble, pour avoir accepté de présider le jury de cette thèse.

Je remercie Mme Marinette REVENU, Professeur à l'Université de Caen, ainsi que Mme Monique THONNAT, Directeur de recherches INRIA à Sophia Antipolis, pour avoir accepté d'être rapporteurs de cette thèse.

Je remercie M. Patrice DALLE, Maître de conférence à l'Université Paul Sabatier de Toulouse, pour sa participation au jury de cette thèse.

Je remercie Anne DOISY, ainsi que les membres de l'équipe DyOGen avec qui j'ai eu l'occasion de travailler pendant cette thèse, pour l'aide apportée, ainsi que pour toutes les séquences d'images que j'ai pu utiliser.

Je remercie toutes les personnes que j'ai pu cotoyer dans les équipes SIC, INFODIS, RFMQ, ainsi que des autres équipes de l'Institut Albert Bonniot, autant pour les échanges et les discussions aux niveaux techniques et scientifiques que nous avons eus que pour tous les bons moments passés ensemble. Je n'ose pas nommer tous leurs noms de peur d'en oublier.

Cette thèse a été possible grâce à un financement du Conseil de Recherches en Sciences Naturelles et en Génie du Canada (Bourse CRSNG 1967).

Enfin, je remercie ma famille, pour l'encouragement constant tout au long de mes études en France, ainsi que tous les amis que j'ai connus pendant ces quatre années passées à Grenoble et qui ont fait de ce séjour une expérience inoubliable.

Table des matières

Introduction générale	1
1 La vision par ordinateur	5
1.1 Différentes philosophies de la vision	5
1.1.1 De la vision bas niveau	6
1.1.2 ... à l'interprétation et la reconnaissance	6
1.1.3 Approche traditionnelle (paradigme de Marr)	6
1.1.4 Vision Active	8
1.1.5 Perception active	8
1.1.6 Vision animée	9
1.1.7 Vision intentionnelle	10
1.1.8 Approche systémique	10
1.1.9 Quelques mots sur la vision humaine	11
1.2 Différents systèmes de vision	13
1.2.1 Le système de Nazif et Levine	14
1.2.2 Le système de Bellet	14
1.2.3 SIGMA	15
1.2.4 BORG	15
1.2.5 VAP	16
1.2.6 MESSIE-II	17
1.2.7 VISIONS	18
1.3 L'intelligence artificielle distribuée	18
1.3.1 La modélisation multi-agents	18
1.3.2 Les approches à base de comportements	21
1.4 Notre approche	23
1.4.1 Concepts de base	23
1.4.2 Vue générale du système	24
1.4.3 Modèle de l'agent	25
1.4.4 Les comportements de l'agent	26

2	La représentation des informations	29
2.1	La représentation des informations en vision	29
2.1.1	Systèmes de vision bas niveau	30
2.1.2	Systèmes de vision haut niveau	30
2.2	L'environnement et les informations dans le système	32
2.3	L'environnement du système	33
2.3.1	Représentation de l'environnement	33
2.3.1.1	Les images sources	34
2.3.1.2	Le gradient	35
2.3.1.3	La variance	36
2.3.1.4	Le mouvement	36
2.3.1.5	Les images résultats de segmentation	36
2.3.2	Apport de la représentation de l'environnement	37
2.3.2.1	Liberté de la représentation	37
2.3.2.2	L'environnement comme un territoire	37
2.4	Utilisation de l'information mouvement	38
2.4.1	Calcul de l'information mouvement	39
2.4.2	Le mouvement par différence d'images	39
2.4.3	Notre méthode de calcul de la différence d'images	41
2.5	Les informations construites sur les entités	42
2.5.1	Représentation des informations	43
2.5.1.1	Informations statistiques sur l'image	46
2.5.1.2	Informations sur les primitives région ou contour	47
2.5.1.3	Informations sur les groupes de primitives (groupement perceptuel)	48
2.5.2	Apport de la représentation des informations	50
2.5.2.1	Niveaux de représentation logique	50
2.5.2.2	Hiérarchisation logique vs organisation planaire	52
3	La gestion des informations	53
3.1	La gestion des informations dans les systèmes de vision	53
3.1.1	Systèmes centrés tâches	54
3.1.2	Systèmes centrés concepts	56
3.1.3	Systèmes hybrides	57
3.2	Stratégie orientée gestion des informations	58
3.3	Les comportements de gestion des informations	59
3.4	L'évaluation des informations	59
3.4.1	Évaluation globale	60
3.4.2	Les critères d'évaluation	60

3.4.3	Critères d'évaluation de pixels	61
3.4.4	Critères d'évaluation de primitives	63
3.4.5	Évaluation floue	63
3.4.6	Observations	64
3.5	Le comportement de perception	64
3.5.1	Principe de l'algorithme	65
3.5.2	Les critères d'évaluation pour les régions	67
3.5.2.1	Critère de seuillage	68
3.5.2.2	Critère d'homogénéité	70
3.5.2.3	Critère de compacité	70
3.5.2.4	Critère d'inclusion	71
3.5.3	Les critères d'évaluation pour les contours	71
3.5.4	Configuration de l'évaluation	73
3.5.4.1	Adaptation selon les informations de groupe	73
3.5.4.2	Évaluation tenant compte de la segmentation précédente	73
3.6	Le comportement d'interaction	75
3.6.1	Fusion	76
3.6.2	Continuité des contours	78
3.6.3	Prémises de négociation	79
3.6.4	La gestion des relations entre agents	81
3.7	Le comportement de différenciation	82
3.7.1	Principe de l'algorithme	84
3.7.2	Les critères d'évaluation	84
3.7.2.1	Critère statistique	85
3.7.2.2	Critère d'intersection	88
3.7.2.3	Critère d'adjacence	89
3.7.3	Caractère dynamique de l'évaluation	90
3.8	L'adaptation comme interaction	91
4	Le contrôle du système	93
4.1	Le contrôle dans les systèmes de vision	93
4.1.1	Contrôle centralisé	94
4.1.2	Contrôle décentralisé	97
4.2	Le contrôle dans les systèmes multi-agents	100
4.3	La liberté de contrôle	103
4.4	Le comportement de reproduction	104
4.4.1	Reproduction dans la même image	105
4.4.2	Une stratégie d'exploration	105
4.4.3	Principe de l'algorithme	106

4.4.4	Observations et améliorations sur l'algorithme	108
4.4.5	Les critères d'évaluation des pixels	109
4.4.5.1	Critère d'extrémum	110
4.4.5.2	Critère d'opposition	112
4.4.5.3	Critère de distance	113
4.4.5.4	Critère d'inclusion	113
4.4.6	Reproduction vers l'image suivante	114
4.4.7	Initialisation des premiers agents	115
4.5	La gestion des comportements	116
4.5.1	Fonctionnement général	116
4.5.2	Événements	118
4.6	Le séquenceur	119
4.7	Spécialisation des agents	120
4.7.1	Utilité de la spécialisation	120
4.7.2	Connaissances fournies par la spécialisation	121
4.7.3	Définition du fichier de configuration	121
4.8	Positionnement du système	124
5	Application à la reconnaissance de cellules vivantes	127
5.1	L'application cytologique	127
5.1.1	Objectif de l'étude	127
5.1.2	L'acquisition des images	128
5.1.3	Approches existantes	130
5.1.4	Difficultés de l'application	132
5.2	Le système réalisé	132
5.2.1	Développement du système	132
5.2.2	Objectifs et spécialisations des agents	133
5.2.3	L'interface du système	136
5.2.4	L'initialisation des premiers germes	137
5.2.5	Prétraitement des images	137
5.2.6	Exemple d'application (cicatrisation <i>in vitro</i>)	138
5.3	Résultats	140
5.3.1	Adaptation et décentralisation de la stratégie de reproduction	141
5.3.2	Adaptation du comportement de perception	143
5.3.2.1	Utilisation d'informations dynamiques pour la perception	144
5.3.2.2	Adaptation et évolution des paramètres	145
5.3.2.3	Coopération pour la reconnaissance	147
5.3.2.4	Combinaison multi-critères et paramétrage	151
5.3.3	Coopération globale pour le comportement de différenciation	153

5.3.4	Gestion réactive des comportements	156
5.3.5	Expérimentation sur un autre type cellulaire	158
5.4	Évaluation	158
5.4.1	Cartes de référence	159
5.4.2	Évaluation des segmentations	162
5.5	Conclusion	165
6	Perspectives et conclusion	167
6.1	L'apprentissage	167
6.1.1	L'apprentissage en vision	168
6.1.2	L'approche réalisée	169
6.1.2.1	Description de l'approche	170
6.1.2.2	Phase d'apprentissage	171
6.1.2.3	Phase de prédiction	173
6.1.2.4	Intégration au système existant	175
6.1.2.5	Perspectives	176
6.2	Modélisation des connaissances	177
6.3	L'étude des cellules	178
6.4	Conclusion	180
	Annexe A	181
	Bibliographie	189

Table des figures

0.1	Composantes identifiées dans les images cellulaires	3
1.2	Modèle conceptuel du système	25
1.3	Modèle général de l'agent	25
1.4	Modèle de fonctionnement des comportements	27
2.5	Structure de l'environnement	34
2.6	Images et caractéristiques de base composant l'environnement	35
2.7	Classification des différentes images d'information de l'environnement . . .	37
2.8	Modèle de l'environnement des agents	38
2.9	Principe général de la différence d'images	40
2.10	Calcul des images de différence (amplitude et signe)	41
2.11	Illustration de notre méthode de différence d'images sur des images de syn- thèse	42
2.12	Représentation simplifiée des différentes structures d'informations existantes dans le système	44
2.13	Exemples tirés de l'application pour l'utilisation des structures de représen- tation de l'information	45
2.14	Organisation des structures d'informations	50
2.15	Hiérarchie des différentes informations du système	51
3.16	Modèle général de l'agent	58
3.17	Principes de la croissance de région et du suivi de contour	65
3.18	Boucle de contrôle du comportement de perception	66
3.19	Croissance de région à partir de germes initiaux	67
3.20	Fonctions d'évaluation du critère de seuillage du comportement de perception	68
3.21	Fonction d'évaluation du critère d'homogénéité du comportement de per- ception	70
3.22	Communications entre deux agents en interaction (fusion) modélisées sous forme de réseau de Pétri	75
3.23	Exploration à partir de l'extrémité d'un contour	79

3.24	Principe du comportement de différenciation	83
3.25	Exemple d'évaluation pour la différenciation d'une région	85
3.26	Principe général de la méthode des k plus proches voisins	86
3.27	Illustration du principe des k plus proches voisins pour le critère d'évaluation statistique du comportement de différenciation	88
4.28	Graphe de lancement des agents (comportement de reproduction)	105
4.29	Positionnement des agents sur l'image	106
4.30	Illustration du principe d'exploration pour le comportement de reproduction	107
4.31	Exploration de l'image pour la reproduction	108
4.32	Boucle de contrôle du gestionnaire interne de l'agent	116
4.33	Exemple de table événements-réponses du gestionnaire de l'agent	117
4.34	Gestion du comportement de reproduction par rapport au comportement de perception	117
4.35	Gestion du comportement de fusion (interaction) par rapport au comportement de perception	118
4.36	Boucle de contrôle du séquenceur	120
4.37	Exemple de configuration pour l'agent pseudopode	122
4.38	Résumé du nombre de critères pour chaque comportement	124
5.39	Séquence d'images de fibroblastes murins L929	129
5.40	Composantes identifiées dans les images cellulaires	134
5.41	Stratégie d'exploration de l'image implantée par le comportement de reproduction	135
5.42	Interface du système	136
5.43	Le processus de cicatrisation <i>in vitro</i>	139
5.44	Reproduction et placement des agents pseudopodes	142
5.45	Gestion et adaptation des informations pour la segmentation des pseudopodes	144
5.46	Fonctions d'évaluation du critère de seuillage du comportement de perception	145
5.47	Évolution du seuil des agents noyaux en fonction du temps	146
5.48	Segmentation et adaptation des agents halos blancs	148
5.49	Segmentation du cytoplasme en fonction des poids des critères de perception	152
5.50	Segmentation et différenciation des agents mouvement	154
5.51	Représentation des statistiques des composantes et différenciation des agents mouvement	155
5.52	Gestion des comportements des agents en fonction du temps	157
5.53	Segmentation d'une séquence d'images de cellules HeLa	159
5.54	Construction des cartes de références par différents opérateurs	160
5.55	Comparaison, par matrice de confusion, des cartes de références établies par différents opérateurs	161

5.56	Résultats du système sur deux images (séquence 1)	163
5.57	Résultats du système sur deux images (séquence 2)	163
5.58	Matrices de confusion pour l'évaluation quantitative des résultats	164
6.59	Représentation des individus de l'apprentissage pour chaque type d'agent .	173
6.60	Espaces de travail pour la prédiction de la configuration d'un nouvel agent lancé	174
6.61	Modèle du système en intégrant les modules d'apprentissage	175

Introduction générale

Un système de vision a pour but de reconnaître une scène qui lui est présentée et d'en décrire les différents éléments. Il constitue, de façon générale, un système de traitement de l'information avec en entrée une image comme information et en sortie, une description symbolique de cette image. Entre les deux extrémités se situe tout un ensemble de traitements qui doivent être effectués afin de réaliser la transformation des informations demandées. Il est également possible de définir un système de vision sur le plan de la connaissance, que l'on peut définir sommairement comme de l'information à caractère utile. Il existe deux points de vue complémentaires sur les connaissances en vision :

- les connaissances comme but en soi : traiter l'image pour extraire de l'information utile à la résolution du problème posé (*traiter pour connaître*) ;
- les connaissances comme instrument : traiter l'image pour mieux aborder son analyse (*connaître pour traiter*).

Cette dualité des points de vue illustre un problème fondamental de la vision par ordinateur. En effet, tout traitement de vision, en segmentation ou en reconnaissance, a besoin de certaines informations pour pouvoir réaliser la tâche demandée. Cependant, ces informations ne sont disponibles bien souvent qu'en fin de ces mêmes traitements. Le système a donc besoin des informations qu'il doit lui-même produire. Ce constat démontre tout le problème de l'utilisation et de la gestion des informations en vision.

Plusieurs approches différentes ont été tentées pour résoudre ce problème au cours des quarante dernières années. Selon les travaux, le problème a été analysé sous l'angle de la représentation des informations en différents niveaux successifs d'abstraction, de la planification, ou plus généralement, du contrôle des méthodes de traitement. Toutes ces approches ont permis certaines avancées dans la recherche d'une solution à la conception d'un système de vision, mais la solution complète n'existe toujours pas, et il reste encore beaucoup d'inconnues dans ce problème.

En fait, beaucoup de travaux ont pris comme postulat de base qu'il fallait trouver la meilleure interconnexion entre différents niveaux de traitements et/ou de représentation. Un système de vision est souvent présenté comme un ensemble constitué de plusieurs couches, avec des liens et des flux d'informations différents en fonction des présentations.

Plutôt que de rechercher quel est le bon nombre de niveaux nécessaires, ou quelles doivent être les interconnexions entre ces niveaux, nous allons effectuer un repliement de ces niveaux en un seul et laisser interagir les différents modules présents entre eux. Nous évitons ainsi de poser le problème en termes de boucle de contrôle ou de cycle de traitement. Le système est vu comme un ensemble de modules, que nous nommons agents, qui forment un tout et sont en constante interaction mutuelle et qui évoluent de manière dynamique.

Nous voulons ainsi briser la chaîne séquentielle des traitements pour faire évoluer ces agents en parallèle. Le problème ne se posera plus en terme de contrôle du système, mais en terme d'informations disponibles à chaque instant. Ces informations évoluent en parallèle sous l'effet d'actions multiples, et la construction de la solution se fait de manière incrémentale avec l'ajout, petit à petit, de toutes les briques apportées par chacun des agents.

Dans cet esprit, nous avons développé une approche distribuée et dynamique de la vision, où l'accent est mis sur un couplage fort entre les différentes étapes de la vision et sur une circulation d'informations permettant l'adaptation et l'accumulation incrémentale de connaissances.

Ce système a été développé dans le cadre d'une application de reconnaissance des cellules vivantes. Il est évident que cette application a influencé le développement de notre système. Nous nous opposons ainsi à la démarche "généraliste" prônée par Aloimonos en ces termes :

Given the lack of success in developing systems which realize either of these goals [recognition and navigation] in a robust manner, it would appear reasonable to consider simpler problems; many researchers have followed this avenue by working on a very specific problem of an industrial application. However, these approaches do not enhance our understanding of vision in general.
[Aloimonos 90, p. 348]

Nous avons tenté de généraliser les concepts issus de cette application pour qu'ils soient applicables à d'autres domaines de la vision. Notre approche a donc été essentiellement ascendante : après une spécialisation en profondeur du système pour résoudre le problème posé par l'application, nous avons généralisé les concepts importants du système en vue d'une ré-utilisation dans d'autres systèmes. Nous croyons à cette démarche, et à l'apport de l'étude d'un problème particulier dans l'avancement de la solution au problème général de la vision, tel qu'il sera démontré tout au long de ce document.

Dans la présentation de nos travaux, il sera donné de nombreux exemples tirés de l'application concernée. Il est donc utile d'en donner une brève description. Le système vise l'étude de séquences d'images cytologiques. Les cellules se déplacent dans le temps tout en se déformant. Cette dimension mouvement est intégrée au système, mais l'accent est cependant mis sur la segmentation et la reconnaissance des composantes dans une image

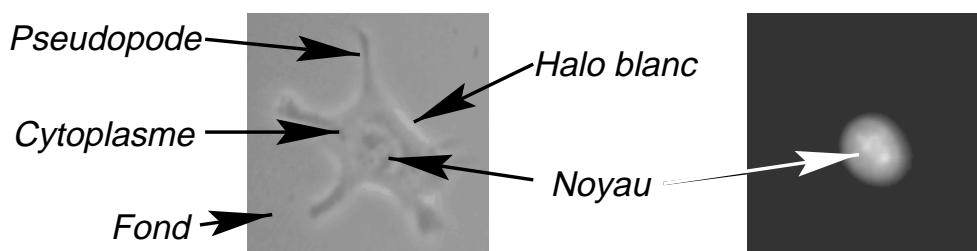


FIG. 0.1 – Composantes identifiées dans les images cellulaires. Les deux images sont mises en commun pour cette identification.

à la fois. Le mouvement constituera une information supplémentaire aidant à reconnaître les cellules. Les images des séquences sont divisées en cinq composantes (figure 0.1) :

- le *noyau*, qui est le coeur de la cellule ;
- le *cytoplasme*, qui représente toute la partie de la cellule qui entoure le noyau ;
- les *pseudopodes*, qui sont les prolongements cytoplasmiques de la cellule ;
- les *halos blancs*, qui entourent la cellule et définissent une frontière floue de celle-ci ;
- le *fond* de l'image, qui entoure les cellules.

Ce document s'articule en 6 chapitres qui nous permettront de présenter les différents aspects de nos travaux.

Le premier chapitre de ce document résumera certains travaux jugés importants dans le domaine de la vision par ordinateur. L'accent sera mis sur les principaux courants de pensées existant dans cette communauté. Ensuite, les travaux issus de l'intelligence artificielle distribuée seront présentés. Ces travaux nous permettront ensuite de définir précisément notre approche de la vision.

Le deuxième chapitre s'attardera sur les mécanismes de représentation que nous avons développés pour le système. L'accent sera mis sur la liberté de circulation des informations utilisées, avec la souplesse des structures employées.

Le troisième chapitre définira les modules de traitements de notre système. Ces modules seront présentés dans l'axe de la gestion des informations, en insistant sur les différentes sources d'informations mises en jeu dans ces traitements.

Le quatrième chapitre abordera les aspects de contrôle présents dans le système. Le contrôle est essentiellement réactif, réalisant un ordonnancement souple et adaptatif des différents modules.

Le cinquième chapitre illustrera les capacités du système à l'aide de nombreux résultats. Les problèmes de l'évaluation de ces résultats seront également discutés.

Enfin, le dernier chapitre présentera quelques perspectives de nos travaux, notamment pour l'apprentissage des connaissances dans le système et pour l'application cytologique étudiée.

Chapitre 1

La vision par ordinateur

La vision par ordinateur est un domaine actif de recherche depuis 40 ans environ. Cette recherche est passée par différentes étapes, évoluant autant avec les découvertes du domaine qu'avec la progression de la puissance des ordinateurs.

Plusieurs courants de pensée existent dans le domaine de la vision par ordinateur. Ils seront présentés en mettant en évidence les concepts qui nous semblent les plus intéressants. Il existe ainsi une pléiade de systèmes de vision différents, avec autant de motivations et modélisations différentes.

Ensuite, les techniques relevant de l'intelligence artificielle distribuée sont abordées. Il s'agit des approches à base de comportements venant des domaines de la robotique ou de la vision par ordinateur qui ont influencé la recherche en proposant des modèles purement réactifs pour résoudre des tâches demandant une forme d'intelligence. Puis les raisons et les buts justifiant une modélisation multi-agents sont présentés, avec les caractéristiques principales de ces systèmes.

Tous ces travaux nous permettent de présenter nos principes de conception, c'est-à-dire les concepts et les idées de notre système de vision, en mettant en avant les idées qui seront développées par la suite.

1.1 Différentes philosophies de la vision

Un bref survol du domaine de la vision par ordinateur est effectué dans cette section. Historiquement, les travaux peuvent être séparés en deux axes, selon qu'ils s'attaquent à la recherche de primitives primaires de l'image (bas niveau) ou qu'ils tentent d'interpréter le contenu de l'image sous une représentation particulière ou dans un but précis. Plusieurs courants de pensée existent sur la conception d'un système de vision. Bien que plusieurs de ces approches aient été développées dans le cadre de la vision tri-dimensionnelle, nous identifierons les concepts qui sont généraux à toute application de la vision, en précisant notre point de vue lorsque cela sera nécessaire.

1.1.1 De la vision bas niveau ...

Jusque dans les années 1980, les efforts se sont surtout concentrés dans la conception de méthodes bas niveau de segmentation et de description des images. C'est à cette époque que fut trouvée la majorité des opérateurs de base que nous utilisons toujours aujourd'hui, que nous pouvons classer en trois catégories : seuillage, détection de contours, extraction de régions [Fu 81, Pal 93]. Cette base, communément admise, se retrouve dans beaucoup de travaux actuels, avec des améliorations ou des combinaisons différentes pour tenter de résoudre les différents problèmes.

Cependant, la vision bas niveau pose toujours de réels problèmes qui n'ont toujours pas été résolus. Il est difficile de segmenter une image en primitives bas niveau sans connaître le domaine d'application de l'image et l'utilisation que l'on veut faire de cette segmentation. Ballard a exprimé d'une façon différente le problème inhérent à la segmentation :

The most severe chicken-and-egg problem of static computer vision is the "segmentation" problem. It is difficult to separate figure from ground without recognizing objects, and it is difficult to recognize objects if they are not separated from the background and other objects.
[Ballard 92, p. 11]

1.1.2 ... à l'interprétation et la reconnaissance

Les travaux proposant des systèmes, ou des approches, complets de vision ont véritablement commencé vers la fin des années 1970. Les travaux les plus remarquables de cette époque restent ceux de Marr, qui a proposé la première méthodologie pour la construction d'un module de vision artificielle. Ces travaux, connus sous le nom d'approche traditionnelle, sont présentés au prochain paragraphe. Par la suite, un certain nombre de chercheurs ont développé de nouvelles méthodes de pensée, regroupées sous l'axe de la vision active, s'opposant aux principes initialement développés par Marr. Les plus connus de ces travaux sont examinés dans les paragraphes qui suivent. Toutes ces approches tentant de se motiver en rapport avec les études sur la vision humaine, nous donneront quelques mots sur celle-ci.

Il est important de préciser que nous ne faisons pas de vision active dans notre travail. Mais cependant, il nous semble important d'identifier les courants de pensée et les concepts importants qui existent dans le domaine de la vision. Ils nous serviront par la suite pour développer notre approche.

1.1.3 Approche traditionnelle (paradigme de Marr)

Vers la fin des années 1970, Marr a proposé la première véritable méthodologie complète pour la conception d'un système de vision [Marr 82].

Plusieurs aspects peuvent être retenus de ces travaux. D'abord, Marr a défini trois

niveaux pour la compréhension et la conception d'un système de traitement de l'information. Un tel système, qu'il soit dans le domaine de la vision ou non, doit être vu sous trois angles :

- la *théorie calculatoire* : il s'agit de la théorie décrivant et structurant le domaine de travail, ses buts, ses principes gouvernants (quoi et pourquoi?) ;
- les *représentations et les algorithmes* : il s'agit de l'implémentation logicielle de la théorie, sa résolution et surtout, les structures de représentation de données utilisées au cours des transformations (comment (*logiciel*)?) ;
- l'*implémentation matérielle* : il s'agit du matériel utilisé pour l'implémentation physique des algorithmes, tant pour l'acquisition que pour la résolution (comment (*matériel*)?).

Pour Marr, ces trois aspects doivent être évoqués distinctement afin de comprendre tout système de traitement de l'information. Ils sont tous aussi importants, mais la notion de représentation demeure centrale chez Marr. En effet, il considère un système de vision comme une suite de transformations, principalement ascendantes (des données vers l'interprétation), d'une représentation en une autre. Il identifiait trois niveaux de représentation principaux :

- la *première ébauche* : l'extraction des informations de base et attributs de l'image bidimensionnelle (régions, contours, ...) ;
- l'*ébauche 2.5D* : le calcul des propriétés tri-dimensionnelles locales relatives à la scène (orientation, distance, ...) ;
- la *représentation 3D* : la description complète de la scène indépendante de l'observateur.

Dans la pratique des travaux qui ont succédé, ces niveaux se sont souvent traduits par trois étapes de traitement : segmentation, reconstruction et reconnaissance. Le but est de transformer le signal image en une représentation symbolique de la scène. L'interprétation finale qui sera faite ne sera valide que si les différentes transformations sont valides.

Le point principal du paradigme de Marr est dans la structure hiérarchique de traitement et de représentation. Il a posé un cadre de traitement rigoureux qui a inspiré nombre de travaux, autant pour la vision bi-dimensionnelle que tri-dimensionnelle.

Marr ne croyait pas à l'utilité des connaissances sur l'application ou le contexte. L'image (ou les images) et une théorie robuste seules permettent de résoudre le problème de la reconnaissance :

The general trends in the computer vision community was to believe that recognition was so difficult that it required every possible kind of information. [...] Clearly, we

do not use such knowledge in real life. [. . .] It seemed clear that the intuitions of the computer vision people were completely wrong and that even in difficult circumstances shapes could be determined by vision alone.
[Marr 82, pp. 35-36]

1.1.4 Vision Active

Le concept de base de la vision active ne peut pas être mieux expliqué que par Aloimonos, qui a initié cette nouvelle approche :

Marr's contributions set the foundations for vision as a scientific discipline. [. . .] However, Marr left out of his theory a very important issue: the fact that all existing visual systems, from insects and frogs to fish, snakes, birds, and humans, are active. Being active, they control the image acquisition process and thus introduce constraints that greatly facilitate the recovery of information about a 3D scene. "I move, therefore I see" is a fundamental true statement. And if I manage to make the human eye stationary, humans start losing perception!
[Aloimonos 94, p. 76]

En effet, Aloimonos considère les problèmes fondamentaux de la vision par ordinateur comme étant des problèmes mal posés dans le cadre de l'approche de Marr, qu'il qualifie de passive [Aloimonos 87]. En fait, tous les auteurs s'accordent pour dire que la vision est un problème difficile, voire insoluble. Selon les textes, on utilise les mots de *problèmes mal posés* [Aloimonos 87] ou *sous-contraints* [Jolion 94] par exemple. Nous reprenons ces termes dans notre étude en ajoutant les termes de *manque d'information* ou de *problème de gestion de l'information*.

L'idée de base d'Aloimonos est que tout observateur est plongé dans le milieu où il vit, et qu'il observe. L'interaction de l'observateur avec son milieu est donc une riche source d'information supplémentaire dans le processus de la vision. Cette interaction est posée ici sous la forme du mouvement des capteurs de l'image. Il s'agit d'utiliser des capteurs passifs (caméras) de manière active. Il s'intéresse aux mouvements des capteurs (caméras) pour contraindre la résolution de problèmes en ajoutant des contraintes, avec pour exemples les problèmes de *shape from X*, *structure from motion* ou du flot optique.

1.1.5 Perception active

Bajcsy a abordé sous un angle différent le problème de la vision, en ne considérant pas la vision comme un problème mal posé, mais plutôt comme un problème de contrôle [Bajcsy 88]. Alors que Aloimonos s'intéresse plus aux capteurs, Bajcsy vise le contrôle entre les différents modules d'un système de vision.

Selon elle, le système a besoin d'une spécification explicite de l'état initial et du but final. L'approche est très axée sur la modélisation, dont les deux points principaux sont :

- les *modèles locaux* : ce sont les modules de traitement et tous leurs paramètres (internes), comme par exemple un algorithme de croissance de région ou de détection

de contour ;

- les *modèles globaux* : ce sont les paramètres réglant les interactions entre les différents modules sous forme de boucles de rétroaction, la combinaison des différents résultats ou les prédictions de ces résultats.

Cette décomposition en modèles locaux et globaux se retrouvera dans notre approche. La modélisation des interactions entre les modules ou avec l'environnement tient une forte place dans la perception active.

Les stratégies d'action définies dans cette approche représentent toujours des compromis entre une action ascendante (guidée par les données vers l'interprétation) et une action descendante (guidée par la connaissance, sous forme de tâche ou de requête). Une différence est faite entre un mode descendant guidé par une tâche d'un autre guidé par une requête. Le système guidé par des tâches interagit avec son environnement en le modifiant et en étant actif dans celui-ci tandis que guidé par des requêtes, celui-ci est observateur de son environnement et ne cherche qu'à réunir les informations dont il a besoin. Selon son étude, une tâche reflète la notion d'*action*, tandis qu'une requête induit la notion d'*observateur*.

Cet aspect de recherche d'informations, qui nous concerne, est au coeur de l'approche définie par Bajcsy :

The fundamental ingredient to the Active Vision theory is the mechanism for decision making, in general, how to choose data information with respect to the perceptual goal. This involves both the method of estimation and the choices of general strategy for control.

[Bajcsy 88, p. 1003]

1.1.6 Vision animée

Une autre approche dérivée de la vision active est la vision animée, présentée par Ballard [Ballard 89]. Elle se fonde sur l'étude des mouvements intentionnels de l'oeil humain pendant une tâche visuelle :

Although some visual phenomena are preattentive, in that the result "pops out" immediately, our contention is that in practice this kind of popout is rare. In fact, most complex tasks have to be performed by the sequential application of simple primitive operations. Such decompositions greatly reduce the combinatorics of the computation.

[Ballard 92, p. 6]

Cette approche est très influencée par les approches comportementales, comme celle de Brooks [Brooks 86]. Elle exclut toute représentation explicite du monde (s'opposant à l'approche reconstructionniste de Marr [Marr 82]). Les modèles utilisés sont dépendants des actions à accomplir.

L'originalité de l'approche, dans notre contexte, est dans les stratégies d'exploration et de focalisation dans l'image. L'exploration de l'image dépend de la tâche à accomplir et du contenu de la scène. Seuls les endroits pertinents de l'image sont explorés et analysés.

1.1.7 Vision intentionnelle

L'idée suivante, tirée de la vision active, est la vision intentionnelle [Aloimonos 90] :

In this new framework we think of vision not as an end to itself, but as part of a bigger process, that does things, i.e. tasks. So, if we need to design a machine with visual sense that will accomplish certain tasks, we just need to be able to solve these particular tasks. And if they are complicated, we should decompose them into simpler tasks, and solve the simpler ones.

[Aloimonos 90, p. 349]

Cette approche met en avant le but pour lequel un système de vision doit être construit. La conception d'un système est donc guidée par son(s) but(s). Les représentations existantes (si elles sont utiles) seront orientées vers la tâche à accomplir. L'idée principale est qu'un système complet de vision est un regroupement de plusieurs sous-systèmes, où chacun de ses sous-systèmes répond à une tâche précise.

Cette approche, et son opposition avec celle de Marr [Marr 82] a fait l'objet d'une discussion qui en analyse les avantages et les inconvénients dans [CVGIP 94].

1.1.8 Approche systémique

Jolion a proposé une approche systémique de la conception d'un système de vision [Jolion 94]. Il part du constat que la vision est un problème sous-contraint. Il suggère de rajouter de nouvelles contraintes, non pas d'une source précise (qui mène parfois à une simplification exagérée du problème), mais de toutes les sources possibles. Cela peut être réalisé en étudiant l'architecture utilisée dans son ensemble et en considérant les interactions des composants internes de celle-ci entre eux. Il s'agit en fait d'une méthodologie de conception d'un système de vision.

Contrairement à d'autres approches précédemment exposées qui insistent sur le fait de construire différents modules pour répondre à différentes tâches, l'accent est mis ici sur une approche globale du système. Les différents modules existent, mais l'union de ces modules doit être considérée comme source supplémentaire d'information.

Sa description comporte 10 règles à suivre pour considérer un système de vision sous une approche systémique :

1. *Conserver la variété* : considérer les contraintes selon plusieurs points de vue en évitant de simplifier outrageusement en voulant généraliser tous les problèmes.
2. *Ne pas ouvrir la boucle de contrôle* : porter attention aux modifications locales qui peuvent avoir des conséquences globales sur le système.
3. *Rechercher les amplifications locales* : rechercher les composants locaux ayant le plus d'influence au niveau global.

4. *Conserver l'équilibre du système au moyen d'actions locales* : maintenir l'harmonie du système par des actions au niveau des modules d'action et non par le contrôle d'un quelconque superviseur / régulateur.
5. *Maintenir les contraintes* : ne pas chercher à trop simplifier le système, ce qui diminuerait ses capacités devant des problèmes réels.
6. *Varier pour mieux unifier* : c'est l'union de plusieurs modules différents qui peut produire de nouvelles informations utiles au système.
7. *Être adaptatif* : le système doit s'adapter aux différents contextes possibles.
8. *Rechercher un but plutôt qu'un comportement détaillé* : bien définir les buts recherchés et les contraintes possibles avant de concevoir les modules pour y parvenir.
9. *Étudier la complexité du réseau de communication* : observer la transformation des informations dans le système n'est pas suffisant ; il faut comprendre le flot de ces informations, leurs conflits possibles et les réactions du système face à ces conflits.
10. *Respecter les contraintes temporelles* : la dynamique du système étant très importante, il faut adapter le système, au moyen de contraintes supplémentaires, aux contraintes temporelles existantes.

Les points importants de ces règles que nous pouvons retenir sont la décentralisation (actions locales), l'importance du flot d'informations et les interactions et interdépendances entre les différents modules du système.

Les contraintes possibles qui peuvent être étudiées sont les *contraintes empiriques* (sur le monde observé), les *contraintes sur les comportements* (contexte et connaissance des modules), les *contraintes idéalistes* (limites de la théorie et de l'équipement utilisé) et les *contraintes systémiques* (reliées à l'utilisation des règles précédemment évoquées).

La force de cette approche tient dans son analyse de la complexité :

Comparing human and computer vision systems, Thorpe argued that the human visual system "beats" the computer because it makes an intensive use of massive parallelism. The systemic paradigm argues that *the power of a vision system is not a particular architecture but its complexity.*

[Jolion 94, p. 69]

1.1.9 Quelques mots sur la vision humaine

Toutes les approches, ou philosophies, de conception de la vision qui ont été présentées se réfèrent de près ou de loin à la vision humaine. En effet, la recherche en vision

peut être classée sous trois axes principaux, selon les communautés scientifiques en cause [Trivedi 89] :

- les *neurophysiologistes* qui étudient les mécanismes biologiques (neuronaux par exemple) de la vision (humaine ou animale) ;
- les *psychologues de la perception* qui cherchent à comprendre les considérations psychologiques qui gouvernent la perception ;
- les chercheurs en *vision par ordinateur* qui tentent de reproduire la vision de façon artificielle.

Ces trois communautés, aux buts différents, s'influencent dans leurs recherches, tant sur le point des idées que des méthodes de travail. La compréhension de la vision humaine donne de nouvelles pistes de recherche pour la vision artificielle, et l'étude des algorithmes et des représentations associées à l'acquisition, au calcul et à la compréhension des images permet de valider ou peut-être même de concevoir de nouvelles théories sur la vision humaine.

Cependant, il faut se rappeler les différences qui existent entre la vision humaine et la vision artificielle. Si un jour cette dernière devient très performante, ce sera sûrement par des techniques et des procédés qui lui sont propres. Il existe de grandes différences entre un cerveau humain et un ordinateur. Les méthodes développées doivent en tenir compte, et exploiter les spécificités de la machine. Les analogies humaines sont très utiles aux développements de nouvelles idées de recherche, mais il faut se rappeler que le but de cette recherche est de s'approcher des résultats de la vision humaine¹ et non pas de ses méthodes. Sandakly a déjà exprimé cette idée en reprenant une analogie très intéressante :

L'observation de la vision biologique peut être une grande source d'informations pour la vision artificielle. Néanmoins, la recherche en vision artificielle ne doit pas se limiter à cette observation et essayer de la recopier exactement dans ses modules de base. Pendant longtemps, l'homme a observé les oiseaux et a essayé de les imiter dans leurs vols. De nos jours, malgré la compréhension totale de vol des oiseaux, les avions n'en partagent que peu de principes dans leur façon de voler et en même temps ils vont plus haut et plus loin. Cela est dû aux difficultés techniques que nous n'arrivons pas à surmonter et à la présence d'alternatives plus simples et bien maîtrisées.
[Sandakly 95, p. 21]

Il nous semble que l'évolution est principalement due à deux facteurs : l'adaptation au milieu et les possibilités du matériel utilisé, c'est-à-dire le substrat biologique pour l'homme ou l'électronique pour l'ordinateur. Si le premier point peut être semblable (un des buts de la vision par ordinateur est de s'adapter aux conditions humaines de vision), le deuxième est complètement différent. L'électronique ne réagit pas de la même façon que le cerveau humain et doit donc trouver sa propre voie pour évoluer vers ses objectifs.

1. quoique cette idée soit discutable

1.2 Différents systèmes de vision

Plusieurs systèmes intégrés de vision ont vu le jour, particulièrement dans les vingt dernières années. Ces systèmes ont été conçus selon différentes philosophies, avec des outils différents et souvent pour des buts différents. Ils ont en commun cependant l'utilisation de méthodes de *raisonnement*, dites *intelligentes*, pour effectuer l'analyse d'une scène.

Ces systèmes peuvent être classés selon différents critères :

- Ils effectuent une analyse de *bas niveau* ou de *haut niveau* selon le contenu sémantique qu'ils apportent à la description de la scène.
- Ils sont *généralistes* ou *spécialisés* pour une application précise.
- Ils illustrent plusieurs domaines de recherches, dont les *tableaux noirs*, les *systèmes experts*, la *planification*, la *coopération* de méthodes, les *systèmes multi-agents*, ...²
- Ils sont *centralisés* ou *décentralisés* dans leur contrôle.
- Ils utilisent une approche *ascendante*, guidée par les données, ou *descendante*, guidée par les tâches ou les requêtes.

Certains systèmes, parmi les plus connus, vont servir à illustrer notre analyse de la vision. Ils représentent différents courants de pensée du domaine, et sans être exhaustif, regroupent les tendances observées dans la recherche en vision par ordinateur. Dans ce chapitre, seule une présentation générale de ces systèmes est donnée. Ensuite, dans les prochains chapitres, nous reviendrons sur ceux-ci pour expliquer les différents aspects sous lesquels nous avons abordé notre travail :

- la représentation des informations (chapitre 2) ;
- la gestion des informations (chapitre 3) ;
- le contrôle dans les systèmes de vision (chapitre 4).

Chaque système n'apporte pas nécessairement de contributions dans chaque aspect que nous traitons, et c'est pourquoi tous les systèmes ne seront pas cités et décrits pour chacun de ces aspects. Pour chaque aspect, seuls quelques systèmes serviront à illustrer notre analyse. Pour plus de renseignements sur les systèmes de vision, une étude très complète a été effectuée dans [Sandakly 95].

2. Nous mélangeons indifféremment ici différentes notions n'ayant rien en commun. Il s'agit d'illustrer, sans être exhaustif, différents axes qui existent et qui ont été utilisés, seuls ou en combinaison avec d'autres, dans la conception de systèmes de vision.

1.2.1 Le système de Nazif et Levine

Nazif et Levine ont développé à l'Université McGill un système expert pour la segmentation bas niveau des images [Nazif 84]. Ce système ne possède aucune connaissance sur les différentes applications et se contente de segmenter une image en régions et contours. La principale contribution de ce système est dans la gestion des informations qui est faite dans les règles de décision.

En effet, ces règles combinent intelligemment des raisonnements sur les différentes entités de l'image, régions et contours, afin d'améliorer petit à petit la segmentation. Il s'agit d'une véritable coopération d'opérateurs bas niveau, et ceci hors de toute sémantique liée au domaine d'application.

Le système est composé de différents modules :

- un module d'initialisation, calculant une première carte de gradient, ainsi qu'une image des régions initiales ;
- un module d'analyse des contours ;
- un module d'analyse des régions ;
- un module d'analyse des aires (portions d'images aux propriétés communes) ;
- un module d'analyse des zones d'intérêt ;
- un séquenceur, guidant les différentes bases de règles utilisées.

Deux zones mémoires distinctes sont globales à tous ces modules :

- une mémoire court terme, contenant les informations sur les différentes entités trouvées dans l'image ;
- une mémoire long terme, contenant les connaissances sur la segmentation et les différentes stratégies possibles.

Les applications présentées pour ce système sont essentiellement des scènes naturelles d'extérieur.

1.2.2 Le système de Bellet

Bellet a développé un système coopératif pour la segmentation de bas niveau des images [Bellet 98]. Ce système est multi-processus, puisqu'une image est segmentée par une multitude de processus indépendants, qui sont basés sur des modèles de croissance de région ou de suivi de contour. Chaque processus est initialisé à un endroit précis de l'image, avec un germe, et il fait croître sa primitive à partir de cet endroit. La croissance (région ou

contour) s'effectue par une évaluation multi-critères de pixels avoisinants et sélection des meilleurs candidats.

Ce système est basé sur une nouvelle forme de coopération entre différents processus de segmentation de natures différentes. Lorsqu'un processus manque d'informations pour mener à bien les décisions qu'il doit prendre, il crée de nouveaux processus, qualifiés de *fil*s pour segmenter de nouvelles primitives et ainsi, récolter plus d'informations sur l'environnement local. Un processus peut également s'appuyer sur les informations récoltées par ses voisins pour mener à bien sa tâche. C'est sur ce modèle de coopération que s'effectue toute la segmentation de l'image.

Le système est généraliste et ne bénéficie d'aucune connaissance sur les applications et les images qu'il segmente. Il a été utilisé pour différents types d'images.

1.2.3 SIGMA

Le système SIGMA a été développé par Matsuyama à l'Université du Maryland vers la fin des années 1980 [Matsuyama 90]. Ce système est basé sur une hiérarchie de trois systèmes experts :

- L'*expert sur le bas niveau* qui détient à la connaissance nécessaire pour l'extraction de primitives.
- L'*expert de sélection des modèles* qui détient à la connaissance sur la transformation des données image en objets haut niveau de la scène.
- L'*expert sur le raisonnement géométrique* qui détient à la connaissance relative au raisonnement spatial entre les objets au niveau de la scène.

Ce système est une des premières réflexions complètes sur un système complet de vision, alliant tous les aspects importants de modélisation de la connaissance, opératoires et descriptives, et ce à tous les niveaux.

1.2.4 BORG

Clouard et al. ont proposé le système BORG [Clouard 94]. Ce système, conçu sur un modèle de planification basée sur un tableau noir, décrit une application de traitement d'images selon cinq niveaux d'abstraction : la *requête* (la définition générale du problème), les *objectifs* (les tâches à accomplir pour résoudre le problème), les *fonctionnalités* (fonctions de traitement d'images à utiliser), les *procédures* (description des opérateurs) et les *opérateurs* (implantation spécifique des opérateurs).

Le but de cette hiérarchisation est de construire dynamiquement le graphe d'opérateurs nécessaire à l'accomplissement d'une requête à partir d'un ensemble de connaissances portant sur plusieurs points : sur le domaine de l'image, sur l'expertise de l'image, sur la

librairie d'opérateurs et sur le contrôle. Cette génération dynamique de la solution peut se faire selon un mode de coopération ou de compétition entre opérateurs, avec la possibilité d'intégrer l'utilisateur pour parfaire la solution.

Sept principes ont été définis et utilisés pour leur système [Clouard 95b]. Ces principes sont complémentaires à l'approche systémique de Jolion précédemment décrite (paragraphe 1.1.8) :

1. Choisir un modèle de résolution basé sur les actions.
2. Utiliser un modèle de raisonnement basé sur une planification hiérarchique des actions.
3. Contrôler l'exécution des opérateurs par un calcul dynamique de leurs paramètres et par des mécanismes de régions d'intérêt sur des sous-ensembles de l'image.
4. Choisir une librairie d'opérateurs composée d'opérateurs *atomiques*.
5. Adopter une évaluation des résultats hiérarchique et délocalisée.
6. Décrire la segmentation comme un ensemble de tâches à accomplir, avec les contraintes nécessaires et une description symbolique et numérique du contexte de l'application.
7. Intégrer l'utilisateur dans la boucle de résolution pour coopérer ou collaborer avec le système en vue de trouver une solution acceptable.

1.2.5 VAP

Le système VAP (Vision As Process) a été développé dans le cadre d'un projet européen dans lequel plusieurs partenaires se sont associés pour construire un système intégré de vision [Crowley 93]. Ses objectifs sont la construction d'un système de vision active fonctionnant en continu et en temps réel. Les partenaires impliqués dans ce projet proviennent de différents laboratoires situés en Grande-Bretagne (Université de Surrey), au Danemark (Université de Aalborg), en Suède (Royal Institute of Technology et Université de Linköping) et en France (Institut National Polytechnique de Grenoble). La répartition distribuée sur tout un continent des différents partenaires du projet a largement influencé la conception du système.

Le système est décomposé en six modules s'inscrivant chacun sur un niveau de représentation :

- l'unité de contrôle de la caméra ;
- la description 2D (un module pour l'image gauche et un autre pour l'image droite) ;
- la description tridimensionnelle ;

- l'interprétation symbolique ;
- le superviseur.

Le dernier module est dédié à la définition du contrôle du système. L'abstraction liée aux niveaux de représentation induit une hiérarchie de contrôle dans le système.

Dans VAP, la vision est perçue comme un processus devant s'intégrer dans une chaîne de traitement beaucoup plus grande et non comme une fin en soi. Le système de vision doit être capable de fournir des informations pertinentes pour une application donnée (dans le cadre de la robotique pour ce projet). L'architecture du système est une collection de modules indépendants communiquant par le biais de canaux prédéfinis. Chaque module est un processus UNIX indépendant lié aux autres par un *squelette* (nommé SAVA - Squelette d'Application pour la Vision Active) définissant l'interface commune pour tous les modules. Les leçons tirées de ce projet sont liées à l'intégration des différents modules au sein d'un même système.

1.2.6 MESSIE-II

Le système MESSIE a été développé par l'équipe PASTIS de l'INRIA [Garnesson 91, Sandakly 95]. Il a été conçu comme une architecture générique d'interprétation de scène fondée sur un tableau noir hiérarchique. La résolution s'effectue par des *spécialistes*, qui sont des bases de connaissances spécialisées pour l'application ou pour un niveau du traitement (bas niveau, niveau intermédiaire ou haut niveau).

L'étude de ce système est portée sur sa deuxième version, développé par Sandakly [Sandakly 95]. Du point de vue fonctionnel, l'architecture de MESSIE-II comporte :

- une modélisation générique des objets sémantiques (les objets de la scène) par quatre points de vue : géométrie, radiométrie, contexte spatial et fonctionnalité ;
- une stratégie générique d'interprétation de scène basée sur le principe du plus évident d'abord. Cette stratégie s'appuie sur la modélisation des objets sémantiques ;
- une modélisation d'un certain nombre d'objets nécessaires à une interprétation, comme la scène, les capteurs, les matériaux et les opérateurs de localisation ;
- une représentation des connaissances incertaines et imprécises.

L'accent est mis sur la représentation des connaissances et les stratégies de résolution. Le système est générique, et spécialisé pour différentes applications par la définition des spécialistes appropriés. Deux applications ont été présentées pour ce système :

- interprétation de scènes de satellite avec fusion multi-capteurs [Clément 93a] ;
- interprétation de scènes tridimensionnelles d'intérieur dans un contexte de robot mobile.

1.2.7 VISIONS

Le système VISIONS (Visual Integration by Semantic Interpretation of Natural Scenes) s'inscrit dans le cadre d'un grand projet de recherche qui a débuté en 1975 à l'Université du Massachussetts [Hanson 78] et qui a bénéficié de l'apport de nombreux travaux depuis ce temps. Il s'agit d'un système généraliste de vision dont le but est la réalisation d'un système de perception en temps réel pour un robot mobile. Pour l'instant, le système a eu comme application des scènes naturelles statiques d'extérieur.

Il s'agit d'un système de vision s'appuyant sur une forte modélisation des connaissances sur plusieurs niveaux de représentation, particulièrement pour le haut niveau du système avec la représentation par *schémas* [Draper 89]. Les auteurs définissent le but des schémas comme une volonté de construire un système généraliste de vision à partir de plusieurs sous-systèmes spécialisés pour leurs domaines.

En fait, chaque schéma repose sur un ensemble de stratégies spécialisées pour la reconnaissance d'une entité précise (un concept) et de ses composantes (routes, maison, arbre, etc.). Il contient les connaissances descriptives et opératoires nécessaires à la reconnaissance des objets de sa classe. Un schéma peut faire appel à des sous-schémas et plusieurs schémas interagissent dans l'interprétation d'une image. Les informations sont représentées dans des tableaux noirs, présents au niveau des schémas et au niveau global.

1.3 L'intelligence artificielle distribuée

L'approche que nous avons développée utilise les principes de l'intelligence artificielle distribuée. Une application de vision par ordinateur se définit bien selon cette approche, du fait de la nature physiquement et fonctionnellement distribuée du problème. Avec les systèmes de vision présentés précédemment, il est facile de concevoir la nature modulaire nécessaire pour analyser une image et en reconnaître les composantes.

Plutôt que de présenter un état de l'art sur les différents travaux existants dans ce domaine, il a été choisi de privilégier certaines approches, ou certains travaux, qui illustrent bien la ligne de pensée qui est la nôtre. Les approches réactives ont servi de base à notre étude, qu'elles proviennent ou non du domaine de la vision. Premièrement, les systèmes multi-agents sont décrits avec les notions importantes qui nous aideront par la suite. Ensuite, il est question d'une lignée de travaux sur les systèmes à base de comportements, issus de la robotique, qui ont beaucoup influencé la conception de systèmes distribués réactifs.

1.3.1 La modélisation multi-agents

Nous désirons renouveler l'approche traditionnelle séquentielle de la segmentation, fondée sur une succession d'étapes. Cela nous a conduit à privilégier une approche distribuée

de type multi-agents [Ferber 95]. Nous insisterons ici sur trois notions importantes de ces systèmes :

- la distribution ;
- l'interaction, comme la coopération ;
- l'adaptation.

Ferber explique, d'une manière générale, la nécessité d'une *distribution* de l'activité et de l'intelligence par les raisons suivantes [Ferber 95] :

- Les problèmes sont physiquement distribués.
- Les problèmes sont fonctionnellement très distribués et hétérogènes.
- Les réseaux imposent une vision distribuée.
- La complexité des problèmes impose une vision locale.
- Les systèmes doivent pouvoir s'adapter à des modifications de structure ou d'environnement.
- Le génie logiciel va dans le sens d'une conception en termes d'unités autonomes en interactions.

Cependant, en voulant distribuer un système, il n'est pas nécessairement question de réaliser un parallélisme entre les modules, mais bien d'exploiter la simultanéité d'exécution de ces modules : "Par opposition au parallélisme qui signifie un déroulement *physique* simultané d'actions, la concurrence est un déroulement *logique* simultané d'actions." [Sandakly 95].

Il existe de multiples définitions pour un agent ou pour un système multi-agents selon le point de vue privilégié [Ferber 95, Wooldridge 95, Boissier 93]. Dans notre contexte, nous retenons les éléments suivants, qui ne constituent pas une définition, mais les concepts importants pour notre exposé :

- un *agent* est une entité autonome en action et en décision ;
- une *société d'agents* regroupe les différents agents et l'environnement dans lequel ils existent ;
- des *interactions* se produisent entre tous les éléments du système (individus, société et environnement).

L'*interaction* est une notion importante dans les systèmes multi-agents, car un objectif dans la définition d'une société d'agents est de créer des échanges entre ces agents afin d'augmenter les capacités du groupe. L'interaction peut être définie comme la mise en relation dynamique de deux ou plusieurs agents par le biais d'un ensemble d'actions réciproques [Ferber 95]. Ces agents peuvent avoir des buts compatibles ou non, des relations différentes face aux ressources, des capacités différentes, ce qui leur donne autant de situations possibles pour interagir. Ces interactions peuvent être de nature positive ou négative selon les plans de différents agents [Von Martial 90].

L'interaction qui nous intéresse le plus est la *coopération*. Il s'agit d'une notion que l'on retrouve fréquemment, et qui prend des formes différentes selon les travaux existants et les opinions des différents auteurs. Il est très difficile donc de la définir clairement, si ce n'est en exprimant les idées d'aide, de collaboration, ou d'apport mutuel qu'il existe entre deux ou plusieurs agents. Elle fait également référence aux notions de dépendances, d'influences et de réciprocités entre les agents autonomes [Castelfranchi 90]. Enfin, elle peut être volontaire ou non de la part des agents, comme elle peut être explicite ou émergente du point de vue d'un observateur extérieur.

Diverses formes de coopération existent, comme par exemple la coopération par les buts [Werner 90], par la résolution de conflits [Galliers 90], par la coordination d'actions [Ferber 95], ou par influences mutuelles des agents [Fenet 98]. Il existe aussi une forme plus opportuniste de coopération, privilégiant l'émergence comme résultat des interactions [Brooks 91a], comme par exemple l'approche de Steels, qui définit une société de robots explorateurs avec des comportements de base où les robots finissent par s'auto-organiser dans leurs actions en fonction de l'influence des autres robots.

Cette notion de coopération n'est pas l'apanage des systèmes multi-agents, puisqu'on la retrouve également dans le domaine de la vision, où l'on retrouve, par exemples, des systèmes de coopération par la fusion de résultats [Charroux 96] ou par l'adaptation mutuelle des méthodes [Kara Falah 94, Kermad 97].

Nous avons privilégié une approche de la coopération basée sur les dépendances mutuelles entre les agents. Elle se retrouve sous deux formes. Premièrement, l'information nécessaire au travail d'un agent est calculée par un autre agent, ce qui induit une dépendance entre ces deux agents [Salotti 94]. Deuxièmement, un agent sera responsable de l'initialisation d'un autre agent en fonction de sa propre exploration et de ses propres résultats. Cette dernière forme de dépendance se situe donc à un niveau de contrôle [Bellet 98].

L'*adaptation* est également un thème important en intelligence artificielle. Il existe plusieurs façons de percevoir cette notion. Ferber la définit de deux façons différentes, soit comme une caractéristique individuelle des agents, et on parle alors d'*apprentissage*, soit comme un processus collectif mettant en jeu des mécanismes reproductifs, ce que l'on appelle *évolution* [Ferber 95]. Maes dit qu'un agent est adaptatif s'il est capable de s'améliorer dans le temps, c'est-à-dire s'il devient meilleur pour accomplir ses buts avec l'expérience

[Maes 94]. Spinu parle plus simplement de l'adaptation comme étant la sélection et le paramétrage automatique des méthodes utilisées [Spinu 97].

Nous utiliserons des idées de chacune de ces définitions. En fait, il sera question d'adaptation (chapitres 2, 3 et 4) et d'apprentissage (chapitre 6). Nous tenterons de distinguer ces deux notions. L'adaptation est vue dans notre contexte comme l'amélioration des capacités de l'agent au cours d'une exécution du système, tandis que l'apprentissage vise les mêmes buts, mais cette fois sur une plus longue période, soit plusieurs cycles d'exécution du système, où les agents peuvent retenir les effets des précédentes exécutions.

Hayes-Roth présente plusieurs aspects d'importance pour l'adaptation d'un agent [Hayes-Roth 95] :

- un agent doit adapter sa stratégie de perception aux exigences de l'information, qui évolue dynamiquement, et aux limitations de ressources ;
- un agent doit adapter ses mécanismes de contrôle aux contraintes qui évoluent lors de son exécution et aux incertitudes provenant de l'environnement ;
- un agent doit adapter ses tâches de raisonnement aux objectifs qui peuvent évoluer dynamiquement, ses objectifs étant locaux ou globaux ;
- un agent doit adapter ses méthodes de raisonnement aux informations disponibles aux différents instants ;
- un agent doit adapter ses stratégies de méta-contrôle aux configurations dynamiques des demandes, des opportunités et des ressources des différents comportements.

1.3.2 Les approches à base de comportements

Brooks définit deux approches différentes de l'intelligence artificielle [Brooks 90]. La première approche, nommée *intelligence artificielle classique*, est centrée sur la manipulation symbolique et la représentation explicite des connaissances. Il reproche à celle-ci son éloignement des réalités du monde physique. La deuxième approche, nommée *nouvelle intelligence artificielle*, est centrée sur les comportements, c'est-à-dire sur des modules individuels réactifs.

Brooks préconise la deuxième approche, pour laquelle l'intelligence n'est pas un concept que l'on peut implanter ou programmer, mais quelque chose qui émerge, vu de l'extérieur, du comportement global d'un agent, ou plus généralement d'une entité. C'est sur ce principe qu'est basée l'architecture qu'il a conçue, nommée *architecture de subsomption*, avec laquelle différents systèmes robotiques ont été développés. Le système est basé sur une série de comportements réactifs et indépendants. Chaque comportement peut se voir comme un automate augmenté à états finis. Il n'existe pas de mécanisme de représentation de l'information, que ce soit au niveau global, ou au niveau de chaque comportement. En fait,

chaque comportement utilise son environnement comme repère pour retrouver ces informations. De cette architecture, il tire la conclusion et l'hypothèse subséquente suivantes :

(Conclusion) When we examine very simple level of intelligence, we find that explicit representations and models of the world simply get in the way. It turns out to be better to use the world as its own model.

(Hypothesis) Representation is the wrong unit of abstraction in building the bulkiest parts of intelligent systems.

[Brooks 91b, p. 140]

Il caractérise son approche par quatre concepts de base³ [Brooks 91a] :

- *Situation* : les agents sont situés dans un monde, et doivent s'en inspirer plutôt que de s'en abstraire ;
- *Réalité physique* : les agents existent et vivent dans un environnement ; ils doivent interagir avec celui-ci⁴ ;
- *Intelligence* : celle-ci est déterminée par la dynamique des interactions des agents avec le monde ;
- *Émergence* : l'intelligence est dans les yeux de l'observateur, elle émerge du comportement global qui est perçu.

L'intelligence, selon cette approche, doit donc émerger des différentes interactions qui existent entre les comportements. Il faut cependant effectuer la sélection des comportements à exécuter pour chaque situation, avec les contraintes suivantes :

- le bon comportement doit être exécuté à chaque instant ;
- il faut laisser libre cours aux interactions entre comportements pour faire émerger une forme d'intelligence.

Maes a proposé une solution à ce problème, en restant dans l'optique de la dynamique entre comportements [Maes 89]. Elle part des hypothèses suivantes :

- une sélection des actions *acceptables* pour le système peut être obtenue en laissant les différents modules de compétences s'activer et s'inhiber entre eux ;
- aucun module *bureaucratique* n'est nécessaire, ni aucune forme globale de contrôle.

Pour résoudre le problème, elle propose donc un modèle où chaque comportement peut inhiber ou activer les autres comportements, c'est-à-dire ses prédécesseurs, ses successeurs et les comportements conflictuels avec lui-même. Il agit sur les potentiels d'activation de ces

3. Les termes originaux, employés par Brooks, sont *situatedness*, *embodiment*, *intelligence* et *emergence*.

4. Ici, Brooks défend son approche pour la robotique, où il préconise un test dans le monde réel pour un système, et non dans un quelconque monde simplifié, comme le *monde des blocs* qui est un exemple connu.

comportements. À chaque intervalle de temps, le comportement ayant le potentiel le plus élevé (et supérieur à un seuil) s'exécute. L'ensemble des comportements peut ainsi s'auto-réguler grâce à ce mécanisme. Il s'agit d'une certaine forme de planification (puisque'il faut fixer les valeurs d'activation/inhibition), mais non-explicite et dynamique.

1.4 Notre approche

Maintenant que les différents courants de pensée qui nous intéressent ont été présentés, autant dans le domaine de la vision que dans celui de l'intelligence artificielle, nous présentons notre approche et le système que nous avons conçu.

Après avoir énoncé les différents éléments de notre approche, le système que nous avons développé est décrit. Cette description est faite en plusieurs étapes. Premièrement, une vue générale du système est donnée. Ensuite, le modèle d'agent que nous avons réalisé est exposé. Et finalement, au dernier niveau, les différents comportements de cet agent sont présentés. Il s'agit ici de donner les grandes lignes de cette approche, puisqu'elle est amplement développée dans les prochains chapitres.

1.4.1 Concepts de base

L'approche que nous avons développée est issue de l'intelligence artificielle distribuée. Nous avons conçu un système multi-agents pour notre application de vision. Ce système est largement inspiré des travaux de Bellet [Bellet 98] et de Salotti [Salotti 94].

Les différents axes de notre approche pour la conception de notre système de vision, répartis en trois catégories, sont les suivants :

- *représentation des informations :*
 - L'accès, la circulation et l'utilisation des différentes sources d'information sont le plus libre et le plus ouvert possible.
 - Les connaissances accumulées par les agents sont partagées par tous et chacun peut s'en servir pour ses propres traitements.
 - Il existe une représentation globale des informations, mais cette représentation a un but utilitaire et doit s'ancrer dans le monde plutôt que s'en abstraire (par rapport à Brooks [Brooks 91b] (paragraphe 1.3.2).
- *gestion des informations :*
 - Les agents construisent incrémentalement les primitives afin de pouvoir adapter leur traitement au fur et à mesure.
 - Cette construction incrémentale permet de construire des connaissances impossibles à acquérir autrement.

- L'action du système est locale, s'adaptant et se spécialisant en fonction des différents contextes présents.
- *contrôle du système* :
 - La stratégie d'exploration de l'image est adaptative aux différents contextes, en plus d'être répartie sur tous les agents de l'image.
 - Le contrôle est décentralisé et adaptatif, évoluant de façon réactive au contexte.
 - L'entrelacement des actions des agents permet de combiner les forces de chacune de ces actions, et ce, à tout moment.

1.4.2 Vue générale du système

La figure 1.2 décrit les différents composants conceptuels du système. L'agent, qui est au coeur de ce modèle, accède à différentes sources d'information et de connaissance. Les principaux éléments de ce modèle sont :

- L'*agent* est la partie active de ce modèle, puisque c'est lui qui traite les images, produit les informations comme résultats, en utilisant les connaissances définies par la configuration.
- L'*environnement* contient les images à analyser, les caractéristiques de ces images et les images résultats de la segmentation. L'environnement est la source d'information initiale et il s'enrichit des différentes segmentations effectuées par les agents.
- Les *informations* sont définies comme toutes les informations construites sur les différentes entités détectées dans les images. Les informations descriptives récoltées par les agents sont sources de connaissances dynamiques que les agents utiliseront par la suite pour effectuer leurs propres traitements.
- La *configuration* contient la connaissance opératoire permettant aux agents de se spécialiser pour les traitements qu'ils ont à effectuer. Cette connaissance opératoire provient d'un fichier de configuration externe au système.

Dans ce modèle, la connaissance opératoire, établie a priori, est séparée de la connaissance descriptive qui est construite au fur et à mesure de l'évolution du système. Cette distinction nette entre ces deux types de connaissances est venue de l'application que nous étudions. En effet, les objets que nous étudions sont très variables dans leur forme et ils se déforment selon des règles méconnues. Il est donc apparu naturel de donner au système les moyens nécessaires pour retrouver ces objets sans pour autant les décrire explicitement, cette description étant jugée trop difficile à fournir.

À partir de configurations initiales (connaissance opératoire), les agents vont donc explorer les images et reconnaître les objets qui s'y trouvent. L'information concernant ces

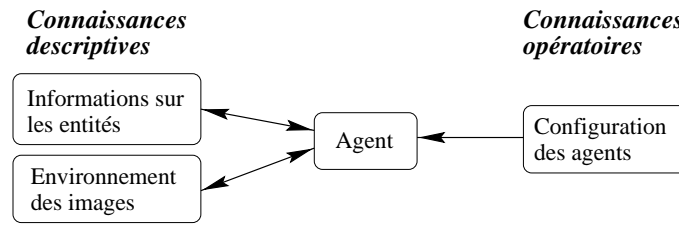


FIG. 1.2 – Modèle conceptuel du système. L'agent possède différentes sources d'informations et de connaissances à sa disposition.

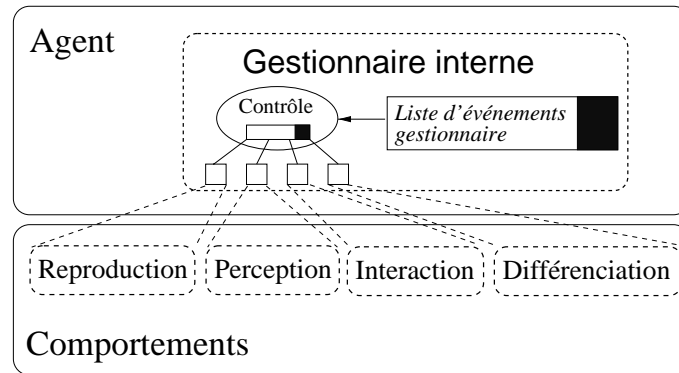


FIG. 1.3 – Modèle général de l'agent. L'agent consiste en une liste d'exécution de différents comportements. Le choix de l'exécution des comportements est soumis aux retours (ou événements) de ces mêmes comportements.

objets (connaissance descriptive) est accumulée par ces agents qui sont ainsi amenés à construire des bases d'exemples leur permettant par la suite de s'adapter en fonction du contexte réel dans lequel ils sont situés. Sans être un apprentissage, il s'agit plutôt d'une spécialisation a priori adaptée par la suite grâce aux informations accumulées.

Ce système est conçu dans un modèle de *co-évolution* entre les agents et l'environnement (au sens large, c'est-à-dire incluant les structures d'environnement et d'information). En effet, les deux structures évoluent en parallèle. D'une part, les agents s'installent dans cet environnement et l'explorent, et d'autre part, les informations s'accumulent au fur et à mesure que la segmentation progresse. C'est le principe de construction incrémentale qui permet cette double évolution des structures. Ainsi, les agents progressent, mais ont besoin de l'environnement pour ce faire, tandis que l'environnement progresse aussi, mais avec le travail des agents. Il y a une dépendance mutuelle entre ces deux structures.

1.4.3 Modèle de l'agent

La figure 1.3 montre l'architecture globale de l'agent. Ce modèle est purement réactif, tous les constituants de l'agent étant conçus pour réagir à des situations selon des modes définis. Ces actions peuvent cependant être modulées et adaptées en fonction de

l'environnement et du contexte. Deux niveaux sont définis dans l'agent :

- Les *comportements* représentent les différentes actions que peut accomplir un agent à tout instant. Ce sont les capacités intrinsèques à l'agent qui définissent son rôle dans le système. Ils sont plus détaillés au prochain paragraphe.
- Le *gestionnaire interne* est responsable, à tout moment, de choisir le comportement le plus apte à s'exécuter en fonction des contraintes extrinsèques et intrinsèques à l'agent. Cette sélection se fait par des règles de priorité gérées en fonction de retours d'événements, par les comportements ou par les autres agents. Cette gestion, purement réactive, permet en tout temps à l'agent de s'adapter à la situation courante et aux modifications de contexte. Cette gestion des comportements de l'agent est expliquée plus en détails au paragraphe 4.5.

Il existe un modèle générique d'agent, qui sera spécialisé par la suite pour accomplir une tâche précise (paragraphe 4.7). Cet agent peut traiter une primitive région ou une primitive contour. Cette primitive sera d'un type spécialisé pour l'application. Cette spécialisation permet d'adapter les comportements de l'agent aux différents types d'informations qu'il doit rechercher. Elle s'effectue par un fichier de configuration externe.

Chaque agent est lié à une primitive d'une image. Il a la tâche de construire cette primitive et de récolter toutes les informations pertinentes à son sujet. Une primitive peut être tout objet ou partie d'objet de l'image, car plusieurs agents peuvent travailler sur la même composante de l'image, qu'ils retrouveront en fusionnant leurs propres primitives.

Un agent possède une vie limitée, avec un début et une fin, comme chacun de ses comportements. Cette vie est rythmée par l'évolution de la vie (création et destruction) de ses comportements. Il est créé à un certain instant par une requête d'un autre agent, et il s'exécute jusqu'à ce qu'il ne possède plus de comportements.

1.4.4 Les comportements de l'agent

Un comportement est une action que peut mettre en oeuvre l'agent, dans le but d'explorer l'environnement ou de modifier celui-ci. Chacun de ces comportements correspond à une étape nécessaire du processus de segmentation. L'entrelacement et la parallélisation de ces comportements permettent d'adapter la stratégie d'exploration et d'analyse de l'image au fur et à mesure des résultats fournis par chacun d'eux. Cette stratégie reflète la notion de gestion des informations, où l'information est utilisée seulement au moment approprié [Salotti 94].

Il est possible d'instancier plusieurs fois le même comportement, selon les situations rencontrées. Par exemple, un agent peut interagir avec plusieurs autres agents à la fois et possèdera autant d'instances du comportement d'interaction qu'il existe d'interactions en cours.

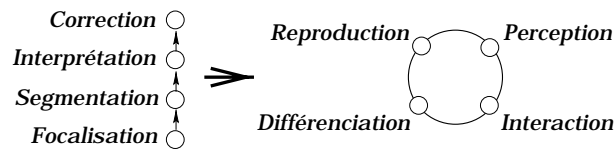


FIG. 1.4 – Le modèle de l'agent reprend les différentes étapes séquentielles traditionnelles de la vision et les recompose dans un modèle distribué et parallèle. Les comportements de l'agent permettent de guider avant tout la recherche de l'information et d'adapter le traitements aux informations disponibles.

Quatre comportements existent, que nous divisons en deux catégories :

- trois comportements de gestion de l'information :
 - *perception* (construction de primitives) ;
 - *interaction* (correction de primitives).
 - *différenciation* (interprétation ou migration d'information)⁵ ;
- un comportement de contrôle, de gestion de la stratégie :
 - *reproduction* (focalisation des agents) ;

Les premiers comportements, de gestion de l'information, sont expliqués au chapitre 3. Ils concernent la construction et la manipulation des différentes structures d'information et de l'environnement du système. Le dernier comportement, qui reflète la stratégie d'exploration de l'image (contrôle) est détaillé au chapitre 4. Il permet au système, et aux agents le composant, d'effectuer un contrôle sur l'allocation des ressources dans l'environnement.

Traditionnellement, la vision consiste en une série d'étapes successives, que nous pouvons représenter comme : focalisation, segmentation, interprétation, correction. Notre représentation, par comportements, vise à remettre en cause la linéarité et l'indépendance de ces étapes de la vision (figure 1.4). C'est l'interdépendance entre les différents comportements, ainsi que leur entrelacement, qui permet d'assurer une cohérence dans le traitements des images. Chaque comportement est concurrent aux autres et interruptible [Guessoum 96a], et son exécution est soumise à un temps limité, suivi d'une réévaluation de la situation.

Ce modèle, parallèle et dynamique, vient du principe de co-évolution évoqué au paragraphe 1.4.2. L'agent peut ainsi alterner entre différentes phases d'exploration de l'image, donc de développement de la société d'agents, et des phases d'analyse d'informations, qui sont le développement de l'environnement et des différentes structures d'informations. Nous pensons que ce schéma peut aider à briser la boucle sans fin, entre le besoin de

5. Le terme de *migration d'informations* peut paraître étrange. Cependant, il sera mieux expliqué au paragraphe 3.7. Il s'agit en fait d'une interprétation de primitives (passage d'un niveau de représentation à un autre).

connaître l'image pour la traiter et traiter l'image pour la connaître (problème de la poule et l'oeuf évoqué par Ballard et présenté au paragraphe 1.1.1).

Chapitre 2

La représentation des informations

Ce chapitre approfondit sur les différents mécanismes de représentation de l'information existants dans les systèmes de vision. Nous décrirons ensuite les représentations présentes dans notre système à partir des différents concepts qui ont été énoncés au premier chapitre.

Tout au long de ce chapitre, les différentes structures de représentation des informations et des connaissances sont examinées dans un double but : quelle est la meilleure façon pour représenter des informations abstraites et surtout, quelle est la meilleure façon de faciliter l'accès et l'utilisation de ces informations. L'accent est mis sur une représentation souple, et fonctionnelle, par rapport aux représentations plus complètes des systèmes de vision traditionnels, mais plus divisées en niveaux, ce qui peut en limiter l'utilisabilité.

En décrivant les différentes informations présentes dans le système, une attention particulière est portée au mouvement. En effet, nous utilisons une information issue du mouvement entre les images comme source supplémentaire d'informations pour les traitements. Cette information de mouvement est intégrée avec les autres sources d'informations à caractère statique et elle est utilisée de façon compatible avec celles-ci.

2.1 La représentation des informations en vision

Il existe plusieurs façons de représenter les informations dans un système de vision. Par le terme *information*, deux notions sont à distinguer :

- les informations récoltées par le système lors de l'analyse de l'image, qui sont véritablement le résultat de l'action du système ;
- les connaissances descriptives, qui sont des informations qui sont fournies, a priori, au système, afin de modéliser les différentes entités qu'il aura à analyser.

Cette séparation est importante à effectuer, même si les informations de la première catégorie tendent à devenir des connaissances descriptives pendant l'exécution du système. Afin de discuter des différents mécanismes de représentation possible, ceux-ci ont été

divisés en deux catégories, selon qu'ils appartiennent à des systèmes de vision bas niveau ou haut niveau.

2.1.1 Systèmes de vision bas niveau

Dans les systèmes de bas niveau, les seules informations représentées sont de la première catégorie, car il n'existe aucune modélisation d'objets sémantiques. Les systèmes sont conçus pour analyser les images et en produire une segmentation bas niveau, c'est-à-dire en terme de primitives générales, telles que les régions ou les contours.

Comme exemple de tels systèmes, nous retenons ici ceux de Nazif et Levine [Nazif 84] et de Bellet [Bellet 98].

Dans le premier de ces systèmes, celui de Nazif et Levine, les notions manipulées sont principalement les régions et les contours, ainsi que les zones d'intérêt et les aires. Les aires sont des portions d'images auxquelles on donne des attributs guidant la segmentation des entités la composant (texturée ou non, fermée ou non, ...). À chacune de ces entités se trouve attaché un nombre important d'attributs qui se divisent en trois catégories :

- des attributs de description numérique ;
- des attributs de description spatiale ;
- des attributs logiques.

Dans chacune de ces catégories, on retrouve des attributs portant sur l'entité elle-même et sur son environnement immédiat composé d'autres entités.

Dans le système de Bellet, nous retrouvons le même genre de structures réunissant les différentes informations trouvées sur les primitives de l'image. Ces informations sont principalement pour ce système :

- des attributs de description numérique (photométrie, taille, position, ...) ;
- des attributs de relation, avec les primitives voisines, pour renforcer la coopération.

2.1.2 Systèmes de vision haut niveau

Dans le cas des systèmes de vision conçus pour une interprétation sémantique de la scène, il existe souvent une très forte modélisation des informations et des connaissances, séparées en plusieurs niveaux de représentation illustrant le degré d'abstraction de celles-ci.

Dans VISIONS [Hanson 78], une méthode de représentation par niveau d'abstraction du système est utilisée :

- Pour le bas niveau, il n'existe pas de formalisme particulier. Ce niveau contient les images de l'environnement, ainsi que des primitives de bas niveau extraites des images.

- Les informations de niveau intermédiaire, c'est-à-dire les différentes primitives extraites des images (segments, régions, surfaces, coins, ...), sont accumulées sous forme de structures de données dans une base de données de représentation symbolique.
- Les informations de haut niveau, sur les objets sémantiques et la scène, sont encapsulées dans un mécanisme de *schémas*, développé par Draper [Draper 89].

Il s'agit de différents formalismes de représentation qui ont peu en commun. De ces trois formalismes, le plus puissant d'un point de vue modélisation est celui des schémas. Un schéma, du point de vue de la représentation des informations :

- représente une entité de l'image (une maison, une route, une scène de ville, etc.) ;
- possède toute la connaissance descriptive nécessaire ;
- peut avoir des liens avec d'autres schémas ou sous-schémas ;
- possède son propre tableau noir local pour organiser les différentes hypothèses et informations sur les composantes qui le concerne.

Les schémas possibles sont instanciés, d'abord sous forme d'hypothèses, selon les primitives trouvées dans l'image. Toute la représentation de la scène se fait par un réseau sémantique hiérarchique de schémas avec les liens les unissant.

Chaque schéma possédant ses propres structures internes de représentation, il est nécessaire d'avoir un tableau noir global. Ce tableau noir permet aux différents schémas d'échanger de l'information. En fait, toutes les informations potentiellement utiles à plus d'un schéma y sont inscrites, où tous les autres pourront l'accéder.

Sandakly propose, pour le système MESSIE-II [Sandakly 95], un formalisme de représentation encore plus poussé, et ce à tous les niveaux du système. Toutes les informations sont maintenues à l'intérieur d'un tableau noir central. Les informations construites par le système sont fortement liées à la connaissance descriptive fournie par l'utilisateur sur les différents objets possibles de l'image. En effet, les objets modélisés a priori sont utilisés pour retrouver les instances présentes dans l'image.

Beaucoup de composantes du système ou de l'environnement peuvent être modélisées, sous forme de connaissance descriptive, par l'utilisateur :

- les *objets sémantiques* de la scène, qui sont définis par quatre points de vue :
 - la *géométrie* (forme de l'objet) ;
 - la *radiométrie* (caractéristiques et matériaux) ;
 - le *contexte* (potentiel et réel) ;
 - la *fonctionnalité* (utile pour la validation) de l'objet ;

- les *capteurs de l'image* (prises de vue, résolution, bande passante) ;
- les *opérateurs de localisation* (relations spatiales entre objets de l'image) ;
- la *scène* (réseau d'objets).

Comme beaucoup d'autres systèmes, MESSIE-II divise les informations selon trois niveaux d'abstraction. À chaque niveau sont définis les traitements nécessaires. À chaque niveau correspond ses propres structures d'informations, tout en conservant une certaine interaction entre les deux premiers niveaux :

- Le *bas niveau* concerne les primitives régions et contours qui sont indépendantes du domaine d'application.
- Le *niveau intermédiaire* contient les primitives de base qui ont été interprétées. À ce niveau, on commence à faire des groupements perceptuels et de l'appariement.
- Le *haut niveau* est la représentation, complète ou partielle, de l'état du monde, permettant la description de la scène observée.

2.2 L'environnement et les informations dans le système

Dans notre système, nous allons tenter de concilier les avantages des approches bas niveau, pour lesquelles l'absence de modélisation haut niveau permet une plus grande souplesse dans l'analyse des entités de l'image et des approches haut niveau, qui ont besoin d'avoir certaines connaissances sur le domaine d'application du système et différents niveaux de représentation des informations récoltées.

La modélisation a priori des entités du domaine d'application présuppose une connaissance parfaite de ce domaine et des mécanismes de vision qui seront mis en oeuvre pour l'étudier. Cependant, l'expérience des différents systèmes de vision existants montre que les résultats ne sont pas à la hauteur des attentes initiales.

Face à ce constat, plutôt que de renforcer la modélisation existante en ajoutant plus d'informations qui devront être définies a priori, nous prêchons pour une modélisation des connaissances descriptives plus souple, voire inexistante, pour donner au système lui-même la possibilité de construire sa base de connaissances à partir des informations qu'il récoltera dans l'image. Les connaissances seront accumulées au gré des différentes interactions des agents entre eux ou des agents avec l'environnement. Les informations provenant de l'image serviront de connaissances descriptives au fur et à mesure de l'exécution du système. Il s'agit donc d'une représentation des informations et des connaissances à caractère utilitaire permettant de s'adapter au monde environnant plutôt que de s'en abstraire.

À cette fin, deux structures distinctes représentent les informations récoltées sur les différentes entités de l'image :

- Une structure nommée *environnement* contient les images sources, leurs caractéristiques et les résultats de segmentation. Elle est exclusivement composée d'images et est décrite au paragraphe 2.3.
- Une structure nommée *information* contient les descriptions des différentes structures que les agents construisent. Elle est composée de structures décrivant essentiellement la géométrie, les statistiques photométriques et les identifications de type et de groupe. Elle est décrite au paragraphe 2.5.

Ces deux structures sont complémentaires dans la description de l'application et des résultats trouvés par les agents. Toutes les informations récoltées par les agents sont accumulées de façon externe à ceux-ci pour deux raisons :

1. Pour permettre aux informations de subsister après la mort de l'agent.
2. Pour être mises en commun et accessibles par tous les autres agents du système.

Ces deux raisons s'inscrivent dans l'axe de la liberté de l'information auquel nous nous attachons. Dans la description qui sera faite, cet axe sera amplement démontré. Il s'agit essentiellement pour la structure d'environnement, de permettre une compatibilité dans la circulation des différentes caractéristiques composant l'image et pour la structure d'information, de permettre une compatibilité entre différentes sources d'information représentant des entités situées à différents niveaux de représentation.

2.3 L'environnement du système

La première structure de représentation de notre système est l'environnement. Malgré le terme employé, il est limité aux différentes images, et leurs caractéristiques, qui servent de base au travail des agents. La particularité de la représentation est de tenter de rendre compatibles, et ouvertes à différents traitements, des informations provenant de différentes sources.

2.3.1 Représentation de l'environnement

Notre système traite des séquences d'images. Pour chaque image, un certain nombre de caractéristiques sont pré-calculées pour enrichir les informations disponibles. Toutes ces informations constituent l'environnement de base de l'agent (figures 2.5 et 2.6) :

- une ou plusieurs *images sources* ;
- le *gradient de l'image* ;

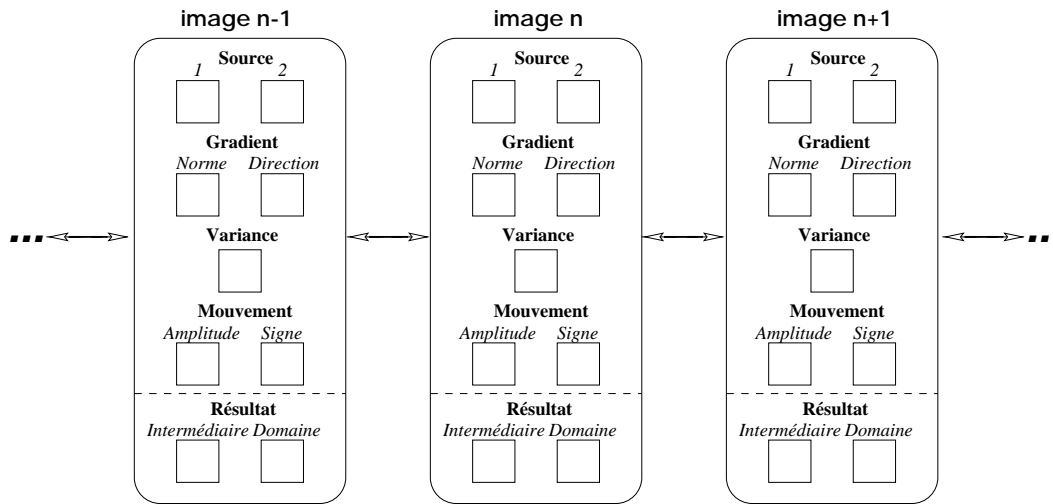


FIG. 2.5 – *Structure de l'environnement, qui contient les images. À chaque image est associée un certain nombre de caractéristiques précalculées et des images résultats qui viennent enrichir cet environnement.*

- la *variance* de l'image ;
- le *mouvement* entre des images successives.

À ces informations de base, d'autres images viennent s'ajouter, représentant les résultats de la segmentation. Ces résultats sont accessibles par tous les agents et viennent enrichir l'environnement.

2.3.1.1 Les images sources

Le système est dédié à l'interprétation 2D de séquences d'images. Ces séquences sont normalement constituées d'images prises à intervalles réguliers. Dans ce système, plusieurs images sources d'une scène peuvent être acquises simultanément, par différents capteurs. Ces images sont complémentaires et donnent des informations différentes sur la scène observée. Par exemple, pour notre application, une image montrant les noyaux des cellules en fluorescence (figure 2.6b) peut être ajoutée à l'image de base montrant les cellules en contraste de phase (figure 2.6a) (paragraphe 5.1.2).

Cependant, les autres caractéristiques qui sont précalculées sont issues de l'image de base seulement. Les images, provenant des autres capteurs, sont considérées comme secondaires, car leur rôle est principalement d'enrichir l'information présente dans la première image. D'ailleurs, ces images secondaires peuvent être présentes ou non. Le système peut avoir à s'adapter à différents contextes, où toutes les informations ne sont pas nécessairement présentes.

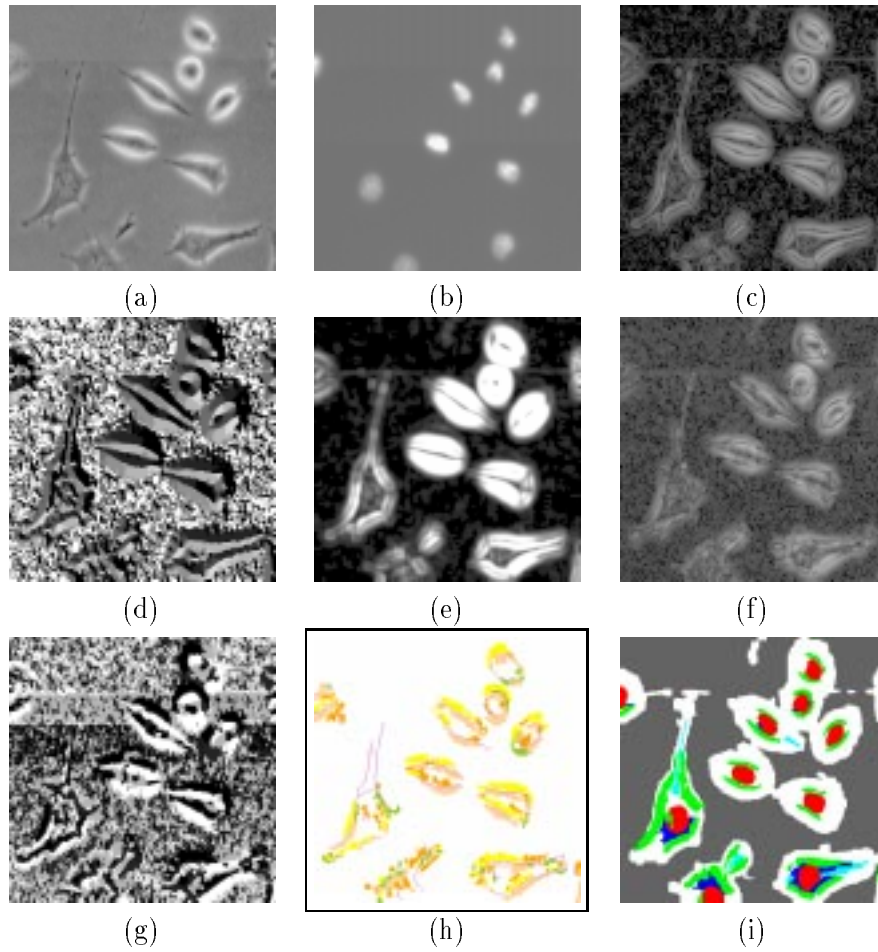


FIG. 2.6 – Images et caractéristiques de base composant l'environnement. (a) Image source principale. (b) Image source secondaire. (c) Norme du gradient. (d) Direction du gradient. (e) Variance. (f) Amplitude de la différence. (g) Signe de la différence. (h) Segmentation niveau primitive. (i) Segmentation niveau composante.

2.3.1.2 Le gradient

Le gradient de l'image donne une information sur les variations existantes dans l'image. Il existe un nombre considérable de techniques pour calculer le gradient discret. Dans la plupart des cas, le gradient est calculé grâce à un produit de convolution. L'évaluation des différentes méthodes permet de constater qu'elles produisent des résultats similaires pour la majorité des applications [Salotti 94]. Le choix de la méthode est donc soumis à des critères plus subjectifs qu'objectifs.

Nous avons choisi une méthode proposée par Salotti, reposant sur une utilisation de 8 masques 5x5, dérivées des masques de Kirsch [Salotti 94, p. 75]. Deux images sont calculées à partir de ces convolutions : une image de la norme (figure 2.6c) et une autre de la direction du gradient représentée par le code de Freeman (figure 2.6d).

2.3.1.3 La variance

La variance donne une indication sur l'homogénéité de l'image. Cette information est calculée pour chaque point de l'image, en utilisant des masques de différentes tailles. La taille du masque utilisée dépend principalement des caractéristiques des images étudiées et de la finalité de l'application envisagée.

Nous avons choisi de calculer la variance sur des masques de taille 9x9 (figure 2.6e). Un masque de cette taille, assez grand, est utile pour lisser l'image et servira à situer cette information dans un contexte plus large. Cela permettra d'effectuer une segmentation plus approximative, mais plus robuste pour certains agents (paragraphe 5.2) [Wu 95].

2.3.1.4 Le mouvement

Les informations présentées ci-dessus ont toutes en commun le fait qu'elles ne concernent qu'une image, ou un instant précis de la séquence. Cependant, le système travaillant sur des séquences d'images, il est important de pouvoir calculer certaines informations liant les différentes images de la séquence entre elles. Le but est de permettre au système de raisonner en couplant la dimension statique des objets de la scène à la dimension temporelle de ces mêmes objets.

La méthode utilisée est une combinaison de différences d'images successives. Cette différence d'images n'est pas une mesure directe du mouvement, mais plutôt un indicateur du déplacement survenu entre deux images successives de la séquence. Deux images sont calculées : l'amplitude de la différence (figure 2.6f) et son signe (figure 2.6g). La méthode de calcul employée est expliquée en détails au paragraphe 2.4.

2.3.1.5 Les images résultats de segmentation

Les résultats de la segmentation font partie de l'environnement comme toute autre image. En fait, il s'agit d'une information dynamique qui évolue par rapport aux autres informations qui sont précalculées, et donc fixes pendant l'exécution. Ces résultats viennent enrichir l'environnement et permettent aux agents de résoudre certaines ambiguïtés par cet apport mutuel de connaissances.

Il peut y avoir plusieurs images résultats de segmentation. Chacune de ces images représente un niveau de représentation de l'information et permet des raisonnements intermédiaires. Dans notre système, il existe deux images résultats, une pour les résultats finaux (figure 2.6i) et l'autre permettant des constructions intermédiaires et complémentaires à la première image (figure 2.6h). Le nombre d'images de segmentation n'a pas de liens avec le nombre d'images sources. Les images sources représentent différentes vues d'une scène qui sont combinées ensemble. Les images de segmentation tirent profit de toutes les images sources, en les combinant, et en triant l'information obtenue par degré d'abstraction. Plus de détails sont données sur ces deux niveaux au paragraphe 2.5.2.

Images \ Types	numérique / symbolique	linéaire / périodique	statique / temporelle	fixe / évolutive
sources	numérique	linéaire	statique	fixe
norme du gradient	numérique	linéaire	statique	fixe
direction du gradient	numérique	périodique	statique	fixe
variance	numérique	linéaire	statique	fixe
amplitude du mouvement	numérique	linéaire	temporelle	fixe
signe du mouvement	symbolique	-	temporelle	fixe
résultats	symbolique	-	statique	évolutive

FIG. 2.7 – *Classification des différentes images d'information de l'environnement. Plusieurs types d'informations sont utilisées par le système, mais celui-ci doit préserver une certaine cohérence et une compatibilité entre ces différentes sources afin d'assurer la liberté de l'information.*

2.3.2 Apport de la représentation de l'environnement

La représentation très simple des images que nous avons choisie comporte plusieurs avantages, au niveau de la souplesse d'utilisation et de la liberté de l'information. De plus, cet environnement est le milieu de vie des différents agents du système, qui y accèdent en s'ancrant en celui-ci.

2.3.2.1 Liberté de la représentation

Un aspect primordial de cette représentation de l'environnement est la compatibilité existante entre les différents types d'informations. Il faut qu'un agent puisse utiliser les différentes sources d'informations sans se soucier, autant que possible, d'adapter ses méthodes à leurs utilisations. Par exemple, un agent peut utiliser les mêmes traitements sur une information statique comme la variance que sur une information mouvement.

Les informations présentes dans l'environnement peuvent être classées selon différents critères qui influenceront leurs manipulations (figure 2.7). Cette manipulation des différents types d'informations ne pourra pas être parfaitement compatible, mais un effort sera fait pour tenter de l'adapter autant que possible. Le but de cette adaptation est de permettre un échange et une combinaison plus fluide entre les différentes sources d'information du système.

2.3.2.2 L'environnement comme un territoire

L'environnement est perçu, dans ce système, comme un univers à explorer et à enrichir pour les agents. Chaque agent est situé sur une image, c'est-à-dire qu'il reçoit au départ une position d'où il s'initialise et autour de laquelle il évoluera, en explorant son espace environnant et en tentant de faire croître le territoire qu'il occupe (la région ou le contour)

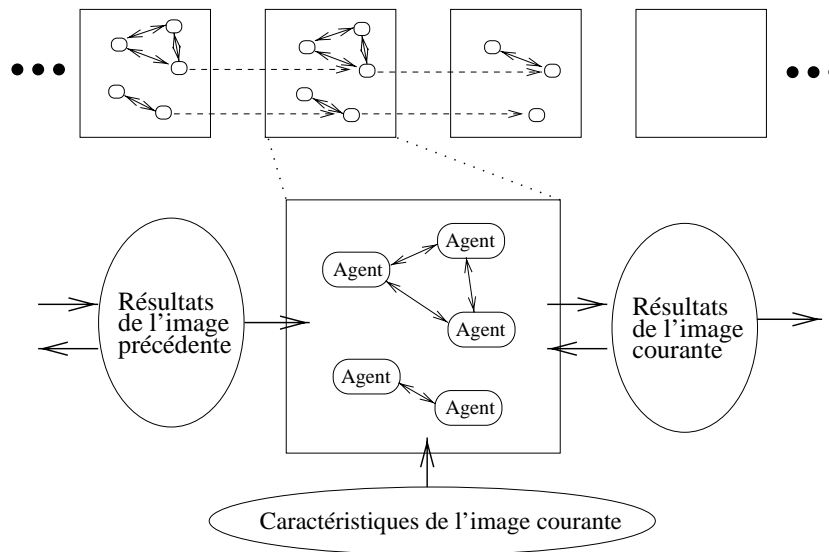


FIG. 2.8 – *Modèle de l'environnement des agents. Chaque agent est situé sur une image et interagit avec les agents de son image et des images voisines. Il a accès à toutes les informations sur son image, sa segmentation et ses caractéristiques, ainsi qu'aux informations sur l'image précédente, qui peuvent l'aider dans son travail.*

(figure 2.8).

Les agents ont accès, sans restriction, aux informations de l'image qu'ils occupent. Toutes les informations (sur la segmentation) sont mises en commun par les agents (images résultats et structures d'information). Elles permettent aux agents d'observer l'évolution de l'action des autres agents et de s'adapter à cette évolution.

Les agents ont également accès aux informations de l'image précédente, afin de tenter de maintenir une cohérence dans la segmentation de la séquence d'images (paragraphe 3.5.4.2). Pour l'instant, il ne s'agit que d'un accès en lecture de cette information, donc en tant qu'observateur. Il n'existe pas, dans le système actuel, de mécanismes de communication ou d'influence mutuelle entre des agents travaillant sur des images différentes.

La propagation des résultats s'effectue donc vers l'avant dans la séquence. D'ailleurs, cette séquence est traitée en pipeline, c'est-à-dire que des agents placés initialement sur la première image, d'autres agents seront initiés sur les images suivantes seulement lorsque le traitement de l'image sera jugé suffisamment avancé (primitives suffisamment grandes).

2.4 Utilisation de l'information mouvement

Le mouvement est une information importante lors de l'analyse d'une séquence d'images, particulièrement dans le cas d'une application où les objets sont difficiles à discerner, du fait d'un trop faible contraste de l'image. Le déplacement des objets d'une image à l'autre

est source de renseignements pour la segmentation. Les informations issues de la segmentation statique sont complémentaires à celles issues de la segmentation du mouvement. Les deux segmentations sont donc traitées ensemble dans le but de se compléter mutuellement.

2.4.1 Calcul de l'information mouvement

Il existe différentes méthodes pour quantifier le mouvement qui existe entre deux ou plusieurs images d'une séquence [Bouthemy 95, Hennebert 96]. Une des plus connues est le flot optique [Horn 80], qui consiste à calculer le champ de vitesses (norme et direction) mesurant le déplacement des objets. Plusieurs techniques sont possibles pour calculer ce champ de vitesses. Cependant, ces techniques présentent plusieurs désavantages pour notre application [Barron 94] :

- le déplacement d'un objet entre deux images doit être faible ;
- les techniques sont principalement adaptées au mouvement rigide et peu aux objets déformables ;
- certaines techniques demandent un grand nombre d'images pour les calculs, ce qui est incompatible avec notre stratégie d'action locale sur les images.

Certains travaux existent sur l'utilisation du flot optique pour notre type d'application [Siegert 94, Doisy 96]. Ces travaux ont utilisé un échantillonnage considérable ou ont réalisé une carte de déplacement sans segmentation ou interprétation des objets.

2.4.2 Le mouvement par différence d'images

Nous avons préféré les méthodes utilisant la différence entre images [Jain 81]. Ces méthodes ne donnent pas une mesure du mouvement au sens de vecteurs de vitesse et direction de déplacement des objets, mais permettent de recueillir des indices sur les endroits de l'image où il y a eu déplacement. Cette notion d'indice est parfaitement compatible avec la philosophie de notre système, où l'on recherche la combinaison de plusieurs sources d'informations compatibles pour raisonner sur les objets de l'image et obtenir une segmentation de l'image.

Le principe général est de calculer la différence entre deux (ou quelques) images, la différence absolue entre deux images étant habituellement considérée. Dans la figure 2.9, la différence entre deux images est considérée pour le déplacement d'un carré uniforme. La figure 2.9c, qui illustre le cas où il n'y a pas de recouvrement entre les positions initiale et finale du carré, permet d'identifier les contours exacts du carré dans les deux images. Avec l'information du signe de la différence il est aisé de séparer les deux carrés et d'identifier l'image d'apparition de chacun d'eux. L'utilisation du signe est utile pour la séparation des différentes zones identifiées par la différence. Cependant, cette information n'est pas

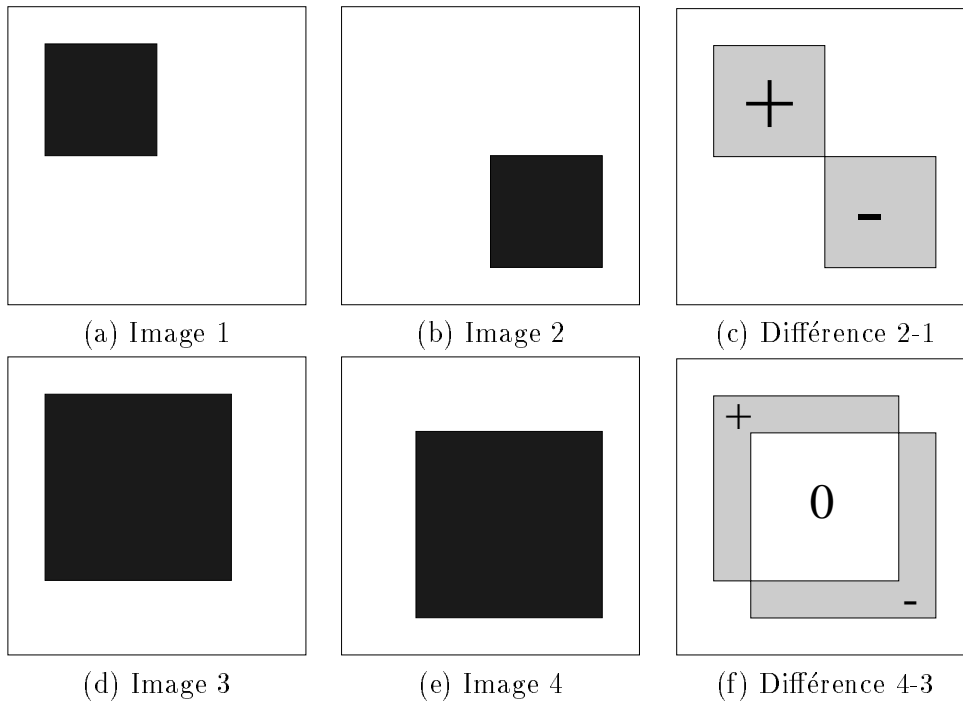


FIG. 2.9 – *Principe général de la différence d'images. (a) (b) Images illustrant le déplacement d'un carré. (c) La différence entre les deux premières images montre deux zones de signes différents. (d) (e) Nouveau déplacement d'un carré, mais avec recouvrement des deux carrés. (f) Différence d'images montrant trois zones différentes. Dans ce cas-ci, la différence ne permet pas de reconstituer entièrement le carré, mais donne des indices visuels sur son déplacement.*

souvent utilisée, les différents auteurs préférant ne retenir qu'une image de la valeur absolue de la différence.

Cependant, la différence d'images, contrairement à d'autres méthodes d'analyse du mouvement, ne permet pas toujours d'identifier complètement les objets, tel qu'illustré à la figure 2.9f. En effet, lorsqu'il y a recouvrement entre les deux positions, seules des zones en périphérie de l'objet permettent de détecter le mouvement. Dans ce cas, la différence apporte un indice visuel pour la segmentation. Il sera alors nécessaire de coupler cet indice avec d'autres pour reconstruire la scène complète.

Cette méthode de différence d'images est utilisée depuis longtemps dans les travaux sur la segmentation du mouvement [Jain 81, Thompson 80]. Certains travaux combinent cette information de différence avec des informations statiques. Leung et Yang recherchent les contours communs entre une image de différence et une image de gradient afin d'identifier les mouvements d'une personne [Leung 95]. Fan et al. ont couplé l'information de différence avec la variance dans une méthode de seuillage basée sur l'entropie [Fan 96]. Dans tous ces cas, la méthode de différence d'images, couplée avec d'autres sources d'information, s'est avérée utile pour détecter des objets déformables.

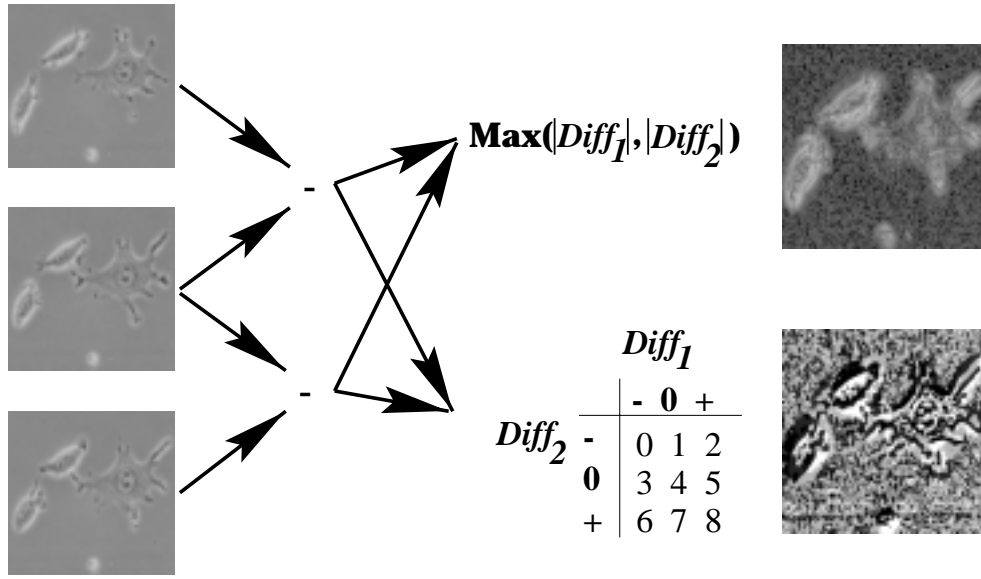


FIG. 2.10 – Calcul des images de différence (amplitude et signe).

2.4.3 Notre méthode de calcul de la différence d'images

Nous avons choisi de combiner trois images pour effectuer une différence d'images, afin d'obtenir des régions plus fines et plus précises sur les régions en mouvement dans une image. Nous avons donc développé une méthode simple de calcul de différence, et qui permet de différencier plusieurs types de zones de mouvement. Nous voulons ainsi distinguer une zone d'une image ayant un déplacement passé (image $n - 1$), futur (image n) ou constant (images $n - 1$, n et $n + 1$). La méthode employée est une amélioration de la méthode de calcul classique de la différence. Elle est basée sur trois images de la séquence ($n - 1$, n et $n + 1$). Deux valeurs sont calculées, que nous nommerons *amplitude* et *signe* de la différence¹.

Le calcul de l'*amplitude* et du *signe* de la différence, pour un pixel (i, j) de l'image n , est effectué de la manière suivante (figures 2.10 et 2.11) :

$$\begin{aligned}
 D_{n,n-1}(i, j) &= \text{Image}_n(i, j) - \text{Image}_{n-1}(i, j) \\
 D_{n+1,n}(i, j) &= \text{Image}_{n+1}(i, j) - \text{Image}_n(i, j) \\
 \text{amplitude}(\mathbf{i}, \mathbf{j}) &= \max(|D_{n,n-1}(i, j)|, |D_{n+1,n}(i, j)|) \\
 S_{n,n-1}(i, j) &= \text{sgn}(D_{n,n-1}(i, j)) + 1 \\
 S_{n+1,n}(i, j) &= \text{sgn}(D_{n+1,n}(i, j)) + 1 \\
 \text{signe}(\mathbf{i}, \mathbf{j}) &= S_{n,n-1}(i, j) + (S_{n+1,n}(i, j) * 3)
 \end{aligned}$$

1. Ces termes sont arbitraires et ne proviennent d'aucune référence. Ils visent à distinguer deux informations que l'on peut calculer à partir de la différence d'images. En particulier, le terme *amplitude* n'est pas employé au sens strict de l'amplitude du mouvement, mais plutôt de l'amplitude de la différence des images.

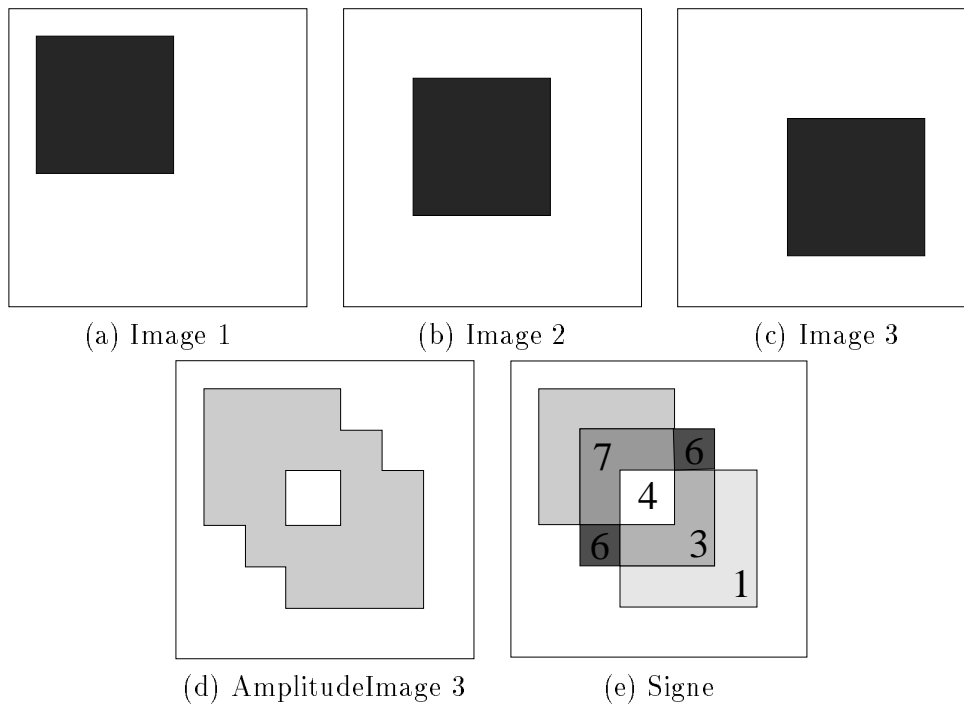


FIG. 2.11 – Illustration de notre méthode de différence d’images sur des images de synthèse. (a) (b) (c) Images illustrant le déplacement d’un carré. (d) L’amplitude de la différence s’obtient en prenant les valeurs absolues maximales des différences 3-2 et 2-1. (e) Le signe des différences permet d’isoler plus précisément les différentes zones de mouvement existantes.

Le calcul de l’*amplitude de la différence* est simple. Il s’agit de prendre le maximum des deux différences comprenant l’image n . Pour le calcul du *signe de la différence*, il s’agit de mettre en relation les signes des deux différences. Pour une différence, il existe trois possibilités (signe négatif, nul, signe positif). Pour deux différences, on obtient donc 9 combinaisons possibles. L’intérêt de cette approche est de permettre l’analyse fine du mouvement de chaque composante, en dissociant les différents types de mouvement entre les images. La figure 2.10 montre un exemple d’une telle analyse. Cette double information sur le mouvement (amplitude et signe) nous permet de découper de façon aisée l’image en différents mouvements homogènes.

2.5 Les informations construites sur les entités

La deuxième structure de représentation de notre système regroupe les informations qui sont issues du travail des agents sur les différentes entités de l’environnement. Les différentes classes d’informations sur les entités sont présentées, avec leur organisation et les liens entre elles. Les avantages de cette représentation, et les différents niveaux de représentation et de communication de l’information sont aussi abordés.

2.5.1 Représentation des informations

Comme beaucoup d'autres systèmes en vision, le système que nous avons conçu réunit des informations de natures diverses sur les entités qu'il construit. Ces informations sont représentées par diverses structures.

Plusieurs notions semblables, mais non-identiques, qui peuvent parfois porter à confusion, sont employées dans la suite pour décrire les informations dans le système. Il s'agit des notions de primitive, de composante, de groupe et d'entité. Dans ce travail, nous définissons ces notions de la manière suivante :

- une *primitive* est une région ou un contour, c'est-à-dire une construction de base de la segmentation ;
- une *composante* est également une région ou un contour, mais porteuse d'une signification sémantique au sens de l'application, comme par exemple, une partie du fond ou un noyau d'une cellule ;
- un *groupe*, ou groupement perceptuel², est un ensemble de composantes et/ou de primitives qui repose sur un concept distinct, à un plus haut niveau, de l'application, comme par exemple, la cellule qui représente un groupe de composantes ;
- une *entité* est la notion la plus générale, puisqu'elle englobe les définitions précédemment données, et qu'elle est utilisée de façon générale pour faire référence indistinctement à l'ensemble des notions précédemment définies.

Les différentes structures d'informations sont décrites dans les paragraphes suivants. Comme pour l'environnement, ces structures vont présenter plusieurs bases communes, afin de faciliter la compatibilité entre les traitements qui seront appliqués. Ces structures sont relativement simples, puisqu'elles consistent principalement en une série de statistiques calculées pour une entité. Elles servent à réunir les informations utiles pour le système dans ses traitements autant qu'à décrire les objets dans un but d'interprétation.

Plusieurs structures d'informations existent, allant dans un ordre croissant de spécificité. Certaines structures servent de bases communes aux autres, afin d'assurer une compatibilité entre les différentes structures. Les différentes structures d'informations sont présentées sur la figure 2.12, avec des exemples concrets à la figure 2.13. Ces structures sont :

- *InfoStat* : les informations statistiques (intervalle, moyenne, écart-type) sur une entité ;

2. Nous utiliserons les deux termes, groupe et groupement perceptuel, selon les cas. Mais il s'agit de la même notion qui est référencée dans les deux cas. Par abus de langage, nous utilisons également la notion de groupe pour des agents associés aux informations reliées à un même groupement perceptuel.

Structures d'informations générales :

InfoStat

byte minimum
byte maximum
float moyenne
float écart

InfoImage

int no_image
int taille
InfoStat gris_1
InfoStat gris_2
InfoStat gradient_norme
InfoStat variance
InfoStat mouvement_amplitude
Histogramme[8] gradient_direction
Histogramme[9] mouvement_signe

Structures d'informations sur les primitives :

InfoPrimitive

Type type
Id no_primitive
Id no_cellule
Pixel germe
Pixel centre_gravité
Pixel[2] boite_englobante
InfoImage

InfoRégion

InfoPrimitive
bool mouvement

InfoContour

InfoPrimitive
Pixel[2] extrémité

Structures d'informations sur les groupements perceptuels :

InfoGroupe

Id no_groupe
int no_image
Pixel germe_groupe
int taille
Pixel[2] boite_englobante
Pixel centre_gravité
int nb_germes
InfoStat germes
Liste composante

InfoComposante

Type type
int nb_germes
InfoStat germe
InfoImage

InfoComposante

--

...

FIG. 2.12 – Représentation simplifiée des différentes structures d'informations existantes dans le système. Les structures *InfoStat*, *InfoImage* et *InfoPrimitive* sont des structures de base n'ayant pas d'instanciations propres, mais qui servent pour définir les autres structures. Seules les structures *InfoRégion*, *InfoContour* et *InfoGroupe* sont associées aux entités de l'image.

- *InfoImage* : les informations statistiques (de type *InfoStat*) par rapport à l'image (niveaux de gris, gradient, variance, mouvement) pour une entité ;
- *InfoPrimitive* : les informations générales³ de description d'une primitive (contient des informations de type *InfoImage* plus d'autres informations de description) ;
- *InfoRégion* : les informations relatives à une primitive région (contient des informations de type *InfoPrimitive* plus d'autres informations de description spécifiques à une région) ;

3. Ces informations sont générales dans le sens qu'elles s'appliquent à une région et à un contour de la même façon.

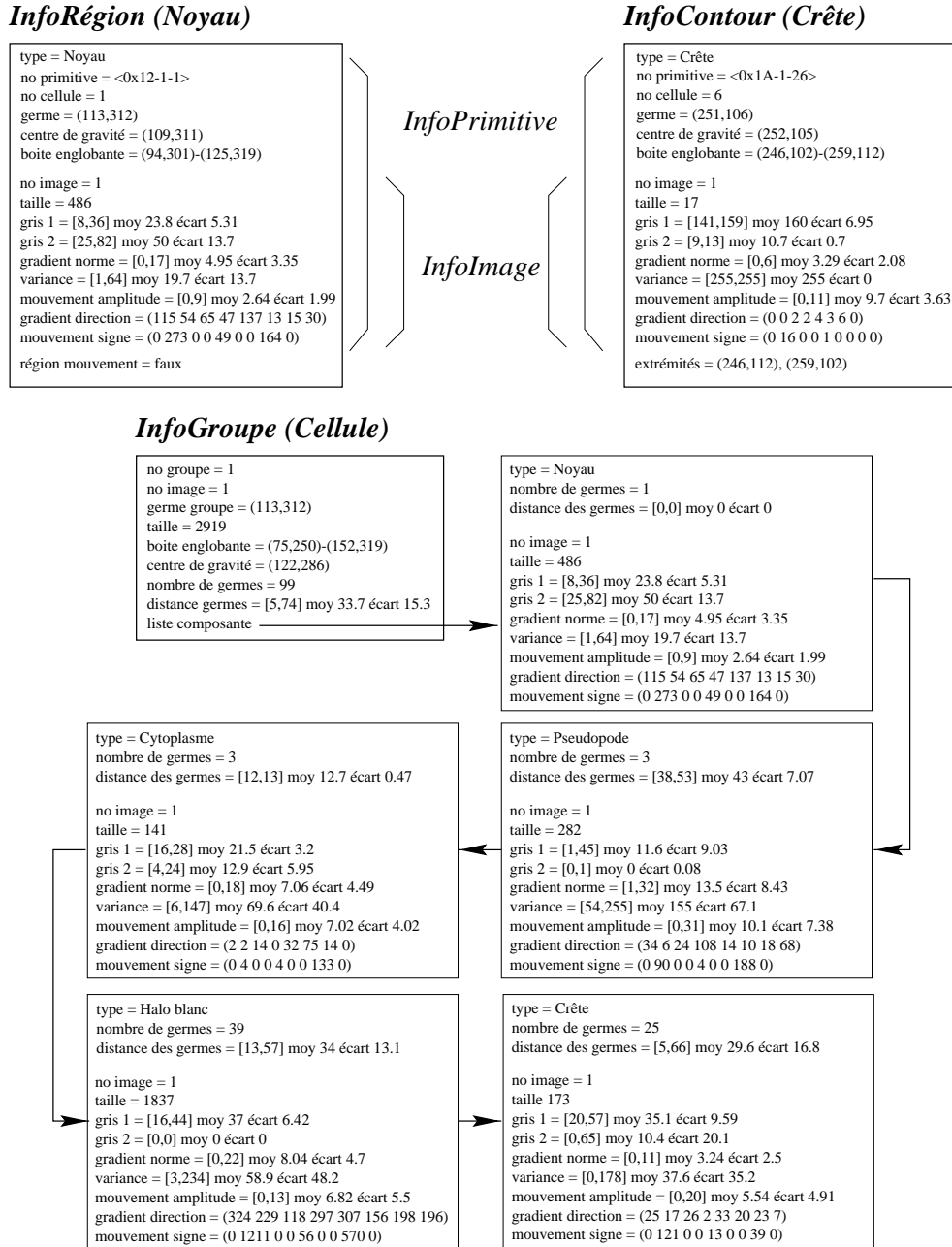


FIG. 2.13 – Exemples tirés de l'application pour l'utilisation des structures de représentation de l'information. Trois exemples sont donnés : les informations sur une région, les informations sur un contour et les informations sur un groupement perceptuel (ou groupe). On identifie dans ces structures les différentes structures de base (InfoStat, InfoImage et InfoPrimitive), ainsi que les champs spécifiques aux entités décrites ici. Dans ces structures, gris1 et gris2 représentent les deux images sources présentes, et entre crochets se trouvent les valeurs minimale et maximale.

- *InfoContour* : les informations relatives à une primitive contour (contient des informations de type *InfoPrimitive* plus d'autres informations de description spécifiques à un contour) ;
- *InfoGroupe* : les informations relatives à un groupement perceptuel (contient des informations sur le groupe plus une structure d'informations (*InfoComposante*) par type de primitive présent dans le groupement).

Ces différentes structures d'informations sont définies avec les mêmes blocs de base, qui sont assemblés et enrichis de structure en structure. Cette représentation permet d'implanter un véritable partage d'informations, qui sera réalisé grâce à l'homogénéité des structures. Par exemple, il est possible d'effectuer la comparaison entre une région et un groupe, par l'intermédiaire de la représentation commune *InfoPrimitive*, qui est présente dans la structure *InfoRegion*, et dans les structures *InfoComposante* décrivant le groupe. Cette *interface* commune permet la communication et l'échange d'informations à tous les niveaux de représentation.

2.5.1.1 Informations statistiques sur l'image

Toutes les entités qui sont extraites des images possèdent certaines caractéristiques photométriques, du simple contour à l'objet plus abstrait qui consiste en un regroupement conceptuel de plusieurs primitives (par exemple une cellule). Un certain nombre de ces caractéristiques sont ainsi compilées pour toutes les entités de l'image, dans une structure appelée **InfoImage**. Il existe une structure pour chaque entité de l'image qui est segmentée par les agents. Cette structure tente d'uniformiser l'accès aux différentes informations statistiques.

La structure **InfoImage** est composée des éléments suivants :

- du *numéro de l'image* dans la séquence où l'entité a été identifiée ;
- de la *taille* en pixels de l'entité ;
- de *statistiques*, telles que la moyenne, l'écart-type⁴, les valeurs minimales et maximales, des informations de niveau de gris (sur une ou plusieurs images sources), de norme du gradient, de variance, d'amplitude du mouvement, et ce, sur tous les pixels qui composent l'entité ;
- d'un *histogramme* des valeurs de la direction du gradient (utilisant le code de Freeman)⁵ ;

4. L'écart-type n'est pas véritablement calculé, mais plutôt la moyenne des carrés, qui permet ensuite, avec la moyenne, de retrouver l'écart-type.

5. Cette idée est inspirée de [Bellet 98, pp.39-40]. Elle peut permettre d'identifier des situations particulières par des valeurs surélevées de l'histogramme.

- d'un *histogramme* des valeurs du signe du mouvement (découpé en 9 zones, tel qu'expliqué au paragraphe 2.4.3).

Ces statistiques sont mises à jour pour chaque nouveau pixel ajouté à l'entité. À première vue, certaines statistiques peuvent sembler inutiles ou redondantes, comme la compilation de l'écart-type des niveaux de gris et la moyenne de la variance des pixels de la région⁶. Ces deux valeurs, qui peuvent être semblables dans certains cas, sont calculées dans un but de compatibilité entre les différentes sources d'informations. L'image des niveaux de gris est considérée au même titre que l'image de la variance (paragraphe 2.3.1).

Certaines informations sont plus difficiles à harmoniser, comme la direction du gradient ou le signe du mouvement, car ces informations possèdent des caractéristiques différentes (paragraphe 2.3.2.1). Certaines approximations ont donc été adoptées afin d'essayer de maintenir une compatibilité entre les différentes informations :

- lorsqu'il sera question de la moyenne de la direction du gradient ou du signe du mouvement, il s'agira en fait du maximum des histogrammes respectifs, car la notion de moyenne n'est pas applicable pour ce genre de données, qui sont périodiques ou symboliques ;
- lorsqu'il sera question des valeurs minimale et maximale de ces mêmes données, il s'agira du minimum et du maximum des histogrammes.

Le but de cette représentation, qui peut être discutable au strict plan mathématique, est d'uniformiser les différentes sources d'information, afin d'arriver à généraliser les méthodes employées dans les traitements. Cette uniformisation est nécessaire sur le plan de la liberté de l'information, pour favoriser tous les échanges possibles entre les différentes sources et permettre la diffusion des informations et des connaissances accumulées.

2.5.1.2 Informations sur les primitives région ou contour

Les primitives de base dans le système sont les régions et les contours. Mais le raisonnement est principalement basé sur les régions, les contours étant perçus comme des informations secondaires (paragraphe 2.5.2.1). La plupart des informations recueillies sur ces deux types de primitives sont identiques.

Les informations générales sur une primitive, région ou contour, contenues dans la structure `InfoPrimitive`, sont (figure 2.12) :

- un *type* d'agent, identifiant la primitive par rapport aux composantes définies pour l'application ;

6. Ces deux valeurs sont calculées de façon différente. Dans le premier cas, il s'agit de l'écart-type des niveaux de gris des pixels de l'entité (donc, véritablement l'écart-type de l'entité), tandis que dans le deuxième cas, il s'agit de la moyenne des niveaux de variance des pixels de l'entité (cette variance étant calculée dans une fenêtre centrée sur le pixel, et non pas sur l'entité).

- un *numéro d'identification de la primitive* ; c'est le même numéro qui sera associé à l'agent lié à cette primitive ; ce numéro permet de retracer aisément le type de primitive, sa spécialisation (interprétation dans l'image) et le numéro de l'image dans laquelle la primitive a été identifiée ;
- un *numéro d'identification du groupe* auquel appartient la primitive, si tel est le cas (facultatif) ; ce numéro sera un lien entre cette structure d'information et la structure d'information sur le groupe ;
- les *informations statistiques générales* de la structure **InfoImage** calculées sur la primitive ;
- une *boîte englobante* permettant de retracer la position de la primitive dans l'image ;
- le *pixel germe* initial de l'agent qui a identifié la primitive ;
- le *centre de gravité* de la primitive.

La structure d'informations sur les régions, nommée **InfoRégion** possède un champ supplémentaire, qui indique si cette région a été identifiée grâce à des *informations issues du mouvement* ou non, et qui sert principalement à des fins d'apprentissage dans le système (paragraphe 6.1).

La structure d'information sur les contours, nommée **InfoContour** possède comme information supplémentaire les *pixels aux extrémités* du contour.

Certaines informations ne sont pas conservées, comme la liste des pixels d'un contour ou du périmètre d'une région, mais plutôt reconstruites au besoin à l'aide des informations contenues dans les structures présentées et dans les images de segmentation. Certaines autres informations ne sont utiles qu'aux agents liés à ces primitives, et n'ont aucun intérêt pour les autres agents, comme la liste des pixels candidats à la croissance de la région ou du contour. Les informations de ce type, servant plus pour agir sur la primitive que pour la décrire, sont définies intrinsèquement à l'agent.

2.5.1.3 Informations sur les groupes de primitives (groupement perceptuel)

Les primitives de base (région et contour) peuvent former des objets, de plus haut niveau, qui ont un sens sémantique pour l'application de vision. Ces primitives forment alors un groupement perceptuel, avec des caractéristiques propres qui peuvent le différencier d'autres groupes existants. Dans notre application, un seul groupe existe, la cellule, qui est composée de différentes régions et contours (pseudopodes, noyaux, etc.). Nous pourrions nous référer soit au groupe de composantes (ou de primitives), soit à la cellule pour désigner la même entité.

Quelques informations de base permettent de décrire un groupe, dans la structure **InfoGroupe** (figure 2.12) :

- un *numéro d'identification* du groupe qui est un lien entre le groupe et chacune des composantes de ce groupe ;
- le *numéro de l'image* où ce groupe a été identifié ;
- le *pixel germe* initial du groupe, qui est le germe du premier agent ayant été lancé sur ce groupe ;
- la *taille* en pixels du groupe, qui est la somme des tailles des composantes ;
- la *boîte englobante* permettant de retracer la position du groupe dans l'image ;
- le *centre de gravité* du groupe ;
- le *nombre de germes* d'agents qui ont été lancés sur ce groupe ;
- les *distances des germes* lancés, mesurées depuis le centre du groupe, et calculées par diverses statistiques, telles que la moyenne, l'écart-type, les valeurs minimales et maximales ; ces mesures peuvent servir pour caractériser le groupe dans sa forme ;
- une *liste d'informations* sur chacun des types différents de composantes qui sont présents dans le groupe, regroupées dans une structure **InfoComposante** ; pour chaque type de composante, différentes informations sont retenues :
 - le *nombre de germes* d'agents de ce type pour le groupe ;
 - les *distances des germes* calculées de la même manière que pour le groupe, mais cette fois classées par type ;
 - les *informations statistiques générales* de la structure **InfoImage** calculées sur l'ensemble des composantes du type.

La structure d'information sur le groupe est donc composée de deux éléments importants, que sont les informations générales sur le groupe et les informations classées par type de composante formant ce groupe. Ces deux séries d'informations se complètent bien, car elles permettent de structurer le groupe comme la somme des composantes. Chaque primitive de l'image peut appartenir à un groupe ou non (par exemple, le fond n'appartient à aucun groupe).

Cette représentation permet aussi de comparer directement une région ou un contour avec le groupe entier, en comparant la primitive avec son type correspondant dans la liste des composantes. Même si les deux structures, pour la primitive et pour le groupe, sont à des niveaux d'abstraction différents, la comparaison entre données est possible, ce qui est en accord avec notre principe de liberté et de compatibilité de l'information.

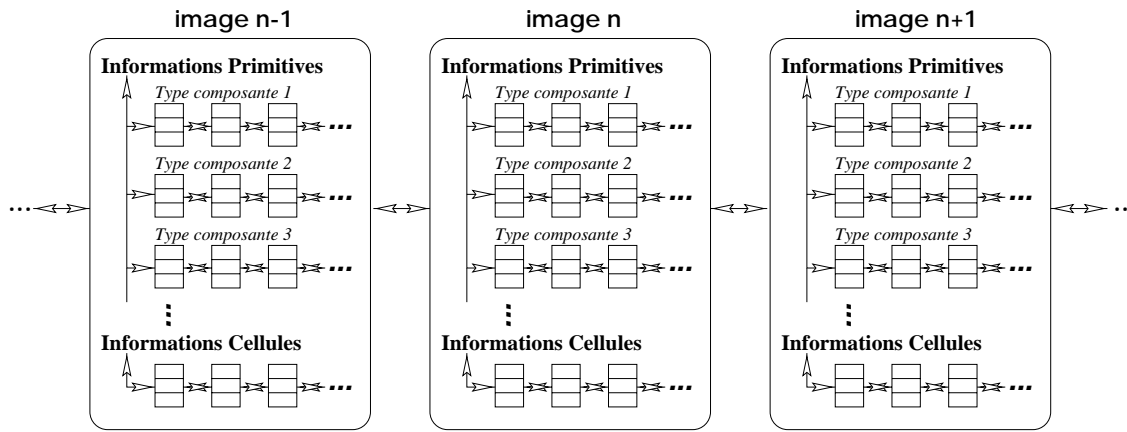


FIG. 2.14 – Organisation des structures d'informations, qui contiennent les données décrivant les objets identifiés dans les images. L'information est triée par image et par type de composante pour des raisons d'efficacité d'accès.

2.5.2 Apport de la représentation des informations

Les structures d'informations sont organisées (figure 2.14) dans le système pour pouvoir passer aisément des images de l'environnement aux informations relatives aux primitives ou aux cellules. Ces structures sont également dissociées des agents (bien que des liens existent), afin de séparer les résultats des traitements, qui doivent être accessibles à tous les agents, lesquels ont une durée de vie limitée. Les agents ont une existence temporaire pendant laquelle ils s'efforcent d'accumuler des informations qui seront utiles pour l'application ou pour les autres agents.

Chaque agent va être responsable d'une entité, c'est-à-dire d'une structure d'information et sa primitive associée. Les structures d'informations sur les cellules n'ont pas d'agents associées puisqu'elles ne sont que la mise en commun des informations contenues dans les structures décrivant les primitives régions et contours. Il n'y a pas de traitement distinct associé à la cellule.

2.5.2.1 Niveaux de représentation logique

Les différentes composantes de l'image peuvent être décrites selon différents niveaux de valeur sémantique. Cette valeur sémantique étant bien sûr définie en fonction de l'application. C'est la définition du problème qui indique la valeur qui sera apportée à chaque information. Les informations les plus proches du domaine d'application auront une valeur sémantique élevée, et les autres constructions qui n'ont de sens que pendant l'exécution auront une valeur sémantique moins élevée.

Notre système possède aussi une telle description des concepts étalée sur plusieurs niveaux. Le niveau de représentation de l'information sur les groupes d'agents, qui a déjà été évoqué, en est un exemple. Cependant, il est possible de définir d'autres niveaux pour

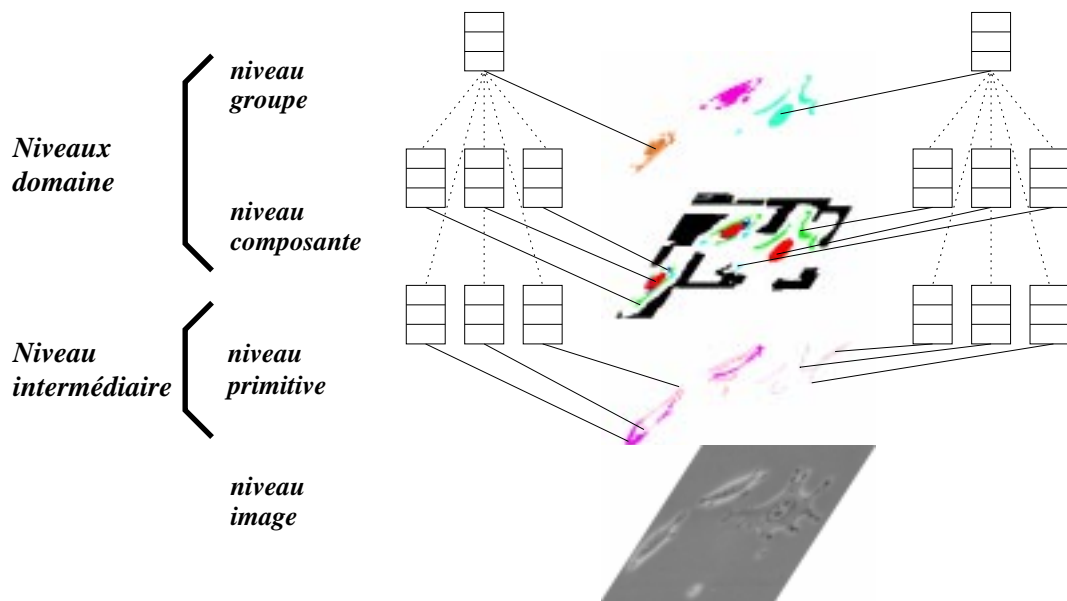


FIG. 2.15 – *Hiérarchie des différentes informations du système, de l'image à la représentation des groupes (ici des cellules). Seules quelques structures, associées à quelques primitives, sont montrés ici dans un but de simplicité. L'image du niveau cellule est reconstruite à l'aide des informations contenues dans les structures d'informations des primitives.*

différentes valeurs sémantiques associées aux primitives identifiées dans l'image.

Cependant, cette division en plusieurs niveaux n'est utile que pour la représentation des concepts manipulés et pour l'interprétation des résultats. Deux notions sont importantes ici :

1. Elle ne change pas le concept de la primitive, ou de l'agent qui lui est associé, car ce dernier travaille de façon identique quelque soit le niveau de représentation de son information.
2. Elle n'impose pas non plus de cheminement de l'information dans la société d'agents.

Trois niveaux de représentation de l'information existent dans le système (figure 2.15) : *primitive*, *composante* et *groupe*. Le niveau central, ou pivot, de notre système est le niveau composante. Le dernier niveau représente le groupe, soit la cellule dans notre application. C'est un concept particulier dans le système, qui est directement relié à l'application, et qui n'est rattaché à aucun agent. Il sert de support aux autres niveaux en regroupant les informations des agents actifs sur un même groupe. Seul les deux niveaux supérieurs possèdent un sens pour le domaine d'application, puisqu'ils regroupent des concepts définis en fonction de celle-ci. Aux niveaux intermédiaires et composantes correspondent deux images de segmentation, soit une par niveau de segmentation (paragraphe 2.3.1.5).

2.5.2.2 Hiérarchisation logique vs organisation planaire

Il est courant dans les systèmes de vision de retrouver une hiérarchisation des concepts, allant des primitives de base issues de l'image aux objets plus complexes ayant un sens précis pour décrire la scène observée. Cependant, la hiérarchisation des informations ne doit pas mener nécessairement à la hiérarchisation des traitements et à leur cloisonnement. Représenter les concepts à différents niveaux d'abstraction a du sens principalement pour l'observateur externe du système, pour lui permettre de mieux percevoir l'action de ce système. Mais pour un système de vision qui manipule des informations, il est important de lui assurer une certaine compatibilité entre les différentes sources d'informations, afin de ne pas restreindre ses possibilités de traitement.

Dans d'autres systèmes de vision existants, il existe une tendance à hiérarchiser l'information manipulée et à imposer un cheminement en sens unique (ascendant) de cette information. C'est-à-dire que l'information de l'image est segmentée, puis elle est interprétée, selon un cheminement séquentiel. La segmentation n'utilise que l'information de l'image, et aucunement celle de l'interprétation, et l'interprétation n'utilise que l'information de la segmentation, d'où une réduction néfaste des sources d'informations.

Dans notre modèle, l'information est libre de circuler. L'information trouvée par un agent du niveau supérieur peut servir à un agent de niveau inférieur et vice-versa. Et une information suffisamment riche et complète pourra être directement interprétée sans passer par l'étape intermédiaire. Cette liberté de circulation permet d'adapter les traitements à l'information, en fonction des buts fixés par l'application. Le but final est d'obtenir un découpage de l'image en concepts interprétables par un spécialiste du domaine d'application. Tous les indices permettant d'arriver à cette fin sont importants et doivent être partagés par tous les agents.

La représentation des différents concepts du système permet de mieux comprendre et interpréter les résultats du système. Cependant, cette hiérarchisation n'est utile que d'un point de vue logique. L'organisation réelle des différentes structures ne tient pas compte des différents niveaux possibles d'abstraction. Une autre façon d'exprimer cette idée est de dire qu'une représentation sur le plan logique ne doit pas cloisonner l'information et restreindre les possibilités de traitement et d'investigation sur cette information. La structure logique des informations n'a pas de conséquence sur le plan fonctionnel.

Ainsi, nous conservons l'idée d'une représentation des informations selon différents niveaux d'abstraction, tel qu'énoncé par Marr [Marr 82], mais nous limitons cette représentation au seul plan logique. Sur le plan fonctionnel, nous orientons l'utilisation de cette représentation vers les traitements qui devront être effectués, comme le préconise Brooks [Brooks 90]. C'est dans cette idée que nous qualifions notre approche comme étant *orientée vers la gestion des informations*.

Chapitre 3

La gestion des informations

Le but d'un système de vision est d'effectuer un travail de reconnaissance, ou plus généralement d'identifier des concepts qu'on lui présente. Cette reconnaissance, ou cette identification, peut prendre de multiples formes selon les objectifs de l'application pour lesquels il est utilisé. Pour réaliser cela, un système de vision doit effectuer plusieurs traitements, qui sont souvent regroupés en modules que l'on nommera par exemple sources de connaissances ou spécialistes selon les cas.

Nous avons tenté, dans ce chapitre et dans le suivant, de séparer les notions de traitement de celles de contrôle. Cependant, il s'est avéré difficile d'effectuer une séparation nette entre ces deux notions, car elles sont imbriquées à tous les niveaux du système, ce qui est un atout dans notre cas puisque le traitement de l'image influence le contrôle et vice-versa. Il existe une dualité et une complémentarité entre ces notions que nous avons voulu mettre en évidence en les réunissant sous le thème de la gestion des informations. Cette gestion permet au système d'effectuer la reconnaissance des objets de l'image et de guider son action lors de cette reconnaissance.

Dans ce chapitre, les éléments du système ayant trait majoritairement aux notions de traitement et de reconnaissance sont présentés. L'agent est d'abord décrit en temps que source de connaissances, puis il est subdivisé en plusieurs comportements qui effectuent les tâches de base de l'agent. Trois comportements sont décrits ici, soient les comportements de perception, d'interaction et de différenciation. Le comportement de reproduction, ayant trait majoritairement avec le contrôle de la stratégie d'exploration, sera décrit au prochain chapitre avec les éléments de contrôle du système.

3.1 La gestion des informations dans les systèmes de vision

Il existe une multitude de méthodes différentes pour gérer les informations dans les différents systèmes de vision. Les méthodes utilisées dépendent souvent de l'application qui est visée par le système (explicitement ou implicitement par les concepteurs).

Nous nous intéressons ici aux méthodes et aux concepts de gestion utilisés dans les systèmes de vision. Ces systèmes présentent des architectures différentes, qui ont influencé le mode de gestion employé. Nous avons donc regardé les différents systèmes de vision selon leur décomposition du problème, c'est-à-dire l'utilité première des modules de traitement définis¹. Trois groupes de systèmes peuvent ainsi être identifiés :

- système orienté *tâche* : le système divise un problème en fonction des différentes tâches, ou actions à accomplir ;
- système orienté *concept* : le système s'attache premièrement à identifier les concepts de l'image, les tâches à exécuter sont définies ensuite pour chacun des concepts ;
- système *hybride* : le système exploite les caractéristiques des deux premiers groupes, souvent en fonction du niveau de représentation.

3.1.1 Systèmes centrés tâches

Le système expert de Nazif et Levine [Nazif 84] est composé de plusieurs modules, qui sont autant de bases de règles distinctes raisonnant sur l'image. Chaque règle du système est équivalente à un opérateur de base. Les trois modules, régions, contours et aires, réalisent l'analyse de l'image et effectuent un certain nombre d'actions, telles que :

- pour les régions : diviser une région en deux, fusionner deux régions ;
- pour les contours : allonger un contour, fusionner deux contours, effacer un contour ;
- pour les aires : définir / sauver une aire texturée, homogène ou fermée.

Dans leur raisonnement, les règles utilisent les principes de base de la théorie gestaltiste : la similarité, la proximité, l'utilité commune, la bonne continuité et la fermeture. Ces principes sont encodés dans les différentes bases de règles utilisées. Afin de réaliser ceci, les raisonnements portent souvent sur les entités voisines de celle qui est analysée, et ce raisonnement inclut en même temps les régions, les contours et les aires. Cette coopération entre différents types d'entités est importante et permet d'augmenter le pouvoir de raisonnement de ce système expert. Cette notion de coopération sera également importante pour notre approche.

Le système expert SIGMA [Matsuyama 90] reprend une partie de ces idées, mais en introduisant une hiérarchie de systèmes experts pour mener l'interprétation des images. Chaque expert correspond à un niveau de représentation et exploite la connaissance opératoire et les raisonnements nécessaires à la reconnaissance :

- l'*expert sur le bas niveau* correspond à la connaissance nécessaire pour l'extraction de primitives ;

1. Selon les systèmes, les modules sont appelés spécialistes, experts, agents, sources de connaissances, ou autres.

- l'*expert de sélection des modèles* correspond à la connaissance sur la transformation des données image en objets haut niveau de la scène ;
- l'*expert sur le raisonnement géométrique* correspond à la connaissance pour le raisonnement spatial entre les objets au niveau de la scène.

La reconnaissance et l'interprétation des différents objets de la scène reposent sur l'utilisation d'hypothèses, qui sont générées et confirmées au fur et à mesure de l'accumulation d'évidences sur celles-ci. Cette notion se retrouvera dans notre système, par l'utilisation de méthodes incrémentales.

Un autre type de système, soit le système de planification BORG de Clouard [Clouard 95a], est basé sur des opérateurs atomiques, c'est-à-dire que chaque opérateur réalise la tâche la plus fine possible. Les opérateurs font partie d'une bibliothèque standardisée de traitement d'images nommée PANDORE. Parmi les nombreux opérateurs, on retrouve des opérateurs de classification, de détection et de segmentation de contours ou de régions, de caractérisation de primitives, de lissage ou d'amélioration d'images, de localisation, de morphologie, etc. Tous ces opérateurs travaillent à partir des informations qu'ils partagent via le tableau noir central. Ce partage, essentiel pour mener à bien les différents traitements, a déjà été identifié dans la présentation de notre approche.

Le projet VAP [Crowley 93] est un regroupement de modules hétérogènes et indépendants réalisant chacun une tâche différente de la vision. Pour chacun de ces modules, différentes approches ont été tentées. Les modules de traitements, ainsi que quelques approches réalisées², sont :

- la description 2D (un module pour l'image gauche et un autre pour l'image droite) :
 - extraction de contours et analyse de pyramides gaussiennes ;
 - algorithmes et filtrage spatiaux-temporels pour extraire l'information ;
- la description tridimensionnelle :
 - appariement stéréo et reconstruction 3D ;
- l'interprétation symbolique :
 - construction et maintenance de modèles de la scène ;
 - définition des zones d'intérêts.

2. Toutes ces approches n'ont pas été intégrées ensemble. En fait, il s'agit de modules interchangeables qui ont été testés avec le squelette d'application défini. Chaque module ayant été testé selon différents algorithmes. Certains modules ont d'abord été réalisés avec des algorithmes de base, puis changés pour des algorithmes plus avancés.

3.1.2 Systèmes centrés concepts

Le système de Bellet pour la segmentation bas niveau suit une approche orientée concept, en exploitant le principe de l'incrémentalité [Bellet 98]. Les primitives de base, région ou contour, sont construites à partir de modèles incrémentaux de croissance de région et de suivi de contour, où un processus est attaché à chaque primitive de l'image. Chaque processus prend les décisions d'agrégation au moment opportun, en rassemblant les informations nécessaires ou en demandant à d'autres processus un nouvel apport d'informations.

Les pixels qui sont ajoutés à la primitive doivent répondre à des évaluations. Ces évaluations combinent plusieurs critères afin de réunir le maximum d'informations pertinentes dans la décision d'agrégation ou non du pixel. Différents critères ont donc été développés dans ce but. La pondération de chacun des critères dans l'évaluation peut varier, selon le résultat recherché.

Pour les contours, les critères d'évaluation utilisés sont les suivants [Bellet 95] :

- la norme de gradient du pixel, qui doit être la plus élevée possible ;
- la norme locale du gradient du pixel, mesurée dans la direction du gradient, qui doit être un maximum local ;
- la continuité du contour, c'est-à-dire que la direction du gradient pour le pixel évalué est perpendiculaire à celle du contour ;
- la taille du contour, car plus celui-ci est long, plus ses informations sont considérées robustes.

Pour les régions, les critères d'évaluation sont les suivants :

- la différence d'homogénéité entre le pixel et la région, qui doit être la plus faible possible ;
- la distance du pixel évalué par rapport au centre de gravité de la région ;
- la compacité de la région, c'est-à-dire que les pixels ayant le maximum de voisins déjà agrégés sont favorisés.

Ces critères permettent de confronter les différents pixels candidats aux propriétés de la primitive et de sélectionner les plus aptes pour l'agrégation. Cette approche, a beaucoup influencé la conception de notre système, principalement sur les principes de l'incrémentalité, de la coopération et de l'adaptativité.

3.1.3 Systèmes hybrides

Le système VISIONS [Hanson 78] intègre les deux décompositions présentées, selon les niveaux de représentation. D'une décomposition dans le bas niveau orientée tâches, les schémas, définis dans le haut niveau, présentent une décomposition orientée d'abord vers les concepts.

L'analyse bas niveau de l'image, dans ce système basé sur un tableau noir, passe par des sources de connaissances. Ces sources de connaissances appliquent des traitements généraux et n'ont aucune connaissance sur l'application. Ce sont des modules exécutant des tâches particulières de la segmentation. Ces sources de connaissances sont présentes aux deux niveaux inférieurs de la représentation :

- les sources de connaissances de bas niveau effectuent des tâches telles que :
 - segmentation et extraction d'attributs pour les régions ;
 - segmentation et extraction d'attributs pour les contours ;
- les sources de connaissances de niveau intermédiaire travaillent sur les résultats bas niveau et effectuent des opérations telles que :
 - classification basée sur les attributs ;
 - organisation et groupement perceptuel ;
 - appariement de graphe sous contraintes (entre primitives et objets modélisés) ;
 - relations entre primitives ;
 - re-segmentation dirigée par le but (paramétrage avec connaissances).

Dans le système MESSIE-II [Sandakly 95], autre système basé sur un tableau noir, les sources de connaissance sont nommées *spécialistes*. Un spécialiste peut être défini selon une tâche particulière à accomplir, ou par un ensemble de tâches relatives à un concept, ce qui fait de ce système un hybride entre les deux types de décomposition. En fait, un spécialiste est une source de connaissances opératoires pour le tableau noir central. Chaque spécialiste possède un ensemble de compétences qui définissent son champ d'action. Il est défini pour un des niveaux d'abstraction de l'information du système. Un sous-problème peut donc être une tâche à accomplir ou un concept à reconnaître.

Le paramétrage des différents opérateurs de traitement d'images se fait selon le but qui leur est donné, c'est-à-dire selon les objets sémantiques qui doivent être retrouvés dans l'image. Cette recherche des objets dans l'image se fait selon le principe du plus robuste d'abord, que nous utiliserons aussi dans nos traitements de reconnaissance de l'image.

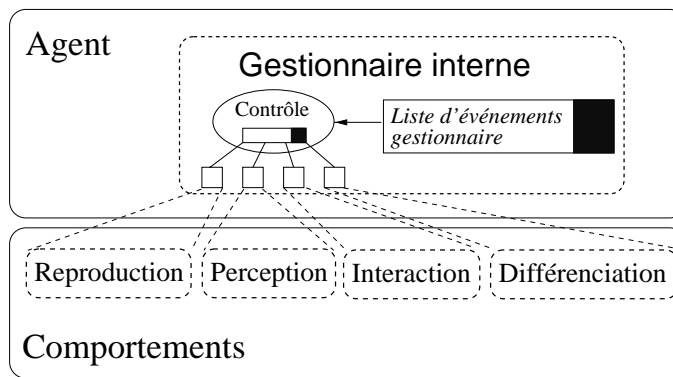


FIG. 3.16 – *Modèle général de l'agent. Un agent est rattaché à un concept de l'image. Au niveau de l'agent, la décomposition s'effectue selon les tâches.*

3.2 Stratégie orientée gestion des informations

Dans notre système, les modules de traitement sont des agents, qui constituent la décomposition première orientée selon les concepts. Un agent est, dès sa création, rattaché à un type de primitive ou de composante, qu'il aura la tâche d'identifier et de reconnaître. Au niveau de l'agent, différents comportements sont définis. Ils représentent les différentes tâches que l'agent devra accomplir dans sa reconnaissance. Il s'agit à ce niveau d'une décomposition en tâches. La figure 3.16 montre le modèle de l'agent, tel qu'il a été décrit au paragraphe 1.4, où l'on peut aisément identifier les deux niveaux de décomposition de notre système.

Tous les agents du système sont basés sur le même modèle générique, et se situent au même niveau d'un point de vue fonctionnel. Aucun agent n'a priorité sur les autres, de même qu'aucun agent ne représente la société d'agents comme une société hiérarchique. En fait, la hiérarchisation n'intervient qu'au point de vue logique de la représentation des informations (chapitre 2). Elle n'est utile que pour l'abstraction des différentes informations, mais ne doit pas entraver la liberté de circulation des différentes informations sur le plan fonctionnel. L'absence fonctionnelle de hiérarchie entre le bas et le haut niveau permet à tout agent d'avoir accès à l'image, quelle que soit sa spécialisation. Il peut ainsi être défini comme ancré dans l'image, et se trouvant toujours défini dans un contexte local.

L'agent possède les caractéristiques propres suivantes, qui ont déjà été identifiées comme essentielles dans les différents systèmes de vision présentés :

- Il *accumule* de l'information, de manière incrémentale, afin de cibler constamment ses traitements en fonction des informations déjà récoltées.
- Il *partage* ses informations avec tous les autres agents, afin d'augmenter la richesse d'informations totale du système.
- Il *coopère* avec les autres agents, de même type ou non, autant par ses actions qui

vont impliquer d'autres agents que par les informations communes qui serviront à l'adaptation mutuelle des traitements.

Ces éléments sont également présents dans les travaux de Salotti, qui a défini ainsi trois principes pour effectuer une bonne gestion de l'information [Salotti 94] :

1. La recherche de toutes les informations théoriquement nécessaires pour décider de la présence d'un indice visuel primaire est un problème majeur qui mérite une grande attention lors de la conception d'un algorithme.
2. Un modèle de description des indices visuels est suffisamment "objectif", s'il permet une exploitation efficace et une émergence d'indices visuels de plus haut niveau.
3. Pour une gestion efficace des informations, il faut préserver la liberté de mouvement de celles-ci.

3.3 Les comportements de gestion des informations

Sur les quatre comportements qui composent l'agent (figure 3.16), trois traitent directement l'information. Leur fonction est de manipuler l'environnement et les structures d'informations associées, c'est-à-dire de lire cet environnement, de traiter l'information et de produire un résultat qui sera inscrit dans ce même environnement et qui viendra l'enrichir. Ces trois comportements de gestion de l'information sont :

- *perception* : évaluation de pixels candidats pour construire une primitive, région ou contour ;
- *interaction* : fusion de primitives décrivant la même entité ;
- *différenciation* : détermination de l'identité d'une primitive, c'est-à-dire détermination du type de composante pour faire migrer l'information vers un niveau de représentation supérieur.

Le principe de base dans l'action de ces comportements est l'évaluation de l'information afin de prendre des décisions. Ces mécanismes d'évaluation sont abordés indépendamment des comportements dans le paragraphe suivant avant d'expliquer les comportements eux-mêmes dans la suite. Le dernier comportement, la *reproduction* qui contrôle la stratégie d'exploration de l'image, sera expliqué au prochain chapitre.

3.4 L'évaluation des informations

Le mécanisme d'évaluation est le principal instrument de décision des comportements. L'évaluation est spécialisée en fonction de l'agent et de la tâche qu'il a à accomplir et elle est adaptée au contexte qui prévaut au moment de la prise de décision.

3.4.1 Évaluation globale

Toutes les évaluations sont conçues selon le même modèle. Il s'agit de combinaisons linéaires pondérées de plusieurs critères. Tous les critères d'évaluation fournissent des valeurs comprises entre 0 et 1. Chaque critère possède un poids dans l'évaluation finale, qui est la somme pondérée des différentes évaluations :

$$Evaluation_{pixel} = \sum_{i=1}^n poids_i critere_i$$

où $critere_i$ représente l'évaluation du pixel selon le critère i et $poids_i$ son poids dans l'évaluation globale.

Le choix des critères, le paramétrage de ceux-ci et la pondération finale sont autant de moyens de spécialisation et d'adaptation de l'évaluation qui donnent aux agents une flexibilité d'action. Tel que déjà mentionné, tous les agents sont conçus selon le même modèle de base, qui est spécialisé ensuite pour une composante précise de l'image. Un des principaux moyens de spécialisation existant est ce paramétrage des évaluations.

3.4.2 Les critères d'évaluation

L'élément de base de l'évaluation est le critère. Chaque comportement utilise plusieurs critères distincts, et chacun de ces critères peut être paramétré en fonction de ce que l'on recherche. Les critères d'évaluation peuvent être divisés en deux catégories, selon qu'ils permettent d'évaluer les pixels, comme pour les comportements de reproduction et de perception, ou les primitives entières, comme le comportement de différenciation.

Un critère met en relation :

- une entité de base qui est évaluée, et qui peut être un pixel ou une primitive (région ou contour) ;
- une mesure, qui peut être la ressemblance, l'opposition ou autre ;
- une ou plusieurs références, constituées par des informations spatiales, des propriétés de la primitive, ou d'autres primitives, ou de l'image ;
- un contexte d'évaluation, local ou global, dépendant de la primitive, du groupe d'appartenance ou de l'image entière ;
- des paramètres, précisant le critère et son poids dans l'évaluation globale.

3.4.3 Critères d'évaluation de pixels

Deux comportements évaluent directement les pixels pour mener leur action. Il s'agit des comportements suivant :

- *perception* : l'évaluation des pixels adjacents à la primitive permet de trouver les meilleurs candidats à l'agrégation ;
- *reproduction* : bien qu'il s'agisse d'un comportement de contrôle (paragraphe 4.4), des mécanismes d'évaluation comme ceux décrits ici sont utilisés pour évaluer les pixels environnants pour trouver les meilleurs germes pour lancer de nouveaux agents.

Ces deux comportements illustrent bien la façon dont le contrôle et le traitement des informations sont imbriqués. Le même mécanisme d'évaluation de pixels qui est utilisé ici pour construire une primitive (comportement de perception) sera utilisé pour mettre en oeuvre la stratégie d'exploration et de focalisation des agents (comportement de reproduction).

Ces comportements utilisent des mécanismes d'évaluation très similaires. La structure de représentation d'un critère d'évaluation des pixels, suivant les éléments de présentation d'un critère, est composée des champs suivants :

- une entité de base qui est le pixel ;
- mesure d'évaluation :
 - le *type* de critère, spécifiant l'opération à effectuer ;
 - un *opérateur* de relation ($<$, $>$, $<=$ ou $>=$), indiquant une relation de grandeur ou un extrémum à rechercher ;
- des références qui sont :
 - l'*image* de référence, qui peut être n'importe quelle image de l'environnement ;
 - une *composante* de référence, qui est un des types de spécialisation définis par le fichier de configuration ;
- un contexte d'évaluation, défini par :
 - un *attribut* indiquant si le critère est local ou global ;
- des paramètres, qui sont :
 - un *paramètre* numérique pour configurer le critère ;
 - le *poids* du critère dans l'évaluation globale.

Tous ces champs ne sont pas obligatoires, chaque type de critère étant défini par quelques combinaisons des champs possibles. Les différents champs décrits ici illustrent la diversité des relations qu'il est possible d'établir entre différentes informations. Des exemples de critères sont donnés avec la définition de chaque comportement.

Tous ces éléments sont spécifiés dans le fichier de configuration, qui indique pour chaque spécialisation d'agent les critères utilisés pour chaque comportement, ainsi que les paramètres et le poids du critère dans l'évaluation globale. Ainsi, chaque critère possède plusieurs variantes, plus ou moins semblables, selon la spécification qui en est faite. Cela se retrouvera pour les comportements de reproduction (paragraphe 4.4) et de perception (paragraphe 3.5).

Différents contextes d'évaluation sont possibles, selon le comportement considéré. Pour le comportement de reproduction, les pixels sont évalués le long d'un profil, qui est un segment de droite perpendiculaire à la primitive (figure 4.30). Seuls les pixels le long de ce profil sont considérés et plusieurs relations sont possibles entre eux. Pour le comportement de perception, les pixels situés à la frontière de la primitive sont considérés, et ce en relation avec les autres pixels du voisinage. Il s'agit de deux contextes d'évaluation ayant des buts différents, selon qu'il s'agit de retrouver le meilleur pixel parmi d'autres (reproduction) ou de considérer tous les pixels remplissant les conditions minimales (perception).

Deux principales catégories de critères existent, soit des critères d'ordre spatial ou d'ordre photométrique. Les premiers mettent en relation les différentes entités de l'image et leurs positions relatives, comme par exemple la distance, l'inclusion entre composantes données, la proximité d'autres pixels déjà segmentés, tandis que les seconds évaluent les différentes caractéristiques des pixels en les mettant en relation, comme par exemple le seuillage, la recherche d'extrémum ou de valeurs opposées ou semblables.

L'évaluation d'un critère suit habituellement deux étapes :

1. L'étape d'*adaptation* permet de paramétrer le critère en fonction de la primitive de l'agent et/ou de l'environnement, dans le contexte et l'instant où l'évaluation se produit. Par exemple, les valeurs de seuillage, de bornes d'évaluation sont précalculées pour l'évaluation, ceci afin de permettre l'adaptation nécessaire du critère en fonction du contexte changeant dans lequel se fait cette évaluation. Cependant, il est possible que certaines valeurs demeurent constantes pour plusieurs évaluations successives avant de devoir être recalculées (pour tout un profil par exemple).
2. L'étape d'*évaluation* oppose le pixel aux autres primitives existantes ou à l'environnement local de l'agent. Cette évaluation permet de cerner une facette de l'évaluation globale (composée de plusieurs critères).

3.4.4 Critères d'évaluation de primitives

Les primitives peuvent être évaluées tout comme le sont les pixels. Seul le comportement de *différenciation* effectue une telle évaluation. Il s'agira d'une évaluation, non pas destinée à choisir telle ou telle primitive, mais plutôt à choisir une identité pour la primitive, tel qu'il sera expliqué au paragraphe 3.7.

Les critères permettant d'évaluer une primitive entière sont différents des critères précédemment évoqués. L'évaluation d'un pixel est un acte qui se produit très souvent pour un agent et l'erreur produite sur un pixel parmi plusieurs ne porte pas trop préjudice au bon déroulement de la segmentation. Cependant, pour une primitive complète, en l'occurrence ici une région, une erreur peut entraîner un résultat final incohérent. Les critères d'évaluation sur les régions comparent la région avec plusieurs autres régions obtenues par d'autres agents. Comme il s'agit d'une comparaison multiple, les temps de calcul pour ces critères sont plus importants que pour les autres critères précédemment évoqués.

L'évaluation de ces critères suit généralement 3 étapes :

1. L'étape d'*adaptation* permet de récolter un certain nombre d'informations disponibles sur les autres primitives nécessaires à l'évaluation.
2. L'étape de *condition* vérifie la présence de suffisamment d'information pour permettre une évaluation fiable du critère. En raison de l'importance de la décision sur une primitive entière, cette condition peut être nécessaire, et le critère, ainsi que l'évaluation globale, ne pourront être réalisés tant que la condition ne sera pas favorable.
3. L'étape d'*évaluation* évalue la primitive, sur certaines informations, par rapport aux autres primitives existantes. Il s'agit de relever une ressemblance, une proximité ou autre.

3.4.5 Évaluation floue

Aucune décision n'est prise lors de l'évaluation d'un critère, quel qu'il soit, il s'agit seulement d'une observation ou d'une mesure. Cette idée de mesurer un fait sans prendre de décision prématurée est importante. Il faut permettre au comportement de rassembler tous les faits avant de prendre la décision finale. Dans cet esprit, les critères ne retournent que très rarement une évaluation binaire (0 ou 1). Dans la majorité des cas, la réponse d'un critère sera un nombre réel situé dans un intervalle $[0,1]$. Le but de cette évaluation *floue* est d'effectuer une mesure plutôt que de prendre une décision.

La combinaison finale pour l'évaluation sera ainsi basée sur l'importance relative des réponses de chaque critère. Cela permet de diminuer l'importance accordée aux différents seuils pouvant intervenir dans le processus. Si chaque seuillage retourne une réponse binaire, le résultat obtenu à la fin est très influencé par le choix du seuil (une légère variation

de celui-ci peut modifier énormément le résultat), et entraîner une instabilité de l'évaluation face aux nombreux paramètres qu'elle comporte.

La figure 3.20 illustre cette idée. Un critère de seuillage est utilisé ici comme exemple, mais l'idée sous-jacente d'évaluation floue est généralisable pour la majorité des critères d'évaluation. Dans cette figure, la fonction de seuillage retourne une valeur dépendante de l'écart existant entre la valeur mesurée et le seuil utilisé pour l'évaluation. On donne ici une importance croissante à une valeur qui s'éloigne du seuil fixé. Cette fonction a pour but de diminuer l'importance relative accordée au seuil, puisque l'évaluation favorisera les valeurs dépassant largement le seuil et soulèvera un doute quant aux valeurs trop rapprochées de ce seuil.

3.4.6 Observations

Il est facile d'établir un parallèle entre cette méthode d'évaluation et les concepts de la logique floue. Elle est en accord avec les principes de base de la logique floue. Cependant, les opérateurs utilisés ici ne sont pas les mêmes que ceux existants dans la logique floue. Mais rien n'empêcherait d'utiliser les opérateurs *et-ou* flous au lieu de la sommation pondérée que nous avons énoncée pour combiner les critères entre eux.

Un autre parallèle peut également être fait, cette fois-ci avec la fusion de données. Chaque critère peut travailler avec des données de nature différente de celles utilisées par les autres critères. Certains manipulent des pixels, d'autres des primitives entières, et le tout sous différentes propriétés, tel que discuté dans les paragraphes 2.3 et 2.5. Mais nous avons tenté de simplifier les problèmes habituels rencontrés en fusion de données en choisissant une représentation commune des résultats de chaque critère, soit une réponse obtenue dans l'intervalle $[0,1]$. Chaque critère, quelles que soient les données qu'il manipule, retournera une évaluation qui pourra être combinée avec les autres.

3.5 Le comportement de perception

Le comportement de perception sert à créer la primitive contour ou région pour laquelle l'agent est spécialisé. Il permet à l'agent de construire, de façon incrémentale, une primitive en évaluant l'environnement et les différentes informations accumulées par lui et par les autres agents. Il est le comportement vital de l'agent, qui détermine sa survie, car c'est sur la construction de la primitive qu'est défini le but (implicite) de l'agent. C'est ce comportement, de par les informations qu'il récoltera, qui permettra aux autres comportements de s'activer.

Ce comportement est basé sur des algorithmes de croissance de région et de suivi de contour inspirés des travaux de Bellet [Bellet 95]. Ces méthodes ont été choisies pour leur exécution incrémentale, permettant l'échange d'informations entre les agents au fur

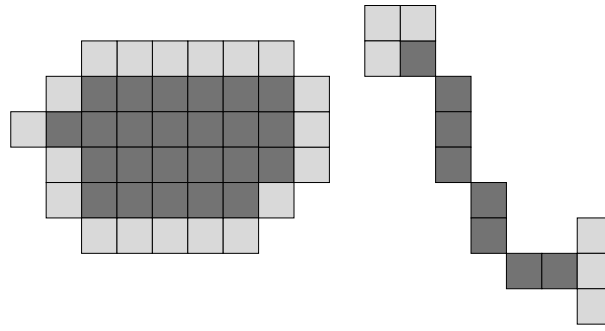


FIG. 3.17 – *Principes de la croissance de région et du suivi de contour. Pour faire croître la primitive (pixels foncés), différents candidats sont possibles (pixels clairs). Le pixel ayant la meilleure évaluation sera ajouté à la primitive.*

et à mesure du processus de construction des primitives et la possibilité de spécialiser l'agrégation des pixels en fonction des composantes recherchées.

3.5.1 Principe de l'algorithme

L'algorithme de base de croissance de primitive est le même, que cette primitive soit une région ou un contour (figure 3.17). Cependant, plusieurs différences existent dans les détails de fonctionnement de cet algorithme.

L'algorithme général est (une fois initialisé) (figure 3.18) :

- Élire le pixel le plus apte pour l'agrégation (évaluation la plus élevée).
- Si l'évaluation est supérieure au seuil d'agrégation :
 - Si le pixel n'est pas encore marqué comme appartenant déjà à un agent, alors ajouter le pixel à la primitive.
 - Mettre à jour les informations sur la primitive.
 - Mettre à jour la liste des pixels candidats à l'agrégation :
 - Évaluer les nouveaux pixels candidats.
 - Ajouter ces pixels à la liste des candidats existants.
- Itérer jusqu'à ce qu'il n'y ait plus de pixels aptes à l'agrégation.

Dans cet algorithme, il est à noter qu'un pixel est évalué avant que l'on vérifie s'il n'a pas déjà été agrégé par un autre agent. Ce choix algorithmique n'est pas sans raison, puisque c'est en voulant agréger des pixels déjà segmentés que l'agent pourra déclencher des comportements d'interaction avec un autre agent (paragraphe 3.6), que ce soit pour fusionner avec lui, puisqu'ils travaillent sur la même entité, ou pour négocier l'échange de ces pixels, auxquels tous deux sont intéressés.

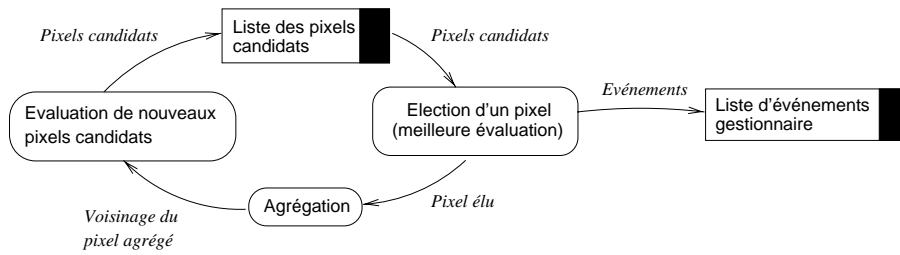


FIG. 3.18 – *Boucle de contrôle du comportement de perception. Ce comportement boucle sur une liste de pixels tant que celle-ci n'est pas vide.*

La différence majeure selon le type de primitive construite est dans la construction de la liste des pixels candidats. Pour une région, la liste sera l'ensemble des pixels situés à la frontière de la région. Lorsqu'un pixel est ajouté à la région, on évalue les 4 voisins (4-connextité) du pixel ajouté pour les insérer dans la liste. La région peut ainsi croître de tous côtés. Pour un contour, seules les extrémités sont considérées. Il y a en permanence seulement 6 pixels candidats pour l'agrégation. Trois pixels à chaque extrémités (en 8-connextité) du contour sont évalués et considérés.

L'initialisation de l'algorithme se fait sur le pixel germe qui a été fourni lors de la création de l'agent. Les voisins de ce pixel sont utilisés pour initialiser la liste des pixels candidats³.

Dans cet algorithme, les principaux paramètres à fournir sont les critères d'évaluation des pixels, qui sont spécialisés pour les différentes composantes à segmenter. Ces critères sont différents selon qu'il s'agit de régions ou de contours. Ils sont détaillés dans les paragraphes suivants.

D'autres paramètres peuvent être spécifiés, soit pour paramétrer l'évaluation (paragraphes 3.5.2 et 3.5.3) ou pour indiquer des actions à entreprendre à la fin de l'algorithme, lorsque la primitive a terminé sa croissance. Dans cette deuxième optique, deux paramètres existent :

- La *taille minimale* pour conserver une primitive : si la primitive est plus petite qu'un seuil à la fin de la croissance, il est possible de l'effacer tout simplement, ainsi que toutes les informations que l'agent a pu récolter.
- Le *remplissage des trous* (pour les régions) : à la fin de la croissance, il peut subsister quelques pixels isolés qui n'ont pas été agrégés mais qui devraient appartenir à la région. Il est possible ainsi d'effectuer un remplissage des interstices subsistants.

Plus de détails sur les méthodes incrémentales de croissance de régions ou de suivi de contours peuvent être trouvés dans [Bellet 98].

3. Pour les contours, seuls 6 des 8 voisins (8-connexes) sont conservés, soit ceux qui se trouvent perpendiculaire à la direction du gradient. Pour les régions, les 4 voisins (4-connexes) sont considérés.

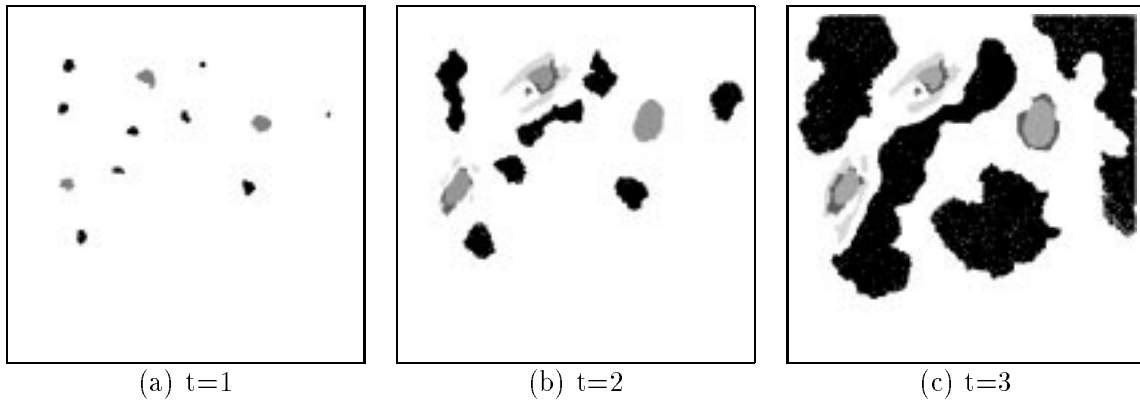


FIG. 3.19 – *Croissance de région à partir de germes initiaux. Différentes stratégies d'évaluation sont mises en oeuvre selon le type de composantes à segmenter.*

3.5.2 Les critères d'évaluation pour les régions

À partir d'un pixel germe initial, l'évaluation de pixels candidats permet de faire croître la région en agrégeant les pixels ayant la plus forte probabilité d'appartenir à celle-ci (figure 3.19).

Quatre critères d'évaluation existent pour les pixels candidats des primitives régions. Chacun de ces critères est adaptable localement et configurable pour spécialiser l'agent dans sa tâche. Ils peuvent être utilisés avec différentes caractéristiques de l'image (niveaux de gris, variance, gradient, mouvement). Ces critères sont :

1. *seuillage* : le seuil est adapté selon les paramètres intrinsèques de la région ou selon les paramètres des autres régions avoisinantes ;
2. *homogénéité* : l'écart du pixel avec la moyenne de la région (d'une caractéristique particulière) est calculé ;
3. *compacité* : il s'agit ici de favoriser les pixels ayant plusieurs voisins appartenant déjà à la région ;
4. *inclusion* : il s'agit de vérifier si le pixel est inclus dans le groupe de l'agent⁴ (situé entre plusieurs agents du même groupe).

Les nombreuses variantes de chacun de ces critères peuvent paraître difficiles à déchiffrer. L'objectif désiré était d'unifier un grand nombre de combinaisons de critères à l'aide d'un langage composé de mots-clés simples. Ces mots-clés permettent de donner, dans le fichier de configuration, les critères d'évaluation pour chaque type d'agent, en exploitant au maximum la complémentarité entre les différents types d'information, et surtout, en initiant une compatibilité entre ces informations d'origines diverses.

4. Ici, nous confondons la notion de groupement perceptuel d'informations (défini au paragraphe 2.5.1.3) et la notion de groupe d'agents, qui sont les agents associés aux primitives d'un même groupement perceptuel.

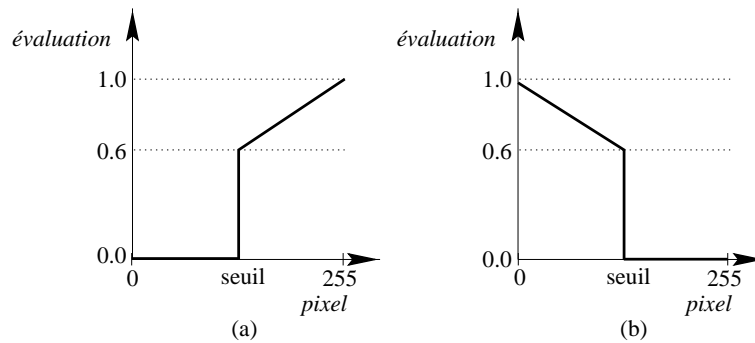


FIG. 3.20 – Fonctions d'évaluation du critère de seuillage du comportement de perception. (a) Seuillage au-dessus du seuil. (b) Seuillage en-dessous du seuil.

3.5.2.1 Critère de seuillage

Le critère de seuillage permet d'accepter ou non un pixel selon qu'il est au-dessus ou en-dessous d'un seuil calculé. La particularité de ce seuillage est dans le calcul du seuil et dans l'évolution dans le temps de ce seuil. En effet, le seuil peut être calculé selon n'importe laquelle des caractéristiques de l'environnement (image source, gradient, variance, mouvement). De plus, ce calcul est adapté en tenant compte des informations récoltées par la segmentation en cours. Comme cette segmentation évolue, ainsi que les informations récoltées sur celle-ci, le seuil évolue tout au cours de la vie du comportement.

Il existe deux méthodes de calcul du seuil :

- en fonction des caractéristiques intrinsèques à la région ;
- en fonction des autres régions de l'image, construites par des agents d'autres types.

Ces deux méthodes sont détaillées ci-après. Le seuil est recalculé à chaque fois que les informations nécessaires évoluent, ce qui permet d'agréger les premiers pixels avec une plus grande incertitude (peu d'informations disponibles) et d'affiner le processus au fur et à mesure que les agents progressent dans leur travail.

Comme le seuil évolue en fonction des pixels agrégés, il est important de choisir en premier les pixels les plus sûrs, afin de renforcer rapidement la confiance dans le seuil. C'est pourquoi la fonction d'évaluation de ce critère n'est pas binaire et permet de valoriser les pixels qui dépassent largement le seuil fixé (figure 3.20).

Seuil calculé sur la région

Cette première variante du critère de seuillage permet de calculer le seuil par rapport aux caractéristiques intrinsèques de la région. Il est configuré par un *facteur de seuillage* permettant de calculer le seuil comme suit :

$$\text{seuil} = \text{facteur} * \text{moyenne}(\text{caractéristique})$$

Le terme *moyenne(caracteristique)* signifie la moyenne de la région selon cette caractéristique. Par exemple, *moyenne(variance)* signifie la moyenne de variance des pixels de la région⁵.

Si *facteur* < 1 alors tous les pixels supérieurs au seuil seront acceptés, et inversement, si *facteur* > 1 alors tous les pixels inférieurs au seuil seront acceptés. Deux rôles sont ainsi attribuées à ce facteur (sens du seuillage et calcul du seuil). Cependant, ces rôles sont complémentaires et indissociables (sinon, on risque d'obtenir des incohérences dans le seuillage⁶).

Par exemple, on peut spécifier pour un agent dans le fichier de configuration :

Critère perception : 0.4 seuillage source 0.9

Ce critère est interprété comme un seuillage à partir de l'image source en niveaux de gris. Un facteur de 0.9 est spécifié, ce qui signifie que le seuil, recalculé au fur et à mesure de l'agrégation de pixels, sera 9/10 de la moyenne des niveaux de gris de la région. Ce critère a un poids de 0.4 dans l'évaluation finale du pixel.

Seuil calculé sur les autres composantes

Il s'agit cette fois de calculer le seuil en fonction des autres régions, définies par d'autres agents travaillant dans l'image. Cette méthode est intéressante, car elle permet de définir une région, non pas en donnant ses propres caractéristiques, mais en opposition à d'autres types de régions existantes. Ce qui permet une définition relative des régions en opposition les unes aux autres.

Cette fois, on spécifiera un type de composante et un ordre de grandeur par rapport à celle-ci (<, >, <=, >=). Une recherche est effectuée parmi les régions du type spécifié pour configurer le seuil. Cette recherche peut porter sur toutes les régions de l'image ou seulement sur les régions appartenant au même groupe, si un tel groupe est défini et qu'il comprend les types d'agents désignés.

L'opérateur de relation utilisé spécifie deux choses : le type de seuil (inférieur, supérieur) et l'information utilisée pour configurer le seuil. L'opérateur < spécifie qu'il s'agit d'un seuil supérieur (tous les pixels en-dessous de ce seuil sont acceptés) et que la valeur minimale des régions du type spécifié sera retenue et vice-versa pour l'opérateur >. L'opérateur = ajouté indique que le seuil sera la moyenne des régions de type composante et non l'extrémum.

Par exemple, on peut spécifier pour un agent dans le fichier de configuration :

Critère perception : 0.4 seuillage variance > Fond

5. Ne pas confondre cette mesure avec la variance globale de la région. La variance est calculée sur un carré entourant chaque pixel sans tenir compte de la région d'appartenance. Il s'agit plus ou moins de la variabilité des niveaux de gris autour d'un pixel. Cette caractéristique est néanmoins très employée dans notre système.

6. En effet, le seuil calculé doit toujours être compris dans l'intervalle des valeurs acceptées au seuillage, sinon le seuil perd tout son sens et la région ne pourrait pas être construite.

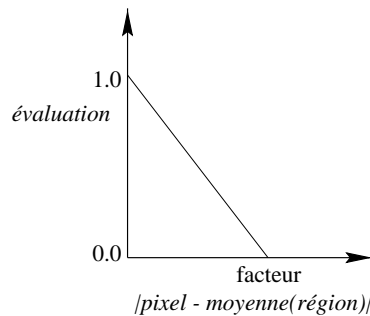


FIG. 3.21 – *Fonction d'évaluation du critère d'homogénéité du comportement de perception.*

Ce critère signifie qu'on seuillera les pixels, pour la région, tel qu'ils aient une valeur de variance supérieure aux valeurs de variances enregistrées pour les agents de type *Fond*. On ignore exactement la valeur de ce seuil, qui sera affinée avec l'évolution du travail des agents de type *Fond*, mais on peut définir une composante en disant qu'elle sera moins homogène qu'une autre. Ce critère a un poids de *0.4* dans l'évaluation finale du pixel.

3.5.2.2 Critère d'homogénéité

Le critère d'homogénéité permet d'accepter les pixels qui sont homogènes, selon une certaine caractéristique, à la moyenne de la région. Il est configuré par un facteur, donnant l'écart admissible par rapport à la moyenne (figure 3.21).

Par exemple, on peut spécifier pour un agent dans le fichier de configuration :

Critère perception : 0.7 homogénéité variance 6

Ce critère permet d'agréger tous les pixels dont le niveau de variance est proche de la moyenne de la région. Une différence de 6 entre la moyenne de la région et le pixel produira une évaluation nulle. Cela ne signifie pas nécessairement que tous les pixels dont la différence est inférieure à 6 seront acceptés, puisque l'évaluation obtenue sera soumise à un seuil d'évaluation, qui peut être plus ou moins permissif. Ce critère a un poids de *0.7* dans l'évaluation finale du pixel.

Il est à noter que la moyenne de la région, pour une caractéristique donnée, évolue au fur et à mesure de l'ajout de nouveaux pixels. La fonction d'évaluation évolue donc tout autant, en se stabilisant lorsque la région atteint une taille critique (un pixel n'a alors que peu d'influence sur les statistiques de la région).

3.5.2.3 Critère de compacité

Le critère de compacité a pour but de favoriser les pixels ayant déjà plusieurs voisins agrégés par la région. Ceci est fait pour lisser la région et surtout éviter la formation de *trous* résultants d'évaluations aux limites du seuil d'agrégation. Ce critère a pour effet

de favoriser une croissance compacte de la région et de remplir les trous. Pour évaluer ce critère, le nombre de voisins (4-connexes) appartenant déjà à la région est compté. Ce nombre divisé par quatre donne le résultat de l'évaluation.

Par exemple, on peut spécifier pour un agent dans le fichier de configuration :

Critère perception : 0.2 compacité

Ce critère, très fréquemment employé dans la définition des critères de perception, permet simplement de favoriser les pixels ayant plusieurs voisins appartenant déjà à la région. Il a un poids de 0.2 dans l'évaluation globale du pixel.

3.5.2.4 Critère d'inclusion

Le critère d'inclusion vise à favoriser des pixels qui sont situés à l'intérieur d'un groupe (inclus entre primitives de ce groupe). Il est également possible de spécifier la proximité obligatoire de régions d'un certain type.

Ce critère est calculé en identifiant le groupe auxquels appartiennent les agents⁷ se trouvant dans 8 directions autour du pixel. On note, pour chaque couple de directions opposées (4 couples au total) :

- si les deux primitives se trouvant dans les deux directions opposées (à une quelconque distance) appartiennent au même groupe ;
- si au moins une de ces deux primitives est du type indiqué (si un type est indiqué).

Le nombre de couples réunissant ces deux conditions divisé par 4 donne l'évaluation pour ce critère. Par exemple, on peut spécifier pour un agent dans le fichier de configuration :

Critère perception : 0.3 inclusion Noyau

Ce critère spécifie que la croissance de la région s'effectuera à l'intérieur du groupe auquel appartient déjà l'agent. Pour plus de sûreté, on ne comptera que les couples de directions opposées où un agent de type *Noyau* sera présent. Ce critère a un poids de 0.3 dans l'évaluation finale du pixel.

3.5.3 Les critères d'évaluation pour les contours

L'évaluation pour un contour ressemble à celle déjà décrite dans le paragraphe précédent pour les régions. Les critères de cette dernière évaluation ont été décrits en détail. Pour cette raison, et aussi parce que nous utilisons plus les régions que les contours dans notre approche (chapitre 5), seul un survol des critères d'évaluation pour les contours est donné dans ce paragraphe.

⁷. ou groupement perceptuel des primitives associées à ces agents.

Nous utilisons le terme *contour* au sens d'une structure linéaire regroupant des points d'un type donné. Ces points peuvent définir une frontière, ou plus généralement, un indice visuel linéaire (paragraphe 5.2).

À partir d'un germe initial, l'évaluation de pixels candidats aux extrémités du contour permet de le suivre en agrégeant les pixels ayant la plus forte probabilité d'appartenir à ce contour (figure 3.17). La sélection des pixels candidats peut se faire selon deux méthodes, configurées en fonction du contour que l'on recherche :

- en prenant les trois pixels perpendiculaires à la direction du gradient du pixel à l'extrémité du contour [Salotti 94, Bellet 98] ;
- en prenant les pixels perpendiculaires à l'orientation des derniers pixels du contour.

L'évaluation des pixels candidats se fait en comparant le pixel par rapport à l'une des trois données suivantes :

- les caractéristiques globales de l'image ;
- les caractéristiques du contour ;
- les caractéristiques de part et d'autre du contour, échantillonnées sur quelques pixels (trois) de chaque côté [Zhou 97].

Les quatre critères définis pour l'évaluation des pixels candidats pour un contour ressemblent à ceux qui ont déjà été définis pour la croissance des régions (paragraphe 3.5.2) :

1. *Extrémum* : il s'agit d'évaluer si le pixel est un maximum ou un minimum selon une caractéristique. Cette évaluation peut être globale (par rapport à l'image) ou locale (par rapport aux pixels de part et d'autre du contour).
2. *Homogénéité* : il s'agit d'évaluer si le pixel est homogène selon une caractéristique donné par rapport à l'ensemble du contour.
3. *Opposition* : il s'agit d'évaluer la différence existant de part et d'autre du contour selon une caractéristique donnée.
4. *Taille* : il s'agit de faciliter l'agrégation de pixels pour un contour ayant déjà acquis une certaine taille, donc jugé plus robuste.

Il n'existe pas de critère permettant la fermeture des contours. La raison est que nous avons utilisé les contours principalement comme indices visuels supplémentaires et non pour rechercher des zones fermées. Mais il serait possible de rajouter un tel critère, évaluant le pixel, par exemple, par rapport à la distance entre les extrémités afin de favoriser la fermeture des contours.

3.5.4 Configuration de l'évaluation

Il existe des variantes de configuration permettant de modifier l'action du comportement de perception. Ces variantes permettent de mieux cibler l'action de ce comportement, et particulièrement de l'évaluation des pixels, en fonction de la spécificité de certains types d'agents. Deux options de configuration d'un type d'agent sont présentées ici :

- l'adaptation de l'évaluation du comportement selon les informations du groupe auquel il appartient ;
- l'évaluation tenant compte de l'étiquetage du pixel dans l'image précédente.

3.5.4.1 Adaptation selon les informations de groupe

Pour certains critères d'évaluation (notamment le seuillage), les informations récoltées par l'agent sont utilisées afin d'adapter certains paramètres de réglage. Ces informations se retrouvent dans une structure (paragraphe 2.5). Tel que déjà expliqué, les structures d'informations sur les primitives régions ou contours ainsi que les structures d'informations sur les groupes d'agents ont une construction similaire permettant une compatibilité et un échange d'informations entre les différents niveaux qu'elles représentent (paragraphe 2.5.2).

Cette compatibilité peut être utilisée afin de permettre au comportement de perception d'adapter ses paramètres avec les informations issues de son groupe plutôt que ses propres informations. En fait, cette adaptation se fera avec les informations des agents de même type que lui et qui appartiennent au même groupe. Cela permet à des composantes assez similaires à l'intérieur d'un groupe (ou groupement perceptuel) de mettre en commun leurs informations et de les utiliser dans l'évaluation de nouveaux pixels. Cela réduit aussi le risque qu'un agent, mal initialisé, interprète mal l'environnement et construise une primitive erronée. On spécifie simplement, pour un type d'agent, si son adaptation s'effectuera selon les informations de sa primitive ou de son groupement perceptuel.

L'adaptation s'effectue par le comportement en utilisant les données contenues dans une structure d'information. Normalement, cette structure correspond à la structure décrivant la primitive sur laquelle travaille l'agent. Il est cependant possible d'utiliser comme référence (uniquement pour l'adaptation) la structure décrivant le groupe auquel appartient l'agent. L'utilisation d'une structure individuelle ou de groupe est spécifié, pour un type d'agent donné, dans le fichier de configuration. Comme la structure du groupe est mise à jour en même temps que la structure de sa primitive, il peut subir l'influence de ses actions.

3.5.4.2 Évaluation tenant compte de la segmentation précédente

L'évaluation telle que proposée jusqu'à maintenant tient principalement compte des caractéristiques statiques des images. Il existe deux informations temporelles que le système

peut utiliser :

- les images d’amplitude et de signe du mouvement, tel que déjà discuté au paragraphe 2.4 ;
- la segmentation de l’image précédente, qui peut aussi donner une piste sur les objets existant dans l’image.

La première information, issue des images directement, est déjà intégrée dans l’évaluation des pixels au même titre que les autres informations de nature statique. Mais la deuxième information, qui est issue du travail effectué par les agents dans les images précédentes, n’a pas encore été exploitée dans la description qui a été faite jusqu’à présent du système.

Cette information est plus difficile à intégrer dans notre système. La raison principale est le caractère déformable des objets de l’application qui nous intéresse. La forme d’un objet dans une image peut être très différente de la forme de ce même objet dans l’image suivante.

À partir seulement d’un pixel segmenté dans une image, il est difficile de prédire quoi que ce soit sur l’étiquetage de ce pixel dans l’image suivante. Néanmoins, en combinant cette information avec les images mouvement déjà calculées, il est possible de faire certaines hypothèses. Si le mouvement est relativement faible entre deux images pour un pixel donné, il est fort probable que ce pixel conservera la même étiquette dans les deux images. Cependant, si le mouvement est fort, aucune hypothèse ne peut être donnée. La mesure du mouvement que nous utilisons donne des indices sur les déplacements entre deux images, mais elle ne saurait être précise puisqu’il s’agit d’une différence d’images fonction de l’intensité des objets. Un objet d’intensité très variable peut donner l’impression de se déplacer beaucoup alors qu’il ne bouge presque pas (paragraphe 2.4).

Donc il est possible d’utiliser certaines informations de la segmentation précédente, mais dans certains cas précis uniquement. L’utilisation ou non de cette information est configurable selon le type d’agent considéré. Dans certains cas, l’utilisation de cette information permet d’accélérer le traitement (en évitant tout le processus d’évaluation d’un pixel) et d’améliorer le résultat, mais dans d’autres cas, cette information peut nuire à la segmentation plus qu’elle ne l’aide. Généralement, les composantes les plus homogènes de l’image peuvent tirer parti de cette information.

Un agent utilisant cette information regardera l’étiquetage du pixel dans l’image précédente, ainsi que l’amplitude mesurée de la différence (mouvement) entre les deux images. Pour l’instant, ce raisonnement n’est effectué que par les agents régions, car le déplacement des contours, plus fins, est difficile à établir dans un raisonnement pixel par pixel. Le raisonnement effectué pour l’évaluation d’un pixel tenant compte de la segmentation

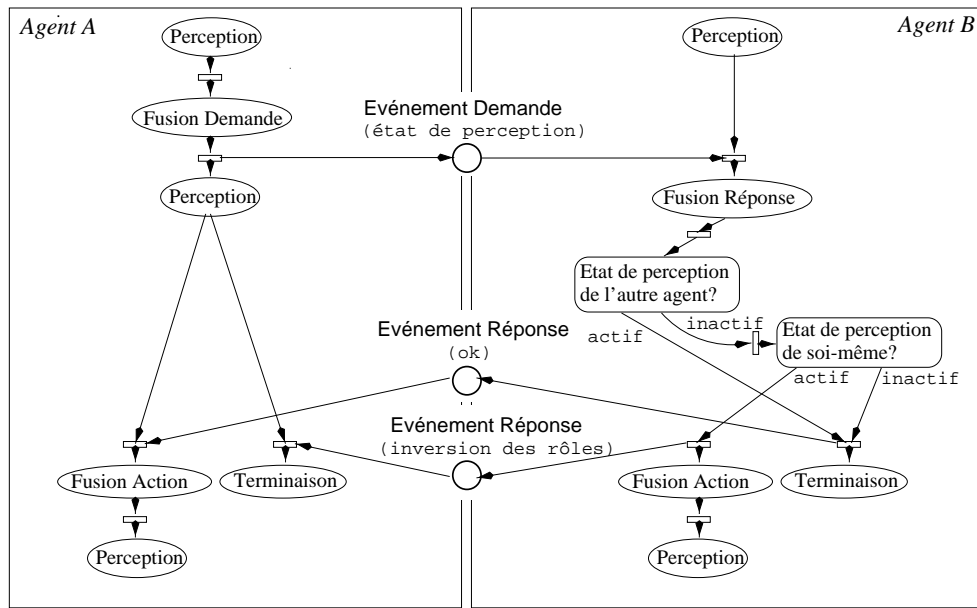


FIG. 3.22 – Communications entre deux agents en interaction (fusion) modélisées sous forme de réseau de Petri. La fusion se découpe en plusieurs sous-étapes (ou sous-comportements).

précédente est :

Si *différence faible* et *segmentation précédente identique*
 alors *évaluation = 1*.
 Si *différence faible* et *segmentation précédente différente*
 alors *évaluation = 0*.
 Si *différence forte* ou *pixel non segmenté dans l'image précédente*
 alors *évaluation selon les critères de perception*.

Cette utilisation de la segmentation précédente est la seule méthode de propagation des résultats qui existe actuellement dans le système. Il peut être jugé regrettable que les agents ne collaborent pas plus dans la segmentation des différentes images de la séquence. Deux questions n'ont pas été résolues pour mener à bien cette tâche :

- Comment propager les bons résultats en évitant de propager les erreurs accumulées?
- Comment tenir compte de la déformabilité des objets lorsque l'on travaille au niveau du pixel pour segmenter les objets?

3.6 Le comportement d'interaction

Le comportement d'interaction gère tous les processus d'interactions bilatérales entre deux agents. Il s'agit d'actions qui résultent en des communications directes entre deux

agents concernés par une situation précise.

Des interactions sont créées par les autres comportements. En fait, il s'agit, dans ces cas, d'échange, ou de partage, de différentes informations récoltées par les agents. Mais ces informations n'étant pas intrinsèques aux instances d'agents (paragraphe 2.5), ces échanges d'informations se font à l'insu des agents concernés. Nous nommons ces processus des interactions indirectes, puisqu'il y a collaboration entre agents, mais cela s'effectue de manière passive.

Dans le cas présent, pour le comportement d'interaction, il s'agit de communications clairement établies entre deux agents devant discuter d'une situation, ou d'un conflit, qui les concerne directement. Les agents ainsi réunis vont échanger des messages et devront poser une action à la suite de leurs discussions. C'est donc ce couple communication-action qui définit ce comportement.

Un seul type d'interaction est défini pour l'instant entre deux agents. Il s'agit de la fusion de leurs primitives. Cette interaction est décrite au prochain paragraphe. Ensuite, une variante permettant aux contours de se rejoindre avant de fusionner est présentée au paragraphe qui suit. Cependant, d'autres interactions sont possibles, comme la négociation de zones étiquetées par un agent. Les idées concernant cette interaction, même si elles ne sont pas complètes, sont décrites ensuite au paragraphe 3.6.3. La gestion des relations multiples que peut avoir un agent sera abordée au paragraphe 3.6.4.

3.6.1 Fusion

L'interaction par fusion permet à deux agents, qui travaillent sur la même composante, de réunir leurs primitives (région ou contour). En effet, tel qu'expliqué avec le comportement de reproduction (paragraphe 4.4), la stratégie employée par le système est de lancer plus d'agents que nécessaire, pour être certain de ne manquer aucune zone de l'image. Les agents ainsi lancés commencent par segmenter leurs composantes et vont se rencontrer au fur et à mesure qu'ils croissent. Deux agents qui se rencontrent et découvrent qu'ils travaillent tous les deux sur la même composante de l'image peuvent fusionner leurs efforts afin de diminuer le nombre d'agents redondants. Un seul des deux agents continuera le travail, l'autre se terminant pour permettre de diminuer la charge du système. La fusion ne concerne donc que des agents possédant la même spécialisation, c'est-à-dire de même type. De plus, si les deux agents appartiennent à un groupe, ce groupe devra être le même pour les deux agents pour que la fusion puisse être lancée.

La formalisation des communications entre deux agents désirant fusionner est présentée à la figure 3.22 [Ferber 95]. Les deux agents sont initialement dans des phases de croissance, c'est-à-dire avec un comportement de perception actif. Les agents évaluant ainsi les pixels candidats vont trouver plusieurs de ces pixels éligibles à l'agrégation mais déjà étiquetés par un autre agent de même type. Il s'ensuit alors une première phase de communication

(étape *FusionDemande*) où un agent demandera la fusion des deux primitives. Après analyse, le second répondra (étape *FusionRéponse*) par l'affirmative ou la négative, et la fusion s'ensuivra ou non (étape *FusionAction*). Il s'agit des trois étapes de communication permettant d'effectuer une fusion de deux primitives.

Voici un exemple d'échange de messages entre deux agents pour fusionner :

```

Agent 155  FusionRequête::exec avec Agent 168
Agent 155  FusionRequête::envoyer_requête à Agent 168

Agent 168  FusionRéponse::exec message de Agent 155
Agent 168  FusionRéponse::fusion_valide Vrai
Agent 168  FusionRéponse::réponse_partenaire Agent 155 : Fusion acceptée
Agent 168  FusionRéponse::envoyer_réponse à Agent 155
Agent 168  FusionRéponse::terminaison

Agent 155  FusionAction::exec avec Agent 168
Agent 155  FusionAction::exec exécuté

```

En théorie, un agent pourrait refuser une demande en fusion. Cependant, il n'existe aucun motif pour l'agent de le faire. Si les deux agents sont de même type et appartiennent au même groupe, alors les deux agents acceptent la fusion. Ces vérifications d'identités ont lieu par le premier agent avant qu'il ne formule sa demande.

Une question demeure pour les deux agents : lequel des deux survivra et continuera le travail. Cette question pourrait être résolue simplement en disant que l'agent initiant la conversation prend le contrôle et s'appropriera la primitive. Mais le problème est de s'assurer que la croissance de cette primitive continue après la fusion. Il s'agit donc de vérifier que les agents peuvent s'acquitter de cette tâche. Un agent ayant terminé sa croissance (et par conséquent son comportement de perception) qui fusionne avec un autre agent en pleine phase de croissance peut résulter en un arrêt de la segmentation si la perception ne se poursuit pas (segmentation incomplète).

La réponse à la question est donc de mesurer la *vitalité* du comportement de perception des deux agents. Cela vise à déterminer si le comportement de perception est apte à continuer la segmentation et qu'il ne se terminera pas avant que la fusion soit terminée (voir la gestion des comportements au paragraphe 4.5). Cette vitalité est mesurée différemment selon qu'il s'agit d'une primitive région ou contour.

Pour les régions, la vitalité peut se mesurer en observant la liste des pixels candidats pour le comportement de perception. Les deux agents sont de même type et possèdent donc la même fonction d'évaluation pour choisir les pixels. La liste des candidats étant triée, il suffit de regarder l'évaluation du pixel de tête de liste pour avoir une idée de cette vitalité⁸. Pour les contours, l'information sur les candidats n'apporte rien sur la

8. Normalement, le comportement de perception commençant à agréger d'abord les pixels ayant la plus forte évaluation, il terminera avec les pixels les plus ambigus, donc ayant des évaluations très proches du seuil minimal d'agrégation. L'évaluation du pixel en tête de liste peut donc donner une idée de la progression (et de l'achèvement) de la croissance de la primitive.

vitalité du comportement. Cependant on peut se fonder sur la longueur du contour, mais l'interprétation en reste délicate.

Dans leurs échanges de messages, les agents vont se communiquer cette mesure de leur activité. L'agent le plus apte (avec la perception la plus forte) survivra et continuera le processus de segmentation.

L'étape de fusion proprement dite demande à l'agent plusieurs actions :

1. fusion des structures d'informations et des statistiques sur les primitives ;
2. fusion des pixels candidats du comportement de perception ;
3. fusion des deux primitives (ré-étiquetage) ;
4. fusion des relations (fusion ou autres) en cours (paragraphe 3.6.4).

Jusqu'à présent, seule la possibilité où les deux agents sont vivants et actifs a été détaillée. Bien sûr, il est possible pour un agent de fusionner avec une primitive même si l'agent associé a terminé son travail. Dans un tel cas, l'envoi de message est supprimé et la fusion est exécutée dès que les vérifications d'usage sur les identités ont été faites⁹. Cette procédure peut aussi être employée si, pour une raison ou une autre, l'interlocuteur disparaît pendant que l'interaction est engagée (cette précaution est nécessaire pour éviter un blocage possible dans un système distribué).

L'utilité de ce comportement est dans le contrôle de la stratégie globale versus l'optimisation des ressources du système. C'est-à-dire que cela permet de lancer beaucoup d'agents dans l'image au début sans se soucier du besoin réel afin d'explorer en détail l'image. Ceux-ci vont s'adapter automatiquement à l'image en se fusionnant. Au fur et à mesure des itérations, seul le nombre adapté survivra et continuera la segmentation sans gaspillage des ressources (temps et système). Ce comportement permet ainsi de réguler l'activité totale du système par des interactions locales entre agents.

3.6.2 Continuité des contours

Une variante du comportement d'interaction de fusion est possible afin de permettre de fermer les contours se trouvant en prolongement les uns des autres. Lorsque deux régions croissent, elles ont une probabilité assez forte de se rencontrer et d'interagir. Cependant, les contours étant plus fins, ils ont une moins forte probabilité de se rencontrer. Il est possible que deux contours décrivant la même entité soient très proches l'un de l'autre sans se rencontrer et donc, sans fusionner.

C'est pourquoi il existe une phase supplémentaire d'interaction pour les contours permettant de rechercher les contours existants et proches, mais non contigus. En fait, le

9. En fait, pour des raisons d'homogénéité, l'agent qui n'a pas d'interlocuteur s'enverra à lui-même un message disant que la fusion est acceptée.

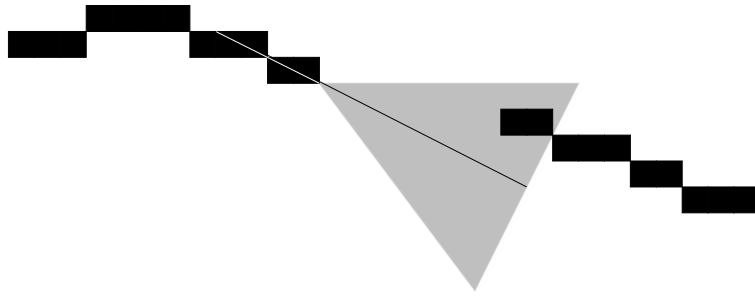


FIG. 3.23 – *Exploration à partir de l'extrémité d'un contour dans le but de le continuer et de le fusionner à un autre contour existant. Un cône de recherche est créé à l'extrémité du contour dans la direction de celui-ci. La taille du cône dépend de la longueur du contour. Si un autre contour est trouvé dans cette surface de recherche, alors les deux sont fusionnés.*

contour explore dans les directions de ces extrémités pour chercher d'autres contours de même type avec lesquels il est susceptible de fusionner (figure 3.23).

C'est à la fin de sa vie, lorsque le comportement de perception est terminé, que l'agent contour lancera cette phase d'exploration, qui fait partie de son comportement d'interaction, pour rechercher d'autres contours. Si cette recherche aboutit, alors le contour est prolongé et la fusion, telle qu'expliquée au dernier paragraphe, s'exécute.

Cette phase préliminaire de la fusion permet un double emploi de ce comportement pour les agents contours. Premièrement, la fusion classique entre deux contours peut s'opérer lorsque les primitives se rencontrent. Et deuxièmement, l'agent peut tenter de rattacher son contour à d'autres contours existants afin d'éviter un résultat avec des petits contours morcelés. Cette opération ne survient qu'à la fin de la vie de l'agent. Si cette phase de raccordement réussit, alors un nouveau comportement de perception peut commencer, sachant que la longueur d'un contour (qui vient d'être augmenté) permet de nouvelles possibilités de croissance (la taille étant un critère de perception).

3.6.3 Prémisses de négociation

La fusion est pour l'instant la seule interaction possible entre deux agents. Cette interaction vise principalement à consolider le travail accompli par les agents. Mais il n'existe pas de possibilités pour un agent de remettre en cause son travail ou le travail d'un autre agent. Dans le système actuel, les agents fonctionnent selon l'adage *premier arrivé, premier servi*. Une fois qu'un pixel est étiqueté, il n'est pas possible de changer cet étiquetage¹⁰.

Ce paragraphe présente quelques idées afin de permettre cette remise en cause de l'étiquetage des pixels. Ces idées sont des pistes permettant de concevoir un comportement de négociation entre agents de primitives région. Elles n'ont pas été implantées dans la

10. Il n'est pas possible de changer le type de composante du pixel. Seul l'identification de l'agent peut être changé lors d'une fusion, ce qui ne remet en cause que des informations de contrôle, et non les informations de segmentation.

version actuelle du système. Un tel mécanisme de correction d'erreurs est difficile à mettre en place, puisqu'il faut d'abord effectuer un diagnostic de l'erreur, donc pouvoir évaluer le résultat d'une opération, ou du moins pouvoir la juger, afin de permettre un retour arrière dans le processus de segmentation [Thonnat 93].

La différence fondamentale entre la fusion et la négociation est dans les types des agents impliqués. Dans le cadre de la fusion, les deux agents considérés doivent être de même type (et de même groupe). L'échange de pixels entre ces deux agents n'a pas de sens, puisqu'ils décrivent déjà la même entité de l'image. La négociation concerne des agents qui sont de types différents, c'est-à-dire qu'ils sont spécialisés différemment, qu'ils ne segmentent pas la même composante et qu'ils possèdent des fonctions d'évaluation différentes pour juger les pixels candidats. La décision de déclencher une interaction de fusion ou de négociation tient donc uniquement à l'identité des agents concernés.

La négociation de pixels peut se décrire selon les étapes suivantes :

1. identification de la zone en litige ;
2. évaluation personnelle de la zone ;
3. communication avec l'autre agent propriétaire de la zone ;
4. contre-évaluation de la zone par l'autre agent ;
5. comparaison des évaluations ;
6. échange de la zone ou statu quo.

Pour négocier l'appartenance de pixels avec un autre agent, il faut d'abord identifier ces pixels. Les agents, dans leurs évaluations des pixels candidats (comportement de perception au paragraphe 3.5), ne tiennent pas compte du fait que ces pixels sont agrégés ou non par un autre agent. C'est seulement lorsque la décision d'agréger ces pixels est prise que l'étiquetage du pixel est vérifié. Si le pixel appartient déjà à un autre agent, alors le type de cet agent est vérifié. Si l'agent est de même type, alors on tentera de déclencher une fusion (voir la gestion des comportements au paragraphe 4.5). Si l'agent est de type différent, alors on tentera d'amorcer une négociation sous certaines conditions.

Avant d'entamer une discussion, l'agent devra d'abord s'assurer qu'il a identifié un nombre critique de pixels à négocier. Il est impensable d'entamer la procédure complète pour chaque pixel en litige. Cela risquerait de gaspiller les ressources du système et de paralyser son action. De plus, on peut tolérer quelques pixels d'erreurs dans une segmentation.

Cette identification de la zone pose certains problèmes dans la philosophie de perception de l'agent. Normalement, un agent n'évalue que les pixels voisins de sa primitive. Ici, l'agent doit évaluer plusieurs pixels qui sortent de son voisinage immédiat. Cette notion

demande de créer une nouvelle variante du comportement de perception, que nous nommons *perception hypothétique*. Cette perception identifie une zone, à l'intérieur de la zone d'un autre agent (elle est initialisée et limitée au territoire de cet autre agent) comme une hypothèse d'appartenance à l'agent. Cette hypothèse sera confirmée ou infirmée par les discussions qui suivront. La construction de cette zone se fait à l'aide de la même fonction d'évaluation que l'agent utilise pour sa perception normale.

Une fois la zone identifiée, il faut l'évaluer, afin d'introduire une mesure unique et quantifiable servant de base de discussion aux agents. Cette mesure pourrait être la moyenne des évaluations de chacun des pixels ou quelque chose de semblable.

Le principal problème apparaît ensuite lorsqu'il s'agit de comparer les évaluations faites par les deux agents. Ces deux agents étant de types différents, ils ont des fonctions d'évaluation différentes avec des seuils d'agrégation différents. Il n'existe pas de comparaison directe possible entre les deux évaluations. Une façon de régler cette incompatibilité serait de relativiser les mesures soit au type (un ratio avec le seuil d'agrégation par exemple) ou à la primitive déjà construite (un ratio avec la moyenne de la région par exemple).

Le deuxième problème est de s'assurer de la stabilité de la zone ainsi échangée. Il faut s'assurer que les agents ne seront pas continuellement en litige pour une même zone, ce qui résulterait en un inter-blocage de ces agents. Une zone ne doit pouvoir être négociée qu'une fois par deux agents donnés. Pour résoudre ce problème, il faut instaurer des seuils plus élevés permettant la négociation d'une zone ou baliser la zone échangée comme inviolable une fois qu'un accord a été conclu.

Les deux problèmes évoqués ci-dessus n'ont pas encore de solutions concrètes permettant d'implanter ce comportement d'une manière fiable et non-bloquante. L'ajout d'un tel comportement vient modifier la philosophie du système. Sans ce comportement, il est possible de prouver que le système aboutit à une solution stable en fin de parcours. Les agents sont consommateurs d'une ressource que sont les pixels et lorsque cette ressource est épuisée, le système, et tous les agents, ne peuvent que terminer. Mais avec un comportement de négociation, une décision n'est plus irréversible. Il n'est plus évident de démontrer que le système peut aboutir à une solution stable. Il est seulement possible d'installer des mécanismes de contrôle garantissant une certaine stabilité, mais qui ne sont pas infaillibles.

3.6.4 La gestion des relations entre agents

Le comportement d'interaction décrit ainsi est basé, comme tous les autres comportements de l'agent, sur un mécanisme très réactif. Il s'agit essentiellement de réactions à des stimuli extérieurs. De ce fait, rien n'empêche un agent de répondre à plusieurs interactions à la fois, comme par exemple être en processus de fusion avec plusieurs agents tiers.

Ceci est possible grâce au mécanisme de gestion des comportements expliqué au paragraphe 4.5. Il n'existe pas de mécanismes bloquants de communication entre deux agents. Tous les échanges entre agents se font sur une base très libre sans aucune contrainte ou réglementation. Ceci est favorisé par la nature distribuée du système.

De ce fait, il existe un problème pour garantir l'identité d'un interlocuteur. Rien n'empêche un agent en cours de discussion de fusionner avec un autre et par le fait même de changer d'identité. Par exemple, l'agent A désire fusionner avec l'agent B qui lui-même est en cours de fusion avec l'agent C. La fusion entre B et C s'effectue et la nouvelle primitive est conservée par l'agent C. L'agent A a donc perdu son interlocuteur (B n'existe plus). Tout doit recommencer à zéro, au risque d'être interrompu encore par d'autres agents extérieurs.

Il faut donc trouver une méthode pour garantir un lien entre deux interlocuteurs donnés. Deux méthodes sont proposées ici, la deuxième étant retenue dans l'implantation actuelle du système pour sa plus grande robustesse :

1. Retenir un pixel (le centre de gravité par exemple) de l'autre agent. Si l'agent change d'identité, il sera toujours possible de le retracer grâce à ce pixel. Cette solution est robuste à la fusion, mais pas à la négociation, où un pixel n'est pas irrémédiablement attaché à une primitive.
2. Conserver un graphe global des relations dans le système. Cette solution, moins esthétique d'un point de vue conceptuel, est cependant plus robuste, car les noeuds du graphe peuvent être mis à jour lors de toutes fusions ou autres interactions. Ce graphe est un reflet des échanges en cours dans le système.

La deuxième solution a été retenue et même étendue à tous types de relations qui peuvent exister entre des agents, comme les relations qui peuvent exister entre un agent et un autre qu'il a créé (lien de création) (comportement de reproduction, paragraphe 4.4).

Cette solution est essentiellement utilisée à des fins de contrôle. Cependant, rien n'interdit son utilisation à des fins de gestion d'informations. Il est possible d'identifier, à l'aide de cette structure, un graphe de relations entre agents appartenant au même groupement perceptuel ou d'autres relations liant différents agents entre eux.

3.7 Le comportement de différenciation

Le comportement de différenciation permet d'interpréter une primitive intermédiaire en lui apportant un sens sémantique pour l'application. Il s'agit d'évaluer la primitive (en l'occurrence ici des régions), afin de déterminer le type de composante dont elle se rapproche le plus. Ce comportement n'est utilisé que par certains types d'agents travaillant sur des informations de niveau intermédiaire, tel qu'établi au paragraphe 2.5.2. C'est le seul comportement se préoccupant du niveau de représentation auquel est liée la primitive.

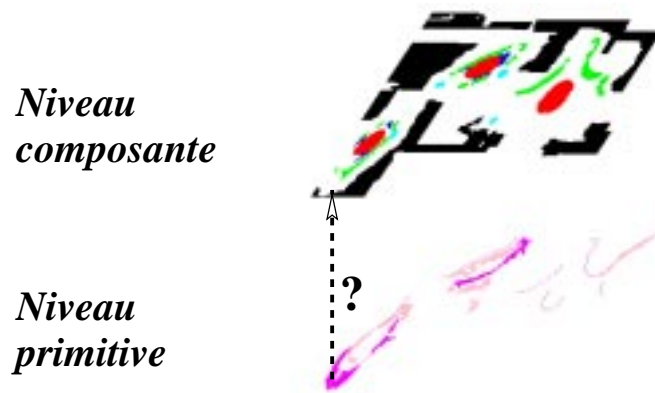


FIG. 3.24 – *Principe du comportement de différenciation. Le but est d'interpréter une primitive de niveau intermédiaire en lui ajoutant une valeur sémantique. La primitive peut ainsi migrer vers le niveau supérieur. Les informations récoltées au niveau composante serviront à effectuer cette différenciation.*

Tel que déjà décrit au chapitre 2 en traitant de la liberté de l'information, les agents travaillent sur des informations situées à différents niveaux de représentation. Ce schéma se retrouve dans beaucoup de systèmes de vision. La nouveauté de notre approche réside dans la souplesse qui existe entre ces niveaux. En effet, plusieurs cheminements d'informations sont possibles, qui n'impliquent pas obligatoirement de transiter par tous les niveaux. Une information suffisamment riche sera directement interprétée, tandis qu'une autre, plus incertaine, devra passer par les différents niveaux de résolution avant d'être interprétée, au sens de l'application (paragraphe 2.5.2). Un exemple est l'information sur le mouvement, qui pourra être traitée par des agents et être découpée en différentes zones de mouvements homogènes (paragraphe 5.2.2). Ces zones devront ensuite être interprétées et rattachées aux autres primitives existantes.

Le comportement de différenciation intervient dans cette idée de migration d'information d'un niveau à l'autre (figure 3.24). Il permet à un agent travaillant sur une primitive de niveau intermédiaire (d'un point de vue sémantique) de faire migrer son information à un niveau supérieur en s'aidant des informations et primitives existantes à ce niveau.

L'agent est configuré, par spécialisation, pour identifier une composante intermédiaire, comme les zones de mouvement par exemple. Son action est alors identique à tout autre agent, sauf lorsqu'il devra différencier son résultat, c'est-à-dire l'interpréter et le faire migrer au niveau supérieur (niveau composante). Les étapes suivies sont alors :

1. Comportement de perception : segmentation d'une primitive.
2. Comportement de différenciation : migration des résultats.
3. Comportement de perception : re-segmentation sur la base des nouvelles informations acquises lors de la différenciation.

La dernière étape peut sembler inutile a priori. Mais lors de la différenciation, la primitive acquiert une nouvelle identité. Le passage d'un niveau à un autre signifie une redéfinition du type de la primitive. Sur la base de plusieurs critères, un nouveau type lui sera donné. Et par la même occasion, l'agent associé verra sa spécialisation modifiée. Le comportement de perception peut donc être spécialisé avec de nouveaux critères d'évaluation, offrant une nouvelle interprétation de l'information acquise.

3.7.1 Principe de l'algorithme

L'algorithme du comportement de différenciation est très simple et similaire au comportement de perception déjà présenté et au comportement de reproduction qui sera présenté au prochain chapitre. En effet, il repose, lui aussi, sur une évaluation multi-critères pondérée. La nature des critères est très différente cependant, car l'évaluation ne porte pas sur des pixels, mais sur la région entière. Avant de pouvoir réaliser cette évaluation, certaines conditions, mesurant la quantité d'information disponible, sont à respecter. L'agent doit attendre que suffisamment d'informations soient présentes afin de garantir la robustesse de l'identification.

Les principales étapes de l'algorithme sont :

- Tester les conditions de présence d'informations. Tant qu'elles ne sont pas respectées, attendre.
- Évaluer les critères de différenciation selon les différents types possibles.
- Trouver l'identité de plus forte probabilité comme identité.
- Modifier l'identité de l'agent, de sa région, de ses informations pour s'accorder avec sa nouvelle identité.

Quelques paramètres permettent de configurer l'action de ce comportement :

- le redémarrage ou non du comportement de perception ;
- la liste des types possibles dans la recherche de la nouvelle identité ;
- le poids des critères d'évaluation.

3.7.2 Les critères d'évaluation

Le but de l'évaluation du comportement de différenciation est de comparer la région obtenue avec l'ensemble des régions de l'image du niveau domaine. La nouvelle identité de la région est déduite par similarité avec les différents types de régions existants. Chaque type possible (donné sous forme de liste en configuration) est évalué et comparé avec la

Identités possibles	Critères			Évaluation globale
	statistique(0.3)	intersection(0.5)	adjacence(0.2)	
<i>Cytoplasme</i>	0.4	0.3	0.4	0.35
<i>Pseudopode</i>	0.3	0.0	0.4	0.17
<i>Halo blanc</i>	0.6	0.6	0.2	0.52
<i>Fond</i>	0.1	0.2	0.2	0.17
Type choisi : Halo blanc				

FIG. 3.25 – Exemple d'évaluation pour la différenciation d'une région. Tous les types possibles sont évalués pour tous les critères. Les nombres entre parenthèses correspondent aux poids des critères dans l'évaluation globale. Cette évaluation globale est utilisée comme la probabilité pour la région considérée d'appartenir à chacun de ces types. Le type ayant l'évaluation la plus élevée est retenu pour identifier la région.

région. Le type ayant la corrélation la plus forte avec la région est choisi comme nouvelle identité de l'agent (figure 3.25).

L'évaluation de la nouvelle identité de la région se fait sur la base de trois critères :

- en comparant les *statistiques* des régions ;
- en évaluant les *intersections* avec les autres régions ;
- en observant l'identité des régions *adjacentes*.

Ces critères sont différents de ce qui a été décrit pour les autres comportements puisque cette fois, ce ne sont pas les pixels qui sont évalués, mais la région entière. Il y a donc une seule évaluation qui est faite, mais elle est plus longue à effectuer. Et comme il s'agit d'une évaluation unique par agent, celui-ci tentera de garantir les conditions optimales permettant d'obtenir le résultat le plus juste. Ces conditions sont évaluées préalablement à l'évaluation globale. Elles contrôlent principalement l'efficacité des critères statistique et d'adjacence. Tant que ces deux conditions ne sont pas satisfaites, le comportement de différenciation se met en attente. Ces conditions sont décrites dans les paragraphes suivants avec les critères d'évaluation.

Les trois critères de différenciation sont bien définis. Ce ne sont pas des modèles génériques qui doivent être paramétrés par le fichier de configuration, comme pour le comportement de perception. Seuls les poids de ces critères dans l'évaluation finale demandent à être définis. L'évaluation finale s'effectue par une combinaison linéaire pondérée des critères selon le modèle déjà présenté au début de ce chapitre (paragraphe 3.4).

3.7.2.1 Critère statistique

Le critère statistique permet d'évaluer le type, ou l'identité, de la région en comparant ses statistiques intrinsèques avec celles des autres régions (de différents types) de l'image. Les types de régions considérés sont indiqués dans la configuration du comportement par

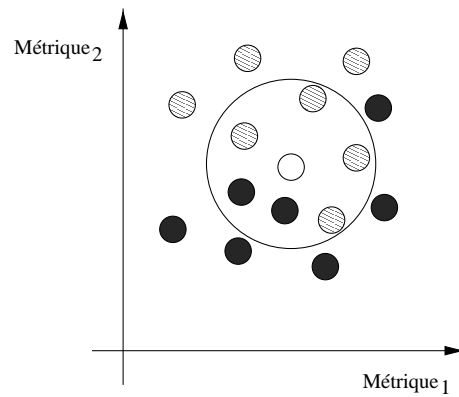


FIG. 3.26 – *Principe général de la méthode des k plus proches voisins pour déterminer la classe d'appartenance d'un nouvel individu. Outre le choix des différentes métriques utilisées, deux paramètres interviennent dans l'algorithme, soit le choix de la distance utilisée et le nombre de voisins considérés.*

une liste d'identités possibles. Chacune de ces identités est un type d'agent du niveau composante.

Cette évaluation statistique revient à faire de la classification. Chaque type d'agent devient une classe, et chaque région de ce type devient un membre de la classe représentée par ses statistiques. La classification opérée vise à déterminer la classe d'appartenance d'un nouvel individu, qui est la primitive segmentée par l'agent. Comme cette classification n'est qu'un critère parmi trois pour l'évaluation, alors une évaluation est calculée pour chaque classe. Cette évaluation devient la probabilité de l'agent d'appartenir à cette classe.

La méthode de classification employée est celle des *k plus proches voisins* (figure 3.26). Cette méthode s'applique principalement lorsque les lois des différentes distributions, ou classes, sont inconnues. Il s'agit de déterminer la classe d'appartenance d'un échantillon en observant les classes d'appartenance de ses voisins proches. Plus de détails sur cette méthode peuvent être trouvés dans [Dubuisson 90].

Les statistiques choisies pour effectuer cette comparaison sont la moyenne des niveaux de gris de la région, ainsi que la moyenne de l'écart-type des pixels de la région. Les raisons guidant ce choix sont :

- ces deux statistiques semblent discriminantes pour les classes observées ;
- un petit nombre de statistiques a été retenu afin de diminuer le temps de calcul déjà important pour ce critère.

Différents problèmes se posent à l'application de cette méthode dans notre cas :

- le nombre d'échantillon est très variable d'une classe à l'autre, certaines classes étant sur-représentées par rapport aux autres ;
- la distribution des échantillons est très hétérogène d'une classe à l'autre ;

- les différentes classes peuvent se chevaucher ;
- les échantillons à classer se retrouvent fréquemment hors de toutes classes, du fait que les informations disponibles peuvent être incomplètes à un instant donné.

Pour résoudre ces difficultés, les différents paramètres de l'algorithme ont été étudiés, ainsi que les possibilités offertes par une implantation distribuée de cet algorithme.

Deux paramètres interviennent dans cet algorithme, soit le nombre de voisins considérés et la distance utilisée. Le nombre k de voisins considérés est calculé automatiquement, en prenant le minimum entre la racine carrée du nombre d'échantillons à considérer et la taille de la plus petite classe [Dubuisson 90].

La distance utilisée entre les échantillons doit être choisie en fonction du contexte. Différentes distances sont possibles, les plus utilisées étant la distance euclidienne et la distance de Mahalanobis. La deuxième est plus complète, puisqu'elle tient compte de la distribution des échantillons dans chaque classe (covariance). Cependant, en raison des différents problèmes déjà énoncés (principalement l'hétérogénéité des classes en nombre d'individus et en distribution), le choix de la distance euclidienne s'est révélé plus justifié au niveau des résultats obtenus.

Donc, l'agent rassemble les informations sur toutes les régions des types considérés selon les statistiques retenues. Ces informations sont récupérées simplement en parcourant la structure globale contenant les informations sur les primitives de l'image (paragraphe 2.5). Seules les k plus proches régions sont alors conservées. Une évaluation de la probabilité pour chaque type d'être la classe d'appartenance est alors effectuée (figure 3.25). Cette évaluation est n/k où n est le nombre de régions d'un type particulier.

Cet algorithme est dynamique, c'est-à-dire qu'il est mis en oeuvre par tous les agents, et qu'au fur et à mesure de la différenciation des agents, de nouvelles informations viennent s'ajouter pour aider les choix à venir. Ce caractère dynamique et distribué est utilisé afin de pallier les différents problèmes énoncés.

Il est possible d'ajouter à cet algorithme des conditions de rejet, c'est-à-dire des conditions minimales à respecter pour garantir la validité du résultat. Ces conditions de rejet ont été intégrées en une condition statistique à respecter. Il s'agit cependant d'un rejet temporaire, ou plutôt d'une mise en attente, puisque les informations disponibles augmentent avec le temps dans le système. Donc, il suffit de laisser le temps aux autres agents d'accroître la quantité d'information sur leurs primitives (figure 3.27).

La condition est basée sur les deux clauses de rejet suivantes [Dubuisson 90] :

- Le *rejet d'ambiguïté* : il s'agit du nombre minimal d'éléments n que la classe majoritaire dans les k voisins doit posséder pour être validée.
- Le *rejet de distance* : c'est la distance maximale aux k voisins autorisée pour que le

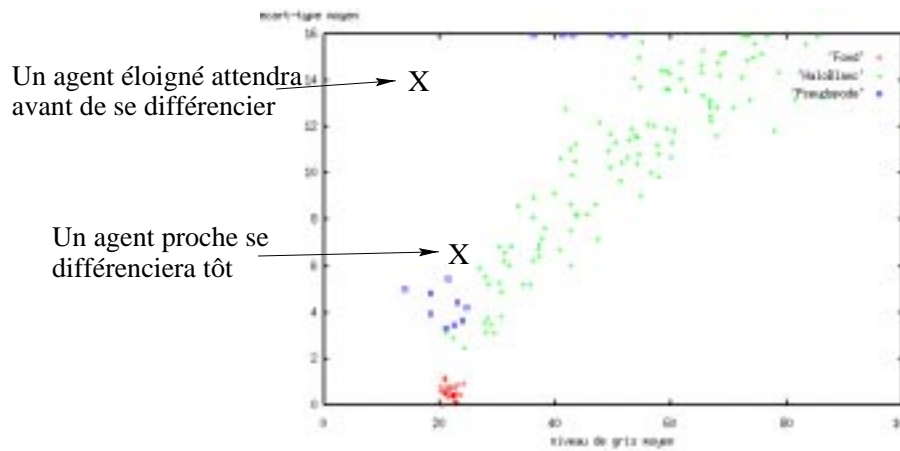


FIG. 3.27 – Illustration du principe des k plus proches voisins pour le critère d'évaluation statistique du comportement de différenciation. L'agent attendra (condition statistique) que suffisamment d'informations soient disponibles pour pouvoir évaluer fiablement son identité.

calcul soit considéré valide¹¹.

Ces deux clauses sont combinées pour obtenir la condition de rejet statistique. Cette condition doit être validée (les deux clauses validées) pour que l'évaluation puisse avoir lieu. Pour éviter que l'agent ne reste en attente indéfiniment d'une condition qui ne pourrait jamais se vérifier, les seuils de distance et de nombre sont relaxés à chaque fois que l'agent teste cette condition, jusqu'à des seuils limites.

3.7.2.2 Critère d'intersection

Le critère d'intersection permet d'évaluer le type de la région en comparant la surface de celle-ci avec les régions du niveau supérieur (niveau composante) (figure 3.24). Ceci est réalisable car, pour chaque niveau de représentation, le système dispose d'une image de segmentation associée (paragraphe 2.3.1.5). Ces images présentes à différents niveaux peuvent être observées en parallèle. Donc, pour chaque type d'agent, il suffit de calculer le rapport de surface entre l'intersection des deux images (sur la partie commune) et la superficie totale de la région évaluée (figure 3.25).

Ce critère pourrait sembler suffisant comme évaluation. Mais il doit être utilisé en conjonction avec d'autres afin de pallier les erreurs commises sur la segmentation et pour éviter les jugements basés sur des intersections trop faibles (dans plusieurs cas).

Il n'existe pas de condition préliminaire pour ce critère. En fait, la condition sur le critère d'adjacence (paragraphe suivant) permet d'assurer la fiabilité du critère d'intersection.

11. En fait, pour des raisons de simplicité, nous ne calculons que la distance moyenne de la classe la plus éloignée.

3.7.2.3 Critère d'adjacence

Le critère d'adjacence repose sur certaines règles de déduction qui permettent d'estimer l'identité de la région en observant le type des régions avoisinantes. Il est cependant plus empirique et moins fiable que les autres critères. Il est exposé ici principalement pour démontrer le genre de raisonnement déductif pouvant être effectué sur les primitives existantes.

Pour une région située sur le niveau de représentation intermédiaire, on observe, dans N directions radiales (avec $N = 16$ par exemple), les régions situées sur le niveau de représentation composante. Il s'agit de noter les régions avoisinantes dans le niveau de représentation supérieur afin d'en déduire l'identité de la région qui permettrait de la différencier et de migrer ses informations vers ce niveau supérieur.

Dans chaque direction, différentes informations comme le type de la région, sa distance et un indicateur permettant de savoir si elle est issue d'une différenciation ou non, sont notées. Ce dernier indicateur indique si la région a été trouvée par un agent de type mouvement (niveau intermédiaire) puis différenciée pour devenir une région d'un type défini au niveau composante.

Le raisonnement est fait à partir des prémisses suivantes :

- si les régions voisines ne sont pas issues d'une différenciation du mouvement, alors la région considérée a des chances d'être de même type ;
- si les régions voisines sont issues d'une différenciation du mouvement, il est probable que la région considérée soit d'un type différent.

Ce raisonnement part d'un principe voulant que deux zones de mouvement voisines représentent deux composantes distinctes s'étant déplacées l'une sur l'autre. Ce raisonnement, cohérent à première vue, pose cependant certains problèmes dans son application. Certaines régions, très hétérogènes en niveaux de gris, se retrouvent décomposées en plusieurs zones de mouvement. Donc ce raisonnement, quoique représentatif de certaines règles de logique possibles, ne doit pas constituer l'essentiel de l'évaluation permettant de différencier la région. C'est cependant le seul critère posant des règles de relations spatiales entre les différents types d'agents.

L'évaluation, pour chaque type, est la somme des évaluations des régions non-issues d'une différenciation du mouvement moins celles qui sont issues du mouvement. Chaque région est pondérée par sa distance (distance maximale observée de 10 pixels).

Tout comme le critère statistique, ce critère possède une condition préalable. Cette condition permet de s'assurer qu'un nombre minimal de régions interviennent dans le calcul de cette évaluation. Pour calculer cette condition, il suffit de compter le nombre d'agents ayant terminé de segmenter leur région (pour chaque région, on teste si l'agent associé est encore en vie). Un seuil minimal est imposé pour permettre d'entamer le calcul

de ce critère. Tout comme pour la condition statistique, le seuil est relaxé d'itération en itération afin d'éviter une boucle sans fin.

3.7.3 Caractère dynamique de l'évaluation

Pour bien comprendre comment fonctionne le comportement de différenciation, et surtout les critères d'évaluation, il faut replacer le tout dans le contexte de la société d'agents qui travaille sur une image. Plusieurs agents, de plusieurs types de spécialisation différents, s'activent à segmenter différentes composantes, et ce, à différents niveaux d'abstraction. Tous ces agents mettent en commun les informations qu'ils récoltent sur les différentes primitives auxquelles ils sont liés. Donc, ces informations sont continuellement mises à jour, au fur et à mesure que progressent les différents agents.

C'est sur cette dernière idée qu'est conçu le comportement de différenciation. Il permet à un agent de segmenter une primitive de niveau intermédiaire, c'est-à-dire une primitive qui peut être segmentée, qui existe comme entité ou partie d'entité dans l'image, mais qu'il ne peut identifier tout de suite. Un exemple, le plus commun, est l'identification de zones de mouvement dans l'image. Il est possible dès le début d'identifier de telles zones, et elles donnent de très bonnes indications sur les limites des différentes composantes de l'image, mais il est difficile, voire impossible, de les identifier (interpréter au sens de l'application) en début de traitement.

L'identification du type de composante, n'est possible que lorsque d'autres agents ont récolté des informations sur l'image. Il est alors possible de comparer la primitive inconnue avec les autres primitives qui ont été identifiées par d'autres agents. Toute cette démarche est dynamique, dans le sens où tous les agents travaillent en parallèle.

L'agent désirant *différencier* (terme utilisé ici pour *interpréter*) sa primitive scrute l'image et les informations récoltées par les autres agents. Il attend que suffisamment d'informations soient disponibles afin de poser un jugement fiable. D'où l'introduction des conditions d'attente préalables à l'évaluation dans le comportement. Elles visent à s'assurer d'une fiabilité minimale dans l'évaluation qui sera portée. Comme ce comportement ne pose qu'une seule évaluation, décisive dans l'avenir de cette primitive, il passe d'abord par une phase d'observation avant de réaliser son évaluation. Cette démarche s'inscrit dans le cadre de la gestion des informations déjà décrite au chapitre 2.

Mais la dynamicité décrite ne s'arrête pas là. Une fois que l'agent a posé un jugement sur l'identité de sa primitive, il inscrit cette identité dans toutes les structures associées (images de segmentation et structures d'informations). Cette primitive, nouvellement identifiée, va maintenant enrichir le bassin d'informations de haut niveau disponibles aux autres agents. Chaque primitive différenciée permet ainsi de différencier d'autres primitives. Dans la figure 3.27, c'est cette notion qui est représentée, où un agent ayant suffisamment d'informations *proches* se différencie, tandis qu'un autre attendra que les informations se

rapprochent de lui avant d'en faire autant.

3.8 L'adaptation comme interaction

Les comportements décrits dans ce chapitre utilisent les différentes informations du système pour adapter leurs traitements et pour prendre des décisions. Deux types d'informations sont utilisés à ces fins :

- les différentes images de l'environnement, c'est-à-dire les informations de base sur l'image et ses caractéristiques ;
- les différentes informations construites par les agents, situées sur le même niveau de représentation ou sur un niveau différent, qui en construisant leurs primitives, enrichissent le potentiel d'informations disponibles.

Ces deux cas illustrent deux sortes d'interactions différentes, de types *agent-environnement* et *agent-agent*. Ce sont par ces interactions, même indirectes, que les agents peuvent adapter leurs différents comportements au contexte dynamique qui prévaut lors de l'exécution du système.

De même, l'adaptation peut se situer à différents niveaux :

- au niveau des pixels, où une méthode est adaptée selon les pixels environnants (par exemple, l'évaluation d'un pixel candidat pour un contour par le comportement de reproduction, paragraphe 4.4) ;
- au niveau de la primitive, où une méthode est adaptée selon les statistiques intrinsèques de la primitive (par exemple, les critères de seuillage du comportement de perception qui se basent sur les moyennes des caractéristiques de la primitive) ;
- au niveau du groupement perceptuel, où une méthode est adaptée en fonction des statistiques réunies des différentes primitives d'un groupe (par exemple, les mêmes méthodes de seuillage du comportement de perception peuvent être basées sur les statistiques des primitives d'un même type appartenant au même groupe) ;
- au niveau d'un ensemble de primitives, sans notion de groupes, où les méthodes sont adaptées à l'aide de toutes les informations disponibles sur un ou plusieurs types d'agents (par exemple, le comportement de différenciation utilise dans ces critères les informations disponibles sur tous les autres types d'agents pour interpréter une région).

Il ne s'agit pas seulement d'utiliser des informations pour en abstraire de nouvelles et ainsi monter dans la hiérarchie des niveaux de représentation. À un même niveau de représentation, des informations, de types différents, sont dépendantes l'une de l'autre.

La construction des primitives étant incrémentales, ces informations se construisent petit à petit. Au début, elles sont moins fiables, parce que bien incomplètes, mais au fur et à mesure de l'exécution du système, ces informations se renforcent et leur utilisation devient de plus en plus fiable. Cette incrémentalité est importante, car deux primitives peuvent dépendre l'une de l'autre. Le premier apport, même incomplet, d'informations permet de construire ces primitives en parallèle et de raffiner les informations qu'elles apportent au système.

Cette dépendance entre les différents agents, pour leurs adaptations comme pour leurs traitements, est une forme de coopération indirecte entre ces mêmes agents. À travers cet échange indirect d'informations, les agents interagissent ensemble et s'appuient les uns sur les autres.

Chapitre 4

Le contrôle du système

Ce chapitre expose les mécanismes de contrôle qui régissent notre système de vision. Ce contrôle permet de réguler la stratégie de résolution et d'interprétation de l'image, par différents mécanismes de décision et d'enchaînement des différents modules existants. Plusieurs façons d'aborder ce problème sont possibles, et certaines d'entre elles sont présentées à partir d'exemples sur des systèmes de vision.

Dans notre système, le contrôle repose sur plusieurs niveaux. La stratégie d'exploration des images est répartie dans tous les agents par la définition d'un comportement de reproduction. Au niveau de l'agent, la gestion des comportements est confiée au niveau supérieur de l'agent et s'effectue de façon réactive aux événements qui surviennent. Elle définit l'articulation entre les fonctions de base du système de vision (perception, reproduction, interaction et différenciation) selon un mode opportuniste et asynchrone. Enfin, au niveau supérieur du système, un séquenceur régule l'ordonnancement global des agents, sans véritable structure décisionnelle ni connaissances a priori du domaine d'application. Ce contrôle est réduit à un contrôle d'exécution aveugle. La configuration des agents du système est également présentée, avec la définition des liens de la société d'agents existants par la spécialisation de ces agents.

4.1 Le contrôle dans les systèmes de vision

Le contrôle, c'est-à-dire le choix à chaque pas de la résolution de l'entité à activer, est un problème central dans tous les systèmes. Il consiste pour un système en l'instauration et la modification de son fonctionnement, de son état et de celui de l'environnement [Boissier 93]. Différentes approches existent pour le contrôle d'un système de vision. Il est possible de qualifier ces approches selon plusieurs critères :

- centralisée ou décentralisée ;
- ascendante ou descendante ;

- planifiée ou opportuniste ;
- guidée par les événements ou par les requêtes ;
- ...

De plus, le contrôle dans les systèmes est souvent présent à plusieurs niveaux, puisqu'il faut coordonner l'ensemble des modules de traitement ainsi que les actions des modules eux-mêmes. Bien souvent, les systèmes présentés ici réunissent plusieurs aspects antagonistes d'un contrôle au sein de leur architecture afin de gagner en polyvalence. Il est rare de trouver un système pouvant, sans ambiguïté, être classé selon les critères énumérés. En fait, il faut observer l'ensemble de la stratégie mise en oeuvre pour comprendre comment se gère un système de vision.

Nous avons choisi de classer les différents systèmes de vision selon le premier critère, c'est-à-dire selon que les éléments de contrôle, regroupant les règles et les données nécessaires pour réguler l'activité des modules de traitement, sont réunis au sein d'une même unité (contrôle centralisé) ou réparti dans le système parmi plusieurs entités, voire au sein des modules de traitement (contrôle décentralisé). Ce choix, bien qu'arbitraire, permet de saisir la liberté d'action qui est attribuée aux différents composants de chaque système.

4.1.1 Contrôle centralisé

Dans le système de Nazif et Levine [Nazif 84], les différentes règles de décision utilisées sont classées en trois niveaux, le premier étant lié aux opérateurs de traitement d'images et les deux autres au contrôle du système :

- les règles de connaissance bas niveau effectuent l'analyse de l'image ;
- les règles de contrôle gèrent les zones d'intérêts ainsi que le séquençement et les priorités données aux différents modules ;
- les règles de stratégie permettent de choisir le type de règles qui seront appliquées prioritairement selon des mesures sur les différentes entités de l'image.

Les règles de contrôle permettent de guider l'action des différents modules d'analyse en privilégiant des zones d'intérêt et des primitives comme cible d'action. Elles contiennent aussi des métarègles permettant de coordonner l'ensemble des actions des différents modules en choisissant les différentes bases de règles qui seront appliquées (ou modules prioritaires dans l'analyse).

Les règles de stratégie réorganisent les bases de règles qui sont utilisées. Elles permettent de privilégier certains types d'actions, comme par exemple la fusion ou la division d'entités, en fonction de certaines caractéristiques mesurées sur l'image.

Il s'agit d'un contrôle centralisé par les deux niveaux supérieurs de règles, même si les actions qui sont effectuées par la suite par les différents modules d'analyse sont très locales, et adaptées aux différents contextes.

Le système expert SIGMA [Matsuyama 90] utilise un contrôle semblable par règles et méta-règles, mais le plan d'interprétation est fixé et défini par deux phases de traitement :

1. Une première segmentation est effectuée pour générer des primitives qui seront étudiées ensuite par les experts des niveaux supérieurs. Il s'agit d'une phase ascendante, c'est-à-dire dirigée par les données afin de déclencher les premières étapes de la reconnaissance.
2. Ensuite, la deuxième phase est descendante, c'est-à-dire dirigée par les buts. Les experts de plus haut niveau génèrent des hypothèses sur la scène et envoient des requêtes de segmentation et de reconnaissance dans le but de vérifier ces hypothèses.

À chaque objet modélisé correspond une série de règles permettant le déclenchement d'actions lorsque cet objet est détecté et instancié. Il s'agit d'une stratégie de résolution axée sur l'information détectée. La connaissance opératoire nécessaire pour cela est regroupée avec la connaissance descriptive de l'objet modélisé.

Le système MESSIE-II [Sandakly 95] illustre quant à lui l'exemple d'un contrôle centralisé par un tableau noir. Ce tableau noir gère et centralise toutes les informations du système. Il est responsable du bon déroulement du système. Deux mécanismes de communication avec ce tableau noir existent :

- Les *requêtes* permettent aux spécialistes de communiquer avec le tableau noir. Par ces requêtes, les spécialistes d'un niveau peuvent commander ceux des niveaux inférieurs, selon un modèle de contrôle hiérarchique descendant. Il s'agit de la méthode la plus utilisée.
- Les *événements* permettent de déclencher les spécialistes (par l'intermédiaire du tableau noir) d'une manière opportuniste selon un mode ascendant. Ces événements proviennent de modules extérieurs ou du langage de programmation utilisé.

Tous les spécialistes définis pour l'application signalent leurs compétences au tableau noir qui se charge par la suite de sélectionner les différents spécialistes en fonction des tâches à accomplir. Un spécialiste dit *généraliste* est lancé lorsqu'il n'y a plus de requêtes pour éviter le blocage du système.

C'est le spécialiste de la scène, situé au plus haut niveau, qui dirige essentiellement la planification et le séquençement des actions dans le système. Il contrôle les informations récoltées sur l'interprétation de l'image, et c'est par ses requêtes que les autres spécialistes vont se déclencher et agir. Il s'agit donc d'un contrôle centralisé et hiérarchisé, mais conservant un caractère opportuniste dans le déclenchement des spécialistes.

Il existe deux types de stratégies, dans MESSIE-II :

- La *stratégie d'interprétation* est commune à toutes les applications. Elle est définie par le spécialiste de la scène qui est situé au plus haut niveau du système. Il a la charge de gérer l'interprétation et la représentation symbolique de la scène. Il consiste en trois actions :
 1. détection des objets sémantiques (requêtes à d'autres spécialistes) ;
 2. validation des objets détectés en leur attribuant des coefficients de certitude ;
 3. résolution finale des conflits pour lever les ambiguïtés d'interprétation.
- Les *stratégies de détection* d'objets sémantiques sont dépendantes des différents contextes d'application. Plusieurs types de spécialistes peuvent exister à cette fin. Ils sont situés aux deux niveaux inférieurs de la représentation et consistent en :
 - un spécialiste minimum, défini par type d'objet sémantique de l'application, et qui possède la stratégie de détection de cet objet ;
 - des spécialistes de groupement perceptuel ;
 - des spécialistes de bas niveau, qui consistent en des algorithmes de traitement d'images pour la détection de primitives.

Toutes les connaissances opératoires, c'est-à-dire les compétences des différents spécialistes définis pour une application, sont modélisées et fournies a priori au système. Cette modélisation des connaissances opératoires doit aller de pair avec la modélisation des connaissances descriptives, c'est-à-dire des objets sémantiques présents dans l'application. Cependant, la connaissance opératoire est détenue par les spécialistes, tandis que la connaissance descriptive est contenue dans la modélisation des objets.

Dans un autre contexte, les approches à base de planification sont souvent centralisées, comme par exemple le système BORG de Clouard [Clouard 94]. Dans ce système, le but est de construire un graphe d'opérateurs, donc de traitements à effectuer pour analyser une image, principalement dans le cadre de la vision bas niveau. La construction de ce graphe d'opérateurs s'effectue avec l'aide de l'utilisateur, car c'est lui qui donne une requête et les connaissances associées, et qui évalue les résultats fournis par le système.

La planification, ou la création d'un prototype de programme pour la réalisation d'une application type, est définie comme l'enchaînement structuré sous forme de graphe de différents opérateurs de la librairie. Les données de traitement sur les différentes entités sont conservées dans un tableau noir global au système.

Cette planification s'effectue selon un mode descendant, entièrement guidé par la requête, c'est-à-dire que celle-ci est raffinée pour être traduite en objectifs, puis en fonctionnalités, en procédures, et enfin, en opérateurs, qui sont les modules de traitements spécifiques. De niveaux en niveaux, certaines connaissances sont nécessaires afin de décomposer

les informations fournies en terme de données plus spécifiques. Au besoin, l'utilisateur peut compléter certaines points d'informations manquants, mais le but initial de ce travail est de réaliser un système possédant toute la connaissance nécessaire pour effectuer de tels raffinements dans le plan de traitement.

Il existe également des approches de planification basées sur les données. Dalle et al. ont développé une approche où l'objectif est spécifié en terme de traitement d'image [Dalle 98]. Plutôt que de formaliser les opérateurs et d'introduire une expertise pour la sélection et l'adaptation de ces opérateurs, ils ont préféré représenter les effets des opérateurs sur les données initiales. Ainsi, il est possible de concevoir des mécanismes d'adaptation et de contrôle d'exécution des plans générés.

Clément et al. ont conçu l'outil OCAPI (Outil de Contrôle Automatique de Procédures Images), utilisé en traitements d'images [Clément 93b]. Cette approche permet le pilotage automatique de programmes. Les auteurs insistent sur la nécessité d'entrelacer planification et contrôle d'exécution, afin de garantir la robustesse du système face à un environnement changeant, non prédictible, et à des actions dont la réussite est variable.

4.1.2 Contrôle décentralisé

Par opposition à MESSIE-II, le système VISIONS [Hanson 78], également basé sur un tableau noir, effectue un contrôle décentralisé, puisque réparti entre les différents schémas du système. Chaque schéma porte en lui les données nécessaires au contrôle. Mais il subsiste un tableau noir global afin de pouvoir adresser différentes requêtes entre les schémas. Ces requêtes sont adressées à tous par cet intermédiaire. Les autres schémas consultent ces requêtes et répondent à celles qui peuvent les concerner.

Une double stratégie d'interprétation est utilisée dans VISIONS. Dans les niveaux inférieurs de représentation, la stratégie est essentiellement ascendante, guidée par les données, donc par les informations qu'on trouve dans les images. Aux niveaux supérieurs, il s'agit d'une stratégie descendante, guidée par le contrôle, provenant des schémas qui indiquent les instances à rechercher.

Les conflits éventuels sont inscrits au tableau noir global. Mais les différentes stratégies de résolution de conflits sont réparties dans les schémas, et elles sont donc adaptées aux différentes instanciations possibles des schémas.

Ce sont les schémas (haut niveau), implantés de manière distribuée, qui contiennent les stratégies et les connaissances opératoires permettant de reconnaître les différentes entités modélisées pour l'application. Chaque schéma est indépendant dans la recherche d'une entité, ou d'un concept de l'image. L'interprétation se fait en générant des hypothèses, qui sont par la suite confirmées ou infirmées par l'activation des sources de connaissances

appropriées. Un schéma possède plusieurs stratégies, qui sont des procédures, ou enchaînements de procédures, pour mener son action, comme par exemple :

- détection d’hypothèses initiales ;
- extension du raffinement d’un objet à partir de son hypothèse ;
- support d’une hypothèse par la recherche d’indices sur les relations spatiales ;
- vérification de la consistance de l’objet, établie sur l’absence de composantes ou l’incohérence d’autres composantes ;
- gestion des conflits avec d’autres schémas ;
- activation d’autres schémas pour la détection des sous-parties.

Au niveau du schéma se trouve une stratégie de gestion des hypothèses sur les objets, qui est responsable de la coordination des hypothèses à l’intérieur du schéma, de l’activation de nouvelles stratégies en fonction des données des différents tableaux noirs (global et local). C’est cette stratégie qui gère l’activité de l’agent, par activation des autres stratégies secondaires.

La connaissance opératoire est directement codée dans les procédures des différentes stratégies. Ces stratégies sont spécialisées, procéduralement, selon les objectifs pour lesquels le schéma a été défini et spécialisé. Les connaissances descriptives et opératoires sont réunies dans les différents schémas.

Dans le projet VAP [Crowley 93], tous les modules sont indépendants les uns des autres. L’intégration de ces différents modules a été possible grâce à la définition d’un *squelette* commun, donnant à chaque module des interfaces communes, ainsi que des moyens de communication et d’échange. Ce squelette, nommé SAVA (Squelette d’Application pour la Vision Active), a servi de base d’intégration pour les composantes développées par des laboratoires différents.

Un des problèmes majeurs du projet VAP a été l’intégration de modules ayant chacun des modélisations très différentes de l’environnement, allant de la description numérique de l’image à la description symbolique de la scène. De plus, ce système devant fonctionner en continu et en temps réel, le problème du cycle de mise à jour des informations, qui est différent d’un module à l’autre, s’est posé. Pour résoudre ce problème, tous les modules sont conçus selon le modèle de prédiction, confrontation et mise à jour des modèles. Chaque module effectue ainsi la construction de modèles utilisables en tout temps.

Il a également été défini les interfaces communes de communication et d’échanges d’informations entre les modules. À la base, chaque module devait communiquer avec les modules situés immédiatement en-dessous et au-dessus de lui dans la hiérarchie des niveaux de représentation. Cependant, ce mécanisme s’avéra trop contraignant pour la réalisation

du système, et un deuxième canal de communication commun à tous a été ajouté, permettant des échanges tant au niveau du contrôle qu'à celui des données. Pour le contrôle, les messages échangés sont principalement des requêtes de commande des modules de niveaux inférieurs (orienté buts). Il existe aussi une communication d'événements remontant au module de supervision pour relater des situations imprévues ou particulières.

Un exemple de contrôle entièrement décentralisé, mais dans un cadre bas niveau, se retrouve dans le système de Bellet [Bellet 98]. Il s'agit d'un système multi-processus, qui est cependant initialisé à partir de quelques germes trouvés selon des critères globaux. Par la suite, le contrôle de ce système est entièrement décentralisé au niveau des processus.

Il existe la capacité pour un processus de lancer de nouveaux processus, de même type ou de types différents, tout autour de lui. En fait, ce mécanisme de lancement intervient lorsqu'un processus manque d'informations sur son environnement pour prendre des décisions. Alors, il fait appel à d'autres processus, qu'il initialise lui-même, pour récolter de nouvelles informations sur les primitives environnantes.

La coopération dans ce système est définie comme orientée contrôle, c'est-à-dire qu'elle est basée sur la gestion des échanges entre deux ou plusieurs méthodes, et sur l'intégration du travail de ces mêmes méthodes (le processus père récupère le travail de ses fils). Deux types de coopération existent :

- Une relation se crée entre un processus *père* et des processus *fils* qu'il a engendrés dans le but de rechercher plus d'informations sur les primitives de son environnement local.
- Une communication s'établit entre deux processus voisins dans l'image, qui peuvent envisager de fusionner ou d'échanger des informations.

Le premier type de coopération est celui qui est le plus employé, car c'est de cette coopération que naît toute la stratégie d'exploration et de résolution de l'image. En effet, les premiers processus germes vont créer les processus suivants en invoquant des demandes d'informations. Et ainsi de suite, jusqu'à ce qu'il se crée une véritable hiérarchie de processus, où chacun est responsable de récolter, par son action de segmentation, des informations pour un autre processus. Il existe des liens entre un processus *père* et ses *fils*. Un processus en attente d'informations peut arrêter son exécution le temps que les informations nécessaires soient trouvées.

Il n'existe pas de connaissances spécialisées, tant descriptive qu'opératoire, sur l'application ou l'image segmentée par le système. Cependant, les processus peuvent être spécialisés, principalement par des paramètres de leur fonction de décision, ce qui peut permettre d'adapter le système pour différents types d'applications.

4.2 Le contrôle dans les systèmes multi-agents

Le problème de la vision étant naturellement distribué, tant au niveau physique que conceptuel, il est important de définir le contrôle pour ces approches distribuées, particulièrement pour les systèmes multi-agents ayant été utilisés pour des applications de vision.

Durfee et al. décrivent trois mécanismes pouvant améliorer la cohérence d'un réseau d'entités dont le contrôle est distribué [Durfee 87] :

1. la planification ;
2. l'organisation ;
3. l'échange de méta-informations.

La planification, selon les systèmes, peut être centralisée ou répartie, fixée a priori ou dynamique. L'organisation peut se définir comme une structure externe aux agents ou comme un objet abstrait réparti sur plusieurs agents. Quant aux méta-informations, elles permettent aux agents de se synchroniser en échangeant des données de contrôle. Selon les systèmes, un ou plusieurs de ces mécanismes peuvent se retrouver.

Guessoum a proposé un modèle d'agent hybride qui allie des capacités cognitives, pour raisonner sur des situations complexes, à des capacités réactives et adaptatives, pour respecter des contraintes temporelles [Guessoum 96a]. Ce modèle d'agent est à deux niveaux : un niveau inférieur comprenant plusieurs modules de traitements, et un niveau supérieur comprenant un module de supervision.

Les trois modules du premier niveau peuvent être réactif (*perception* et *communication*) ou cognitif (*raisonnement*) et opèrent de façon asynchrone et concurrente. Les interactions entre ces modules, ainsi que la cohérence des traitements, sont gérées au second niveau qui consiste en un module de supervision. Il permet à l'agent de raisonner sur les états des modules du premier niveau. Ceci est fait en utilisant un ATN (*Augmented Transition Network*, basé sur l'utilisation d'états et de transitions), qui décrit les réactions de l'agent aux événements qui marquent sa vie. Deux besoins motivent cette supervision :

- ordonnancer les actions indépendamment du fait qu'elles s'exécutent de façon concurrente ou non ;
- synchroniser les actions concurrentes afin de préserver la cohérence de leur traitement.

Il n'existe pas de niveau central de contrôle dans la société, c'est chaque agent qui possède le contrôle de ses activités et gère ses interactions avec les autres agents. Le module de supervision d'un agent, qui se comporte comme un méta-module, se base sur les états possibles des modules et des actions possibles correspondantes. Cette gestion

tient compte du caractère asynchrone des modules et des modalités différentes de chaque module, selon qu'il est réactif ou cognitif.

Au niveau global de la société d'agents, le contrôle est géré à deux niveaux. Premièrement, une coordination inter-agents par envoi de messages permet à chaque agent de maintenir un graphe de ses dépendances envers les autres agents et un graphe d'interférences entre ses règles de décision et celles des autres agents. Deuxièmement, une gestion à un niveau supérieur du parallélisme externe est effectué sur les processus associés aux modules de supervision de chaque agent. Il s'agit d'un contrôle explicite de la société réparti sur plusieurs agents.

Plusieurs applications ont servi à valider cette architecture d'agent, dont la simulation de procédés industriels de fabrication [Guessoum 94], le monitoring de la ventilation artificielle des patients en soins intensifs [Guessoum 96b] et la modélisation de l'évolution des entreprises [Guessoum 96c].

Cette approche a influencé notre conception par la gestion souple de multiples modules de natures différentes à l'intérieur d'une même architecture. Le caractère réactif, asynchrone et adaptatif de la gestion des actions est apparu comme une solution au problème de la coordination des traitements. Cependant, les comportements de notre système étant entièrement réactifs, la gestion de ceux-ci se doit de l'être également. Le but est de laisser les comportements appropriés émerger au sein de l'agent pour être ensuite exécutés par celui-ci.

Boissier a travaillé sur la problématique du contrôle dans un système intégré de vision [Boissier 92], qu'il a appliqué au système VAP [Crowley 93]. Il définit le contrôle d'un système intégré de vision par rapport aux niveaux de représentation, comme étant "lié à la recherche des conditions régissant le regroupement des traitements agissant sur un niveau en contrôleurs de traitements d'un niveau inférieur, aux règles spécifiant les interactions entre les traitements intra-niveaux et inter-niveaux" [Boissier 93, p. 4]. Deux points sont à noter dans cette définition. Premièrement, le contrôle est essentiellement descendant, c'est-à-dire que les agents d'un niveau commandent ceux des niveaux inférieurs. Deuxièmement, les règles régissant les interactions entre les agents, travaillant au même niveau de représentation (intra-niveaux) ou à des niveaux différents (inter-niveaux), doivent être bien définies afin de contraindre et de cibler les traitements du système.

Son architecture, ayant pour but l'étude du contrôle dans un système intégré de vision, décompose le système en sous-systèmes. Chacun de ces sous-systèmes est dédié à la construction de la description de la scène à un niveau de représentation donné et correspond à un agent du système global. Un agent peut être défini par son niveau (sur un plan vertical qui donne un agent-niveau), par son focus (sur un plan horizontal qui donne un agent-focus) ou par l'intersection des deux (appelé un agent de base). Ces trois types d'agents correspondent à des structurations différentes du contrôle, qui selon les cas se trouve intégré dans l'agent ou réparti dans la société d'agents. Toute relation entre

agents situés sur des niveaux de représentation différents se définit en terme de processus-contrôleur, mais cependant, les protocoles d'interactions prévoient des échanges ascendants et descendants.

Trois couches de contrôle sont définies et sont dédiées à la résolution des problèmes de décision, d'adaptation et de commande. Chacune de ces couches manipule des problèmes à des échelles temporelles différentes. Pour toutes les instances du contrôle dans le système, il sera exprimé en termes de ces trois couches.

Dans les travaux de Boissier, le contrôle est un élément actif du système, dans le sens où il modifie consciemment l'état des processus, ou modules de traitement. De plus, les interactions et les échanges entre agents sont encadrés et contraints, selon qu'ils sont sur le même niveau ou à des niveaux différents. Nous avons préféré à cette approche une approche plus flexible, qui pose le problème du contrôle, au niveau de l'agent ou de la société, en des termes réactifs et libérés de toute contrainte. L'intelligence du système se trouve dans les comportements de base. Elle peut aussi être vue, de manière émergente, dans l'observation d'un comportement global des agents, et de ce fait, elle n'est pas inscrite de manière explicite dans des couches de contrôle.

Lefèvre a proposé un système multi-agents pour la vision, dans le cadre d'une application de photo-interprétation militaire [Lefèvre 94]. Chaque agent possède une expertise qui repose sur trois notions :

- l'estimation a priori de la capacité de l'agent à résoudre le problème posé (score d'aptitude) ;
- le contrôle des paramètres d'exécution que peut effectuer l'agent (adaptation de ses paramètres internes) ;
- l'analyse a posteriori de la qualité d'un résultat (score de réussite).

Cette expertise permet à l'agent, dans sa couche de contrôle, d'évaluer sa pertinence à répondre à une requête extérieure, de déterminer les valeurs de traitement correspondantes et d'établir un compte-rendu final sur la qualité d'un résultat. Les requêtes extérieures prennent la forme d'actions de coopération, si elle sont envoyées par des agents de même niveau, ou de sous-traitance, si elle sont envoyées par des agents de niveau supérieur.

Le principal inconvénient de cette approche est dans la modélisation des scores, d'aptitude ou de réussite, qui demande une connaissance a priori sur les opérateurs et sur l'évaluation des résultats qui est bien souvent impossible à établir. L'auteur propose pour cela une modélisation selon la théorie de Dempster-Shafer d'un *univers des paramètres du contexte* pour définir les scores d'aptitude et d'un *univers des paramètres du traitement et des paramètres calculés sur le résultat obtenu* pour définir les scores de réussite.

Plutôt que de poser le problème en termes de choix d'opérateurs, effectuant des transformations successives des données, nous avons préféré une approche incrémentale, procédant

par construction progressive.

4.3 La liberté de contrôle

Salotti a proposé des directives pour la conception d'un système de vision soucieux d'optimiser la gestion de l'information [Salotti 94] :

- Retarder la décision problématique le plus longtemps possible, tant que les informations disponibles ne permettent pas de trancher de façon sûre.
- Aider la décision en allant chercher toutes les informations qui sont nécessaires.
- Adapter la décision.
- Centraliser les décisions problématiques (décision globale avec concertation).
- Décentraliser les décisions simples (décision locale individuelle) .
- Faire savoir le cas échéant qu'une décision peu sûre a été prise.

C'est ce modèle qui a inspiré le contrôle dans notre système. En fait, il s'agit d'un contrôle très réactif visant une souplesse d'action dans la stratégie de façon à mieux s'adapter au contexte de l'image plutôt qu'une planification rigide des actions.

Le contrôle existe à trois niveaux dans le système :

- Le *comportement de reproduction* des agents permet d'effectuer l'ordonnancement des différents agents de la société. En effet, la stratégie d'exploration de l'image est distribuée dans les différents agents par le biais de ce comportement. Il s'agit d'une planification réactive dépendant de l'environnement local à chaque agent.
- Le *gestionnaire interne* de l'agent régule les priorités des différents comportements d'un agent et choisit le plus approprié pour chaque situation. Il s'agit d'un contrôle réactif de l'activité des agents, basé sur les événements retournés par les différents comportements.
- Le *séquenceur* est le seul niveau central et global du système. Son action se résume à donner la main tour à tour aux différents agents du système, et à contrôler les temps et les priorités d'exécution de chaque agent.

Au niveau de la société d'agents, le contrôle effectué par le biais du comportement de reproduction consiste à effectuer le bon choix d'objet et à se focaliser au bon endroit. Ce contrôle, qui définit la stratégie d'exploration de l'image, peut s'apparenter dans le principe à ce que l'on retrouve dans des systèmes comme VISIONS (dans les schémas qui représentent les concepts de l'image).

Par contre, au niveau de l'agent lui-même, le contrôle effectué par le gestionnaire interne consiste à effectuer le bon choix d'action, comme dans les approches à base de planification (ex: BORG) ou dans le système de Lefèvre. Les concepts sont déjà identifiés, selon ce principe, et il faut choisir la bonne méthode à appliquer. À la différence de ces approches néanmoins, ce contrôle n'est pas planifié.

Le contrôle, ainsi réparti sur différents niveaux, est décentralisé au niveau de la société d'agents, mais centralisé au sein de l'agent. Des différents niveaux de contrôle, seul le gestionnaire de l'agent n'a qu'une portée limitée à l'agent (un seul individu). Tous les autres éléments de contrôle influencent l'action de la société d'agents. Cette société d'agents fonctionne selon deux modes complémentaires, selon le point de vue observé :

- En *concurrence*, puisque chaque agent tente de s'approprier une ressource qu'est l'image en agrégeant des pixels et en interagissant avec les autres agents. Il s'agit d'une concurrence de territoire.
- En *coopération*, puisque les agents sont tributaires des informations récoltées par les agents pour adapter leurs propres méthodes. Il s'agit d'une coopération indirecte (les agents ne cherchent pas explicitement à aider les autres) pour la gestion des informations.

Nous appelons donc liberté de contrôle le fait que la gestion des différentes actions dans le système s'effectue selon un modèle réactif. La liberté vient dans les notions d'adaptativité au contexte et de souplesse d'exécution dans la stratégie.

4.4 Le comportement de reproduction

Le comportement de reproduction sert à créer des agents en certains lieux de l'image. Il permet à un agent d'explorer son environnement local et de lancer des germes d'agents qui continueront le travail de segmentation. Par l'action de ce comportement, c'est la stratégie d'exploration de l'image qui est définie et distribuée dans chacun des agents. Son action dans le contrôle se situe donc au niveau de la société d'agents. Les types et les conditions de lancement des agents sont définis par le fichier de configuration.

Deux variantes de ce comportement existent :

- pour lancer des agents dans la même image, afin de continuer la segmentation de l'image courante :
- pour lancer des agents dans l'image suivante, afin de poursuivre la segmentation de la séquence d'images.

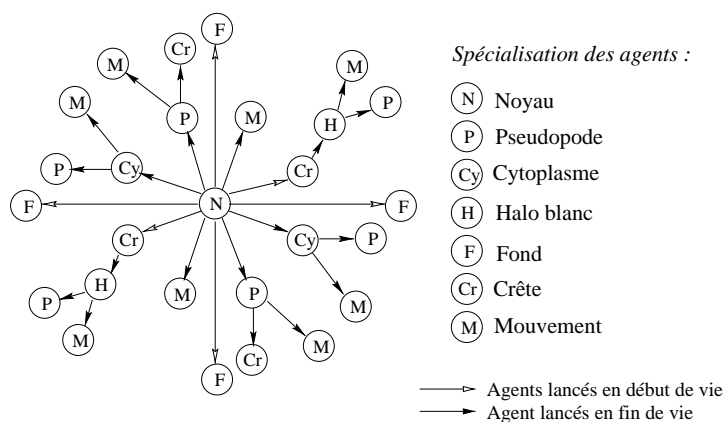


FIG. 4.28 – *Graphe de lancement des agents (comportement de reproduction). Tous les agents contribuent à l'exploration de la cellule. Le nombre d'agents indiqué ici est indicatif, et ne sert qu'à illustrer la configuration de lancement d'agents. Par exemple, l'agent noyau peut lancer plus de quatre agents fond. Ce nombre est indiqué dans la configuration de l'agent.*

4.4.1 Reproduction dans la même image

La reproduction dans la même image a pour but de lancer des agents, de spécialisations différentes, aux endroits stratégiques de l'image. Un agent étudie son environnement local en fonction du type d'agent qu'il veut lancer et choisit les endroits les plus appropriés pour initialiser de nouveaux germes. Cette exploration permet l'adaptation de la création des agents au contexte local de l'image et au type d'agent à créer (par spécialisation du comportement).

4.4.2 Une stratégie d'exploration

Il existe deux moments dans la vie d'un agent où celui-ci peut lancer un comportement de reproduction, soit au début de sa vie ou à la fin de celle-ci. Selon le cas, les nouveaux agents créés travailleront en parallèle avec l'agent qui les a lancés, ou en série. Ce choix de l'instant où cette reproduction est faite permet de mieux adapter la stratégie globale de la société d'agents aux différents types d'informations présents dans l'image. Une information facile à trouver sera recherchée par des agents dès le début, tandis que pour une information plus délicate, le traitement sera retardé, le temps que d'autres agents aient avancé leur travail et que la recherche de l'information devienne plus aisée.

C'est donc une véritable stratégie au niveau de la société d'agents qui peut être définie grâce à ce comportement, pourtant défini très localement. Par spécialisation, il est possible de spécifier pour un agent :

- quand lancer de nouveaux agents ;
- quels types d'agents lancer ;

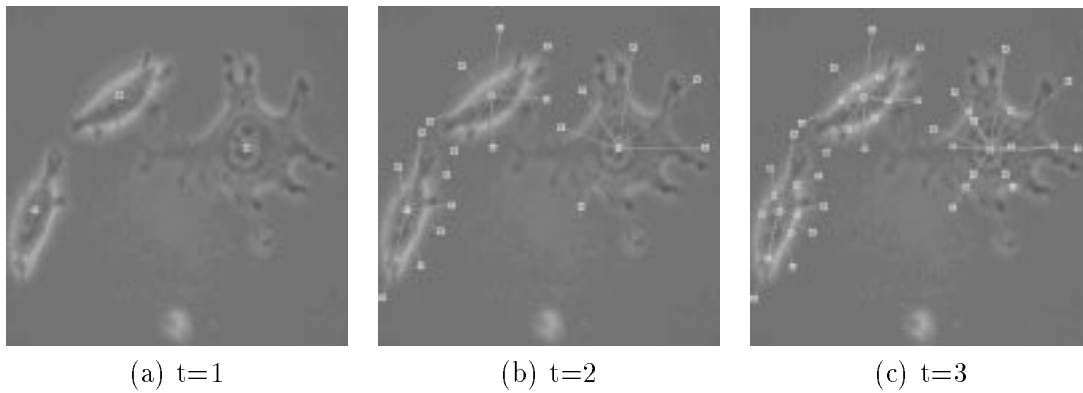


FIG. 4.29 – *Positionnement des agents sur l'image. La stratégie de reproduction (création d'agents) influence le résultat final de la segmentation. Cette création d'agents s'effectue en plusieurs étapes, permettant de définir une stratégie complète de segmentation d'une séquence d'images.*

- comment trouver les bons germes.

La figure 4.28 donne un aperçu du graphe de lancement des agents dans l'exploration de la cellule. Ce graphe, qui représente les configurations de reproduction des différents agents, permet de définir une stratégie complète d'exploration de la cellule. Tous les agents participent à cette stratégie en recherchant les composantes qui peuvent se retrouver dans leur environnement immédiat. Il a été défini expérimentalement selon l'application. La figure 4.29 illustre l'application de ce graphe pour notre application. À partir d'un agent noyau initial sur chaque cellule, d'autres types d'agents sont lancés en fonction des configurations pour ces agents et de l'exploration de l'image qui est effectuée par les agents.

Dans la stratégie d'exploration, un type d'agent peut être recherché et des agents créés par plusieurs autres, qu'ils soient de même type ou non. Ainsi, le but est d'accroître les chances de succès pour trouver toutes les composantes des cellules. C'est l'entrelacement de plusieurs explorations menées par plusieurs agents qui permet d'augmenter la fiabilité de cette stratégie.

4.4.3 Principe de l'algorithme

Le principe du comportement de reproduction est le même que l'agent travaille sur une primitive région ou sur une primitive contour (figure 4.30). Il suit les étapes suivantes :

- Extraire la liste des pixels de la frontière de la primitive (dans le cas d'un contour, cette liste sera composée des pixels du contour) ;
- Diviser cette frontière en n sections ;
- Pour chacune des sections :
 - Calculer le profil de pixels de direction normale à la frontière ;

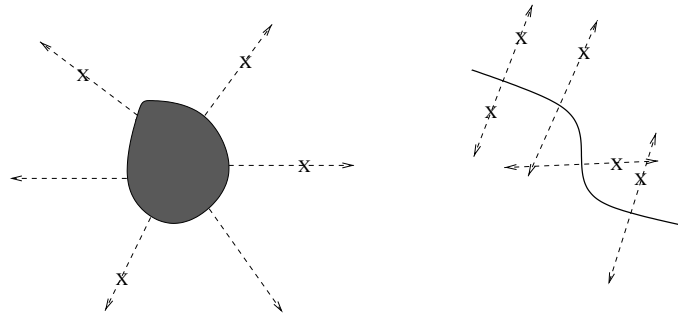


FIG. 4.30 – Illustration du principe d'exploration pour le comportement de reproduction dans le cas d'un agents région et d'un agent contour. Le principe est d'explorer l'environnement selon des profils normaux à la région ou au contour. Un germe peut être trouvé ou non sur chaque profil selon les résultats de l'évaluation des différents pixels du profil.

- Limiter la longueur du profil à la première primitive rencontrée dans l'image ou à une longueur maximale de 100 pixels ;
- Évaluer chacun des pixels du profil en fonction du type d'agent que l'on cherche à lancer ;
- Retenir le pixel ayant l'évaluation la plus élevée, si cette évaluation est supérieure à un seuil (configuré pour le type d'agent) ;
- Lancer un germe d'agent sur tous les pixels retenus.

Cet algorithme présente des similitudes avec celui développé par Zhou et Pycok pour trouver la frontière d'une cellule [Zhou 97]. Ils utilisent en effet un principe de recherche de profils et d'évaluation des pixels afin d'établir le contour d'une cellule. L'analyse des profils qu'ils ont effectuée permet d'obtenir le pixel ayant la plus grande probabilité d'appartenir à la frontière de la cellule. Parmi les critères utilisés, la cohérence entre les candidats relevés sur des profils adjacents est aussi utilisée.

Les paramètres qui doivent être fournis pour le comportement de reproduction sont spécialisés en fonction du type des agents et apparaissent dans le fichier de configuration. Ces paramètres relèvent soit de l'agent qui explore soit de l'agent à initialiser :

- paramètres dépendants de l'agent *lanceur* :
 - type d'agent à lancer ;
 - nombre d'agents à lancer (détermine le nombre de sections et de profils) ;
- paramètres dépendant de l'agent à lancer :
 - critères d'évaluation pour les pixels du profil ;
 - seuil minimal de l'évaluation.

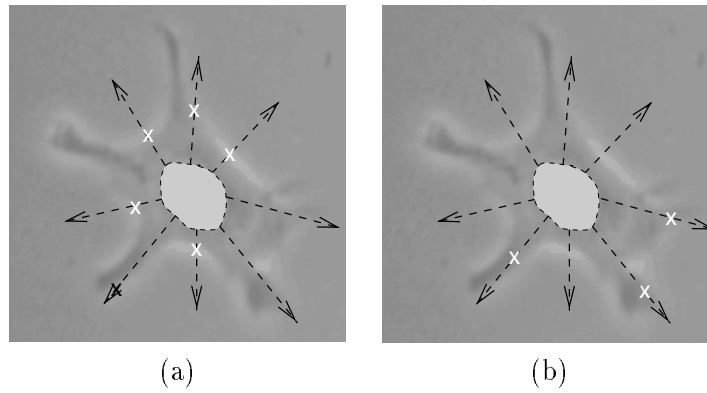


FIG. 4.31 – *Exploration de l'image pour la reproduction. L'agent explore son environnement et évalue les pixels sur des profils perpendiculaires à sa surface. Les X indiquent les germes effectivement lancés. (a) Recherche de germes pour les halos blancs entourant la cellule. (b) Recherche de germes pour les pseudopodes.*

La distinction entre ces deux catégories provient du type d'agent pour lequel sera spécifié cette spécialisation. Dans le fichier de configuration, nous retrouvons par exemple pour l'agent de type *Noyau* les spécifications suivantes :

Reproduction même image début : *Fond* x 7
Reproduction même image début : *Crête* x 18
Reproduction même image fin : *Pseudopode* x 14
Reproduction même image fin : *Mouvement* x 15
Reproduction même image fin : *Cytoplasme* x 11

Le signe x indique le nombre d'agents à lancer (7 agents fond, 18 agents crête, ...). Ensuite, chaque type d'agent possède ses propres critères pour retrouver les germes dans l'image. Par exemple, quel que soit l'agent qui veut initialiser des agents de type *Fond*, les critères et le seuil utilisés pour trouver les germes seront les mêmes.

4.4.4 Observations et améliorations sur l'algorithme

La figure 4.31 illustre ce principe de reproduction. Pour un agent qui désire lancer d'autres agents, deux situations sont montrées, en fonction du type d'agent considéré. Les profils sont illustrés avec la position des germes lancés. Ce mécanisme permet à un agent d'explorer son environnement immédiat en fonction de ce qu'il recherche. Plusieurs directions sont explorées et seuls les profils où un pixel présente une évaluation suffisamment élevée permettront de lancer un germe d'agent. Il peut y avoir au plus un germe par profil, et il est possible qu'il n'y en ait aucun. Si plusieurs germes sont lancés sur une même composante de l'image, ils se rencontreront et fusionneront ensemble.

Cependant, l'efficacité de ce comportement est largement dépendante du nombre de profils considérés. Si le nombre de directions explorées n'est pas assez important, des régions de l'image peuvent être oubliées, tel que montré à la figure figure 4.31b, où des

pseudopodes sont manqués parce que trop peu de directions sont explorées. Dans ce cas-ci, le problème principal est la faible largeur de ces composantes par rapport à leur distance de l'agent central. Elles sont facilement manquées lors de la phase d'exploration.

Il existe trois solutions possibles pour résoudre ce problème :

1. La solution la plus simple consiste à augmenter le nombre de directions explorées. Cependant, cette solution n'est pas sans inconvénients, puisqu'elle augmente le temps de calcul du comportement, et surtout, elle augmente, de façon exponentielle, le nombre d'agents actifs dans l'image. Chaque agent lancé aura la possibilité lui aussi de lancer un ou plusieurs autres agents, dans autant de directions. La charge du système s'en trouve donc accrue. Même si plusieurs des agents surnuméraires fusionneront, ils causent, pendant une période de temps, un ralentissement important du système.
2. La deuxième solution consiste en un compromis entre les deux situations extrêmes, soit beaucoup de directions explorées, et un nombre plus restreint de germes lancés. Il s'agit de comparer entre eux les différents germes obtenus par la phase d'évaluation. Le principe veut que l'on ne retienne qu'un germe pour m profils étudiés. Donc plusieurs directions sont effectivement explorées, mais pour un ensemble de directions, on ne retiendra qu'un seul germe. Le temps de calcul pour trouver ces germes sera important, mais le nombre d'agents résultant sera limité. Ce nombre de profils rassemblés ensemble peut être configuré où il peut être adapté en fonction de l'environnement.
3. La dernière solution consiste à distribuer la stratégie d'exploration et de reproduction sur plusieurs agents de types différents. C'est-à-dire que des germes d'agents difficiles à trouver par l'exploration d'un agent pourront être beaucoup plus faciles à trouver par un autre agent. Dans l'exemple de la figure 4.31b, il est possible de confier la recherche des pseudopodes aux agents *Noyau*, *Cytoplasme* et *Halo blanc*. Les dispositions combinées de ces trois types d'agents renforcent les chances de retrouver tous les pseudopodes de la cellule.

La meilleure solution au problème réside dans la combinaison des deux dernières possibilités énoncées, soit l'optimisation de la recherche par un seul agent et la coopération de plusieurs agents pour renforcer les chances de réussite. Cependant, dans le système actuel, seules la première et troisième solution cohabitent.

4.4.5 Les critères d'évaluation des pixels

Pour trouver les germes d'agents à lancer, il faut évaluer les profils de pixels perpendiculaires à l'agent. Cette évaluation est dépendante du type d'agent à lancer, et sera

spécialisée en conséquence. Quatre critères d'évaluation principaux existent, chacun pouvant se décliner sous plusieurs variantes :

1. recherche d'un *extrémum* le long du profil ;
2. recherche d'une *distance* précise ;
3. recherche de pixels *opposés* selon une caractéristique ;
4. recherche de pixels *inclus* entre deux agents d'un même groupe.

Ces critères possèdent quelques variantes, en particulier pour le critère d'extrémum. L'évaluation de plusieurs critères pour prendre une décision suit le modèle présenté au paragraphe 3.4, et s'apparente à l'évaluation déjà présentée pour le comportement de perception au paragraphe 3.5. Il s'agit dans les deux cas de méthodes d'évaluation reposant sur la combinaison de plusieurs critères. Cette évaluation peut aussi bien servir à la construction de primitives comme dans le comportement de perception que dans la mise en place d'une stratégie d'exploration, comme dans le cas du comportement de reproduction.

4.4.5.1 Critère d'extrémum

Le critère d'extrémum évalue le pixel en fonction d'un minimum ou d'un maximum existant. Il s'agit d'évaluer si le pixel est près ou loin de cette valeur d'extrémum. Cette valeur d'extrémum peut être fonction de l'image ou des autres agents existant dans l'image. Dans le premier cas, elle peut aussi être calculée globalement dans l'image ou localement pour le profil.

Extrémum global fonction de l'image

Il s'agit ici de rechercher la valeur extrémum dans l'image, de façon globale, c'est-à-dire qu'elle sera la même pour tous les profils explorés¹. Cette valeur peut être calculée selon n'importe laquelle des caractéristiques présentes dans l'environnement (image source, norme et direction du gradient, variance, amplitude et signe du mouvement). Selon la caractéristique considérée, l'évaluation peut prendre différentes significations.

Une fois la valeur extrémum trouvée, il faut comparer cette valeur à la valeur (pour la même caractéristique) du pixel considéré. L'évaluation peut conduire à interpréter la valeur extrémum comme un minimum ou un maximum.

$$\text{Pour un minimum : } Evaluation_{critere} = valeur_{extremum} / valeur_{pixel}$$

$$\text{Pour un maximum : } Evaluation_{critere} = valeur_{pixel} / valeur_{extremum}$$

1. En fait, la valeur sera recherchée dans une boîte 200x200 centrée sur l'agent, où seuls les pixels non-segmentés seront considérés.

Par exemple, on peut spécifier pour un agent dans le fichier de configuration :

Critère reproduction : 0.4 extréma global < variance

Une fois décodée, cette ligne précise qu'on évalue le pixel pour rechercher le minimum de variance (on l'évalue par rapport au minimum de l'image). Ce critère a un poids de 0.4 dans l'évaluation finale du pixel.

Extrémum local fonction de l'image

Il s'agit ici de rechercher un extrémum local dans le profil étudié. Pour chaque pixel, on évaluera si celui-ci est un extrémum local ou non (sur une distance de 3 pixels de chaque côté du pixel évalué), en comptant le nombre de pixels répondant à l'exigence (sur un maximum de 6 pixels)². Les autres nuances de ce critère sont dans la façon de compter les pixels voisins. Il est possible de rechercher un extrémum (minimum ou maximum) franc, c'est-à-dire sans aucun voisin ayant la même valeur, ou d'accepter les plateaux comme extrémum, avec le pixel central au plateau comme germe.

Par exemple, on peut spécifier pour un agent dans le fichier de configuration :

Critère reproduction : 0.5 extréma local <= variance

Ici, on recherche un minimum de variance le long du profil. Si un plateau (plusieurs valeurs identiques) survient et qu'il est minimal, il sera considéré comme valide au sens de cette évaluation. Ce critère a un poids de 0.5 dans l'évaluation finale du pixel.

Extrémum fonction des autres agents

Il s'agit ici de rechercher l'extrémum parmi les informations récoltées par les autres agents de l'image. La valeur extrémum ne sera plus recherchée dans l'environnement spatial de l'agent, mais parmi les informations sur les structures de l'image. En fait, cette valeur sera recherchée dans les structures d'information des agents d'un type donné (spécialisation). La notion de *local* ou *global* conserve un sens pour ce critère, puisqu'elle spécifie si la valeur sera recherchée parmi tous les agents de l'image (*global*) ou seulement parmi les agents appartenant au même groupe d'agents (*local*).

Une fois la valeur extrémum trouvée, il faut comparer cette valeur à la valeur (pour la même caractéristique) du pixel considéré. L'évaluation peut être faite pour interpréter la valeur extrémum comme un minimum ou un maximum³.

Pour un minimum : $Evaluation_{critere} = 1 - valeur_{pixel} / valeur_{extremum}$

Pour un maximum : $Evaluation_{critere} = 1 - valeur_{extremum} / valeur_{pixel}$

2. Ceci dans le but d'avoir une évaluation floue plutôt que binaire, tel que discuté au chapitre 2.

3. Il est à noter que la fonction d'évaluation donnée ici est différente de celle présentée dans le cadre du critère d'extrémum global fonction de l'image. Le but recherché est de conserver un minimum de sens lorsqu'on interprète le critère dans le fichier de configuration.

Par exemple, on peut spécifier pour un agent dans le fichier de configuration :

Critère reproduction : 1.0 extréma global source < Fond

Ici, la valeur extrémum recherchée est la valeur de niveau de gris minimale parmi tous les agents de type *Fond* de l'image. Ces agents n'appartenant à aucun groupe, cette recherche est globale dans l'image, mais spécialisée aux agents *Fond*. Ensuite, on recherchera dans le profil des pixels inférieurs à cette valeur (plus sombre que le fond). Ce critère a un poids de 1.0 dans l'évaluation finale du pixel, ce qui signifie qu'il est le seul critère utilisé pour évaluer les pixels.

4.4.5.2 Critère d'opposition

Le critère d'opposition permet de rechercher une transition le long du profil. Il permet d'évaluer un pixel sur la moyenne des valeurs situées en aval et en amont sur le profil. L'évaluation compare ces deux moyennes sur la base de leur opposition ou de leur similarité.

Les moyennes sont calculées (sur 3 pixels) selon n'importe laquelle des caractéristiques présentes dans l'environnement (image source, norme et direction du gradient, variance, amplitude et signe du mouvement). Selon la caractéristique considérée, l'évaluation est adaptée et peut prendre différentes significations. Par exemple, le calcul de la moyenne des directions du gradient tient compte de la périodicité des valeurs pour permettre d'obtenir une évaluation cohérente avec la situation observée.

La notion importante utilisée pour ce calcul est la différence maximale qui peut exister entre deux valeurs. Deux valeurs de variance peuvent être considérées très proches ou très éloignées selon la distribution des valeurs qui existent dans l'image ou à proximité de l'agent. Le critère doit tenir compte de cette distribution. L'opposition entre les deux valeurs est donc relative à l'écart maximal possible. Cet écart maximal peut être calculé de deux manières, en tenant compte de la totalité de l'image (global) ou seulement sur le profil considéré (local). Dans les deux cas, il s'agit de rechercher les valeurs minimale et maximale pour calculer cet écart maximal. Pour la direction du gradient, l'écart maximal est fixe, avec une valeur de 4.

Un opérateur (< ou >) spécifie si une bonne évaluation est donnée en cas de forte opposition des moyennes ou de similarité entre celles-ci. Par exemple, on peut spécifier pour un agent dans le fichier de configuration :

Critère reproduction : 0.3 opposition global > graddir

Ce critère permet la recherche d'une transition dans les valeurs de la direction du gradient le long du profil. On recherche un pixel où la différence entre les moyennes en aval et en amont calculées selon la direction du gradient est maximale. Dans ce cas, le notion de global ou local n'a pas d'importance puisqu'il s'agit de la direction du gradient

et que l'écart maximal dans tous les cas est de 4. Ce critère a un poids de 0.3 dans l'évaluation finale du pixel.

4.4.5.3 Critère de distance

Le critère de distance permet de choisir le meilleur endroit sur le profil pour lancer le nouvel agent. Ce profil a une longueur limitée, qui est souvent raccourcie par la proximité d'une autre primitive (tel qu'expliqué au paragraphe 4.4.3). Il est possible de spécifier le meilleur endroit de lancement d'un agent simplement par une mesure de distance sur le profil.

L'évaluation de ce critère est binaire (0 ou 1), selon que le pixel évalué sur le profil répond ou non au critère de distance. Pour préciser ce critère, il faut spécifier :

- un *opérateur* ($<$, $>$, $=$, $<=$ ou $>=$) ;
- une *position de référence* sur le profil, qui peut être absolue (si la valeur est supérieure ou égale à 1) ou relative à la longueur du profil (si la valeur est inférieure à 1 qui représente une fraction de la longueur totale du profil).

Par exemple, on peut spécifier pour un agent dans le fichier de configuration :

Critère reproduction : 0.5 distance > 0.5

Ce critère est interprété par le comportement comme la recherche d'un pixel se situant dans la deuxième moitié du profil. Tous les pixels proches de l'agent reçoivent une évaluation nulle (0) et les pixels situés au-delà de la moitié du profil reçoivent une évaluation parfaite (1). Ce critère a un poids de 0.5 dans l'évaluation finale du pixel. Normalement, ce critère n'est pas utilisé seul, mais en conjonction avec un autre critère. Il permet de spécifier que l'on recherche une caractéristique proche ou loin de l'agent.

4.4.5.4 Critère d'inclusion

Le critère d'inclusion permet d'évaluer les extrémités du profil étudié. En fait, tous les pixels du profil recevront la même évaluation. Comme le critère de distance, il permet d'ajouter des contraintes spatiales et il doit être utilisé en conjonction avec d'autres critères pour être efficace. Il *valide* ou non un profil dans la recherche de germes pour de nouveaux agents.

Pour ce critère, on regarde les deux agents se trouvant aux extrémités. Si les deux agents appartiennent au même groupe, alors l'évaluation est parfaite (1). Dans le cas contraire, elle vaut zéro (0). Cette évaluation est donc strictement binaire.

Une variante de ce critère permet de spécifier le type d'agent (spécialisation) qui doit se trouver en bout de profil. Le type de l'agent est alors utilisé en plus de son appartenance à un groupe.

Par exemple, on peut spécifier pour un agent dans le fichier de configuration :

Critère reproduction : 0.5 inclusion HaloBlanc

Ce critère permet la recherche de germes le long d'un profil où les deux extrémités (l'agent qui recherche les germes et celui situé en bout de profil) appartiennent au même groupe. Ici, on spécifie en plus que l'agent en bout de profil devra être de type *HaloBlanc*. Ce critère a un poids de 0.5 dans l'évaluation finale du pixel, et il devra être utilisé avec d'autres critères pour spécifier le pixel précis où le germe pourra être lancé.

4.4.6 Reproduction vers l'image suivante

La reproduction vers l'image suivante a pour but de poursuivre le traitement des images de la séquence. Elle permet de démarrer le traitement de l'image suivante, même lorsque l'image courante n'est pas entièrement traitée. Ainsi, le traitement complet de la séquence d'images s'effectue en pipeline.

L'idée est de permettre à un agent de lancer un autre agent à la même position (ou presque) dans l'image suivante. Lorsque l'agent a terminé ses propres traitements (segmentation terminée), il peut lancer un autre agent (du même type ou non) dans l'image suivante. Cependant, cette possibilité n'existe que pour les agents de type région, car il est beaucoup plus facile de retracer une région d'une image à l'autre grâce à la surface de celle-ci qu'un contour mince.

Ce lancement s'effectue normalement à la position courante du centre de gravité de la primitive. Il est possible également de prendre en compte le déplacement des objets dans la séquence, soit en tenant compte du déplacement moyen entre les images de toute la séquence ou seulement dans les dernières images traitées. Un agent a accès aux informations des agents de l'image précédente. Il peut donc les utiliser pour calculer un déplacement moyen (ou pour le mettre à jour). Ce déplacement moyen permet de déterminer, avec plus de confiance, la position probable de la primitive dans l'image suivante afin d'initier un nouvel agent pour la traiter. Cela permet d'éviter de perdre un objet au cours de la séquence. Pour l'instant, cette possibilité de prendre en compte le déplacement n'est pas implantée dans le système développé.

Ce mécanisme n'est pas infaillible, mais en ce qui concerne notre application, où les déplacements ne sont pas si brusques, ce système est satisfaisant. Il permet de suivre les composantes tout au long de la séquence d'images, en créant des liens de création d'agents et de transmission d'informations d'une image à l'autre.

La stratégie utilisée jusqu'à présent pour traiter notre application est de centraliser le passage d'une image à l'autre sur une composante (en l'occurrence le noyau de la cellule). Ce choix a été fait à cause de la simplicité de la segmentation du noyau et pour sa position centrale dans la cellule. La reproduction d'une image vers la suivante est de la responsabilité de ces agents et ensuite, la recherche des autres composantes (moins stables

d'une image à l'autre) se fait par les mécanismes du comportement de reproduction dans la même image. Cette stratégie est bien adaptée pour les applications cytologiques.

Dans le cas d'une application différente, il pourrait être utile de développer plus ce comportement afin d'effectuer une vraie recherche des composantes d'une image à l'autre. Cette recherche pourrait être effectuée en développant des critères d'évaluation des pixels similaires à ceux existants. Cependant, ils devraient tenir compte en plus d'une probabilité plus forte de trouver l'objet à un endroit estimé par le déplacement moyen entre images.

4.4.7 Initialisation des premiers agents

Le comportement de reproduction présenté permet de rechercher et d'initialiser automatiquement les agents sur une image et les suivantes, sans aucune intervention de l'utilisateur. Le problème est d'initialiser les agents de la première image qui lanceront le processus. Deux solutions sont possibles :

1. De façon *automatique*, réalisable dans le cas où les premiers agents sont faciles à déterminer.
2. De façon *manuelle*, en demandant à l'utilisateur d'indiquer les premiers germes sur la première image, dans le cas où ceux-ci sont plus difficiles à détecter.

La deuxième solution sera traitée avec l'interface au paragraphe 5.2.4. La première solution, plus élégante mais plus complexe, peut être résolue partiellement en adaptant les critères déjà développés pour le comportement de reproduction dans la même image à une recherche globale dans l'image. Cette recherche est prise en charge par le système, et non par des agents, puisqu'aucun n'existe encore.

Pour cela, les mêmes critères (ou presque) peuvent être utilisés. Dans le fichier de configuration, ils sont précédés par le mot *init*, limitant leur utilisation au cas d'une initialisation automatique dans la première image. Cette adaptation est faite pour tenir compte de la recherche dans une image et non plus sur un simple profil. Aucune information de segmentation n'est disponible, seules les images sources et leurs caractéristiques peuvent être employées.

Par exemple, deux des critères précédemment évoqués peuvent être adaptés au contexte de la recherche initiale de germes :

- le critère d'extrema ;
- le critère de distance (entre les germes trouvés).

Il s'agirait cette fois de parcourir l'image entière à la recherche de germes, et de comparer les germes trouvés entre eux. Ce dernier aspect n'est toutefois pas encore implanté dans la version actuelle du système.

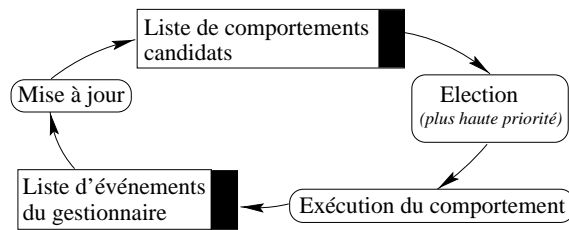


FIG. 4.32 – Boucle de contrôle du gestionnaire interne de l'agent. Celui-ci choisit le meilleur comportement et l'exécute. Les comportements retournent des événements guidant le choix du gestionnaire interne.

4.5 La gestion des comportements

L'agent possède un gestionnaire interne qui est responsable du choix du comportement à exécuter à chaque instant en fonction du contexte qui évolue. En fait, chaque comportement qui s'exécute possède une période de temps allouée pour son exécution au-delà de laquelle il doit rendre la main au gestionnaire. Et c'est à ce moment que le gestionnaire effectue ses choix pour l'exécution du comportement qui semble le plus adapté à la situation présente.

4.5.1 Fonctionnement général

La gestion interne des comportements d'un agent est inspirée de celle du séquenceur. Elle est toutefois généralisée pour tenir compte des disparités existantes entre les différents types de comportements. Cette gestion se fait par un mécanisme couplant événements et priorités d'exécution des comportements : les événements sont analysés par le gestionnaire en vue d'ajuster, de façon réactive, la priorité des comportements susceptibles d'être déclenchés.

Les comportements d'un agent sont interruptibles : ils peuvent ainsi s'exécuter de manière concurrente. Le temps alloué pour leur exécution est contrôlé par le gestionnaire interne. Les comportements sont donc déclenchés de manière asynchrone et concurrente, selon le schéma proposé par Guessoum [Guessoum 96a]. Le modèle de comportement adopté s'apparente également au principe des tâches compétitives mis en oeuvre par Drogoul et Ferber [Drogoul 92]. La gestion des comportements s'effectue de façon réactive, selon un schéma d'activation programmé en fonction des événements, semblable aux approches de Brooks [Brooks 91a] et Maes [Maes 89]. Cependant, contrairement à ces approches, les activations sont réglées par une couche supérieure aux comportements, et non par les comportements eux-mêmes.

La figure 4.32 illustre le fonctionnement du gestionnaire de l'agent. Le gestionnaire maintient à jour une liste des instances des différents comportements et leur priorité associée. À chaque cycle, le gestionnaire choisit le comportement ayant la plus haute priorité et

Événements		Réponses	
Type	Émetteur	Action	Comportement
Début de vie	Perception	Priorité max.	Reproduction (même image)
Fin de vie	Perception	Priorité max.	Reproduction (même image)
Fin de vie	Perception	Priorité max.	Reproduction (image suivante)
Pixel marqué	Perception	Incrément priorité	Fusion (demande)
Début de vie	Fusion (demande)	Priorité min.	Fusion (action)
Demande fusion	Autre agent	Instanciation et priorité max.	Fusion (réponse)
Réponse fusion	Autre agent	Priorité max.	Fusion (action)

FIG. 4.33 – Exemple de table événements-réponses du gestionnaire de l'agent. À chaque type d'événement reçu par un comportement ou un autre agent correspond une réponse qui se traduit en termes de priorité ou d'instanciation de comportements si l'instance recherchée n'existe pas.

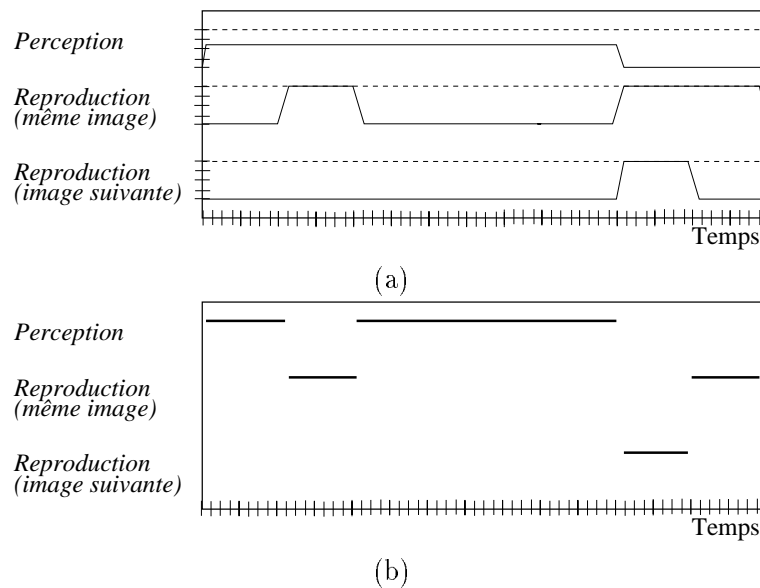


FIG. 4.34 – Gestion du comportement de reproduction par rapport au comportement de perception. (a) Évolution des priorités de ces comportements. (b) Comportements élus en fonction du temps. Le comportement de perception possède une priorité moyenne tout au long de la vie de l'agent. Le comportement de reproduction s'active lorsque des événements de début ou de fin d'activité sont envoyés par le comportement de perception.

l'exécute pour une période de temps limitée. Le comportement est responsable de rendre la main au gestionnaire une fois ce temps écoulé. Ensuite, en fonction des événements accumulés, la liste des comportements est mise à jour et le cycle recommence. La notion de priorité signifie l'adéquation du comportement à répondre au contexte actuel. Plus la priorité est élevée, plus le comportement est apte à s'exécuter.

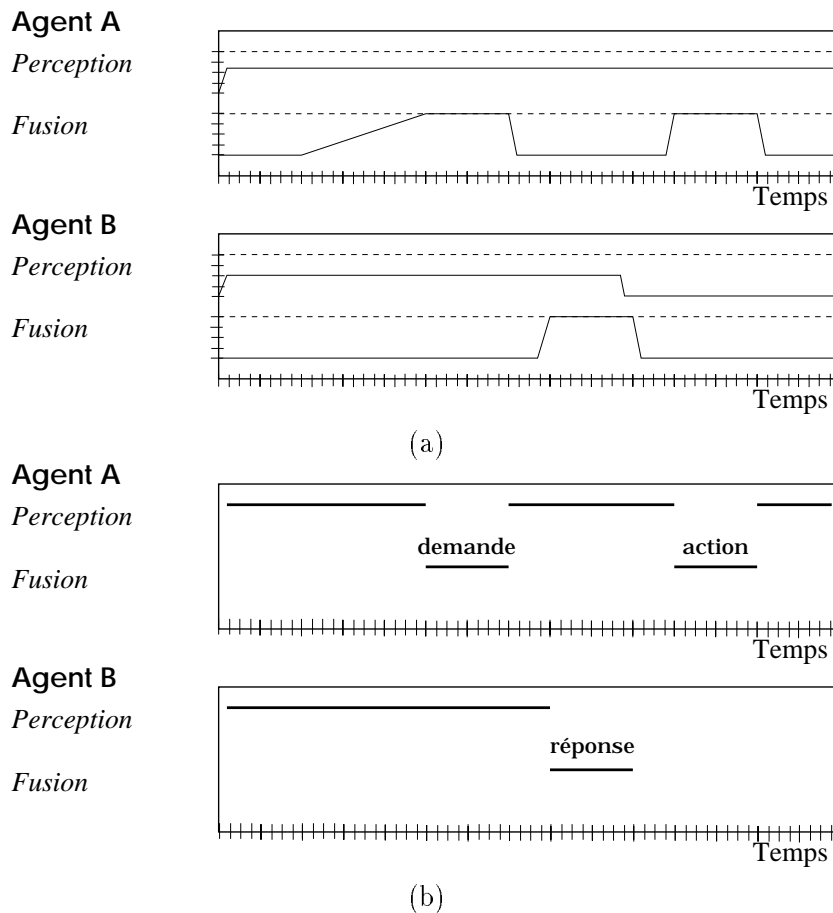


FIG. 4.35 – Gestion du comportement de fusion (interaction) par rapport au comportement de perception. (a) Évolution des priorités de ces comportements. (b) Comportements élus en fonction du temps. Le comportement de fusion (interaction) s'active lorsque des événements envoyés par le comportement de perception suggère qu'une fusion est possible avec un agent voisin. La discussion entre les deux agents s'effectue en trois étapes, puis un des deux agents se termine pour laisser l'autre fusionner les deux entités et continuer la perception.

4.5.2 Événements

Un événement peut être vu comme une communication locale, à courte durée de vie. Il est transmis par un comportement pour exprimer un état ou une situation particulière ; il conduit en retour le gestionnaire à privilégier d'autres comportements, selon une logique qui lui est propre. Cette technique permet ainsi l'instauration de relations d'influences mutuelles entre comportements.

Contrairement aux requêtes, qui sont des commandes complètes et muries envoyées au séquenceur, les événements ne sont que de simples états de fait. Un comportement retourne un événement pour marquer un fait, sans se soucier de l'utilité de cette action. Le comportement n'a pas d'influence directe sur l'utilisation ultérieure de cet événement.

Le gestionnaire est le seul à même de connaître les événements importants et leur utilité. Pour cela, il utilise une table de liens entre les événements reçus et les réponses à donner (figure 4.33). Cette table est fournie a priori par le concepteur. Un comportement ne perçoit que lui-même au sein de l'agent et n'a aucune connaissance de l'existence des autres : c'est le gestionnaire qui associe les différentes tâches en vue de les coordonner dans l'accomplissement d'un objectif (implicite) unique.

Les figures 4.34 et 4.35 montrent l'évolution des priorités des comportements, selon les événements retournés par chacun, en fonction du comportement de perception, qui est le principal émetteur d'événements. Sur réception d'un événement, le gestionnaire décide de l'action à entreprendre (selon une table interne et l'état courant de l'agent). Cette action se traduira par une modification (augmentation ou diminution) de la priorité d'un des comportements de la liste ou par la création d'une nouvelle instance d'un comportement.

Dans l'exemple donné dans la figure 4.35, le comportement de perception signale la proximité d'un autre agent, par l'envoi d'événements *Pixels marqués*. Ces événements se traduiront par la création d'un comportement de fusion (plus exactement, d'une demande de fusion) et par l'augmentation de la priorité de celui-ci. Lorsque la priorité de ce comportement sera suffisamment élevée (traduisant une jonction sur plusieurs pixels entre deux régions), il se déclenchera et communiquera avec l'agent voisin pour fusionner avec lui.

4.6 Le séquenceur

Le dernier niveau de contrôle du système est le séquenceur. Il s'agit d'un contrôle plus bas niveau que les autres niveaux de contrôle présentés dans la mesure où il n'est pas régi par des connaissances dépendant du domaine d'application. Le rôle du séquenceur est de simuler une action parallèle à partir d'une exécution séquentielle. En fait, chaque agent est exécuté avec une période de temps limitée au-delà de laquelle il doit rendre la main. Cet ordonnancement des agents s'apparente à celle d'un système d'exploitation.

À chaque agent sont associées deux variables explicites, gérées par le séquenceur : un *quantum* de temps d'exécution et une *priorité* d'exécution. Le temps d'exécution d'un agent est le produit de ces deux quantités. Le quantum est le même pour tous les agents, et varie en fonction de la charge du système, tandis que la priorité d'un agent est modifiable par requête ou par configuration. L'agent utilise cette période de temps à son gré, mais il est responsable de rendre la main au système lorsque son temps est écoulé.

Dans ce contexte, la priorité d'un agent signifie l'importance donnée à l'agent pour son exécution (traduite en temps d'exécution plus ou moins long). En fait, un agent dont les résultats sont considérés robustes ou dont les résultats sont nécessaires rapidement se verra attribuer une priorité plus élevée. Ainsi, l'information facile à traiter le sera rapidement et les décisions plus problématiques auront accès à un surplus d'informations.

Le contrôle du système s'appuie sur une liste d'agents qui est maintenue à jour avec

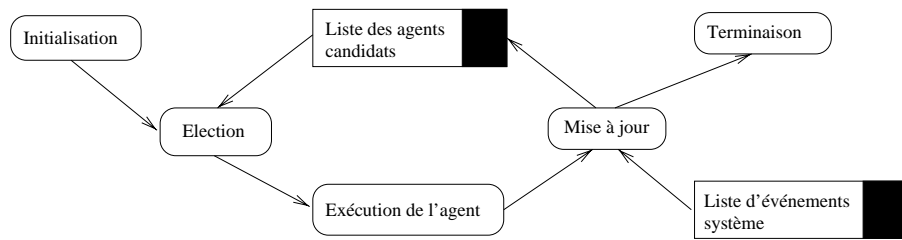


FIG. 4.36 – Boucle de contrôle du séquenceur. Celui-ci exécute les agents à tour de rôle en leur attribuant un temps limité d'exécution.

les informations provenant d'une liste de requêtes, qui proviennent de l'utilisateur ou des agents du système (figure 4.36). Cette boucle de contrôle ressemble à celle du gestionnaire interne de l'agent. Cependant, la mise à jour est plus simple dans ce cas, puisqu'elle n'est que la réponse à des requêtes précises. Le séquenceur ne possède pas de pouvoir de décision sur ces requêtes. Ces requêtes sont :

- créer / détruire un agent ;
- modifier la priorité d'un agent.

4.7 Spécialisation des agents

Le modèle d'agent décrit dans ce travail et explicité au paragraphe 1.4 est générique. Tous les agents du système, quel qu'ils soient, sont basés sur ce modèle. Cependant, il est spécialisé pour l'application. Cette spécialisation réside dans la spécification de paramètres pour chacun des comportements et pour l'agent lui-même. La spécialisation de l'agent permet d'adapter son action au type d'information qu'il recherche. Elle s'effectue en spécifiant des paramètres pour chacun des comportements de l'agent.

4.7.1 Utilité de la spécialisation

La spécialisation permet de définir plusieurs catégories, ou types, d'agents différents. Chacun de ces agents est défini selon les besoins de l'application. Il faut discerner deux besoins différents dans cette définition des agents :

1. Reconnaître les *composantes* de l'application.
2. Retrouver des *entités intermédiaires* pour aider la reconnaissance des composantes.

Le premier besoin est facile à comprendre. Toutes les composantes utiles pour l'application concernée doivent être définies. Ces composantes représentent la finalité du traitement, du point de vue de l'utilisateur du système. Il s'agit des différents constituants des images qui sont importants pour sa compréhension. Cependant, pour arriver à reconnaître ces composantes, il faut parfois passer par des constructions intermédiaires, d'où

le deuxième besoin exprimé. Ces constructions intermédiaires permettent en fait de rassembler certaines informations avant de pouvoir identifier précisément les composantes de l'application. Il peut s'agir d'entités qui seront interprétées par la suite dans le cadre du comportement de différenciation ou simplement d'entités sur lesquelles viendront s'appuyer la recherche des composantes dans le cadre du comportement de reproduction. Ces deux besoins ont donné lieu aux différents niveaux de représentation de l'information présentés au paragraphe 2.5.2.

4.7.2 Connaissances fournies par la spécialisation

Deux types de connaissances sont utilisés pour définir l'agent dans son action [Crevier 97] :

- la *connaissance opératoire* : elle provient d'un fichier de configuration et fournit les paramètres nécessaires aux agents pour comprendre l'environnement dans lequel ils sont immergés (*spécialisation a priori*) ;
- la *connaissance descriptive* : elle provient des structures d'informations (paragraphe 2.5) et permet d'adapter finement au contexte la spécialisation indiquée pour l'agent (*adaptation dynamique*).

Les notions de spécialisation et d'adaptation sont complémentaires dans la définition de l'agent. En effet, il est impossible de spécifier d'avance tous les paramètres dont aura besoin l'agent pour accomplir son travail. Ceux-ci sont trop dépendants du contexte précis dans lequel l'agent se trouvera, c'est-à-dire l'image, qui peut être différente d'une acquisition à l'autre et également la position dans l'image, puisque les composantes présentent des formes et des luminosité variables à l'intérieur même d'une image. Les paramètres qui ne pourront pas être fixés par spécialisation le seront plus tard par adaptation (tel que discuté au chapitre 3). La spécialisation donne cependant des indications sur l'adaptation qui devra être effectuée, en spécifiant les paramètres ambigus et en donnant les principes d'adaptation qui devront être utilisés en cours d'exécution.

La connaissance qui est fournie au système relève essentiellement du domaine du traitement d'images [Boucher 98b]. Il s'agit d'une connaissance opératoire de bas niveau. Certaines connaissances du domaine d'application sont définies, comme la morphologie globale de la cellule pour aider le système dans sa stratégie. Les différents types de connaissance sont entremêlés dans la définition des agents, pour leur donner un caractère opératoire plutôt que déclaratif.

4.7.3 Définition du fichier de configuration

Plusieurs types de paramètres peuvent être spécifiés par voie de configuration. Seuls les paramètres relatifs à l'action du système sont abordés dans ce paragraphe. Il existe d'autres paramètres permettant de réguler l'interface qui sont expliqués au paragraphe 5.2.3.

```

Agent Région                                :   Pseudopode

# PARAMÈTRES GLOBAUX
Composante groupe                           :   oui
Segmentation intermédiaire                  :   non
Priorité de l'agent                         :   1

# PARAMÈTRES PERCEPTION
Comportement perception actif               :   oui

# perception évaluation
Seuil de l'évaluation                       :   0.5
Critère perception                         :   0.4 seuillage variance > Fond
Critère perception                         :   0.4 seuillage source < Fond
Critère perception                         :   0.2 compacité

# perception autres
Perception précédente utilisée              :   oui
Remplissage des trous                      :   oui
Adaptation selon info cellule               :   non
Taille minimale pour conserver             :   10

# PARAMÈTRES INTERACTION
Comportement fusion actif                   :   oui

# PARAMÈTRES DIFFÉRENCIATION
Comportement différenciation actif          :   non

# PARAMÈTRES REPRODUCTION
Comportement reproduction actif             :   oui

# reproduction évaluation
Seuil de l'évaluation                       :   0.35
Critère reproduction                       :   1.0 extréma global source < Fond

# reproduction autres
Reproduction même image fin                :   Cytoplasme x 7
Reproduction même image fin                :   Crête x 12
Reproduction même image fin                :   Pseudopode x 12
Reproduction même image fin                :   Mouvement x 15

```

FIG. 4.37 – Exemple de configuration pour l'agent pseudopode.

La configuration permet de définir autant de types différents d'agents qu'il est utile pour l'application. L'action de ces paramètres se situe au niveau de l'agent dans sa globalité et au niveau de chacun de ses comportements. La figure 4.38 donne un aperçu du nombre de paramètres existants pour chacun des comportements, ainsi que des paramètres globaux au niveau de l'agent.

La figure 4.37 donne un exemple de spécialisation par configuration pour un type d'agent défini dans notre application. La configuration complète de tous les agents est donné en annexe A. Plusieurs de ces paramètres ont déjà été expliqués avec les comportements associés (chapitres 3 et 4). Mais il est utile de résumer les principaux critères ou catégories de critères que l'on retrouve dans cette configuration :

- Paramètres globaux à l'agent :
- *composante groupe* : indique si les informations de l'agent appartiennent ou non

- à un groupement perceptuel ;
- *segmentation intermédiaire* : indique le niveau de représentation de l'information produite par l'agent (intermédiaire ou domaine) ;
- *priorité de l'agent* : indique la priorité donnée à l'agent par le séquenceur, donc le temps accordé à l'agent par rapport aux autres.
- Paramètres des comportements :
 - *comportement actif* : permet d'inhiber certains comportements, donc d'interdire à l'agent de les utiliser ;
 - *paramètres d'évaluation* : pour les comportements de gestion de l'information, il est possible de spécifier les critères, les poids de ces critères ainsi que le seuil de la fonction globale d'évaluation ;
 - *autres paramètres du comportement* : plusieurs autres paramètres, dépendant de chaque comportement, peuvent être donnés afin de spécialiser celui-ci.

Les paramètres globaux de l'agent permettent principalement de déterminer sa place et son importance dans la société d'agent. Cela permet de situer ses liens par rapport aux autres (groupe), de situer le niveau de ses résultats et l'importance donnée à son action (priorité).

Chaque comportement de l'agent peut être inhibé, même le comportement de perception, qui a pourtant été présenté comme le comportement vital de l'agent (paragraphe 3.5). Même sans comportement de perception, il peut encore être possible d'exécuter le comportement de reproduction. Il est tout à fait possible d'imaginer un agent dont le but est d'explorer l'environnement et lancer d'autres agents. Cela sera discuté au chapitre 5. Certains autres comportements n'ont de sens que pour des situations précises, comme le comportement de différenciation qui ne vaut que pour des agents travaillant sur des informations d'un niveau intermédiaire de représentation.

Tous les paramètres des comportements vont spécifier l'action précise qu'aura l'agent. Il est possible d'accorder plus d'importance à un comportement pour un agent et le contraire pour un autre. La spécialisation des comportements permet de fixer le rôle qu'aura l'agent par rapport aux autres : certains seront plus aptes à explorer l'image en mode reproduction tandis que d'autres ne feront que de la perception (segmentation). À partir du même modèle d'agent, plusieurs finalités d'agents peuvent être ainsi créées.

L'adaptation n'est pas évoquée directement dans les paramètres de configuration. En fait, c'est en observant certains paramètres que l'adaptation qui sera faite peut être perçue. Par exemple, en observant les critères d'évaluation du comportement de perception dans la figure 4.37, on remarque que les critères font référence à une autre spécialisation se trouvant dans le même fichier, soit ici le *Fond*. Il est spécifié que les critères seront évalués par rapport à des informations de cette composante, qui n'est pas encore connue précisément.

Comportement	Nb de critères	Critères paramétrables (nb de paramètres)	Paramètres de contrôle
<i>Perception</i>	région : 4 contour : 4	oui (2-5)	région : 5 contour : 3
<i>Interaction</i>	-	-	région : 1 contour : 2
<i>Différenciation</i>	région : 3 contour : 0	poids seulement	région : 3 contour : 0
<i>Reproduction</i>	4	oui (5-6)	4
<i>Agent (global)</i>	-	-	3

FIG. 4.38 – Résumé du nombre de critères pour chaque comportement, ainsi que du nombre de critères pour l'agent (autres que ceux des comportements). Les paramètres de présentation de l'interface sont exclus de ce tableau. Seuls les paramètres influençant le fonctionnement du système (et ses résultats) ont été comptabilisés. Les paramètres globaux du système n'ont pas été comptabilisés. Ces paramètres servent essentiellement à la présentation de l'interface graphique.

Sa présentation dans le fichier est semblable à celle présentée ici, mais avec des valeurs différentes pour les paramètres. C'est à l'exécution que seront connues les valeurs exactes permettant ensuite d'utiliser les critères d'évaluation. Le fichier de configuration permet ici de paramétrer un comportement en référence avec une autre composante. C'est seulement à l'exécution que le comportement s'adaptera en recherchant les valeurs exactes pour se configurer.

4.8 Positionnement du système

Dans les quatre premiers chapitres de ce travail, nous avons décrit le système conçu selon les différents aspects de la représentation des informations, des méthodes de traitement et du contrôle du système. Diverses philosophies de vision, ainsi que plusieurs systèmes existants, ont été présentés. Notre système se situe de façon originale à l'intersection de plusieurs concepts évoqués précédemment.

Le découpage du système en agents, où chaque agent représente un concept de l'image, s'apparente à la philosophie du système Schéma (dans VISIONS), où chaque schéma est associé à un concept et possède et spécialise les méthodes de traitement associées. La logique d'exploration de l'image s'apparente également à ce système, puisque dans les schémas, des hypothèses sont générées pour la recherche d'autres concepts associés aux schémas instanciés, tandis que dans notre système, la reproduction permet à un agent de rechercher d'autres concepts avec une logique précise.

La différence principale avec ce système, et avec plusieurs autres systèmes existants, réside dans le repliement du bas niveau sur le haut niveau. En effet, dans notre système, la représentation par niveaux n'est effectuée que sur un plan logique de représentation

de l'information et n'a pas d'incidence sur le plan fonctionnel du système. Un agent est associé à un concept de l'image ou de l'application, et quel que soit ce concept, il conserve la même architecture, les mêmes capacités et surtout, les mêmes rapports avec le reste des agents. Plus précisément, il articule pour chaque concept la mise en oeuvre des méthodes de perception, fusion, reconnaissance et focalisation associées définissant ainsi une stratégie complète de vision au niveau de chacun d'eux.

En fait, si le découpage en concepts de notre système s'apparente à des systèmes haut niveau, le contrôle repose sur des principes émanant de systèmes bas niveau, comme celui du système expert de Nazif et Levine, du système multi-processus de Bellet, ou bien de l'approche comportementale de Brooks.

L'originalité du système se situe dans cette articulation entre le bas niveau et le haut niveau. Alors que dans les systèmes existants, ces deux niveaux sont souvent séparés, autant dans la représentation des informations que dans les méthodes et traitements employés, nous voulons les réconcilier et pallier le manque d'informations d'un niveau par les indices trouvés dans l'autre et vice-versa.

La séparation entre les méthodes de gestion de l'information et les stratégies de contrôle n'est pas évidente dans notre système. Le meilleur exemple est sans doute le comportement de reproduction présenté au paragraphe 4.4 qui utilise une évaluation définie pour la gestion de l'information à des fins de stratégie et de contrôle de la société d'agents. Le but recherché est d'assouplir la rigidité souvent employée dans la définition des entités, afin d'assurer une plus grande réactivité des agents, tant au niveau décisionnel que fonctionnel. Un objectif espéré de cela, mais pas forcément atteint, serait de voir l'émergence de nouvelles aptitudes de cette formulation libre du système.

Cet aspect de liberté, nous l'avons placé au début pour le traitement de l'information. Il est associé au problème du choix de l'information et de la prise de décision qu'évoquait Bajcsy [Bajcsy 88]. Par contre, au lieu de poser ce problème en termes de boucles de contrôle et de rétroaction, nous avons préféré l'exprimer en termes d'accessibilité de l'information et de complémentarité de celle-ci, en reliant les niveaux sur eux-mêmes afin d'ouvrir de nouvelles possibilités de traitement.

Le parallèle peut également être fait avec la vision intentionnelle proposée par Aloimonos [Aloimonos 90]. Dans cette approche, le système exprime ses buts à un haut niveau de description du système et la représentation des informations est orientée directement vers ses buts. Notre conception est dirigée selon cette approche, tout en permettant de décrire ces buts haut niveau avec des moyens propres au bas niveau, et ainsi faire le lien entre ceux-ci. Chaque agent décrit une intention de reconnaissance d'un concept, quel que soit son niveau, et l'expression de cette intention s'effectue en termes de paramétrage de comportements bas niveau.

Cette définition d'un système par une société de plusieurs agents réunit plusieurs principes énoncés par Jolion dans son approche systémique [Jolion 94]. Parmi les principes

évoqués, nous nous reconnaissons aisément dans les suivants : *conserver de la variété dans le système ; conserver l'équilibre global par des actions locales ; varier pour mieux unifier ; être adaptatif*. Les objectifs ainsi poursuivis sont de définir un système complet comme étant plus que la somme de ses composantes, et de miser sur la complexité du système, par ses interactions, pour obtenir le résultat désiré.

Chapitre 5

Application à la reconnaissance de cellules vivantes

Le système réalisé a été décrit de façon générale dans les précédents chapitres. Ce chapitre présente maintenant les fondements de l'application qui ont permis son élaboration. L'application étudiée est la reconnaissance de cellules vivantes, incluant la segmentation, l'interprétation en composantes, ainsi que le suivi de ces cellules.

Après avoir décrit cette application et les images associées, le système sera présenté par rapport à celle-ci, c'est-à-dire en définissant les différents concepts de l'application et leur réalité dans le système. Ensuite, l'application servira à illustrer différents aspects du système qui ont déjà été discutés auparavant, tels que l'adaptation et la coopération. Les résultats permettent également d'évaluer ce système, ce qui est fait à l'aide de cartes de références établies par des opérateurs humains.

5.1 L'application cytologique

L'application cytologique existe en vision par ordinateur depuis les années 1960. Malgré les travaux qui se sont succédés, les difficultés posées par celle-ci n'ont toujours pas été résolues. Après avoir exposé les objectifs de notre système pour cette application et présenté les modes d'acquisition des images que nous traitons, quelques approches liées à cette application sont présentées. Ensuite, les principales difficultés posées par les images sont énumérées.

5.1.1 Objectif de l'étude

Ce travail de recherche s'inscrit dans un projet global d'étude quantitative du comportement migratoire cellulaire. La déformation d'une cellule et la modulation de ses propriétés migratoires mettent en oeuvre des processus dynamiques aux niveaux moléculaire et cellulaire. L'analyse quantitative de ces déformations et des mouvements de la cellule

apparaît comme un outil essentiel pour améliorer la compréhension des phénomènes migratoires cellulaires ainsi que pour différencier et contrôler les situations physiologiques et pathologiques affectant cette migration.

Les outils actuellement disponibles pour étudier à un niveau microscopique les différents aspects de la motilité et des déformations cellulaires restent extrêmement limités car de nature essentiellement statique. Outre les méthodes de dénombrement et de caractérisation de cellules à un instant donné, les méthodes utilisant des marqueurs de l'activité cellulaire (division, migration) sont généralement mises en oeuvre à partir de cellules fixées. À l'opposé, les approches macroscopiques de la migration proposent une quantification basée sur l'estimation de densités moyennes de cellules qui exclut une analyse fine de la dynamique cellulaire dans l'espace. Le dépouillement de l'ensemble de ces expériences est actuellement très long et d'une fiabilité limitée compte tenu de la quantité d'informations à analyser et du faible degré d'automatisation des environnements de traitement des images.

L'analyse des cellules à partir d'une séquence d'images constitue donc un élément clé pour l'interprétation de scènes complexes en biologie et en médecine. L'utilisation des techniques d'analyse du mouvement reste cependant très marginale dans le secteur biomédical, alors même qu'elles apparaissent comme déterminantes pour l'étude de la déformation et de la migration de cellules. On peut avancer comme explication de cet état de fait que contrairement au contexte industriel où les objets suivis sont le plus souvent rigides, les objets biologiques que sont les cellules se déforment fréquemment et rapidement, aussi bien dans les phases de division cellulaire (les cellules s'arrondissent au préalable) que dans les phases de migration active (projection d'extensions cytoplasmiques, lamellipodes, filopodes, ...). La difficulté méthodologique est donc accrue par le caractère déformable des objets étudiés.

Ce projet est mené en collaboration avec Anne Doisy et Xavier Ronot, biologistes de l'équipe DyOGen (INSERM U309 et EPHE) de l'Institut Albert Bonniot. Le but est de fournir des méthodes quantitatives permettant de suivre la cellule en migration et d'analyser ses mouvements et de ses déformations. Cette recherche s'inscrit dans un cadre plus large d'étude quantitative du comportement migratoire cellulaire. À cet égard, d'autres travaux sont en cours au sein du laboratoire TIMC/IMAG, portant notamment sur les contours actifs [Leitner 95] et sur le flot optique [Germain 98, Doisy 89]. Le système présenté dans ce travail s'ajoute aux différents moyens d'études qui sont développés.

5.1.2 L'acquisition des images

Pour l'acquisition des séquences d'images, une station d'imagerie, comprenant un microscope à inversion (Axiovert 135M, Carl Zeiss) équipé d'une chambre d'incubation, et d'une caméra SIT (C2400-08, Hamamatsu), a été utilisée. Le tout est relié à une station d'imagerie (SAMBAs 2640, Unilog).

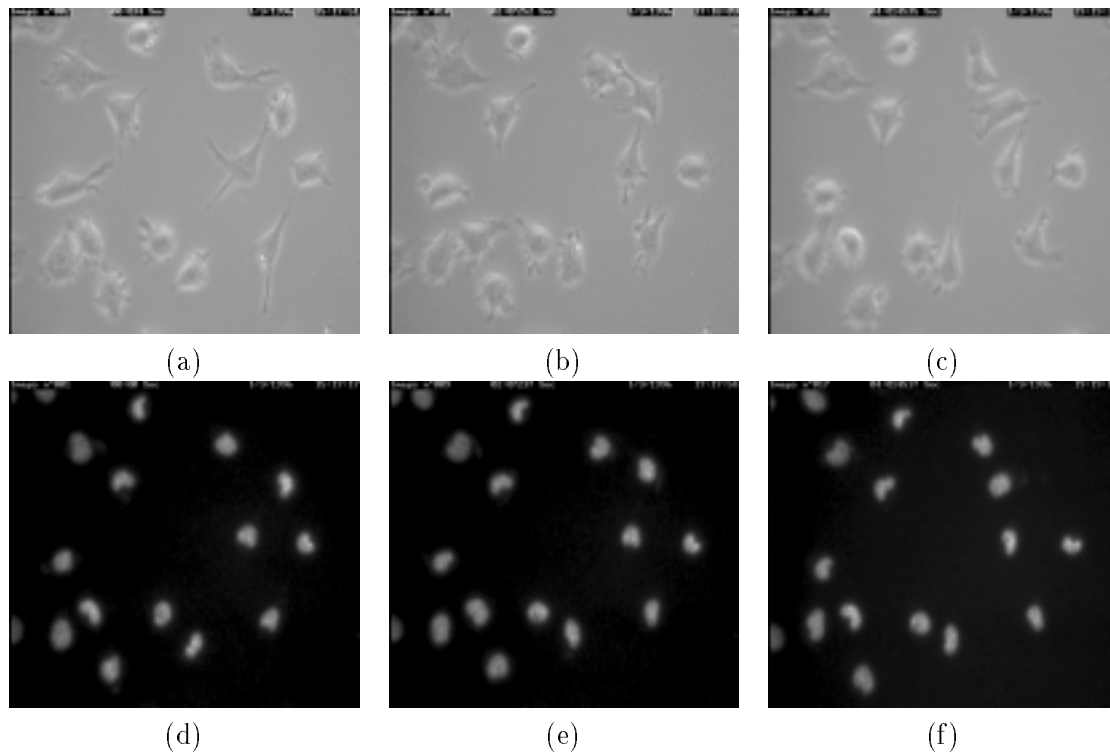


FIG. 5.39 – Séquence d'images de fibroblastes murins L929. (a) (d) Images des cellules et des noyaux à $t=0$. (b) (e) Images des cellules et des noyaux à $t=2h$. (c) (f) Images des cellules et des noyaux à $t=4h$. Les deux images (cellules et noyaux) sont acquises simultanément. Une séquence d'images est constituée d'un ensemble de paires d'images (cellules et noyaux). On note que les déplacements des cellules sont moins importants que leurs déformations.

Deux types de cellules en culture ont été étudiés pour ce travail :

- la majorité des expériences ont été menées sur des cultures de fibroblastes murins (L929, ATCC, CCL1) (figure 5.39) ;
- quelques expériences complémentaires ont été effectuées sur des cultures de cellules de carcinome épithéloïde humain (HeLa, ATCC, CCL2) (figure 5.53).

Les images sont acquises à des intervalles de temps réguliers (de 10 à 30 minutes selon les séquences) sur plusieurs heures. La figure 5.39 illustre quelques images d'une séquence. Deux images sont prises à chaque acquisition :

- Une image en contraste de phase montre les cellules entières (figure 5.39a, b et c.). Cette image est toujours présente et constitue la première source d'information utilisée.
- Une image en fluorescence montre les noyaux de ces cellules marqués avec un fluorochrome de l'ADN (Hoechst 33342) (figure 5.39d, e et f.). Cette image complémentaire

peut être présente ou non, selon les séquences étudiées.

Nous effectuons l'analyse des séquences d'images en post-traitement sur des stations de travail SUN. De plus amples détails sur les méthodes d'acquisition des images peuvent être trouvés dans [Boucher 98a] et [Leitner 95].

5.1.3 Approches existantes

Les premiers travaux effectués sur des applications cytologiques utilisaient des méthodes de segmentation simples et bas niveau pour tenter d'extraire automatiquement les cellules, comme des techniques de seuillage, de détection de contours ou de régions [Fu 81]. Cependant, ces méthodes ont obtenu des résultats limités parce que trop générales et parce qu'elles ne prenaient pas assez en compte l'application. Ensuite, il a été question d'insérer des connaissances a priori sur l'application [Liedtke 87] et de développer certains mécanismes de raisonnement [Lockett 94] pour réussir la tâche d'identification des cellules.

Les approches précédemment citées ont en commun la détection de cellules composées d'un noyau avec cytoplasme limité sans extensions, il s'agit donc de cellules assez simples à étudier du point de vue du traitement d'images. Les travaux qui ont suivis concernent des cellules plus complexes, d'un point de vue morphologique, avec un ou plusieurs prolongements. L'analyse de telles cellules requiert la mise en oeuvre de méthodes de segmentation et de raisonnement beaucoup plus complexes.

Wu et al. ont utilisé un opérateur de variance qu'ils adaptent pour la segmentation de cellules vivantes dans un milieu semi-solide constitué de collagène [Wu 95]. La méthode qu'ils ont développée comporte deux étapes :

1. Détection et extraction d'une zone approximative contenant la cellule et le fond entourant celle-ci. Ce masque approximatif de la cellule est trouvé en utilisant un opérateur de variance sur une fenêtre de taille 41x41 sur toute l'image, pour détecter la position approximative de la cellule.
2. Ensuite, la séparation de la cellule du fond est fondée sur une analyse statistique de la variance, permettant de calculer un seuil différenciant la cellule du fond, dans le masque précédemment établi.

Zhou et Pycock ont proposé une méthode d'exploration de l'image et de segmentation des frontières des cellules dans l'image [Zhou 97]. Ils modélisent d'abord la frontière de la cellule par une série de points échantillonnés sur celle-ci. La distribution des niveaux de gris sur ces différents points est prise en compte. Ce modèle déformable est invariant aux changements d'échelle, de translation, de rotation et de contraste. Le but de la segmentation est de retrouver, à partir du centre de la cellule, les points correspondant le plus au modèle existant. Pour réaliser cela, une méthode d'exploration par évaluation de profils

est utilisée. Les pixels sont évalués le long des profils, en tenant compte des caractéristiques internes et externes de la cellule. L'originalité de cette approche est la définition de plusieurs critères statistiques adaptatifs pour évaluer les frontières des cellules et séparer les différentes cellules. Les points trouvés sur les différents profils sont mis en relation, afin d'assurer la cohérence de la frontière.

Cloppet-Oliva a présenté une approche coopérative région/contour pour segmenter et surtout identifier les différents prolongements de cellules neuronales [Cloppet-Oliva 96a]. La première étape de son approche, après amélioration du contraste, consiste à extraire une image de contours, à l'aide de l'opérateur de Deriche. Ensuite, la deuxième étape est d'effectuer une croissance de région par déformation géométrique de contours. En fait, il s'agit d'une variante des approches par contours actifs, employée cette fois pour obtenir une région, que l'on obtient par expansion du contour. Un mécanisme de gestion des intersections permet de faire croître la région et de la diviser en sous-régions au croisement de plusieurs objets. L'étape suivante consiste à combiner les résultats de l'approche contour avec ceux de l'approche région afin de bien réaliser la segmentation des cellules et de leurs prolongements, les deux méthodes se complétant au niveau des résultats. L'étape d'identification qui suit a pour but de rattacher les prolongements aux cellules et de séparer les cellules. L'utilisation d'un réseau bissecteur pour squelettiser les prolongements, combiné à des règles heuristiques de gestion des fourches permet d'obtenir la séparation des neurones des cellules étudiées. Outre la méthode coopérative employée, la contribution de ce travail se situe dans le mécanisme élaboré permettant de reconnaître individuellement chaque cellule et ses prolongements.

Enfin, ces dernières années, un certain nombre de méthodes ont été développées afin de traiter des séquences d'images de cellules, pour suivre leur déplacement. Gwydir et al. identifient trois catégories de méthodes pour étudier le mouvement non-rigide [Gwydir 94] :

1. les techniques différentielles géométriques ;
2. les modèles contraints globalement sur la forme et/ou le mouvement ;
3. les approches par minimisation d'énergie, comme les contours actifs.

La majorité des travaux présentés sur l'application des cellules en mouvement ont utilisé la troisième approche. Leymarie et al. utilisent un modèle de contour actif qu'ils initialisent manuellement en traçant, sur la première image, une courbe proche de la cellule [Leymarie 93]. Le contour peut ensuite se réinitialiser d'image en image d'après le résultat de l'image précédente. Gwydir et al. s'intéresse aux cellules neuronales avec une approche semblable [Gwydir 94]. Enfin, Leitner et al. ont proposé une fonction d'énergie plus robuste pour la croissance du contour actif [Leitner 95]. Celui-ci est moins sensible aux conditions d'initialisation. Ils ont appliqué leur approche aux cultures de cellules de fibroblastes murins, qui sont de même type que les cellules qui nous intéressent.

De tous ces travaux, on peut retenir trois idées essentielles pour un système de segmentation et de détection de cellules. Ces idées sont les mêmes que pour toutes les applications de vision, telles que nous les avons présentées précédemment :

- l’*adaptation* des méthodes de segmentation [Wu 95, Zhou 97] ;
- la *coopération* entre différentes méthodes [Cloppet-Oliva 96a] ;
- la *spécialisation* des méthodes pour l’application cytologique [Cloppet-Oliva 96a, Zhou 97].

5.1.4 Difficultés de l’application

Les différents auteurs des approches présentées ci-dessus reviennent fréquemment sur les difficultés et sur la complexité des images étudiées. Parmi les principales caractéristiques de ces images, notons [Cloppet-Oliva 96b, Zhou 97, Wu 95] :

- le contraste entre les cellules et le fond de l’image est très faible (identifiable par un histogramme unimodal des niveaux d’intensité des images) ;
- l’intensité du fond de l’image est non-uniforme, en raison de la source de lumière utilisée dans les microscopes optiques ;
- à l’intérieur même d’une cellule, des variations importantes d’intensité existent ;
- la frontière exacte de la cellule est rendue floue par la présence d’un effet optique de réfringence de la source lumineuse sur les bords de la cellule qui se traduit par la présence d’un halo lumineux masquant une partie de la cellule et du fond de l’image ;
- la déformation de la cellule dans le temps rend plus difficile sa reconnaissance temporelle.

5.2 Le système réalisé

Les paragraphes qui suivent replacent le système dans le contexte de l’application qui nous intéresse. Après avoir traité des aspects d’implantation du système, les objectifs et définitions des agents pour l’application seront expliqués. Un exemple d’application est présenté comme étude possible grâce au système développé.

5.2.1 Développement du système

Le système multi-agents que nous proposons est implanté dans un seul processus UNIX. Nous avons choisi de regrouper tous les agents à l’intérieur d’un même processus plutôt

que de les distribuer sur plusieurs processus pour deux raisons :

1. Le nombre d'agents dans le système est très élevé. Il peut y avoir plusieurs dizaines d'agents de segmentation par cellule et par image. À ce rythme, le nombre maximum de processus pour un processeur est rapidement atteint et les différentes ressources disponibles dans le système d'exploitation pour gérer un tel système (sémaphore, mémoire partagée, file de messages) ne suffisent plus.
2. Cette approche nous procure l'avantage de contrôler les conditions d'exécution et de simulation du système.

Ce choix nous a contraint à implanter notre propre système d'exploitation, ou séquenceur, à l'intérieur du système. C'est pour cette raison que le séquenceur décrit au paragraphe 4.6 ne réalise que des tâches d'ordonnancement simples non-dédiées au système développé.

Le système a été réalisé en C++ sur des stations SUN SPARC. La modélisation par objets nous a permis de concevoir et de maintenir aisément le système. En effet, notre philosophie distribuée multi-agents s'adapte naturellement à une modélisation par objets, avec un héritage aisé des différents concepts, tant au niveau des agents qu'au niveau des comportements.

Le traitement des séquences d'images est le but de ce projet. Par contre, le système réalisé est plus focalisé sur les aspects impliquant le traitement d'une image à la fois. Il s'agit d'un point qui peut être amélioré, afin de prendre mieux en compte l'aspect temporel de la séquence d'images. La séquence d'images est pour l'instant utilisée pour le calcul d'informations mouvement, mais pas pour créer des liens entre agents travaillant sur des images différentes afin d'assurer la cohérence des résultats.

Compte tenu des contraintes imposées par l'application (taille des séquences, intervalle d'acquisition), le système peut effectuer le traitement des séquences en *temps réel*. En supprimant l'interface du système, la segmentation complète d'une image 320x320 prend environ une minute, sur une station Ultra Sparc. Ce temps n'est pas considéré comme prohibitif pour l'application qui nous concerne.

5.2.2 Objectifs et spécialisations des agents

Le but de l'application est de segmenter et d'interpréter les différentes cellules présentes dans les séquences d'images. Pour reconnaître ces cellules, il est apparu judicieux de les diviser en différentes composantes, et d'utiliser ainsi une société d'agents pour la reconnaissance de chaque cellule, où chaque agent est dédié à la détection d'une primitive (ou composante) d'un type donné.

Les images des séquences sont divisées en cinq composantes (figure 5.40) :

- Le **noyau** est le coeur de la cellule. Sa segmentation est facilitée par l'image en

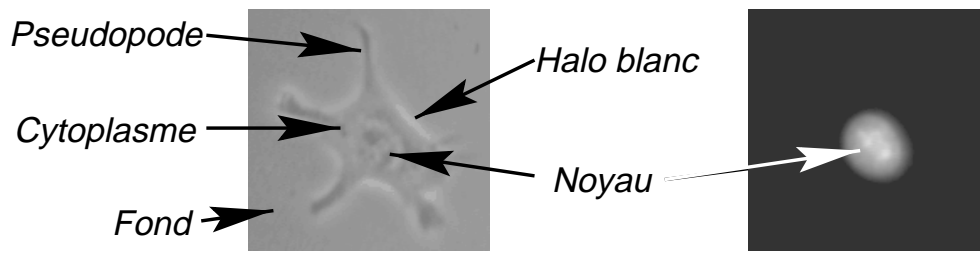


FIG. 5.40 – Composantes identifiées dans les images cellulaires. Les deux images sont mises en commun pour cette identification.

fluorescence. Le mouvement du noyau, dans la séquence d'images, est assez faible.

- Le **cytoplasme** représente toute la partie de la cellule qui entoure le noyau. Sa segmentation est difficile, car il est très peu homogène et le contraste entre celui-ci, les pseudopodes et le fond est très faible.
- Les **pseudopodes** sont les prolongements cytoplasmiques de la cellule. Le contraste entre le fond et les pseudopodes est très faible, mais détectable (ils sont normalement plus sombres que le fond). Leur mouvement est important. La frontière entre les pseudopodes et le cytoplasme est très faible, voire inexistante. Elle ne pourra pas être identifiée précisément, mais cela n'est pas gênant pour l'application.
- Les **halos blancs** entourent la cellule et définissent une frontière floue de celle-ci. Ils résultent de la réfraction de la lumière sur la cellule. Ils sont inclus dans notre définition de la cellule.
- Le **fond** de l'image entoure les cellules. Il est bruité et non-uniforme sur l'image.

Cette décomposition de la cellule en composantes a été réalisée sur une base expérimentale, afin d'optimiser la coopération possible entre des agents de différents types sur la base d'une variété d'informations qu'ils produisent. En effet, cette décomposition est un objectif des traitements et en même temps un moyen d'y arriver. Certains types d'agents ont été définis d'abord comme fonctionnalité, plutôt que comme objectif à atteindre.

La définition des pseudopodes et du cytoplasme en est un exemple. Ces deux composantes sont difficiles à discerner précisément dans les images. Un pseudopode correspond à un prolongement du cytoplasme et ce dernier est un constituant majeur de la cellule. Mais la frontière entre les deux est difficile à fixer précisément. Nous avons choisi tout de même de définir deux types d'agents, pseudopode et cytoplasme, en leur donnant des caractéristiques différentes, et d'observer leurs interactions et le résultat obtenu. Selon les cellules étudiées, l'interaction entre ces deux types d'agents est différente, les agents pseudopodes segmentant habituellement les zones sombres de la cellule et les agents cytoplasmes les zones moins homogènes et plus claires.

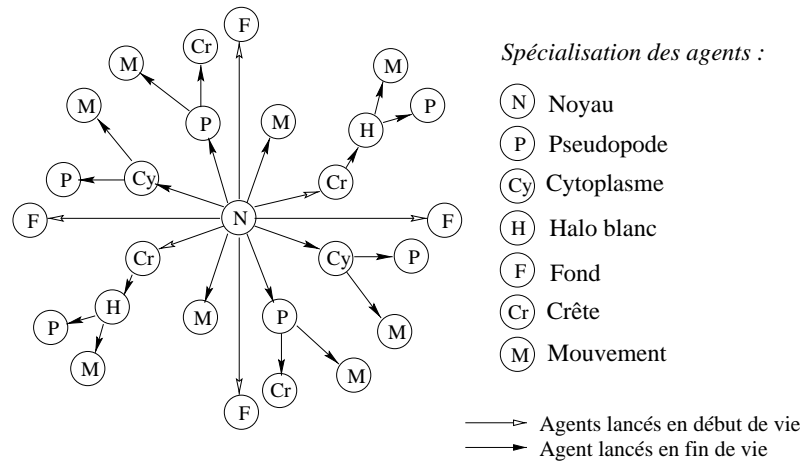


FIG. 5.41 – *Stratégie d'exploration de l'image implantée par le comportement de reproduction. Tous les agents contribuent à l'exploration de la cellule.*

Aux agents définis selon les composantes énumérées ci-dessus viennent s'ajouter deux autres types d'agents, basés sur des primitives intermédiaires, c'est-à-dire qui apportent une aide à la société d'agents, mais dont le résultat n'est pas utile pour l'application :

- Les **crêtes** sont les arêtes centrales des halos blancs. Il s'agit d'une ligne centrale aux halos qui aidera principalement à la focalisation des agents halos blancs.
- Le **mouvement** permet d'identifier des zones homogènes au sens du mouvement et de les segmenter avant de pouvoir les identifier en tant que composantes de l'application (paragraphe 3.7).

Tous les agents spécialisés pour les primitives et composantes nommées ci-dessus, sauf l'agent crête, sont basés sur des primitives régions. L'agent crête est le seul agent basé sur une primitive contour. Les régions ont semblé, pour cette application, plus riches en informations sémantiques que les contours pour décrire une réalité de l'application. Les contours, soit les agents crêtes, vont servir d'aide au processus de décision, en apportant un complément d'information aux autres agents existants. Il s'agit d'un choix de développement, car d'autres configurations d'agents seraient possibles.

La notion de groupement perceptuel pour notre application se définit naturellement par les cellules. Chaque cellule consiste en un groupement perceptuel qui inclut les types d'agents suivants : noyau, cytoplasme, pseudopode, halo blanc et crête. Même si les crêtes ne sont qu'une information intermédiaire, nous les considérons dans la cellule, puisqu'il s'agit d'une information propre à celle-ci. Les agents mouvement ne peuvent être considérés dans la cellule avant leur différenciation, car certains d'entre eux prendront l'identité du fond, qui n'est pas une composante cellulaire.

La stratégie d'exploration de l'image, définie dans le comportement de reproduction (paragraphe 4.4), est fortement centrée sur l'agent noyau (figure 5.41). Ce choix s'explique

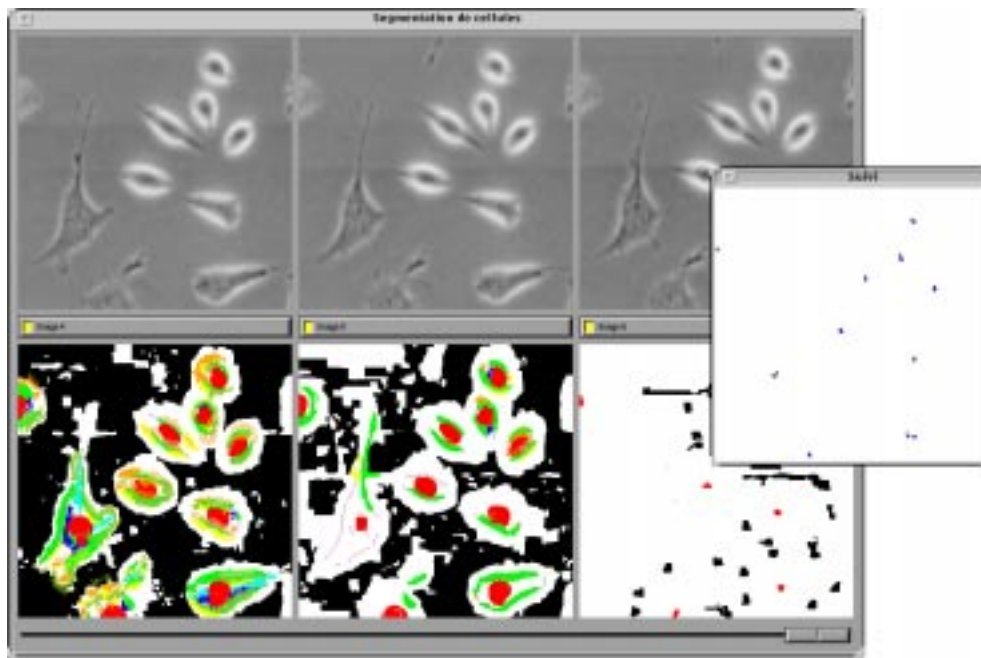


FIG. 5.42 – *Interface du système. Deux fenêtres sont affichées, l'une permettant de visualiser la séquence d'images en cours de segmentation et de reconnaissance et l'autre permettant de suivre le centre de gravité des composantes (ici les noyaux) sur la séquence complète. Diverses options sont possibles pour utiliser cette interface.*

par deux raisons :

- le graphe d'exploration est inspiré de l'application, des objets observés dans l'image, et le noyau, en étant central dans la cellule, semble normal comme pivot de l'exploration, d'autant plus qu'il n'existe qu'un seul noyau par cellule ;
- la segmentation du noyau est assez aisée à réaliser grâce à l'image source montrant les noyaux en fluorescence.

L'image en fluorescence des noyaux peut être présente ou non selon les séquences d'images étudiées. En cas d'absence de ces images, le comportement de perception des agents noyaux sera inhibé, mais il conservera son rôle central dans la reproduction. L'initialisation de tels germes noyaux est discuté au paragraphe 5.2.4.

5.2.3 L'interface du système

Une interface a été réalisée pour permettre à l'utilisateur de mieux suivre et de contrôler l'exécution du système. Elle repose sur deux composantes :

1. un *fichier de configuration* permettant de donner les paramètres des différents agents et de l'exécution (détaillé au paragraphe 4.7) ;

2. une *interface visuelle* permettant de suivre l'exécution du système et de visualiser différents paramètres.

L'interface visuelle est composée de deux fenêtres complémentaires (figure 5.42), l'une permettant de suivre la segmentation et l'autre de voir le déplacement des composantes tout au long de la séquence (paragraphe 5.2.6). La séquence d'images est traitée en pipeline et l'utilisateur peut suivre le déroulement des traitements selon différentes options de configuration :

- affichage ou non du schéma de lancement de germes par les différents agents ;
- superposition ou non de la segmentation sur l'image source ;
- tracé des périmètres des différentes primitives segmentées ;
- affichage superposé ou non de la segmentation de niveau intermédiaire sur la segmentation de niveau composante.

5.2.4 L'initialisation des premiers germes

Le problème de l'initialisation des premiers germes d'agents sur la première image a déjà été abordé au paragraphe 4.4.7, avec le comportement de reproduction. Deux solutions sont possibles :

1. Initialisation automatique, réalisable dans le cas où les premiers germes sont faciles à déterminer. Il s'agit de donner avec la configuration d'un agent les critères permettant de retrouver de façon globale les germes dans l'image (paragraphe 4.4.7).
2. Initialisation manuelle en demandant à l'utilisateur d'indiquer les premiers germes sur la première image, dans le cas où ceux-ci sont plus difficiles à détecter. Dans ce cas, l'utilisateur peut démarrer lui-même l'exécution en indiquant, sur l'interface, la position de ces premiers germes. Par la suite, le système continue de façon automatique.

Ces deux options ne valent que pour la première image, où il n'existe aucun agent, ni aucun résultat de traitement. Pour la suite des images de la séquence, les agents sont initialisés automatiquement à l'aide des informations récoltées dans l'image précédente. Dans notre application, ce sont les agents noyaux qui se chargent d'initialiser les agents noyaux dans l'image suivante lorsque leur propre segmentation est terminée.

5.2.5 Prétraitement des images

Un seul prétraitement est effectué sur les images sources en contraste de phase, dans le but de corriger la non-homogénéité du fond. Cette dégradation des images est due à

la source de lumière des microscopes optiques, qui se retrouve dans les images obtenues. Les images sont plus claires en leur centre et plus sombres en périphérie. La correction de ces images par un prétraitement permet de rétablir l'homogénéité des images, rendant ainsi les traitements plus cohérents et surtout, indépendants de la position de l'objet dans l'image.

La méthode que nous avons choisie est basée sur les travaux de Sibarita et de Mayet et al. [Sibarita 96, Mayet 96]. Elle consiste à interpoler le fond non-homogène des images par une fonction en forme de *cloche*, à partir de points préalablement ajustés par l'utilisateur. L'opération demande à cet utilisateur d'indiquer un certain nombre de points appartenant au fond pour ensuite interpoler une fonction approximant ce fond. Les paramètres obtenus peuvent être conservés et sont valides pour une séquence entière d'images.

La fonction décrite par Sibarita pour approximer le fond comporte six paramètres, et elle est de la forme [Sibarita 96] :

$$\exp(Pol(x, y))$$

où $Pol(x, y)$ est une fonction polynomiale du second degré, de la forme

$$Pol(x, y) = \sum_{i=0}^2 \sum_{j=0}^{2-i} k_{ij} x^i y^j$$

où les k_{ij} représentent les six paramètres de la fonction à déterminer.

Une fois les paramètres déterminés, on obtient une *image* du fond et de son illumination. La méthode consiste ensuite à soustraire le fond des images de la séquence. Cette soustraction est effectuée selon le modèle LIP (*Logarithmic Image Processing*), développé pour les opérations mathématiques sur des images en lumière transmise [Mayet 96]. Le fondement de ce modèle est que les images observées ne sont pas linéaires, autant d'un point de vue physique, que du point de vue de la perception visuelle humaine. La soustraction LIP de deux images est définie par :

$$f(x, y) - g(x, y) = 256 \frac{f(x, y) - g(x, y)}{256 - g(x, y)}$$

Dans notre cas, $f(x, y)$ est l'image source et $g(x, y)$ l'image du fond que nous avons calculé par interpolation. Le résultat de cette soustraction nous donne une image source plus homogène, avec un fond plus uniforme.

5.2.6 Exemple d'application (cicatrisation *in vitro*)

La figure 5.43 montre un exemple d'application pour lequel nous avons utilisé ce système. Il s'agit d'étudier la cicatrisation *in vitro*, c'est-à-dire la migration des cellules pour recoloniser une plaie artificielle. L'importance de cette application est dans l'utilisation d'outils de quantification de la migration cellulaire. Les données recueillies pendant de telles expériences *in vitro* permettent de développer des modèles de la cicatrisation qui

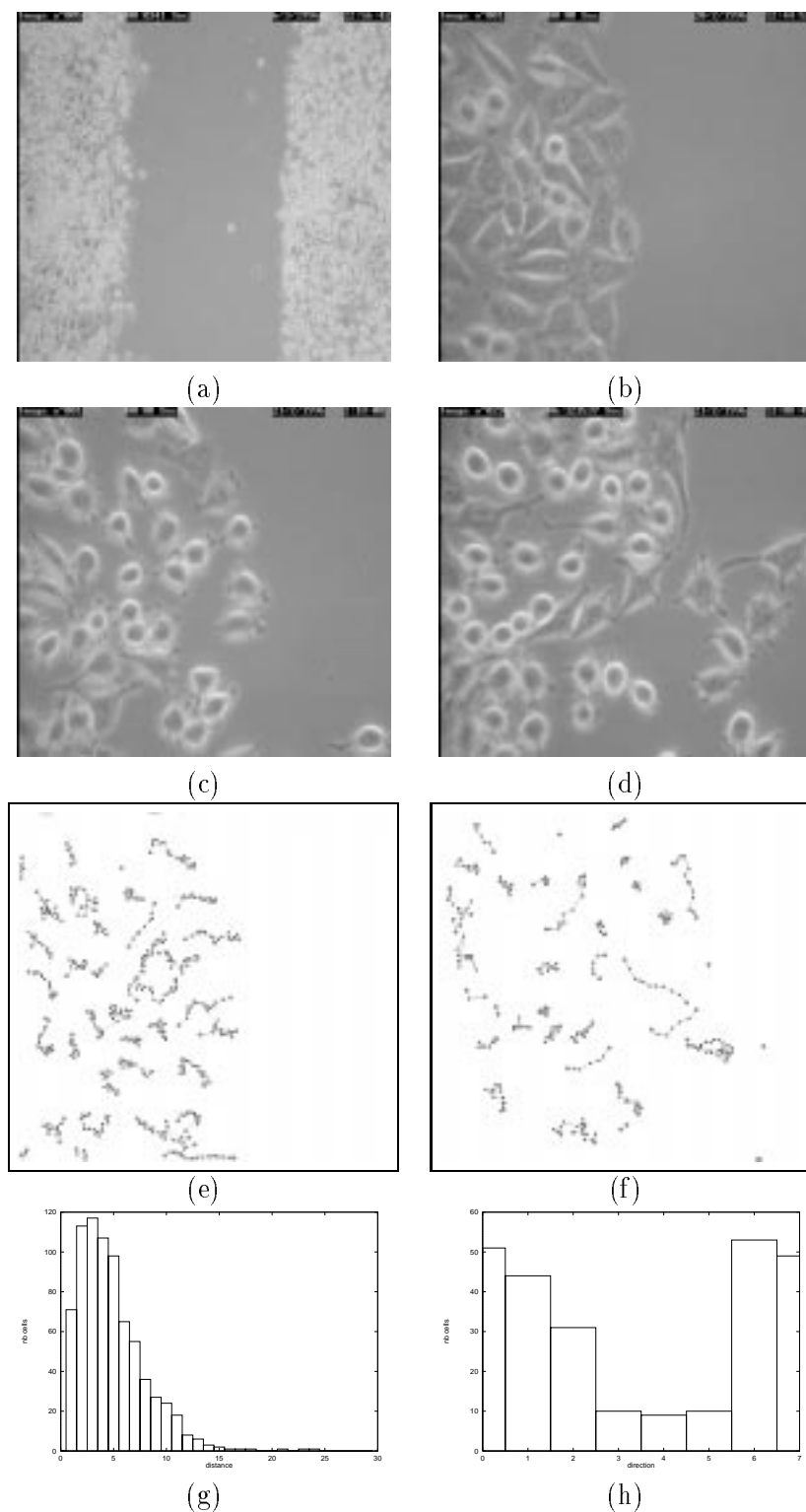


FIG. 5.43 – Le processus de cicatrisation *in vitro*. (a) Exemple de plaie artificielle dans une population de L929. (b) Position des cellules à $t=0$. (c) Position des cellules à $t=13$ h. (d) Position des cellules à $t=24$ h. (e) Suivi des noyaux des cellules pendant les 13 premières heures. (f) Suivi des cellules pendant les 7 dernières heures. (g) Distance parcourue par les cellules pendant les 13 premières heures. (h) Histogramme des directions du déplacement (codées selon Freeman) des cellules proches de la plaie pour les 13 premières heures.

pourront éventuellement servir pour des traitements *in vivo*, comme par exemple pour le traitement des grands brûlés.

La plaie est créée en retirant une bande de cellules d'une culture adhérentes à un support de verre (figure 5.43a). Ensuite, le mouvement de recolonisation de l'espace vide par les cellules a été observé en prenant une image toutes les 30 minutes pendant 24 heures (figures 5.43b, c et d).

Différentes expériences de segmentation et de suivi ont ensuite été réalisées. Les figures 5.43e et f illustrent le suivi des cellules pendant différentes phases de la migration. Dans cet exemple, seuls les noyaux des cellules avaient été suivis, mais de futures simulations prévoient le suivi de cellules entières. La figure 5.43g montre l'histogramme des distances parcourues par les différentes cellules pendant la première moitié de l'acquisition (les cellules proches du bord de la plaie se déplaçant plus que les autres), et la figure 5.43h montre l'histogramme des directions des cellules au bord de la plaie, où l'on peut observer des directions privilégiées du déplacement dans la direction du vide causé par la plaie. Plus de détails sur cette expérience peuvent être trouvés dans [Boucher 98a].

5.3 Résultats

Afin de mettre en valeur, ou de mieux expliquer, les aspects importants de notre système, différents résultats illustrant le système dans la reconnaissance des cellules sont montrés. Les différentes expériences qui ont été menées dans ce but sont :

- description de la stratégie de reproduction pour lancer les agents pseudopodes, afin d'illustrer les principes d'adaptation et de décentralisation de cette stratégie ;
- description des différentes facettes du comportement de perception :
 - utilisation d'informations dynamiques et autonomie des agents, illustrées avec l'exemple des agents pseudopodes ;
 - adaptation et évolutivité des paramètres utilisés pour les critères de perception, démontrés avec l'exemple des agents noyaux ;
 - coopération pour la reconnaissance des halos blancs, autant entre les différentes sources d'informations, qu'entre différentes catégories d'agents ;
 - avantages et désavantages de la combinaison multi-critères employée dans les différentes fonctions d'évaluation, illustrés avec l'exemple de la segmentation du cytoplasme des cellules ;
- coopération globale entre les différentes informations accumulées pour le comportement de différenciation, qui est utilisé par les agents mouvement ;

- avantages de la gestion réactive et concurrente des comportements, avec l'illustration de l'entrelacement existant entre les différentes compétences des agents ;
- applicabilité du système sur un autre type cellulaire, soient les cellules de carcinome épithéloïde humain (HeLa).

Pour ces différentes expériences, une configuration de base des agents et du système a été établie. Cette configuration de base est donnée en annexe A. À partir de cette configuration, certains paramètres ont été modifiés dans chacune des situations afin de mettre en évidence les aspects que nous voulons démontrer. Les paramètres importants pour chaque situation seront donnés pour renforcer la discussion.

5.3.1 Adaptation et décentralisation de la stratégie de reproduction

Le comportement de reproduction, et avec lui la stratégie d'exploration de l'image, fait l'objet de la première expérience. Ce comportement a été décrit au paragraphe 4.4. L'exemple utilisé ici est le placement des germes d'agents pseudopodes. Pour lancer de tels agents, des profils sont explorés, dans plusieurs directions radiales à l'agent. Les pixels le long de ces profils sont évalués afin de trouver les meilleurs endroits pour initialiser de nouveaux germes pseudopodes. Les critères d'évaluation permettant de trouver les meilleurs emplacements de germes pseudopodes sont :

Critère reproduction: 1.0 extréma global source < Fond

Critère reproduction: + Noyau 0.5 distance > 10

Le premier critère signifie que l'on recherche des pixels plus sombres (en niveaux de gris) que le fond. Les informations accumulées sur les primitives fond par les agents associés serviront à adapter la valeur de seuil nécessaire à cette condition. Par la suite, le pixel le plus sombre répondant au minimum à la condition de seuil sera retenu.

Le deuxième critère ne s'applique que si la reproduction s'effectue à partir d'un agent noyau (exprimé par *+ Noyau*). Dans ce cas seulement, un deuxième critère est utilisé, afin de choisir des germes ayant une distance d'au moins dix pixels de la primitive noyau. Ce critère a pour but d'empêcher des germes d'être initialisés sur des tâches sombres se trouvant à côté du noyau. La pondération totale des critères, qui dépasse 1.0 dans ce cas, sera réajustée automatiquement. Ces deux critères sont spécifiés avec la configuration des agents pseudopodes. Tout agent désirant lancer des agents pseudopodes fera référence à cette configuration pour connaître les critères à appliquer.

La figure 5.44 montre le principe de la reproduction appliquée aux pseudopodes. Dans la figure 5.44a, le lancement des germes s'est effectué seulement depuis les agents noyaux, le long des profils perpendiculaires à la surface qui sont représentés. La configuration de lancement de ces germes par le noyau est spécifié de la façon suivante :

Reproduction même image fin: Pseudopode x 12

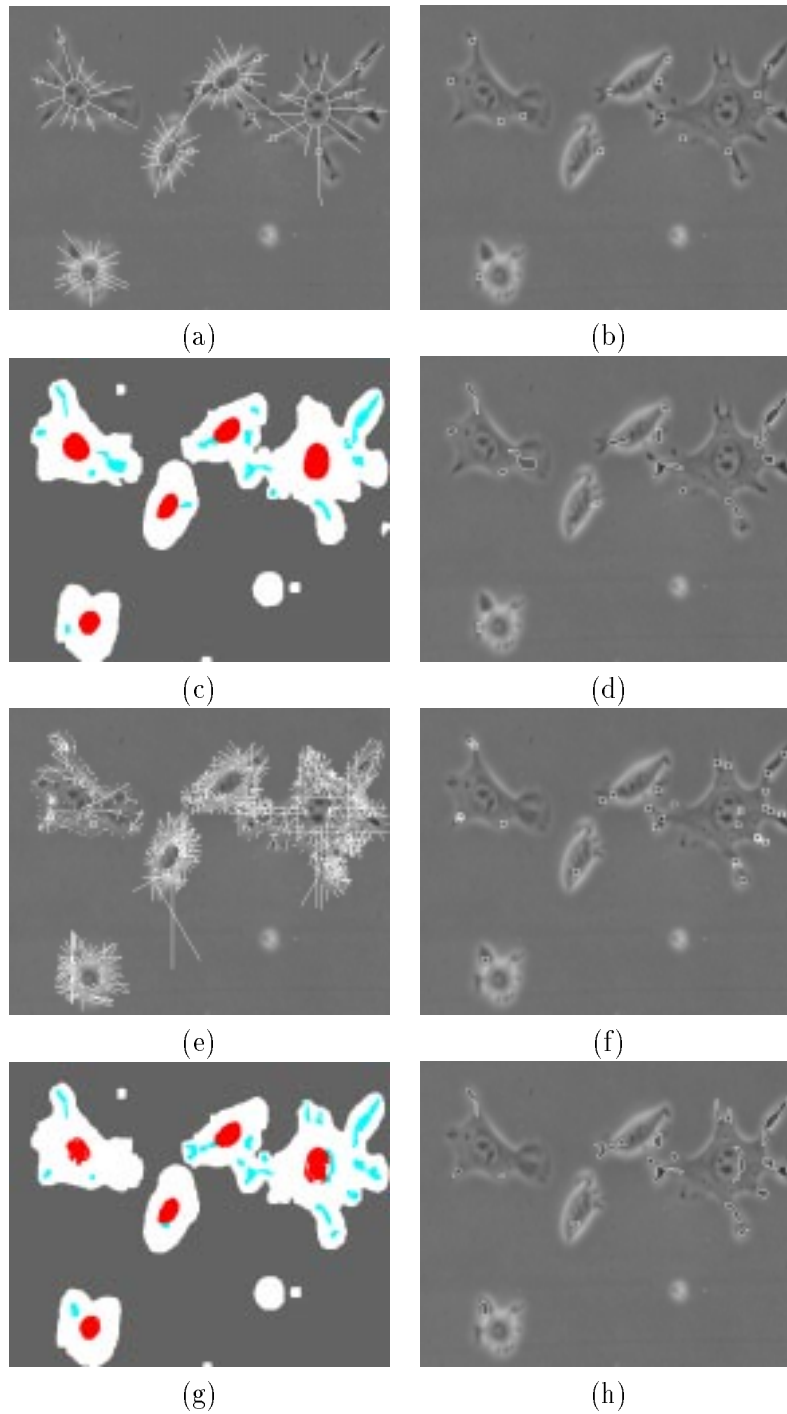


FIG. 5.44 – *Reproduction et placement des agents pseudopodes. (a) Exploration et placement des germes pseudopodes initiés depuis les agents noyaux uniquement. (b) Position des germes trouvés. (c) Résultats de la segmentation des pseudopodes. (d) Tracés des périmètres des régions pseudopodes trouvées superposés sur l'image source. (e) Exploration et placement des germes pseudopodes initiés depuis tous les agents. (f) Position des germes trouvés par tous les agents (noyaux, pseudopodes, cytoplasmes, halos blancs). (g) Résultats de la segmentation des pseudopodes. (h) Tracés des périmètres des régions pseudopodes trouvées superposés sur l'image source.*

Cette configuration signifie que l'agent noyau lancera, en fin de vie, des agents pseudopodes en explorant 12 directions, où un germe pourra être trouvé ou non dans chacune des directions. Les positions des germes effectivement trouvés sont montrées dans la figure 5.44b, et la segmentation correspondante effectuée par la suite par les agents pseudopodes dans les figures 5.44c et d.

Il est à noter que certains pseudopodes n'ont pas été détectés par cette exploration, et que certains autres ont été trouvés sur une cellule différente de l'agent noyau d'origine. Les agents situés sur la mauvaise cellule (appartenance au mauvais groupement perceptuel) auront la possibilité en fin de vie de corriger leur groupement d'appartenance simplement en observant l'appartenance majoritaire parmi les agents situés autour d'eux.

Pour régler le problème de la localisation exacte des germes, et pour éviter de manquer certains pseudopodes mal situés par rapport à l'agent qui désire lancer les germes d'agents pseudopodes, il suffit de distribuer cette reproduction parmi plusieurs types différents, en laissant le soin à chaque agent d'essayer de retrouver le maximum de germes possibles. C'est cette stratégie qui a été utilisée dans la figure 5.44e où les configurations des agents noyaux, cytoplasmes, halos blancs et même pseudopodes ont été modifiées afin d'inclure la commande de lancement de germes pseudopodes dans 12 directions à la fin de la vie de l'agent. L'agent pseudopode participe également à cette stratégie, en essayant de retrouver d'autres pseudopodes, ou en recherchant des morceaux qui ont pu être manqués lors de la phase de perception. En utilisant ces configurations décentralisées et distribuées sur plusieurs agents, l'exploration de la cellule est mieux réalisée et plus de germes ont pu être trouvés (figure 5.44f). La segmentation des pseudopodes est aussi améliorée, tel qu'illustré dans les figures 5.44g et h.

Cette expérience démontre l'importance de l'adaptativité de la stratégie de reproduction dans la recherche de germes d'un type précis et la nécessité de la décentralisation de cette stratégie pour en augmenter l'efficacité.

5.3.2 Adaptation du comportement de perception

Plusieurs aspects du comportement de perception sont maintenant étudiés. L'utilisation d'informations dynamiques, et l'autonomie des agents dans l'adaptation de leurs paramètres sont présentées avec l'exemple de l'agent pseudopode. Ensuite, l'évolution des paramètres dans le temps en fonction des informations récoltées est illustrée avec l'exemple de l'agent noyau. L'agent halo blanc permet de démontrer plusieurs points importants du système, soit la coopération de différentes sources d'informations et de différents agents dans l'accomplissement d'un but et l'adaptation à plusieurs niveaux de représentation. Certains aspects de l'évaluation multi-critères, avec l'exemple de l'agent cytoplasme, montrent les avantages et les désavantages d'un tel mécanisme.

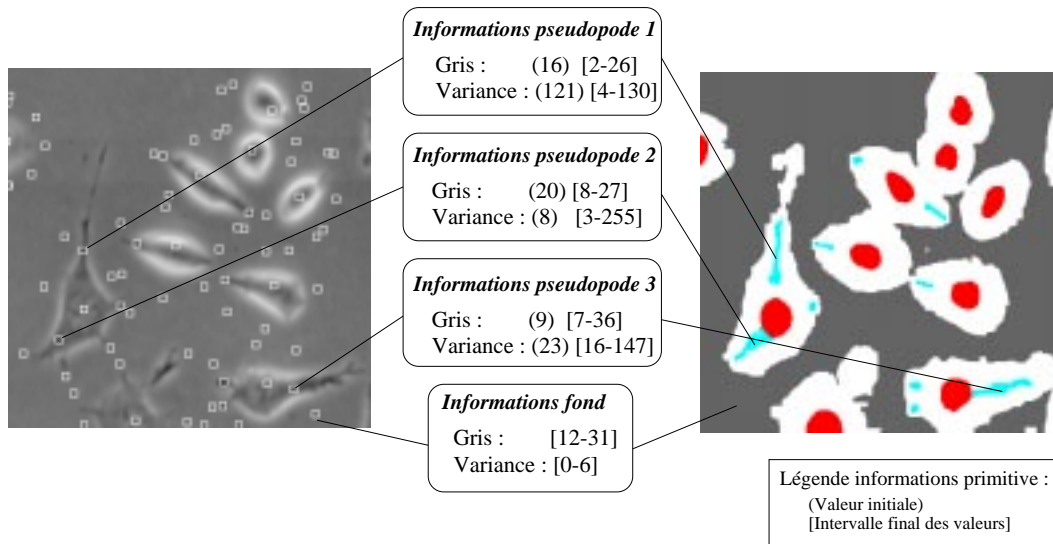


FIG. 5.45 – *Gestion et adaptation des informations pour la segmentation des pseudopodes. Les agents pseudopodes utilisent les informations produites par les agents fond pour s'adapter et produire leurs segmentations.*

5.3.2.1 Utilisation d'informations dynamiques pour la perception

Les agents, une fois lancés, sont autonomes dans leur processus d'agrégation de pixels pour la construction de leurs primitives. Ils ont la possibilité d'adapter eux-mêmes les différents paramètres des comportements en fonction du contexte local. Cela permet, pour la définition d'un type d'agent, d'obtenir la segmentation de plusieurs variantes d'une même composante. L'exemple choisi pour illustrer cela est l'agent pseudopode, qui possède les critères de perception suivants :

Critère perception : 0.4 seuillage variance > Fond

Critère perception : 0.4 seuillage source < Fond

Critère perception : 0.2 compacité

Ces critères rappellent ceux utilisés pour la reproduction de cette composante, c'est-à-dire les critères permettant la localisation des germes d'agents pseudopodes présentés au paragraphe 5.3.1. En effet, dans les deux cas, les critères sont définis par rapport au fond, plutôt que de manière absolue. La définition la plus fiable d'un pseudopode, d'un point de vue photométrique, est son opposition au fond. Il est plus sombre que le fond et d'une variance plus élevée. Ce sont les deux principaux critères utilisés, le troisième critère (compacité) permettant essentiellement de lisser la région trouvée et d'en boucher les trous.

La figure 5.45 montre l'application de tels critères par différents agents de type pseudopode. Trois agents pseudopodes sont détaillés avec les informations qu'ils utilisent et les résultats obtenus. Les informations utilisées par tous ces agents sont les mêmes, c'est-à-dire les statistiques des agents fond de l'image. Ces agents fond définissent deux seuils qui

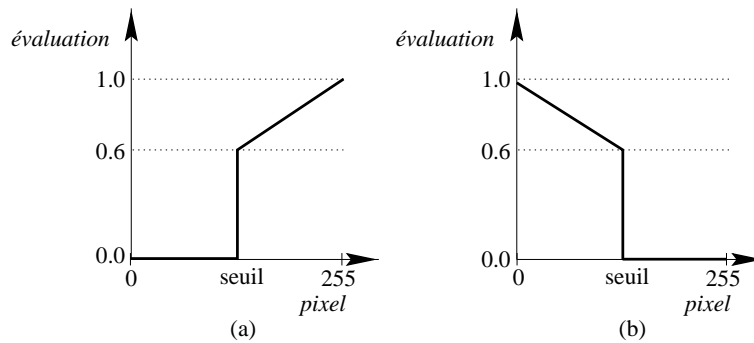


FIG. 5.46 – *Fonctions d'évaluation du critère de seuillage du comportement de perception.* (a) *Seuillage au-dessus du seuil.* (b) *Seuillage en-dessous du seuil.*

seront utilisés par les agents pseudopodes, soient un pour le niveau de gris (< 12) et un pour la variance (> 6).

Cependant, dans la pratique, il ne suffit pas de retenir tous les pixels vérifiant obligatoirement les deux seuils pour bien segmenter les pseudopodes, mais bien de les adapter aux différents contextes possibles. Cela se voit dans les informations obtenues sur différents pseudopodes qui ne respectent pas toujours ces seuils. En fait, l'évaluation s'effectuant sur une base multi-critères, il n'est pas essentiel que toutes les conditions soient réunies, mais seulement un nombre suffisant, avec une robustesse suffisante. Par robustesse, nous entendons que la valeur d'un pixel dépasse largement le seuil fixé. En effet, comme notre fonction de seuillage n'est pas binaire (figure 5.46), cela permet d'accepter plus facilement un pixel dépassant beaucoup l'un des seuils même si il ne vérifie pas l'autre.

Le principe de l'autonomie des agents est ici illustré avec l'adaptation des différentes informations de paramétrage obtenues en fonction des contextes locaux. Les fonctions d'évaluation des agents utilisent différentes sources d'informations, provenant parfois du travail d'autres agents. C'est dans cette forme de coopération indirecte dans le partage d'informations que les agents peuvent définir les données dont ils ont besoin.

5.3.2.2 Adaptation et évolution des paramètres

Les critères d'évaluation utilisés peuvent s'adapter en fonction des différents contextes locaux. L'exemple d'une telle adaptation est exposée ici avec les agents noyaux. Les noyaux sont les composantes les plus faciles à segmenter de notre système, du fait de la présence d'une image spécifique les représentant en fluorescence lorsqu'ils sont marqués à l'avance par un fluorochrome de l'ADN (figure 5.47b). Sans cette image, ils sont presque impossibles à distinguer selon les cellules, car ils se confondent, d'un point de vue photométrique, avec le cytoplasme des cellules.

Avec l'image des noyaux, un seuillage suffit à retrouver ces composantes (figure 5.47). Cependant, le seuillage ne peut pas être effectué à l'aveugle, ou avec un seuil fixe et

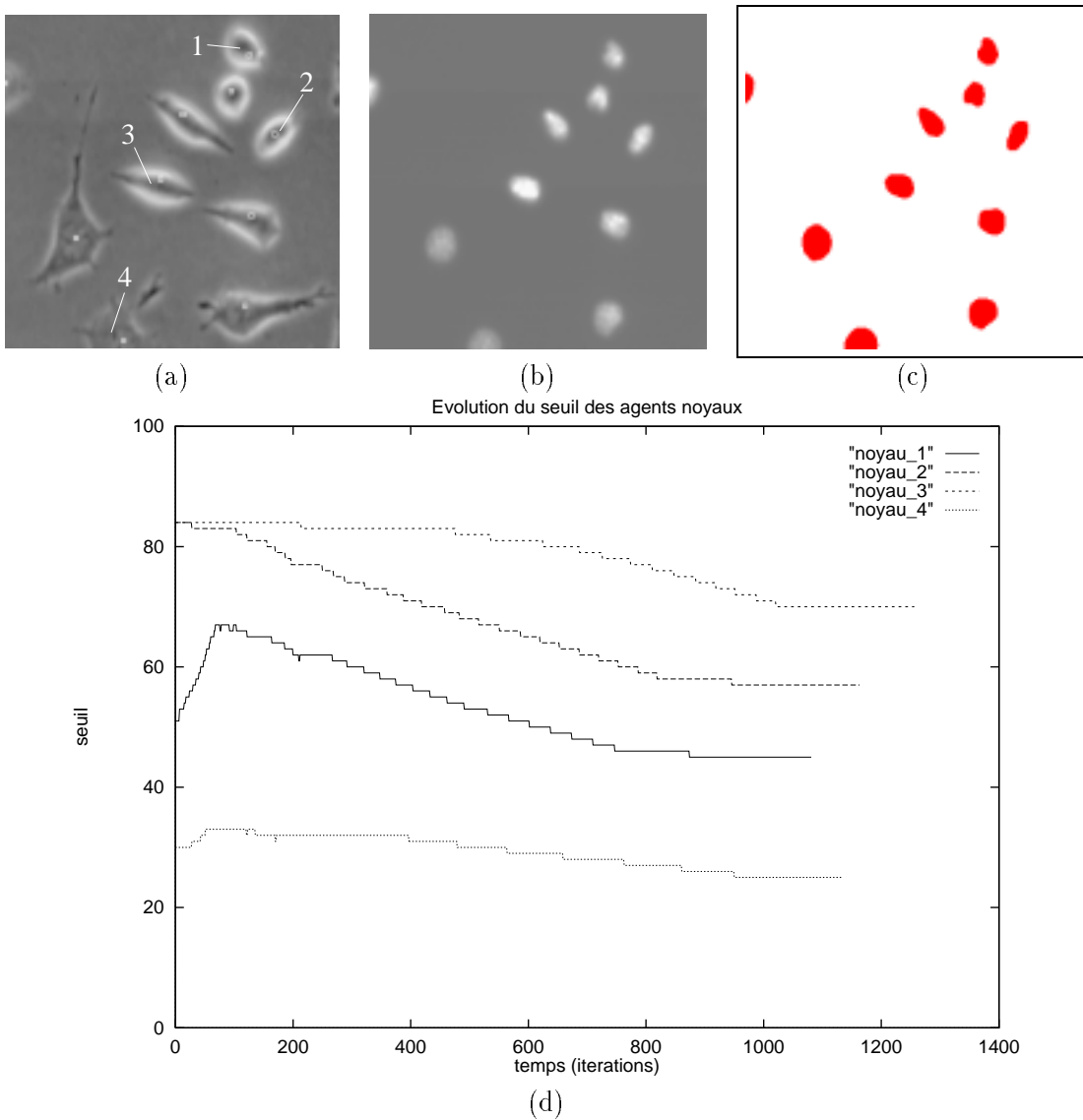


FIG. 5.47 – Évolution du seuil des agents noyaux en fonction du temps. (a) Image source acquise en contraste de phase. (b) Image noyau correspondante acquise en fluorescence (deuxième image source). (c) Résultats de la segmentation des noyaux. (d) Évolution du seuil (critère de seuillage) pour 4 agents sélectionnés (indiqués sur (a)).

global, car chaque noyau, même si il est visible, possède sa propre intensité, en fonction de sa réaction au marqueur utilisé. Chaque agent noyau devra donc tenir compte de cette caractéristique dans la segmentation de sa composante.

Les critères de perception de l'agent noyau sont :

Critère perception: 0.6 seuillage noyau 0.5

Critère perception: 0.4 compacité

Le premier critère est le plus important, car c'est lui qui fixe le seuillage utilisé. Le second critère ne sert qu'à éviter les trous qui pourraient apparaître dans le résultat final.

Le seuil défini par le premier critère est le suivant :

$$seuil = 50\% \text{ moyenne}_{\text{région}}(source_{\text{noyau}})$$

Ce seuil signifie qu'on va utiliser les statistiques internes à chaque composante pour réaliser le seuillage. Dans ce cas, la moyenne des niveaux de gris calculée sur l'image source des noyaux va être la référence dans le calcul du seuil. À chaque nouveau pixel ajouté à la région en croissance, cette moyenne va être modifiée et le seuil utilisé aussi.

La figure 5.47d présente l'évolution des seuils de différents agents noyaux au cours de leur segmentation. Le seuil utilisé dans la fonction d'évaluation évolue en fonction de la moyenne des pixels déjà agrégés. Le cas de l'agent n° 1 est intéressant à étudier. Le seuil utilisé par cet agent augmente subitement au début avant de diminuer par la suite. Ce phénomène s'explique par la fonction de seuillage utilisée qui n'est pas binaire, mais qui favorise les pixels dépassant largement le seuil fixé (figure 5.46). Les premiers pixels agrégés vont donc entraîner l'augmentation de la valeur du seuil rapidement, jusqu'à ce que des pixels de valeur moindre soient ensuite agrégés. Cette fonction de seuillage permet de s'assurer de la robustesse de cette adaptation.

En regardant les différents seuils utilisés, par différents agents autant que pour un seul agent au cours du temps, l'utilité du principe d'adaptation de la perception au contexte local devient facile à cerner. L'information que les agents accumulent en cours de travail leur permet de mieux analyser l'environnement local dans lequel ils évoluent et d'en tirer profit en adaptant leurs traitements à cet environnement.

5.3.2.3 Coopération pour la reconnaissance

La coopération, qu'elle soit directe ou indirecte, est un aspect important du système. Elle se décline sous différentes formes, que ce soit par une mise en commun d'informations ou par la complémentarité existant entre comportements, par exemple entre la reproduction et la perception (focalisation et segmentation).

Dans la figure 5.48, quatre expériences sont confrontées sur ce thème de la coopération :

1. Segmentation des halos blancs par les agents s'adaptant selon leurs informations intrinsèques (figures 5.48a, b et c). Ces agents ont été initialisés par les agents noyaux.
2. Coopération entre agents, où les agents crêtes permettent de mieux cibler les vrais halos blancs et d'éviter les segmentations manquées (figures 5.48d, e et f).
3. Coopération entre agents halos blancs, en adaptant les paramètres de perception selon l'ensemble des informations des groupements perceptuels, et non sur la base de chaque agent individuel (figure 5.48g).

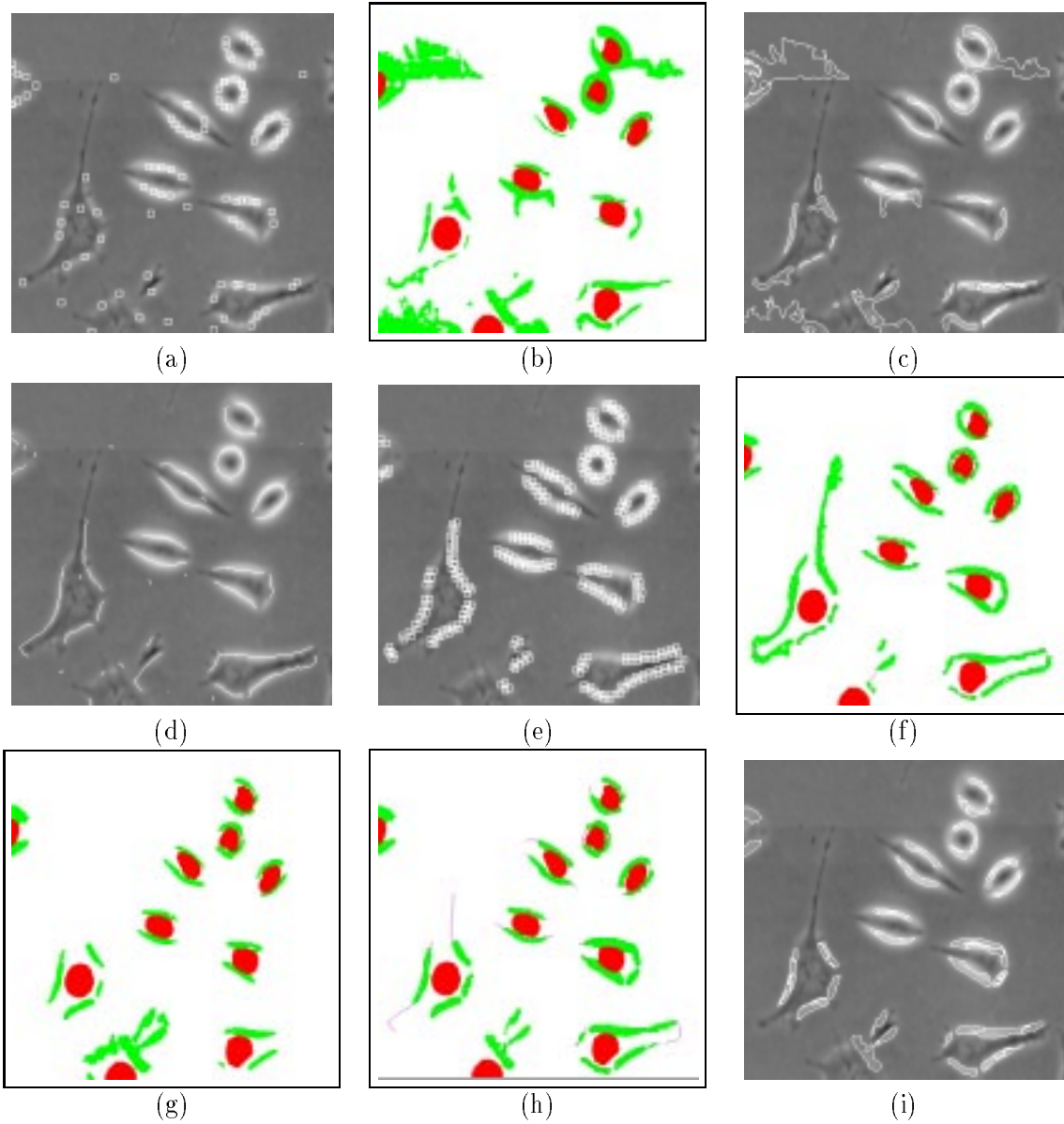


FIG. 5.48 – *Segmentation et adaptation des agents halos blancs. (a) Placement des germes halos blancs depuis les agents noyaux. (b) Segmentation des halos blancs avec adaptation du seuillage selon les informations de la primitive. (c) Tracés des périmètres des halos blancs pour le cas (b). (d) Segmentation des crêtes pour situer les halos blancs. (e) Placement des germes halos blancs depuis les agents crêtes. (f) Segmentation des halos blancs avec le lancement depuis les crêtes et l'adaptation du seuillage selon les informations de la primitives. (g) Segmentation des halos blancs lancés depuis les noyaux avec adaptation du seuillage selon les informations de la cellule (groupement perceptuel). (h) Segmentation des halos blancs avec lancement depuis les crêtes et adaptation du seuillage selon les informations de la cellule. (i) Tracés des périmètres des halos blancs pour le cas (h).*

4. Combinaison des apports des expériences 2 et 3, les agents crêtes aident à l'initialisation des agents halos blancs et la segmentation des halos blancs s'adapte selon les informations de groupement perceptuel (figures 5.48h et i).

Différents critères permettent de trouver le meilleur emplacement des germes d'agents halos blancs. Dans la figure 5.48a, c'est l'agent noyau qui a initialisé les germes d'agents halos blancs en employant les critères suivants :

Critère reproduction: 0.35 extréma global > source

Critère reproduction: 0.35 extréma local > source

Critère reproduction: 0.15 extréma local < variance

Critère reproduction: 0.15 opposition global > graddir

Ces critères peuvent être traduits comme la recherche de maximum de niveau de gris (au niveau local et global). On recherche également un minimum de variance se trouvant habituellement au centre du halo et une opposition de la direction du gradient de part et d'autre du halo (ces conditions ayant été observées expérimentalement).

Avec ces critères, la majorité des germes sont bien initialisés, mais pas tous, comme on peut l'observer sur la figure 5.48a, où certains germes ont été trouvés dans le fond de l'image. Ces quelques germes mal initialisés posent beaucoup de problèmes par la suite (figures 5.48b et c), car la segmentation des halos se résume à un seuillage adaptatif semblable à celui déjà présenté pour les agents noyaux au paragraphe précédent. En effet, les critères de perception pour les agents halos blancs sont :

Critère perception: 0.8 seuillage source 0.95

Critère perception: 0.2 compacité

Ce seuillage réussit bien lorsque l'agent est bien focalisé, mais donne des résultats erronés lorsque l'agent ne se trouve pas sur la bonne composante (figures 5.48b et c), à cause des informations intrinsèques aux agents qui sont alors faussées. Le seuil est calculé selon l'expression :

$$seuil = 95\% \text{ moyenne}_{\text{région}}(source_{\text{contraste de phase}})$$

Pour résoudre ce problème, deux solutions sont possibles, selon qu'on s'attaque au problème de l'initialisation ou au problème des informations intrinsèques qui sont fausses. La première solution consiste à s'assurer que l'agent est bien initialisé sur le halo en recherchant la segmentation d'une autre caractéristique du halo. Pour ce faire, des agents crêtes seront initialisés par les agents noyaux au lieu des agents halos blancs comme précédemment (figure 5.48d). Les critères de reproduction sont toujours les mêmes, sauf que les agents initialisés sont de type différent.

La différence est dans les caractéristiques recherchées lors de la perception. Une crête est définie comme une ligne centrale du halo blanc présentant un maximum de niveau gris, un minimum de gradient et une opposition dans les directions du gradient de part et

d'autre¹. Les critères de perception de l'agent crête sont définis comme :

Critère perception: 0.25 extréma local < gradnorme

Critère perception: 0.35 extréma local > source

Critère perception: 0.3 opposition local > graddir

Critère perception: 0.1 taille 10

Cette fois, l'agent segmente une ligne avec des caractéristiques propres, en évitant les pièges du seuillage. Si la crête est présente, alors les agents halos blancs peuvent être initialisés (figure 5.48e). Cette fois, les fausses initialisations des agents sont évitées par la robustesse de la crête comme information intermédiaire, et la segmentation des halos blancs est améliorée (figure 5.48f). Les critères de reproduction des halos blancs depuis les agents crêtes sont beaucoup plus simples puisque les agents sont initialisés à distance fixe tout le long de la crête :

Critère reproduction: Crête 1. distance 2

La deuxième solution possible pour améliorer les résultats des agents halos blancs consiste à concerter les efforts et les informations de tous les agents halos blancs travaillant sur une même cellule, donc appartenant à un même groupement perceptuel. Les halos sont assez semblables à l'intérieur d'une même cellule, mais ils peuvent être différents de cellule en cellule.

Si un agent halo blanc utilise maintenant l'ensemble des informations accumulées par tous les agents de son groupe² pour adapter ses traitements, il diminue les risques de manquer sa cible. En fait, un agent mal initialisé ne pourra pas effectuer de segmentation erronée, car le seuil calculé selon les informations accumulées par les autres agents sera trop élevé, et cet agent se terminera tout simplement. Cette adaptation selon les informations de la primitive ou selon les informations du groupe se fait aisément grâce à l'homogénéité de ces structures d'informations (paragraphe 2.5.1).

La figure 5.48g montre les résultats d'une telle adaptation par groupe. Les germes d'agents halos blancs pour cette expérience ont été lancés depuis les agents noyaux (placement des germes identiques à la figure 5.48a), et les agents mal placés se sont terminés avant d'effectuer une segmentation erronée. Les paramètres de configuration modifiés pour cette expérience sont les suivants :

Adaptation selon informations groupe: oui

Critère perception: 0.8 seuillage source 0.8

Le premier paramètre indique à l'agent halo blanc d'effectuer ses calculs d'adaptation selon les informations de la cellule³, et non selon ses informations intrinsèques. De plus, il a été possible de diminuer le paramètre du critère de seuillage utilisé. Le nouveau seuil

1. Le nom de *crête* vient de l'observation des profils de niveaux gris. En observant les halos comme des sommets de niveaux de gris, la ligne que nous suivons s'apparente à une ligne de crête.

2. Des agents de même groupe sont des agents dont les informations appartiennent au même groupement perceptuel.

3. Par informations de la cellule, nous entendons les informations cumulées de tous les halos blancs de la cellule.

calculé est plus robuste et s'exprime par :

$$seuil = 80\% \text{ moyenne}_{\text{cellule}}(source_{\text{contraste de phase}})$$

Enfin, il est possible de cumuler les deux solutions présentées ci-dessus pour augmenter la robustesse de la segmentation des halos blancs, c'est-à-dire initialiser les agents halos blancs depuis les agents crêtes et adapter les paramètres de perception selon les informations de la cellule. Le résultat de cette double coopération est montré aux figures 5.48h et i.

Ces différentes combinaisons illustrent le potentiel de la coopération dans ce système. Cette coopération se retrouve à plusieurs niveaux, que ce soit dans les sources d'informations ou entre les différents types d'agents présents dans le système. Les informations intermédiaires comme aide dans des contextes difficiles sont également une forme de coopération entre des indices visuels de nature différente. Et là où l'agent seul risque de commettre des erreurs dues à sa sensibilité aux conditions d'initialisation, la réunion de plusieurs agents d'un même groupe permet d'augmenter les chances de succès et de faire ressortir les informations les plus robustes.

5.3.2.4 Combinaison multi-critères et paramétrage

Dans les paragraphes précédents, différents exemples de critères ont été montrés, autant pour le comportement de perception que pour le comportement de reproduction. Plusieurs types de critères existent, avec chacun leurs paramètres et leurs poids dans l'évaluation globale. La combinaison de plusieurs types de critères peut être essentielle pour bien cerner la définition d'une composante.

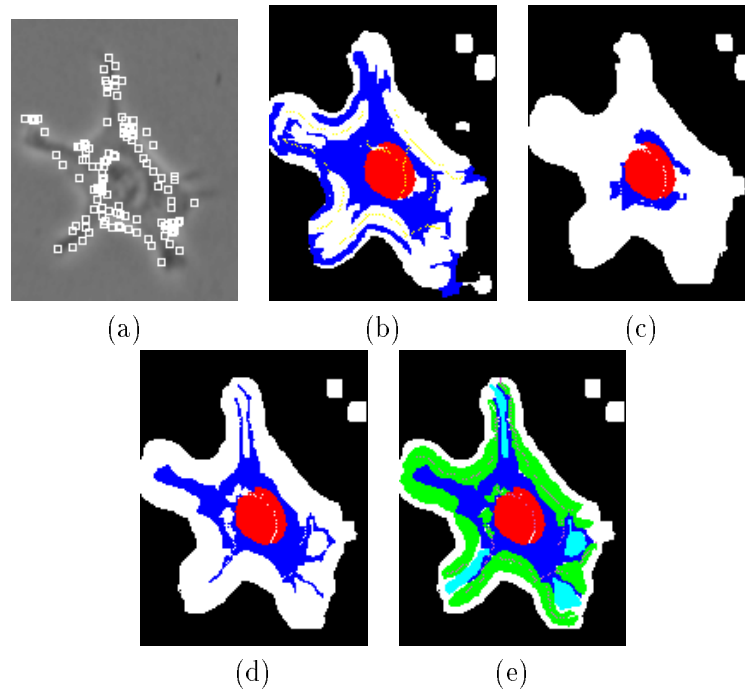
L'exemple choisi pour illustrer cette notion est celui de l'agent cytoplasme. Bien qu'elle paraisse assez simple, il s'agit de la composante la plus difficile à définir d'un point de vue critères de perception. On retrouve de grandes variations des différentes caractéristiques photométriques dans cette composante. Elle est très difficile à définir en opposition à d'autres composantes. C'est pourquoi il a été décidé de la définir comme étant au centre de la cellule (ou groupement perceptuel).

Le lancement d'agents cytoplasmes est relativement aisé. Ces germes peuvent être lancés par tous les autres agents simplement en recherchant les positions à mi-chemin entre deux composantes appartenant au même groupe (figure 5.49a). Il s'agit donc exclusivement de critères spatiaux qui sont utilisés :

Critère reproduction: 0.5 inclusion

Critère reproduction: 0.5 distance 0.5

Par contre, la segmentation, par le comportement de perception, est plus délicate. Voici les différents critères utilisés pour cela, avec les poids correspondants permettant d'obtenir le résultat montré en figure 5.49b :



Critères	(b)	(c)	(d)	(f)
Homogénéité source 8	0.5	0.3	0.2	
Inclusion noyau	0.1	0.5	0.1	
Inclusion	0.1	0.1	0.5	
Compacité	0.3	0.1	0.1	

FIG. 5.49 – Segmentation du cytoplasme en fonction des poids des critères de perception. (a) Placement des germes d’agents cytoplasmes. (b) Segmentation du cytoplasme en donnant la priorité au critère d’homogénéité. (c) Segmentation du cytoplasme en donnant la priorité au critère d’inclusion avec contrainte sur la présence du noyau. (d) Segmentation du cytoplasme en donnant la priorité au critère d’inclusion dans le même groupement perceptuel sans contrainte. Segmentation réalisée en (e) ajoutée aux résultats des autres agents de tous types. (f) Poids des critères utilisés dans chaque segmentation. Pour les résultats montrés en (b), (c) et (d), seules les segmentations des noyaux, des cytoplasmes et du fond sont affichées, les autres segmentations étant masquées pour mettre en évidence les cytoplasmes.

Critère perception : 0.5 homogénéité source 8

Critère perception : 0.1 inclusion Noyau

Critère perception : 0.1 inclusion

Critère perception : 0.3 compacité

Un seul critère photométrique est utilisé, les autres étant d’ordre spatial. Le critère d’inclusion est décliné deux fois, avec un paramètre différent. Ce critère signifie qu’un pixel est accepté si il est inclus à l’intérieur de composantes appartenant au même groupement perceptuel (à la même cellule). Dans le premier cas, cette inclusion est contrainte par

la nécessité d'avoir une des composantes avoisinantes de type noyau. Dans le deuxième cas, aucune contrainte n'est imposée. Le poids de ces différents critères peut varier (figure 5.49f), avec des grandes variations sur les résultats obtenus (figures 5.49b, c et d). À chaque fois, un critère est prépondérant dans l'évaluation globale.

Cette expérience met en évidence la nécessité d'une combinaison multi-critères dans l'évaluation, mais également un problème posé par cette évaluation. De nombreux paramètres interviennent dans la configuration des agents, et la définition des valeurs exactes de tous ces paramètres peut devenir très difficile, à cause principalement des multiples influences de chaque paramètre dans le comportement global. Ce point est discuté plus amplement au prochain chapitre, où des notions d'apprentissage sont ajoutés au système pour tenter de pallier ce problème.

5.3.3 Coopération globale pour le comportement de différenciation

Le comportement de différenciation n'est employé que par un type d'agent, c'est-à-dire les agents mouvement (paragraphe 3.7). Ces agents travaillent sur une information de niveau intermédiaire, qui doit par la suite être différenciée, c'est-à-dire interprétée pour lui donner une valeur sémantique au sens de l'application.

Les agents mouvement recherchent donc les zones de l'image où il y a eu un déplacements entre deux ou plusieurs images consécutives. Le lancement de ces agents s'effectue par tous les autres types d'agents simplement en recherchant les maxima sur l'image d'amplitude de la différence d'images (figure 5.50c) :

Critère reproduction: 0.5 extréma global > diffamplitude

Critère reproduction: 0.5 extréma local > diffamplitude

Par la suite, ces agents segmentent les zones homogènes de mouvement, au sens du signe de la différence d'images, en s'assurant d'un seuil suffisant d'amplitude de cette différence (figure 5.50e) :

Critère perception: 0.5 homogénéité diffsigne 0.1

Critère perception: 0.5 seuillage diffamplitude 0.5

Ces deux étapes (focalisation, segmentation) sont assez faciles à réaliser. L'étape la plus cruciale pour les agents mouvement est de trouver l'identité sémantique de la primitive qu'ils ont trouvée. C'est le comportement de différenciation qui est utilisé pour cela. Pour réaliser cette tâche, ce comportement doit s'appuyer sur les différentes informations accumulées par les autres agents travaillant dans l'image. Les paramètres de ce comportement, qui sont les poids des critères et les identités possibles de cette différenciation, sont les suivants :

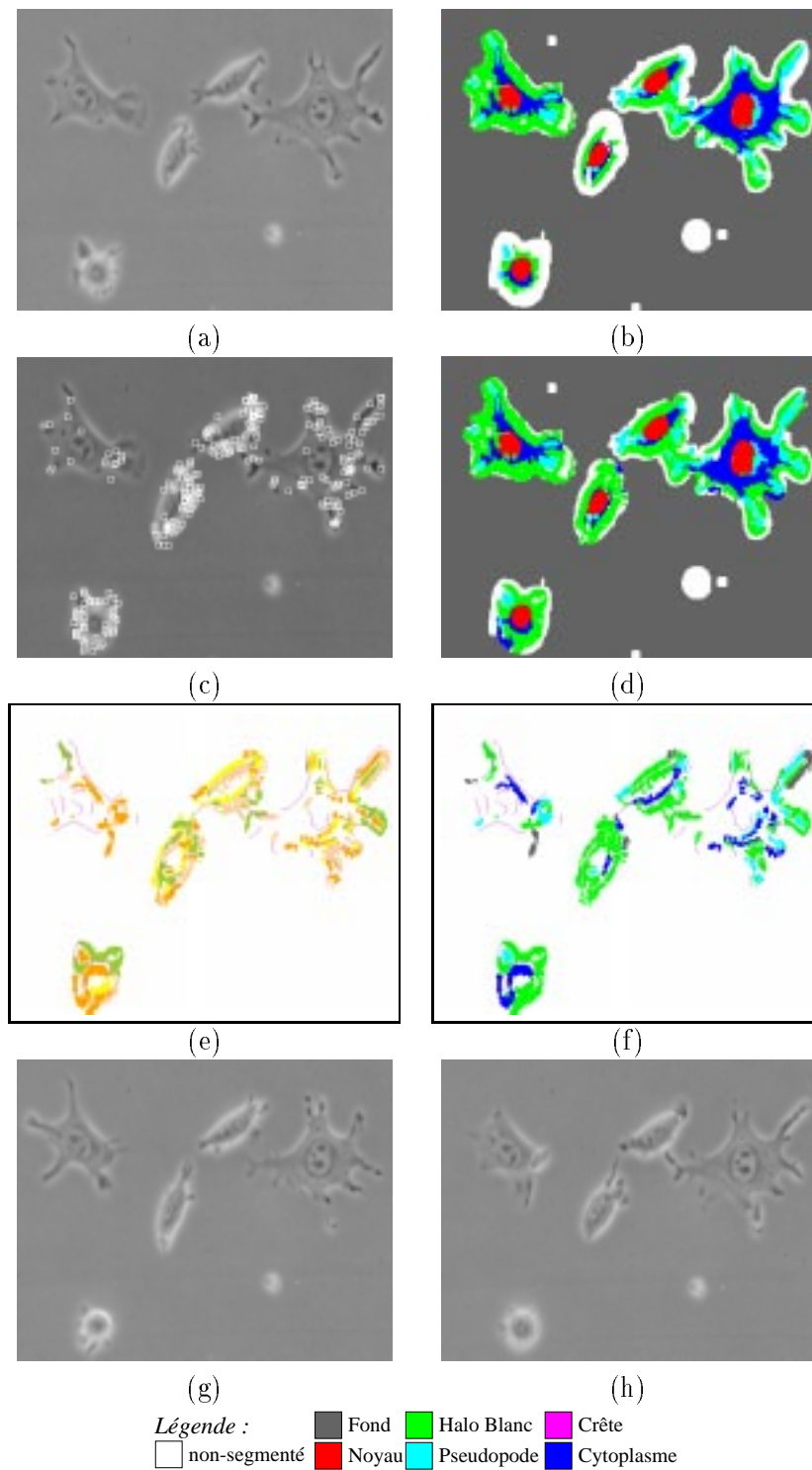
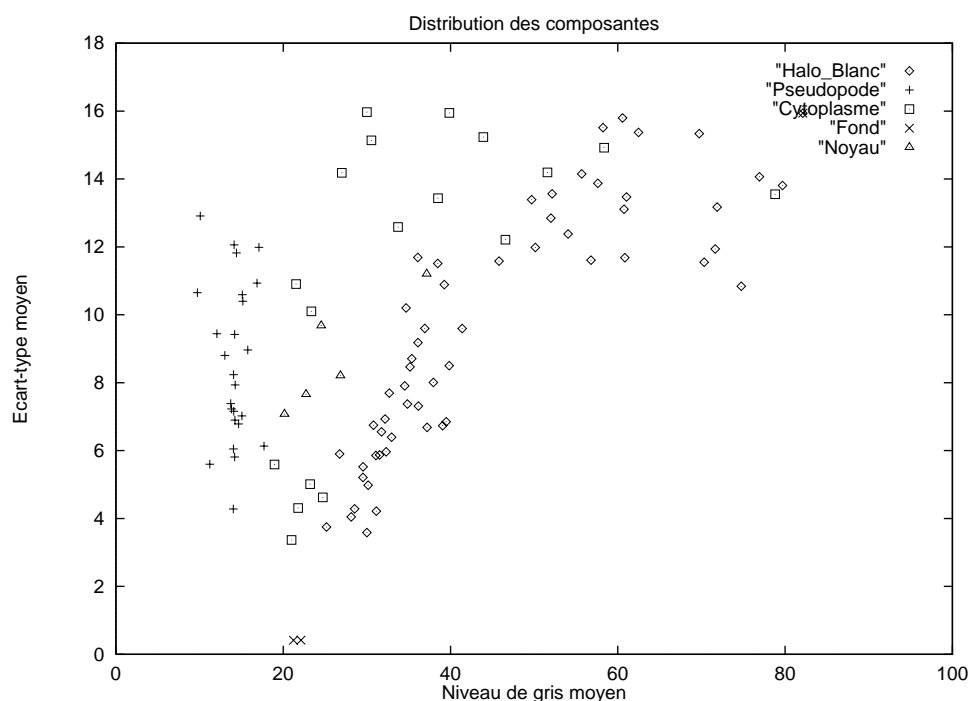
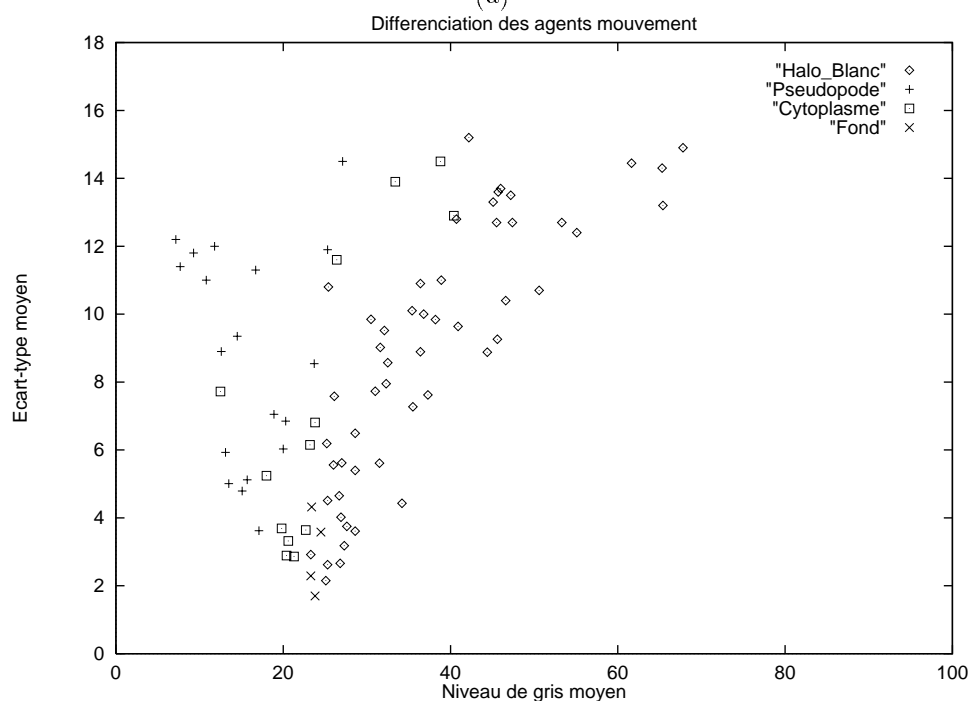


FIG. 5.50 – Segmentation et différenciation des agents mouvement. (a) Image source. (b) Segmentation sans utiliser l'information mouvement. (c) Germes d'agents mouvement lancés par tous les autres agents. (d) Segmentation complète utilisant l'information mouvement. (e) Niveau de représentation primitive (illustrant les régions de mouvement). (f) Niveau de représentation primitive montrant la différenciation effectuée par les agents. (g) Image précédente de (a) dans la séquence. (h) Image suivante de (a) dans la séquence.



(a)



(b)

FIG. 5.51 – Représentation des statistiques des composantes et différenciation des agents mouvement. (a) Graphique des statistiques des agents de niveau composante (primitives segmentées en figure 5.50b). (b) Résultat de la différenciation des agents mouvement en utilisant tous les critères (et non pas que le critère statistique) (primitives segmentées en figure 5.50f).

Poids du critère statistique : 0.35

Poids du critère intersection : 0.55

Poids du critère adjacence : 0.1

Identités possibles : Fond HaloBlanc

Identités possibles : Pseudopode Cytoplasme

Le critère statistique utilise la méthode des k plus proches voisins pour trouver l'identité de la primitive. Le critère d'intersection consiste à calculer les recouvrements existant entre les deux images de segmentation (une par niveau de représentation, figures 5.50b et e). Le critère d'adjacence, moins important, consiste à déduire l'identité de la primitive selon les primitives voisines dans l'image.

Les informations utilisées pour ces différents critères sont dynamiques, puisqu'elles sont modifiées au gré des traitements effectués par tous les différents agents travaillant sur l'image. L'agent mouvement tient compte de ce caractère dynamique en observant les informations disponibles à un instant donné et en décidant de se différencier ou d'attendre plus d'informations. La figure 5.51a montre les informations telles qu'elles se distribuent à la fin de l'exécution des agents (correspondant à la segmentation montrée à la figure 5.50b). Sur cette figure, on peut observer les classes formées par les différents types d'agents. La difficulté du traitement vient du recouvrement existant entre ces classes, du nombre d'échantillons variables d'une classe à l'autre, d'où la nécessité d'utiliser plusieurs critères dans la décision et d'attendre le moment opportun pour prendre cette décision.

La figure 5.51b illustre le résultat de la différenciation effectuée par les agents mouvement sur la base de tous les critères. Le résultat de segmentation de cette différenciation est montré à la figure 5.50f. L'apport de ces agents au résultat final de segmentation est perceptible en comparant la figure 5.50b, sans agents mouvement, avec la figure 5.50d, qui comporte les résultats des agents mouvement différenciés. Cette différence se note particulièrement sur les cellules au centre de l'image, en sous-segmentation dans la figure 5.50b, mais segmentées entièrement dans la figure 5.50d.

Le comportement de différenciation démontre un cas de coopération au niveau de l'échange d'informations entre agents. Le travail des agents mouvement en différenciation s'appuie exclusivement sur les contributions des autres agents présents dans l'image. L'information mouvement de ces agents, confrontée avec les informations statiques des autres agents, permet de raffiner la segmentation et d'améliorer la reconnaissance des cellules.

5.3.4 Gestion réactive des comportements

Différents comportements ont été décrits dans ce travail. Chaque comportement est une action que peut poser un agent pour répondre à un stimulus, provenant de l'intérieur ou de l'extérieur de l'agent. L'entrelacement et la coopération entre les actions menées par ces comportements peut se visualiser autant au niveau de l'agent qu'au niveau global de

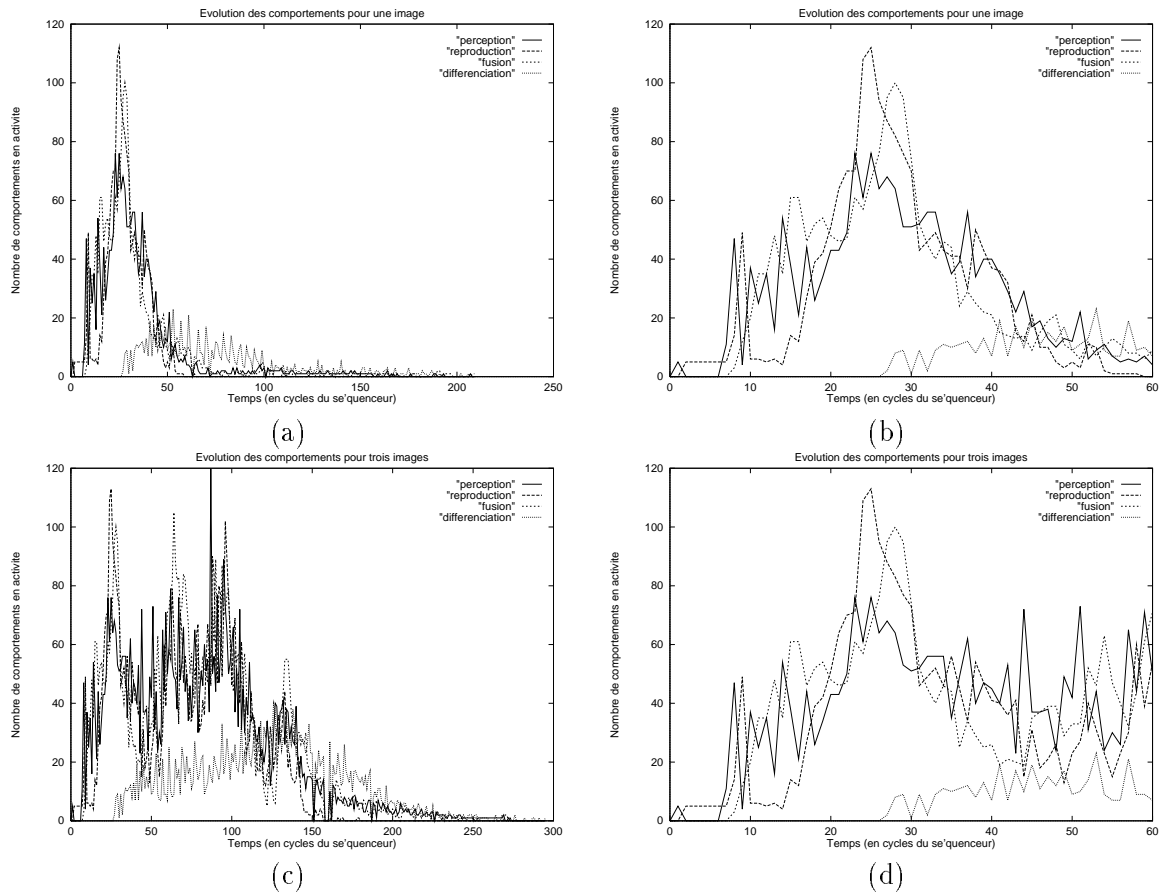


FIG. 5.52 – *Gestion des comportements des agents en fonction du temps. (a) Comportements actifs en fonction du temps pour le traitement d'une image. (b) Mise en évidence de l'activité du système pendant les 60 premiers cycles. (c) Comportements actifs en fonction du temps pour le traitement de trois images d'une sequence. (d) Mise en évidence de l'activité du système pendant les 60 premiers cycles.*

la société d'agents.

La figure 5.52 montre l'activité du système en termes de comportements actifs dans la société d'agents. À tout moment dans le système, plusieurs comportements différents interviennent pour répondre à différentes demandes. Dans la figure 5.52a, qui illustre l'activité du système pour le traitement d'une image, on remarque que tous ces comportements sont constamment en activité, par des agents différents, ce qui permet d'observer, au niveau global, une véritable dynamique entre d'une part les comportements en action et d'autre part les informations que ces comportements échangent grâce à leurs dépendances mutuelles.

Il peut être noté sur cette figure que chaque pic de reproduction est suivi d'un pic de fusion, dû aux agents surnuméraires qui ont été lancés et qui fusionnent dès leur rencontre. Le comportement de différenciation est utilisé par quelques agents seulement, uniquement lorsque les segmentations des primitives mouvement sont terminées et que des informations

sont disponibles pour effectuer la différenciation.

Dans la figure 5.52c, l'entrelacement entre les comportements est montré sur une séquence de trois images. La segmentation des images s'effectue en pipeline, c'est-à-dire que lorsque la segmentation d'une image est suffisamment avancée, des agents sont créés sur l'image suivante et ainsi de suite. Dans la figure 5.52b, alors que l'on observe un ralentissement de l'activité apparaissant autour du 50^{ème} cycle, l'activité repart avec la segmentation de la deuxième image dans la figure 5.52d.

L'étude de cette activité au niveau global de la société d'agents permet de mieux saisir l'importance de cet entrelacement des différentes actions possibles dans le système. Chaque comportement est réactif, de même que la gestion des différents comportements effectuée par les agents. Le système est constamment en activité pour répondre à des stimuli de toutes sortes, et pour compléter ses différentes sources d'informations par différentes actions complémentaires.

5.3.5 Expérimentation sur un autre type cellulaire

Les différents résultats montrés dans ce chapitre proviennent de différentes séquences d'images représentant la même lignée cellulaire. Un des buts d'un système de vision serait de pouvoir s'adapter à différents types de données, à l'intérieur de certaines limites données par l'application. C'est pourquoi nous avons testé le système, avec sa configuration de base habituelle, sur une lignée différente de cellules, soit une culture de cellules HeLa.

L'intérêt, d'un point de vue biologique, de comparer ces deux types de cellules, est que la première lignée provient d'un tissu normal (L929), tandis que les cellules HeLa sont d'origine cancéreuse. Les recherches sur la migration et la déformation cellulaire ont, entre autres buts, l'objectif de d'identifier des différences de comportements entre les cellules normales et pathologiques.

Du point de vue de la vision par ordinateur, donc photométrique essentiellement, ces deux types de cellules sont assez semblables pour qu'on tente d'appliquer les mêmes traitements de reconnaissance. La figure 5.53 montre les résultats de segmentation obtenus sur cette lignée cellulaire. Ces résultats sont similaires à ceux déjà obtenus sur la lignée précédente. Les cellules présentent quelques différences mineures des points de vue photométriques et morphologiques, mais le système s'est assez bien adapté à ces différences. D'autres essais sur ce type de cellules ont déjà été présentés dans [Boucher 98b].

5.4 Évaluation

Après avoir décrit les différents aspects du système, il est maintenant proposé une évaluation des résultats. L'évaluation proposée est effectuée sur une base quantitative, grâce à des cartes de référence qui serviront de comparaison pour évaluer les résultats de

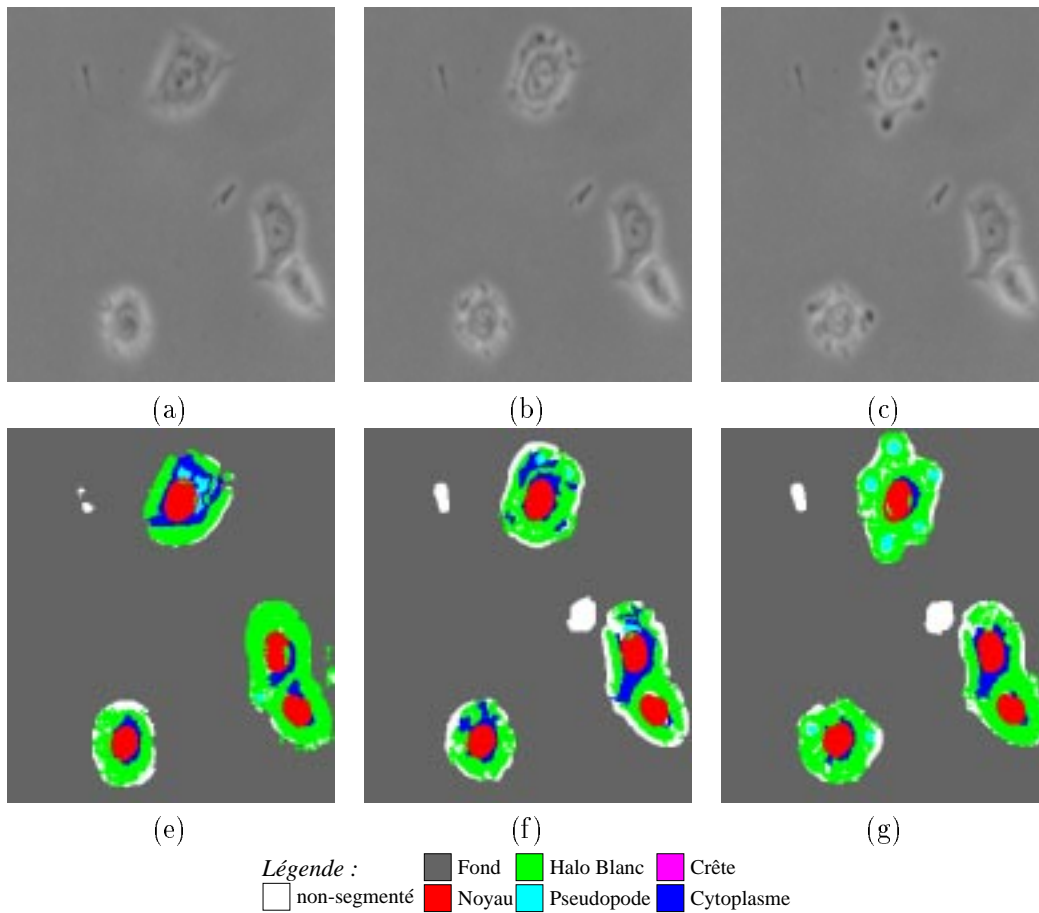


FIG. 5.53 – *Segmentation d’une séquence d’images de cellules HeLa. (a) (b) (c) Images sources prises à 30 minutes d’intervalle. (d) (e) (f) Segmentations réalisées avec les mêmes configurations de base que pour les fibroblastes murins. La configuration des agents n’a pas été ajusté pour ces images.*

notre système. Une fois la méthode de construction de ces cartes exposée, les résultats seront évalués puis analysés.

5.4.1 Cartes de référence

L’évaluation du résultat d’un système doit se faire en fonction des buts et des moyens qui lui sont donnés. Tel qu’il a déjà été énoncé, cette connaissance du but final à atteindre par le système est primordiale. Dans notre application, nous nous intéressons à l’interprétation de l’image au sens du domaine. La segmentation qui est faite de l’image est couplée et intégrée à l’interprétation recherchée par l’application. Comme le système possède des connaissances haut niveau sur les composantes de l’image, son évaluation portera donc sur les objets, au sens de l’application, qu’il aura réussi à trouver.

Nous avons opté pour une méthode d’évaluation quantitative, par construction de cartes

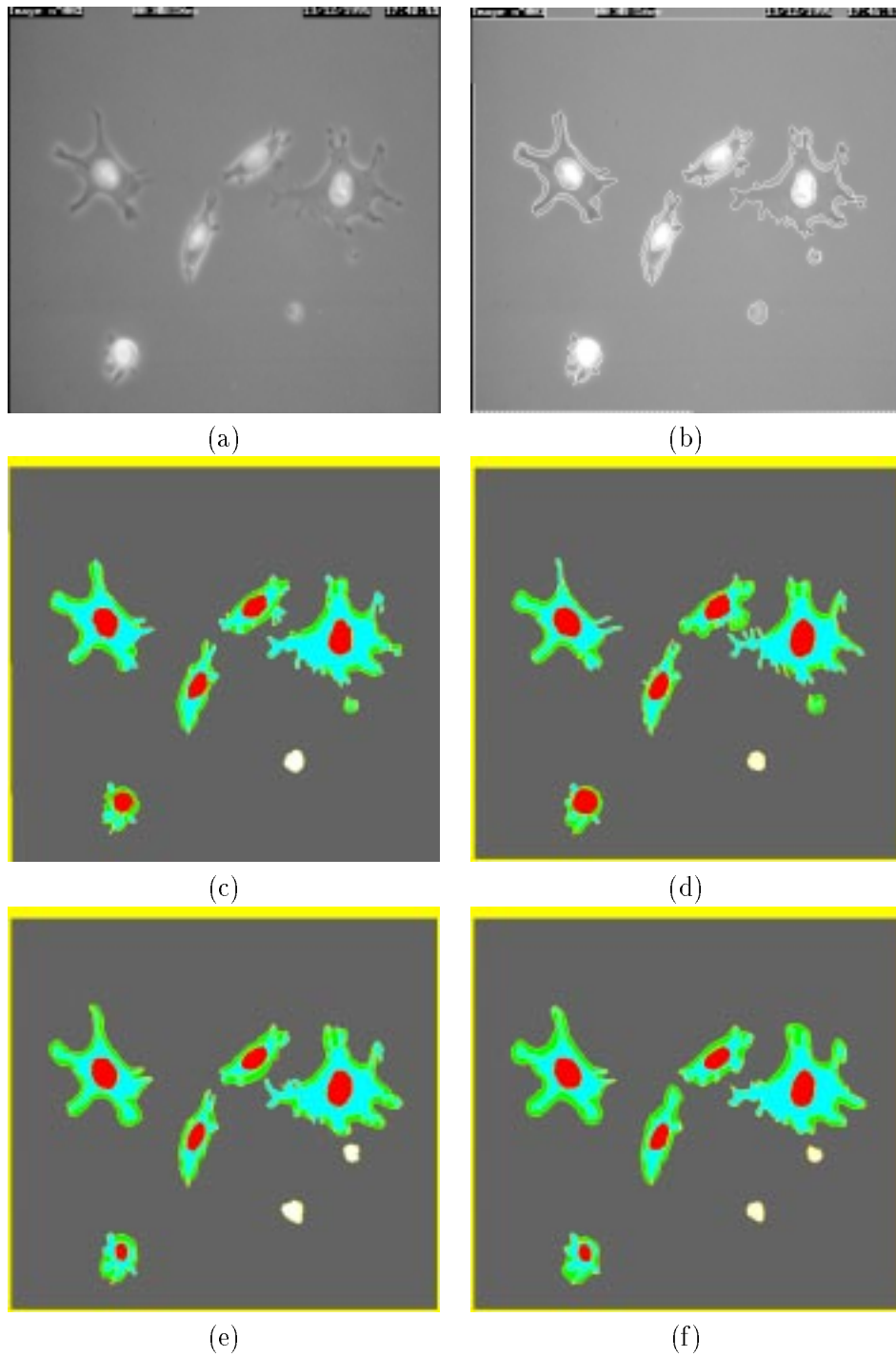


FIG. 5.54 – Construction des cartes de références par différents opérateurs. (a) Image source cumulant les cellules et les noyaux présentée aux opérateurs. (b) Exemple de carte des contours dessinée par un opérateur. (c) Carte de référence obtenue par l'opérateur 1 (non-expert du domaine). (d) Carte de référence obtenue par l'opérateur 2 (non-expert du domaine). (e) Carte de référence obtenue par l'opérateur 3 (expert du domaine). (f) Carte de référence obtenue par l'opérateur 4 (expert du domaine).

Opérateur 1\Opérateur 2	<i>Fond</i>	<i>Noyau</i>	<i>Halo</i>	<i>Cytoplasme</i>	Total	%
<i>Fond</i>	209336	1	686	156	210179	99.6
<i>Noyau</i>	0	3135	0	19	3154	99.4
<i>Halo</i>	947	126	2418	183	3674	65.8
<i>Cytoplasme</i>	172	270	443	9177	10062	91.2
Total	210455	3532	3547	9535	227069	-
%	99.5	88.8	68.2	96.3	-	-

(a)

Opérateur 3\Opérateur 4	<i>Fond</i>	<i>Noyau</i>	<i>Halo</i>	<i>Cytoplasme</i>	Total	%
<i>Fond</i>	207632	0	1208	187	209027	99.3
<i>Noyau</i>	0	3017	0	36	3053	98.8
<i>Halo</i>	439	0	4221	383	5043	83.7
<i>Cytoplasme</i>	84	43	105	9835	10067	97.7
Total	208155	3060	5534	10441	227190	-
%	99.8	98.6	76.3	94.2	-	-

(b)

Opérateur 2\Opérateur 4	<i>Fond</i>	<i>Noyau</i>	<i>Halo</i>	<i>Cytoplasme</i>	Total	%
<i>Fond</i>	207439	0	2504	222	210165	98.7
<i>Noyau</i>	0	3109	69	396	3574	87.0
<i>Halo</i>	419	14	2731	410	3574	76.4
<i>Cytoplasme</i>	115	64	152	9178	9509	96.5
Total	207973	3187	5456	10206	226822	-
%	99.7	97.6	50.1	89.9	-	-

(c)

FIG. 5.55 – Comparaison, par matrice de confusion, des cartes de références établies par différents opérateurs. (a) Comparaison entre les cartes établies par deux opérateurs non-experts du domaine. (b) Comparaison entre les cartes établies par deux opérateurs experts du domaine. (c) Comparaison entre les cartes établies par un opérateur non-expert du domaine et un opérateur expert du domaine.

de référence de certaines images. Cela consiste à demander à un utilisateur humain de segmenter manuellement les images que l'on donnera au système. Bien sûr, cette méthode est limitée, puisqu'elle oblige l'utilisateur humain à effectuer la même tâche que le système, ce qui semble limiter l'utilité du système. Mais il est très difficile, voire impossible, de disposer d'une méthode d'évaluation quantitative, fiable et automatique des résultats. Si une telle méthode existait, c'est qu'elle posséderait toutes les heuristiques haut niveau nécessaires à la compréhension de l'image, et le problème de la vision serait donc résolu puisqu'il suffirait d'implanter ces heuristiques dans le système lui-même. L'intérêt de la méthode d'évaluation par cartes de référence est de choisir quelques images tests, de quelques séquences, qui permettront d'évaluer le système.

L'image que nous présentons à l'opérateur humain est montrée dans la figure 5.54a. Il s'agit en fait de l'image en fluorescence des noyaux superposée à celle des cellules en contraste de phase. Ce choix de superposer deux images a été fait pour obtenir de l'opérateur les contours de toutes les composantes. La figure 5.54b illustre une carte de contours tracés manuellement par un opérateur. Cette carte est ensuite utilisée pour obtenir une carte de référence semblable aux cartes montrées dans les figures 5.54c, d, e et f. Les deux premières cartes ont été construites par des opérateurs humains non-experts du domaine d'application, tandis que les deux suivantes l'ont été par des biologistes experts du domaine d'application.

Les cartes présentées divisent les images en cinq régions : noyau, cytoplasme/pseudopode, halo blanc, fond et *autre*. Le cytoplasme et les pseudopodes d'une cellule ne sont pas séparés dans ces cartes. Vu la difficulté de fixer une limite entre ces deux concepts, car un pseudopode est en fait une extension du cytoplasme, et non une composante à part, nous avons préféré les réunir pour l'évaluation. Une classe *autre* réfère aux objets de l'image qui ne sont pas modélisés dans le système. Ils seront simplement ignorés.

Les principales difficultés de la construction de ces cartes ont été identifiées. Premièrement, la définition des halos blancs pose un réel problème, puisqu'il s'agit d'un concept immatériel, et sa délimitation n'est pas bien définie, même entre biologistes. Ensuite, lors de la segmentation manuelle des images, il a été révélé difficile, même pour un opérateur humain, expert ou non, de bien découper, avec précision, toutes les zones de l'image, en raison du faible contraste existant.

La figure 5.55 compare les cartes de référence établies par différents opérateurs à l'aide de matrices de confusion entre les cartes. Les comparaisons sont établies pour les cartes faites par deux opérateurs non-experts du domaine (figure 5.55a), par deux opérateurs experts du domaine (figure 5.55b) et entre une carte établie par un opérateur non-expert et une carte établie par un opérateur expert du domaine (figure 5.55c). Il est intéressant de noter les différences existantes entre toutes ces cartes. La composante posant le plus de problèmes est, tel que nous venons de le mentionner, le halo blanc, que chaque opérateur, expert ou non, interprète différemment. Il existe une très grande différence de reconnaissance de cette composante selon l'opérateur. Néanmoins, comme nous pouvions le présumer, l'écart entre les cartes de référence effectuées par les experts du domaine d'application est inférieur à celui qu'on peut observer entre les cartes effectuées par les non-experts du domaine.

5.4.2 Évaluation des segmentations

Les figures 5.56 et 5.57 montrent quelques résultats obtenus par le système. Ici, le système a été testé sur les deux premières images de deux séquences différentes, afin de montrer des cellules de formes variées. Il s'agit toutefois du même type de cellules dans

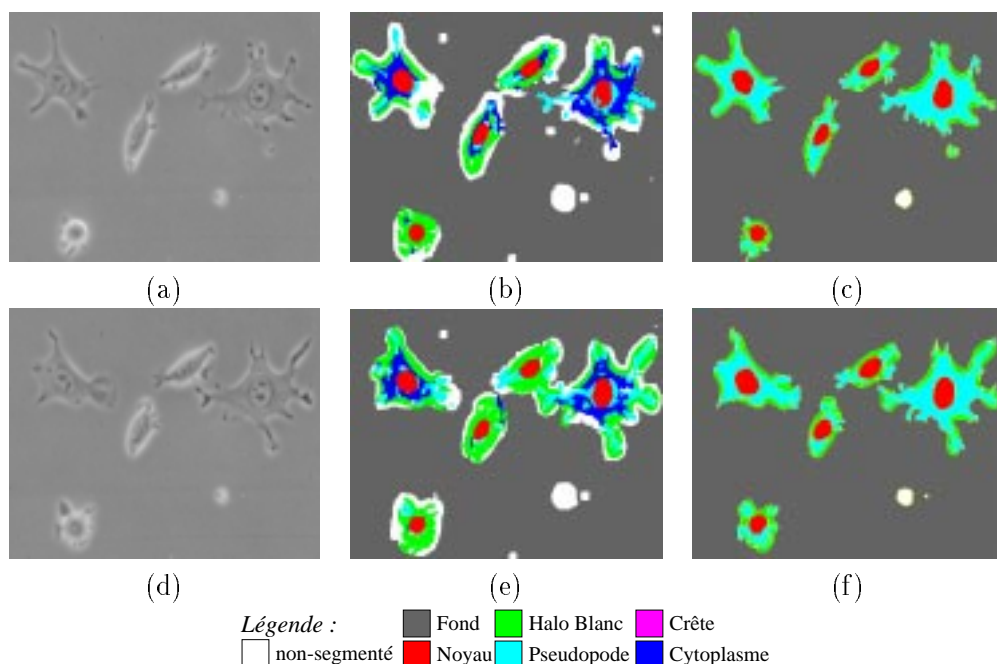


FIG. 5.56 – Résultats du système sur deux images (séquence 1). (a) (d) Images 1 et 2. (b) (e) Résultats obtenus. (c) (f) Cartes de référence. L'intervalle de temps entre les deux images est de 10 minutes. Les zones obtenues sont le fond, les noyaux, les cytoplasmes, les pseudopodes, les halos blancs et les parties non-segmentées. Sur la carte de référence, les cytoplasmes et les pseudopodes sont regroupés.

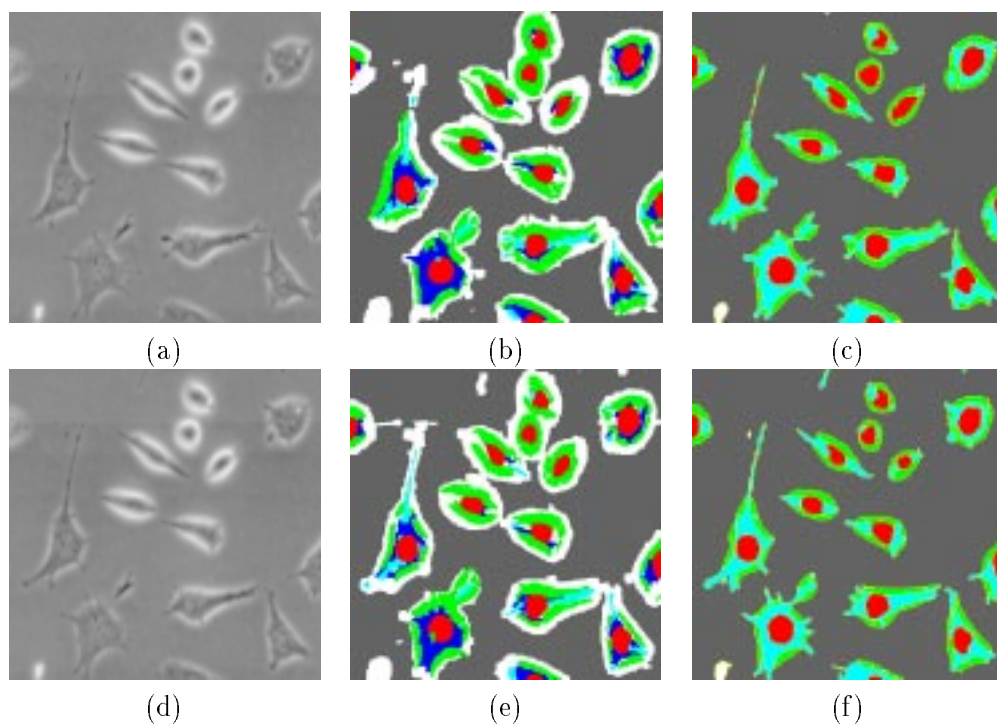


FIG. 5.57 – Résultats du système sur deux images (séquence 2). (a) (d) Images 1 et 2. (b) (e) Résultats obtenus. (c) (f) Cartes de référence. Les mêmes conditions que pour la séquence 1 ont prévalu.

Segmentation\Référence	Fond	Noyau	Halo	Cytoplasme	Autre	Total	% correct
<i>Fond</i>	114213	0	41	147	0	114401	99.8
<i>Noyau</i>	0	2828	20	109	0	2957	95.6
<i>Halo</i>	2443	79	3240	1482	2	7246	44.7
<i>Pseudeupode</i>	237	127	180	2809	1	3354	83.8
<i>Cytoplasme</i>	712	175	215	4788	1	5891	81.3
<i>Non-segmenté</i>	11175	16	845	2135	408	14579	2.8
Total	128780	3225	4541	11470	412	148428	-
% détecté	88.7	87.7	71.3	66.2	99.0	-	-

(a)

Segmentation\Référence	Fond	Noyau	Halo	Cytoplasme	Autre	Total	% correct
<i>Fond</i>	85257	37	9	70	0	85373	99.9
<i>Noyau</i>	29	5753	197	302	1	6282	91.6
<i>Halo</i>	4571	353	7073	1937	9	13943	50.7
<i>Pseudeupode</i>	29	21	30	2139	0	2219	96.4
<i>Cytoplasme</i>	1	208	69	4180	1	4459	93.7
<i>Non-segmenté</i>	21723	55	730	3449	278	26235	1.1
Total	111610	6427	8108	12077	289	138511	-
% détecté	76.4	89.5	87.2	52.3	96.2	-	-

(b)

FIG. 5.58 – Matrices de confusion pour l'évaluation quantitative des résultats. (a) Évaluation quantitative de la première image de la séquence 1. (b) Évaluation quantitative de la deuxième image de la séquence 2. Dans la colonne de gauche figure la classification réalisée par le système et en haut la classification de référence. Les pseudopodes sont inclus dans le cytoplasme dans cette dernière classification. L'évaluation effectuée en tient compte. La ligne non-segmenté indique les pixels manqués par les agents, tandis que la colonne autre indique les pixels qui n'appartiennent à aucun objet.

les deux séquences (fibroblastes murins L929).

Il est possible d'effectuer certaines remarques qualitatives sur les résultats obtenus. Le système conçu travaille en sous-segmentation. Ce qui signifie qu'un pixel n'est annexé par un agent que lorsque le degré de confiance envers ce pixel est élevé. Tous les pixels de l'image où un doute suffisant demeurerait, selon les évaluations des agents, n'ont pas été étiquetés. Ce qui explique les zones non segmentées des résultats obtenus.

Visuellement, le résultat est différent selon le type de composante considéré. Les noyaux sont bien segmentés, grâce à l'image en fluorescence. Les pseudopodes sont bien délimités. Par contre, toutes les tâches sombres de la cellule ont été marquées comme pseudopode, l'agent cytoplasme étiquetant le reste de la cellule. Ceci n'est pas gênant pour l'évaluation, puisque les deux composantes sont regroupées pour l'évaluation (qui est donc complaisante sur ce sujet). Les halos blancs sont cependant beaucoup plus importants qu'ils ne devraient l'être (notable par simple comparaison visuelle entre les segmentations et les références). Le fond entoure sommairement les cellules pour délimiter leurs surfaces.

La figure 5.58 montre l'évaluation quantitative effectuée entre les segmentations et les références pour deux des quatre images exposées. La méthode d'évaluation compare tous les pixels étiquetés des régions des deux images et produit une matrice d'évaluation (ou de confusion) sur la qualité des résultats. L'interprétation de ces évaluations, et des pourcentages calculés, peut porter à certaines confusions cependant. Par exemple pour le tableau 5.58a, deux pourcentages sont associés au type halo blanc, indiquant deux critères d'évaluation. Celui-ci a réussi à trouver 71,3 % des pixels halos de la référence. Cependant, seuls 44,7 % de tous les pixels qu'il a trouvés sont véritablement des halos. Le premier pourcentage illustre la conformité à la référence (trouver ce que l'on doit trouver), tandis que le deuxième est une mesure du discernement de l'agent (ne pas trouver plus que ce que l'on doit). Dans cet exemple, les agents halos blancs ont faussement segmenté plusieurs pixels qui n'étaient pas de type halo blanc.

Si on regarde l'ensemble des tableaux, on remarque que ce sont les agents cytoplasmes et pseudopodes qui ont manqué le plus de pixels (sous-segmentation), tandis que les agents halos blancs en ont segmenté beaucoup trop (sur-segmentation).

5.5 Conclusion

Ce chapitre a montré quelques résultats de notre système, en illustrant autant les différents aspects de conception de notre approche que les résultats de segmentation. Les aspects de l'adaptation, de liberté de mouvement de l'information et de souplesse du système vis-à-vis de l'application ont pu être démontrés sur des exemples précis. La méthode retenue pour évaluer les résultats de l'application permet de confronter le système au jugement des experts. Cependant, cette évaluation ne peut que mesurer la qualité du résultat final, sans tenir compte des succès individuels de chaque agent. Les résultats obtenus sont encourageants, même si certaines difficultés restent, comme la difficulté de trouver les bons paramètres de configuration pour les agents.

Différentes approches sont à l'étude au sein du laboratoire TIMC-IMAG pour l'étude quantitative du comportement migratoire cellulaire. En début de chapitre, il a été mentionné les approches par contours actifs [Leitner 95] et par flot optique [Germain 98, Doisy 89]. Il est utile de comparer l'apport de ces méthodes et de la nôtre pour cette étude.

L'approche par flot optique est réalisée sans segmentation de l'image et produit une carte des déplacements (vitesse et direction) pour chaque point de l'image et ce, sur une séquence d'images entière. Cette approche est fiable tant que les déplacements et les déformations entre images successives ne sont pas trop importantes. L'avantage de cette approche est dans l'étude des mouvements que peut subir une population entière de cellules (translation, rotation, expansion, contraction). Comme elle est réalisée sans segmentation, elle est robuste quelle que soit la densité de la population étudiée. Par contre, elle

est limitée, pour la même raison, à l'étude du mouvement global d'une population (ou d'une cellule si elle est isolée manuellement), sans possibilité d'étude de la morphologie particulière des cellules.

L'approche par contours actifs permet, tout comme notre approche, de retrouver la morphologie des cellules. Ces deux approches sont, à la base, des méthodes de segmentation et de reconnaissance des cellules. Elles possèdent donc les avantages et les désavantages inhérents aux approches par segmentation. Le point fort de l'approche par contours actifs développée, comparée à notre approche, est la stabilité des résultats tout au long de la séquence d'images (initialisation sur une image avec les résultats de l'image précédente), et le point faible est le manque d'adaptation locale aux endroits peu contrastés ou très texturés (sur certains pseudopodes par exemple). Le contour actif fournit également une bonne description de la morphologie globale de la cellule (contour fermé par définition), tandis que notre approche fournit une meilleure description des différentes composantes de la cellule.

Globalement, ces trois approches sont perçues comme étant complémentaires pour l'étude envisagée. Il est également possible de faire coopérer ces méthodes dans un même système. Par exemple, Germond a développé une approche de coopération entre un modèle de contours actifs et un système basé sur des agents de croissance de région [Germond 97]. Cette approche a été appliquée pour la segmentation d'images du cerveau.

Chapitre 6

Perspectives et conclusion

Après avoir présenté et discuté tous les aspects du système développé dans les précédents chapitres, ce chapitre va s'attarder à présenter quelques perspectives d'évolution afin d'améliorer le système. Trois idées de recherche sont présentées en relation avec le système, mais selon des axes différents. Nous décrirons d'abord une évolution du système en tant que système général de vision, en présentant une approche d'apprentissage qui a été réalisée afin de pallier les problèmes déjà énoncés sur la configuration du système. Ensuite, quelques idées sont exposées pour améliorer la modélisation des connaissances de configuration. Finalement, une perspective d'utilisation du système pour l'application à la reconnaissance des cellules est présentée, afin de montrer l'importance d'un tel système dans l'étude des comportements cellulaires. Enfin, suite à ces perspectives, une conclusion générale sur le système et sa contribution est présentée.

6.1 L'apprentissage

Tout au long de ce travail, l'importance de l'adaptation, pour un système de vision, a été discutée sous de multiples facettes. Cette adaptation permet au système de s'affranchir des difficultés locales au contexte, mais certains problèmes demeurent, comme le réglage des multiples paramètres nécessaires pour *calibrer* le système adaptatif (paragraphe 5.3.2.4). Afin de résoudre ce problème, nous avons étudié les moyens possibles pour que le système réalise lui-même ce calibrage des paramètres.

Cela nous a amené à considérer les méthodes d'apprentissage [AFIA 95]. L'apprentissage consiste, dans notre contexte, à développer une expérience sur les exécutions précédentes du système, afin d'en déduire et d'estimer les meilleures configurations des agents pour la résolution du problème posé. Cette notion d'apprentissage est très voisine de l'adaptation. Nous différencions ces deux notions par deux critères : l'échelle temporelle et la référence utilisée. L'adaptation est pour nous la faculté pour un système d'estimer ses paramètres pendant une exécution, mais sans mémoire des exécutions précédentes, tandis

que l'apprentissage conserve cette dimension temporelle en se rappelant les expériences passées. De plus, l'adaptation permet d'estimer les paramètres en fonction du contexte de l'agent, tandis que l'apprentissage fait plutôt référence à une base de connaissances représentant l'expérience accumulée.

Avec l'adaptation, les agents sont capables de s'affranchir de plusieurs situations inattendues et variées, mais ils n'obtiennent aucun retour de leur travail [Maes 94]. Ils ne peuvent ainsi s'améliorer dans l'accomplissement de leur tâche avec l'expérience. Cette expérience permettrait à l'agent de devenir encore plus robuste dans son exécution.

6.1.1 L'apprentissage en vision

Maes identifie cinq objectifs pour une méthode d'apprentissage afin qu'elle puisse s'intégrer à un agent autonome [Maes 94]:

1. l'apprentissage doit être incrémental, afin de permettre à l'agent de bénéficier de chaque expérience ;
2. l'apprentissage doit être orienté vers les connaissances utiles pour accomplir le but de l'agent ;
3. l'apprentissage doit être robuste face au bruit, aux variations d'environnement, aux erreurs des capteurs, etc. ;
4. l'apprentissage doit être non-supervisé, pour conserver l'esprit d'autonomie de l'agent ;
5. l'apprentissage devrait pouvoir bénéficier d'un certain degré de connaissances a priori, afin d'éviter de partir de zéro.

Ces cinq objectifs correspondent à l'apprentissage idéal pour un agent afin de lui permettre de conserver son autonomie et ses capacités d'adaptation. Cependant, il est très difficile de concevoir une telle approche, particulièrement dans le cadre d'une application de vision, où l'agent a de la difficulté à évaluer chacune de ses expériences. En effet, le problème de l'évaluation d'un résultat de segmentation n'est pas résolu, particulièrement dans un cadre distribué comme le nôtre où il est difficile de séparer le travail effectué par un agent de celui effectué par les autres. Pour cela, nous utiliserons les cartes de référence de segmentation que nous avons construites (paragraphe 5.4). Mais cette fois, il ne s'agira pas d'évaluer le résultat global de la segmentation, mais bien d'adapter la mesure au travail individuel de chaque agent.

Différentes approches ont été utilisées pour l'apprentissage dans un système de vision [Bhanu 94b, Crevier 97]. Plusieurs méthodes d'apprentissage ont été utilisées, comme, entre autres, l'apprentissage par induction, analytique, par étude de cas, par méthodes génétiques ou par méthodes connexionnistes. Ces méthodes ont été appliquées sur différents

aspects du problème de la vision, comme par exemple les modèles des objets à reconnaître, les paramètres des algorithmes utilisés, les stratégies d'interprétation et de reconnaissance des objets, ou encore les stratégies d'évaluation des résultats.

Bhanu et al. ont cherché à optimiser les paramètres de segmentation à l'aide d'algorithmes génétiques [Bhanu 94a]. Pour l'apprentissage, les paramètres de segmentation ont été corrélés avec les caractéristiques de l'image, ainsi qu'avec des variables relatives à leur application (scènes d'extérieur), comme l'heure d'acquisition de l'image et la nébulosité. Cette corrélation a servi à déterminer les paramètres de segmentation adaptés pour chaque situation.

Draper a développé des techniques d'apprentissage de stratégie d'interprétation dans le cadre du système Schéma (VISIONS) [Draper 96]. La méthode consistait à générer un arbre des hypothèses et stratégies possibles pour la reconnaissance. Ensuite, l'apprentissage effectué consistait en une optimisation de ce graphe.

Guarda a traité le problème de la génération automatique de détecteurs visuels [Guarda 98]. Son approche vise l'apprentissage de détecteurs de classes d'objets, appliqués à la reconnaissance d'éléments du visage. L'apprentissage procède par induction sur un ensemble d'exemples, et combine les techniques du calcul évolutionniste comme l'algorithme génétique et la programmation génétique. Les détecteurs ainsi appris sont composés de masques dont les paramètres sont appris et de plans, appris également, pour combiner les différents masques de reconnaissance.

Ainsi, les travaux de recherche sur l'utilisation de méthodes d'apprentissage en vision deviennent de plus en plus nombreux. Il s'agit d'une voie naturelle vers laquelle se dirigent les chercheurs, car il devient évident qu'un système de vision performant devra, dans un proche avenir, pouvoir apprendre à développer par lui-même ses capacités :

It seems to be a common feeling that animals learn to see, and this feeling, together with the re-emergence of computer learning paradigms that mimic many forms of human learning, has raised hopes that learning is the key to the computer vision problem. Indeed, it seems clear that Nature does not "program" all our visual capabilities into the genome, and we certainly know that programming a computer with a closed-form solution to the vision problem is a daunting task.

[Brown 94, p. 1]

6.1.2 L'approche réalisée

Une expérience d'apprentissage a été réalisée avec le système que nous avons développé. L'idée de base est d'aider l'utilisateur qui se voit confronté aux réglages de nombreux paramètres lorsqu'il veut définir des types d'agents. L'influence de ce réglage a déjà été démontrée au paragraphe 5.3.2.4. Beaucoup de paramètres existent et ils ont des influences très variées sur le comportement du système.

6.1.2.1 Description de l'approche

Cet apprentissage a été réalisé dans le cadre d'un projet de DEA effectué par Long-Hai Le Huynh [Le Huynh 98] au sein du laboratoire TIMC-IMAG, dont les aspects principaux sont reproduits ici. Il a porté sur les paramètres de configuration des agents, tels que définis dans le fichier de configuration. Trois objectifs ont été définis pour l'apprentissage de ces paramètres :

- conserver les paramètres ayant conduit à une bonne segmentation ;
- modifier les paramètres ayant conduit à des segmentations erronées ;
- déterminer l'importance relative de chaque paramètre dans la configuration de l'agent.

Le but final d'un tel apprentissage est de pouvoir donner à chaque agent, lors de son initialisation, le jeu de paramètres optimaux pour qu'il puisse mener à bien sa tâche. Donc, il ne s'agit plus d'avoir des configurations standard par type d'agent, mais d'adapter la configuration de chaque agent en fonction de son contexte.

Pour l'instant, seul un apprentissage numérique sur les paramètres de configuration du comportement de perception a été effectué. Il est limité à trois critères de perception (paragraphe 3.5) : seuillage, homogénéité et compacité. Les données symboliques ont été adaptées pour être compatibles avec les méthodes numériques employées.

L'approche développée comporte deux phases :

1. l'*apprentissage*, supervisé, où le travail des agents est évalué et représenté dans une base d'exemples ;
2. la *prédiction*, où les agents sont configurés à partir des exemples précédemment appris.

Même si les deux phases de l'apprentissage sont séparées, il est possible d'intégrer de nouveaux exemples dans la base pendant la phase de prédiction. Cependant, une méthode d'évaluation, c'est-à-dire une carte de référence de segmentation, devra être disponible afin d'évaluer ces nouveaux exemples.

Les problèmes principaux rencontrés lors de ce projet, et qui existent dans toute approche d'apprentissage, sont les suivants :

- déterminer ce qui doit être appris ;
- déterminer les méthodes d'évaluation pour les nouveaux individus qu'on va insérer dans la base d'apprentissage ;
- utiliser les données apprises pour améliorer les prochains résultats.

6.1.2.2 Phase d'apprentissage

Le but de la phase d'apprentissage est d'initialiser plusieurs agents avec des configurations différentes, définies au hasard, et ensuite d'observer et de répertorier le travail qu'ils ont pu ainsi réaliser. Ainsi, certaines configurations se révéleront plus adaptées que d'autres en fonction des différents contextes présents. À la fin, il sera possible d'avoir une configuration adaptée pour chaque contexte possible pour un agent, plutôt qu'une configuration par type d'agent. L'adéquation entre une configuration et un contexte est menée par le biais d'une carte de référence.

Nous disposons donc de trois groupes de variables caractérisant des situations différentes. Chaque situation devient par la suite un échantillon, ou individu, pour la phase d'apprentissage. On espère ainsi établir des corrélations entre ces trois groupes pour donner, par la suite, le bon ensemble de paramètres à un nouvel agent qui sera lancé. Ces trois groupes de variables sont :

- les paramètres de configuration de l'agent ;
- les informations statistiques sur la primitive, définissant le contexte ;
- l'évaluation de la segmentation de l'agent.

Les *paramètres de configuration* retenus sont les paramètres du comportement de perception comme le seuil de l'évaluation globale et les poids et facteurs des différents critères de perception (voir la définition de ces critères au paragraphe 3.5). Afin de tenir compte de la nature symbolique des différents critères, ceux-ci sont représentés sous plusieurs combinaisons, comme autant de critères différents. De trois critères génériques¹, on obtient ainsi 28 critères spécifiques² qui peuvent être appris de manière numérique, pour être compatible avec les autres paramètres de l'apprentissage. Par exemple, les variables retenues sont (27 variables au total) :

- le seuil de l'évaluation globale ;
- le poids du critère de compacité³ ;
- le poids du critère de seuillage sur l'image source ;
- le facteur du critère de seuillage sur l'image source ;
- le poids du critère de seuillage sur l'image de variance ;
- le facteur du critère de seuillage sur l'image de variance ;

1. Seuls les critères de seuillage, d'homogénéité et de compacité avaient été retenus dans le cadre de ce travail. Le critère d'inclusion n'a pas encore fait l'objet d'apprentissage.

2. Chacun de ces critères est spécifique puisque l'image de référence pour appliquer le critère, lorsqu'il y a lieu, est déjà spécifiée. Chaque combinaison fait ainsi référence à une image différente.

3. Il n'y a aucun paramètre à donner pour ce critère autre que son poids dans l'évaluation globale.

- ...
- le poids du critère d’homogénéité sur l’image source ;
- le facteur du critère d’homogénéité sur l’image source ;
- ...

Les *informations statistiques* sur une primitive sont calculées à partir des informations accumulées sur la primitive. Il comprend les informations suivantes (9 variables) :

- niveau de gris (moyenne, moyenne des carrés, écart-type) ;
- norme du gradient (moyenne, moyenne des carrés, écart-type) ;
- variance (moyenne, moyenne des carrés, écart-type).

L’évaluation de la segmentation demeure le problème le plus délicat. Il ne s’agit pas ici d’évaluer la segmentation globale d’une image, mais de celle d’un agent, dans un contexte distribué. L’agent n’est pas seul pour accomplir la tâche de segmentation donnée, car plusieurs agents travaillent en parallèle sur chaque composante de l’image. Il est donc difficile d’évaluer s’il a bien réussi ou non la portion qui lui incombait. L’évaluation retenue a cependant été la comparaison entre la segmentation de l’agent et la carte de référence définie pour cette image (paragraphe 5.4). Il faut néanmoins noter que cette évaluation peut favoriser les agents travaillant en sous-segmentation, puisqu’un agent n’ayant segmenté que quelques pixels qui sont justes sera valorisé, au sens de cette évaluation, par rapport à un autre agent qui aura segmenté entièrement la composante plus quelques autres pixels. Il s’agit d’une limite de cette évaluation dont il faut être conscient.

Une analyse en composante principale est effectuée par type d’agent (*Noyau*, *Pseudopode*, *Cytoplasme*, ...). Il existe donc un espace de représentation par spécialisation d’agent. Dans cet espace, tous les individus d’un même type sont donc répertoriés, avec leur configuration particulière, leur segmentation et leur taux de réussite. La figure 6.59 représente quelques types d’agent avec la représentation des trois premiers plans principaux de chaque analyse.

Dans l’état actuel du travail, ces données demeurent difficiles à interpréter (figure 6.59). Mentionnons simplement l’homogénéité des individus de type fond par opposition aux individus de types halo blanc et pseudopode, qui montre la présence pour ces derniers de situations bien différentes.

Avec cet ensemble de paramètres de natures diverses, on obtient un vecteur de 37 variables caractérisant une situation donnée de la segmentation. La prochaine étape est de représenter ce vecteur dans un espace de paramétrisation où il sera plus aisé de retrouver les corrélations entre les différentes variables et d’identifier des éléments caractéristiques pour chaque situation. Pour ce faire, la méthode de l’analyse en composantes principales

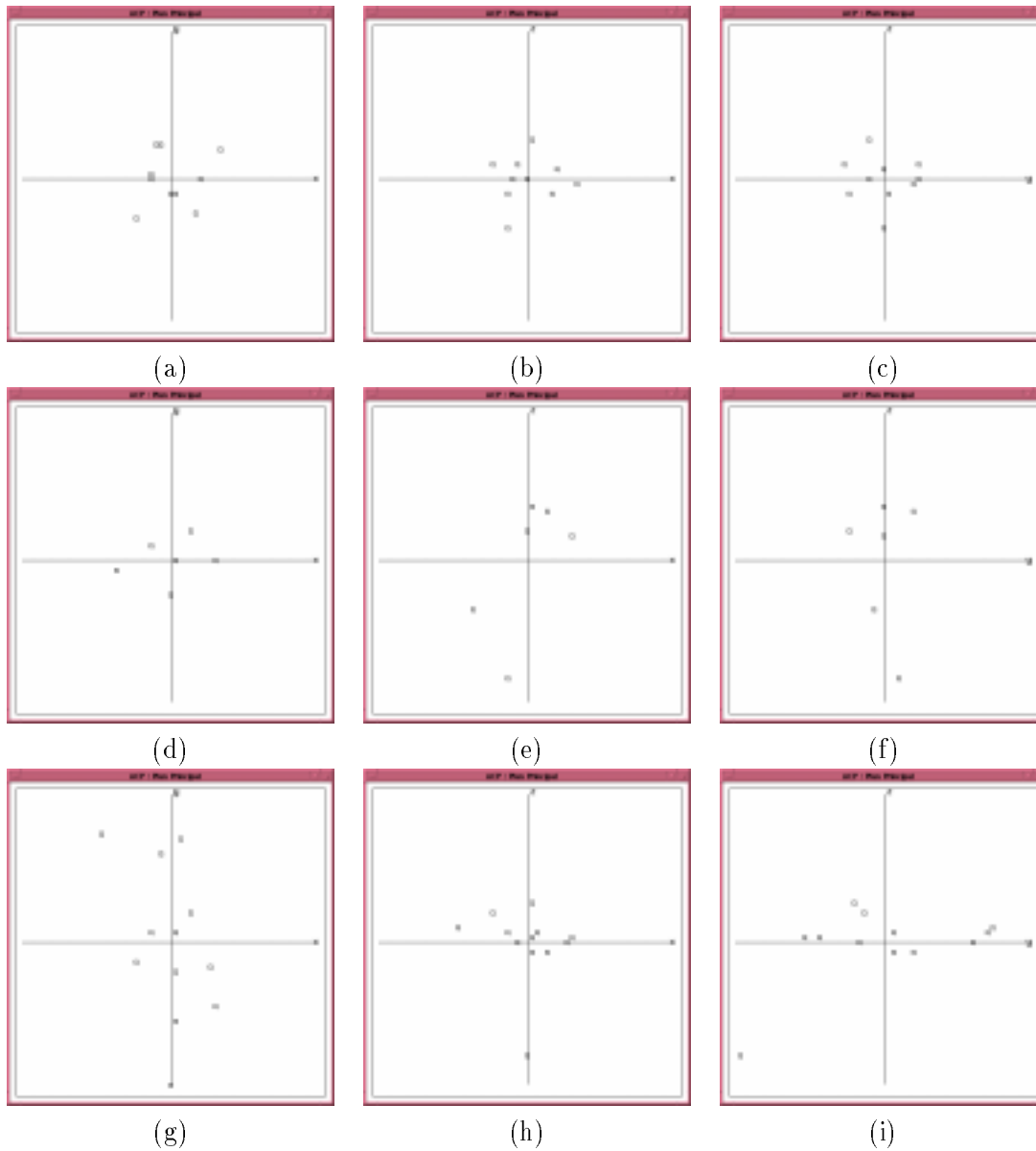


FIG. 6.59 – Représentation des individus de l'apprentissage pour chaque type d'agent. Chaque type a fait l'objet d'une ACP dont les trois premiers plans principaux sont montrés ici. (a) (b) (c) Individus de référence de type Fond. (d) (e) (f) Individus de référence de type Halo blanc. (g) (h) (i) Individus de référence de type Pseudopode.

(ACP) a été retenue afin de représenter les données et identifier des classes de regroupement de ces données.

6.1.2.3 Phase de prédiction

La phase suivante, après avoir récolté des données pendant la première phase d'apprentissage, est de tenter de prédire, pour un nouvel agent d'un certain type initialisé en un

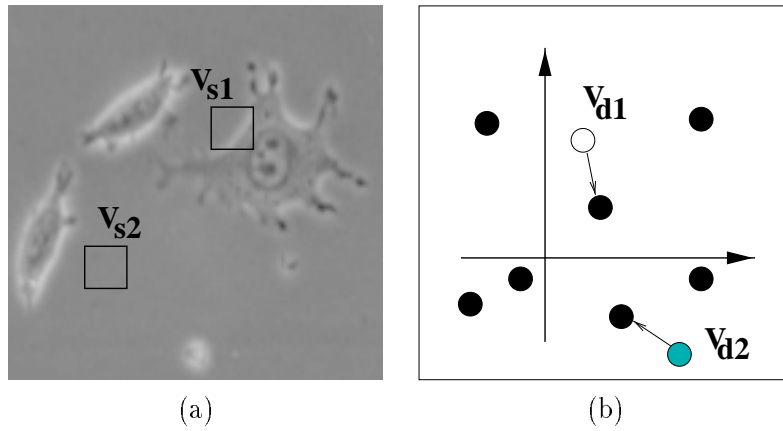


FIG. 6.60 – *Espaces de travail pour la prédiction de la configuration d'un nouvel agent lancé. (a) Analyse des informations contextuelles au germe créé dans l'environnement. (b) Recherche dans l'espace des paramètres (ACP), la configuration la plus adaptée au germe nouvellement créé.*

certain endroit de l'image, quels seront les paramètres de configuration les plus adaptés pour sa tâche de segmentation⁴. Pour cela, il faut construire un vecteur de données sur ce nouvel agent et le comparer avec la base d'apprentissage déjà établie pendant la première phase afin d'établir la meilleure configuration possible pour cet agent.

Lorsqu'un agent est initialisé en un certain lieu de l'image, il est possible d'observer le contexte dans lequel il devra travailler, c'est-à-dire les valeurs de niveaux de gris, de variance et de norme de gradient autour du nouveau germe. Ces caractéristiques photométriques sont mesurés dans une fenêtre 9x9 autour du germe (figure 6.60a). Ces données vont constituer la première partie du vecteur de comparaison, sur le contexte. On procède ensuite à une initialisation par défaut des autres variables du vecteur, afin de pouvoir projeter celui-ci dans l'espace de représentation des données de l'apprentissage. Ensuite, il est possible de trouver, par une mesure de distance dans le nouvel espace, les paramètres de configuration les plus appropriés pour le nouvel agent (figure 6.60b). Dans cette mesure de distance, les individus ayant déjà, par le passé, réalisé de bonnes segmentations (bonnes évaluations) sont favorisés.

Le nouvel agent possède maintenant sa configuration propre établie en fonction des caractéristiques locales de son environnement. À la fin de la segmentation, si une carte de référence existe pour l'image segmentée, il pourra être décidé d'évaluer les résultats obtenus par cet agent et de les insérer dans la base d'apprentissage si l'évaluation est suffisamment élevée. Ainsi, on s'apparente à un apprentissage incrémental.

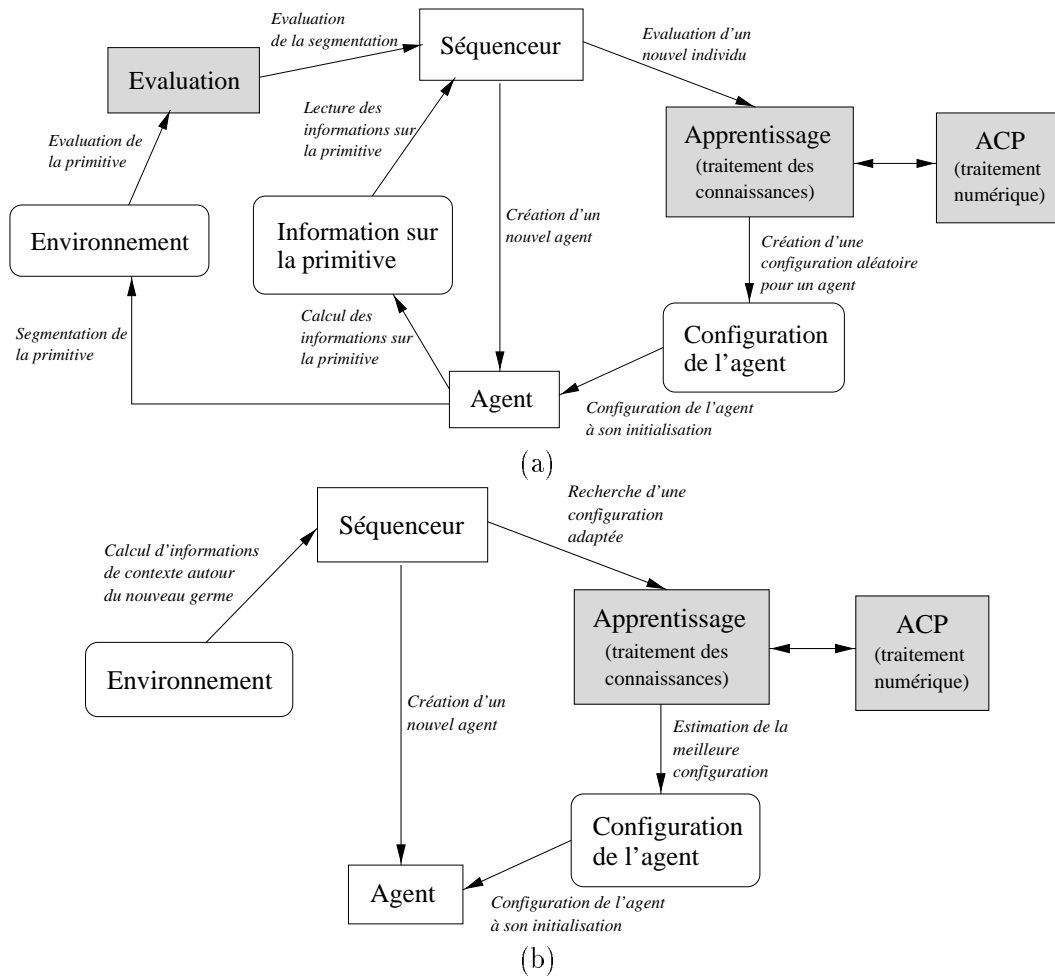


FIG. 6.61 – Modèle du système en intégrant les modules d'apprentissage (boîtes grises). (a) Fonctionnement du système pendant la phase d'apprentissage. (b) Fonctionnement du système pendant la phase de prédiction.

6.1.2.4 Intégration au système existant

L'intégration de l'apprentissage au système existant s'est faite en ajoutant deux nouveaux modules dédiés à cette tâche (figure 6.61) : le module d'apprentissage et le module d'ACP [Le Huynh 98].

Le module d'apprentissage traite les connaissances, c'est-à-dire :

- les critères pour décider de garder un nouvel individu ou non ;
- les statistiques sur le flux des individus et leur utilisation ;
- la répartition des données dans les vecteurs de l'ACP ;

4. On ne parle ici que de segmentation, puisque seuls les critères du comportement de perception ont fait l'objet d'un apprentissage.

- la signification de chaque variable des vecteurs.

Le module d'ACP traite le calcul numérique, c'est-à-dire qu'il gère, pour chaque type d'agent :

- la population des individus, en insérant les nouveaux, ou en remplaçant certains individus ;
- l'espace des 3 premiers vecteurs principaux de l'espace de représentation ;
- la transformation permettant la projection d'un individu dans cet espace ;
- la représentation de chaque individu dans cet espace.

De plus, un autre module traitant de l'évaluation doit être ajouté pour la phase d'apprentissage. Ce module réalise les tâches d'évaluation, tel que présenté au paragraphe 5.4.

L'intégration de l'apprentissage s'est effectuée d'une manière globale, sans modifier les agents existants. En fait, le rôle d'interface avec les modules d'apprentissage est affecté au séquenceur. En effet, que ce soit en phase d'apprentissage ou en phase de prédiction, les appels aux modules dédiés s'effectuent lors de la création (affectation de paramètres) et de la destruction (récolte de paramètres). Ces deux opérations étant réalisées par la séquenceur, c'est par lui que les informations nécessaires sont transmises au module d'apprentissage (figure 6.61).

6.1.2.5 Perspectives

Pour l'instant, il n'est pas possible de conclure à une amélioration de la segmentation avec la méthode d'apprentissage réalisée [Le Huynh 98]. En fait, le système d'apprentissage retrouve par lui-même les paramètres de configuration qui avaient été définis par un opérateur expert du système. Ce dernier point est encourageant pour la suite des recherches.

Dans la suite des travaux, il serait intéressant d'optimiser les paramètres de configuration en fonction du contexte local de l'agent dans l'image. Par exemple, deux agents pseudopodes travaillant sur deux cellules différentes exploiteraient des paramètres de configuration différents. Cet aspect pourrait également être intégré à un niveau cellulaire, en définissant des configurations adaptées en fonction de la nature de chaque cellule (taille, brillance, forme).

Les paramètres définissant d'autres comportements peuvent également être appris. Par exemple, le comportement de reproduction possède une logique de paramétrisation de critères semblables au comportement de perception. Il serait intéressant de tenter d'appliquer cette méthode pour ce comportement. Ainsi, c'est la stratégie d'exploration de l'image qui pourrait être améliorée en optimisant les paramètres utilisés.

L'apprentissage effectué dans la version actuelle est purement numérique avec l'utilisation de l'analyse en composantes principales. Cependant, afin de tenir compte de la nature des paramètres existants dans la configuration des différents comportements, il faudrait intégrer une nature symbolique à cet apprentissage. Les aspects symboliques des critères ont été contournés en développant les combinaisons possibles, et en éliminant à la fin les critères de poids nul. Certaines particularités de ces critères ont dû être omises pour parvenir à cette fin. La réalisation d'un apprentissage intégrant les aspects numériques et symboliques permettrait de prendre en compte toute la richesse des paramètres existants.

Enfin, l'intégration de l'apprentissage s'est effectuée, pour l'instant, au niveau global du système, avec une interface orientée vers le séquenceur plutôt que vers les agents. Cependant, la gestion des données et l'interaction avec les modules d'apprentissage pourraient être réalisées au niveau des agents. En distribuant ainsi l'approche, il serait intéressant d'étudier les interactions possibles entre agents qui pourraient contribuer à développer l'apprentissage. Plusieurs catégories d'apprentissage peuvent ainsi être envisagées pour s'appuyer sur le caractère distribué du système selon la définition des échanges réalisés entre les agents ([Brazdil 91] cité par [Vercouter 98]) :

- *avant l'apprentissage* : les agents communiquent d'abord leurs données et l'apprentissage est ainsi centralisé dans chaque agent ;
- *pendant l'apprentissage* : les agents apprennent en même temps qu'ils envoient leurs données, ce qui nécessite un algorithme d'apprentissage spécialisé ;
- *après l'apprentissage (intégration de la connaissance)* : les agents apprennent localement sur leurs données avant de communiquer aux autres leur résultat et les conclusions ainsi obtenues sont intégrées dans chaque agent ;
- *après l'apprentissage (résolution distribuée du problème)* : les agents apprennent d'abord localement puis c'est pendant la résolution d'un problème qu'ils font part de leurs différents point de vue si nécessaire.

6.2 Modélisation des connaissances

Dans notre travail, l'accent a été mis sur une modélisation opératoire de l'application plutôt que sur une modélisation descriptive des objets. Les différents paramètres de configuration fournis pour les comportements permettent de définir les traitements à effectuer pour reconnaître les composantes de l'application. Les connaissances descriptives sur ces composantes sont construites pendant l'exécution du système, plutôt que d'être fournies a priori par la configuration des agents.

Afin de pouvoir proposer un modèle générique de conception d'un système de vision s'appuyant sur ces idées, il reste à effectuer un effort de formalisation des principes définis

dans notre modèle. Cet aspect fait déjà l'objet de travaux qui ont été entamés par Touboul [Touboul 98].

Touboul a proposé un modèle de représentation des connaissances, autant opératoires que descriptives, permettant de généraliser certains concepts présents dans notre système. Selon son approche, les composantes que nous avons identifiées dans notre application sont maintenant modélisées selon la répartition des valeurs selon les différentes caractéristiques de l'image (niveau de gris, variance, gradient, ...). L'hypothèse sous-jacente de cette modélisation est la distribution selon une loi normale des différentes valeurs pour chacune de classes (composantes) identifiées. Cependant, son modèle ne requiert pas une modélisation complète des composantes, en fonction de la connaissance qui est disponible et des traitements envisagés.

La localisation des différentes composantes, qui était effectuée selon une recherche sur des profils de pixels dans notre système, est maintenant formalisée selon quatre paramètres :

- direction de recherche de la composante (ou de son germe) ;
- l'angle de recherche (ouverture) de la composante par rapport à la première ;
- les distances, minimales et maximales, de recherche.

En fait, il s'agit dans ce cas de rechercher les nouveaux germes dans une zone de recherche plutôt que sur une ligne (profil) définie. Ce système de représentation permet de définir les relations spatiales retrouvées dans plusieurs systèmes de vision, comme MESSIE-II [Sandakly 95].

La reconnaissance des composantes dans l'image se fait selon deux étapes : localisation d'un objet (hypothèse de présence d'un échantillon de l'objet) et segmentation de celui-ci (ou rejet de l'hypothèse si la segmentation n'est pas possible). La suite de ce travail sera d'intégrer une modélisation multi-agents pour compléter l'effort de modélisation des connaissances effectuée. Le principal intérêt de cette approche pour notre travail reste dans la définition de mécanismes de représentation des connaissances selon des critères génériques, ce qui permet d'envisager l'utilisation de notre système pour d'autres domaines d'application.

6.3 L'étude des cellules

L'adhérence et la migration cellulaires sont largement impliquées dans de nombreux processus physiologiques normaux et pathologiques :

- La formation d'une tumeur maligne met en jeu une succession d'événements moléculaires et cellulaires qui aboutissent à des modifications des propriétés d'adhérence

de cellules en cours de prolifération. Les cellules cancéreuses peuvent alors envahir les tissus adjacents, être disséminées à distance du site d'origine et former des métastases. Toute modification des propriétés d'adhérence cellulaire est susceptible d'entraîner une évolution vers une pathologie cancéreuse.

- Les phénomènes d'adhérence et de migration cellulaires sont aussi très largement impliqués lors du développement embryonnaire, la cicatrisation des blessures tissulaires, et les réponses immunes et inflammatoires. Des progrès considérables ont été accomplis ces dernières années concernant les molécules impliquées dans le processus migratoire, les protéines d'adhérence présentes à la surface des cellules ; cependant, les mécanismes moléculaires qui gouvernent ces processus sont encore très largement incompris. L'étude du rôle des molécules d'adhérence associée à la détermination des caractéristiques migratoires constituent des éléments-clefs pour expliquer le comportement des cellules tumorales. Une meilleure compréhension des mécanismes par lesquels les récepteurs d'adhérence cellulaires modulent la capacité migratoire des cellules pourrait également faciliter le développement de nouveaux agents thérapeutiques.

L'analyse quantitative conjointe du rôle des molécules d'adhésion sur les paramètres de la migration cellulaire constitue un outil essentiel pour mieux comprendre les processus de différenciation normale (ontogénèse) et pathologique (oncogénèse).

La mise au point d'un outil d'analyse dynamique devrait avoir un impact important sur les recherches dans ce domaine. Après un développement sur un modèle cellulaire bien caractérisé, il sera appliqué à divers types cellulaires dans différents contextes physiologiques ou pathologiques et fournira les critères dynamiques permettant de différencier leur pouvoir métastatique.

Ces approches novatrices d'analyse sur cellules vivantes permettront notamment d'établir les relations entre les propriétés d'adhérence et de migration cellulaire au cours du processus de cancérisation.

Par exemple, une application est envisagée avec le système actuel pour analyser le mouvement et la déformation des pseudopodes. Le but ici est de caractériser le mouvement des pseudopodes par rapport au mouvement global de la cellule, et d'étudier, parmi les pseudopodes, lesquels permettent à la cellule de progresser dans son environnement et lesquels sont destinés à explorer cet environnement.

En culture et suivant leur type, les cellules ont la propriété de se déplacer par émission de prolongements (ou pseudopodes) tout en se déformant. En cancérologie, il est important de pouvoir analyser et moduler la capacité migratoire des cellules pathologiques. Les outils actuellement disponibles pour étudier à un niveau microscopique les différents aspects de la motilité et des déformations cellulaires restent extrêmement limités car de nature essentiellement statique. Ainsi, l'utilisation de ce système, combiné avec le développement

de méthodes d'analyse, permettrait le développement de modèles de la cellule et de ses composantes.

Cet axe de travail peut être combiné avec le développement de modèles théoriques destinés à la simulation des capacités migratoires de la cellule [Dugnolle 98]. Ainsi, le trio reconnaissance, modélisation et simulation peut permettre l'étude de nouvelles théories en biologie sur le développement de cellules pathologiques ou encore sur la cicatrisation de plaies ou de brûlures.

6.4 Conclusion

La présentation, tout au long de ce document, de l'approche que nous avons réalisée, a permis de démontrer qu'il est possible d'opérationnaliser des principes de conception originaux, tels que la gestion des informations et la construction incrémentale de connaissances.

Afin de préserver la liberté de mouvement des informations, et de s'assurer d'une coopération fructueuse entre les différentes sources de connaissances, il est important de maintenir la compatibilité et la souplesse dans la représentation des informations dans un système de vision. Ce constat vise un équilibre entre le besoin de connaissances pour résoudre le problème de la vision et la réalité voulant que ces connaissances ne soient disponibles qu'une fois le traitement effectué.

Un système de vision se doit d'être réactif et adaptatif aux différents contextes présents dans les images qu'il aura à traiter. Ces deux qualités sont nécessaires tant au niveau des traitements que le système doit effectuer qu'au niveau du contrôle de ces traitements. Ces deux aspects que sont le traitement et le contrôle sont difficilement dissociables dans le système. Plutôt que de tenter cette séparation, nous proposons de les entrelacer pour réaliser incrémentalement le travail de reconnaissance des différents objets de l'application, selon un mode d'exécution parallèle pour mettre en avant la complémentarité des différents modules du système.

Notre approche a été développée dans le cadre d'une application de reconnaissance des cellules vivantes. Nous avons tenté de généraliser les concepts de cette approche afin de montrer son apport dans le cadre général de la vision par ordinateur. Cependant, elle n'a pas la prétention d'être fonctionnelle pour tout type d'application de vision. Mais à la lumière des différentes perspectives présentées dans ce chapitre, il est possible d'imaginer son utilisation future dans des domaines d'application similaires par la représentation des objets de l'image et par la définition du type d'étude envisagé sur les images.

Annexe A

Fichier de configuration

```
# Configuration générale
Démarrage automatique          : oui
Suivi de cellules               : oui

# Configuration du séquenceur
Quantum (minimum et maximum)   : 30 150

# Configuration de l'interface visuelle
Interface visible               : oui
Facteur gamma pour l'affichage : 3.5
Nombre d'images affichées      : 3
Affichage segmentation primitive : oui

# Configuration pour l'affichage sur l'image source
Schéma des germes              : non
Largeur du carre d'affichage   : 3
Affichage des traits           : non
Agents marqués en fin de vie   : non

Affichage de la segmentation   : oui
Affichage des périmètres       : oui

# Sauvegarde des résultats
Sauvegarde segmentation composante : oui
Sauvegarde segmentation primitive : oui
Sauvegarde segmentations superposées : oui
Sauvegarde cellules            : oui
Sauvegarde informations        : oui
```

Agent Région	:	Fond
 <i># PARAMÈTRES GLOBAUX</i>		
Composante groupe	:	Non
Segmentation intermédiaire	:	non
Priorité de l'agent	:	2
Suivi de l'agent	:	non
Affichage sur l'image source	:	non
Affichage du germe	:	non
Tracé du périmètre	:	non
 <i># PARAMÈTRES PERCEPTION</i>		
Comportement perception actif	:	oui
 <i># perception évaluation</i>		
Seuil de l'évaluation	:	0.5
Critère perception	:	0.7 homogénéité variance 6
Critère perception	:	0.3 compacité
 <i># perception autres</i>		
Perception précédente utilisée	:	non
Remplissage des trous	:	oui
Taille minimale pour conserver	:	10
 <i># PARAMÈTRES INTERACTION</i>		
Comportement fusion actif	:	oui
 <i># PARAMÈTRES DIFFÉRENCIATION</i>		
Comportement différenciation actif	:	non
 <i># PARAMÈTRES REPRODUCTION</i>		
Comportement reproduction actif	:	non
 <i># reproduction évaluation</i>		
Seuil de l'évaluation	:	0.8
Critère reproduction	:	0.5 extréma global < variance
Critère reproduction	:	0.5 extréma local <= variance 5

```

Agent Région : Noyau

# PARAMÈTRES GLOBAUX
Composante groupe : oui
Segmentation intermédiaire : non
Priorité de l'agent : 1

Suivi de l'agent : oui
Affichage sur l'image source : non
Affichage du germe : non
Tracé du périmètre : non

# PARAMÈTRES PERCEPTION
Comportement perception actif : oui

# perception évaluation
Seuil de l'évaluation : 0.4
Critère perception : 0.5 seuillage noyau 0.5
Critère perception : 0.5 compacité

# perception autres
Perception précédente utilisée : non
Remplissage des trous : oui
Adaptation selon info cellule : non
Taille minimale pour conserver : 10

# PARAMÈTRES INTERACTION
Comportement fusion actif : non

# PARAMÈTRES DIFFÉRENCIATION
Comportement différenciation actif : non

# PARAMÈTRES REPRODUCTION
Comportement reproduction actif : oui

# reproduction évaluation
Seuil reproduction : 0.35
Critère reproduction : 1.0 extréma global source < Fond
Critère reproduction : + Noyau 0.5 distance > 10
Critère reproduction : init

# reproduction autres
Reproduction même image début : Fond x 7
Reproduction même image début : Crête x 18
Reproduction même image fin : Pseudopode x 14
Reproduction même image fin : Mouvement x 15
Reproduction même image fin : Cytoplasme x 11
Reproduction image suivante : Noyau

```

Agent Région : HaloBlanc

PARAMÈTRES GLOBAUX

Composante groupe : oui
Segmentation intermédiaire : non
Priorité de l'agent : 1

Suivi de l'agent : non
Affichage sur l'image source : non
Affichage du germe : non
Tracé du périmètre : non

PARAMÈTRES PERCEPTION

Comportement perception actif : oui

perception évaluation

Seuil de l'évaluation : 0.6
Critère perception : 0.8 seuillage source 0.8
Critère perception : 0.2 compacité

perception autres

Perception précédente utilisée : non
Remplissage des trous : oui
Adaptation selon info cellule : oui
Taille minimale pour conserver : 10

PARAMÈTRES INTERACTION

Comportement fusion actif : oui

PARAMÈTRES DIFFÉRENCIATION

Comportement différenciation actif : non

PARAMÈTRES REPRODUCTION

Comportement reproduction actif : oui

reproduction évaluation

Seuil reproduction : 0.4
Critère reproduction : 0.35 extréma global > source
Critère reproduction : 0.35 extréma local > source
Critère reproduction : 0.15 extréma local < variance
Critère reproduction : 0.15 opposition global > graddir
Critère reproduction : Crête 1. distance 2

reproduction autres

Reproduction même image fin : Pseudopode x 18
Reproduction même image fin : Mouvement x 10
Reproduction même image fin : Cytoplasme x 11
Reproduction image suivante : Noyau

```

Agent Région : Pseudopode

# PARAMÈTRES GLOBAUX
Composante groupe : oui
Segmentation intermédiaire : non
Priorité de l'agent : 1

Suivi de l'agent : non
Affichage sur l'image source : non
Affichage du germe : non
Tracé du périmètre : non

# PARAMÈTRES PERCEPTION
Comportement perception actif : oui

# perception évaluation
Seuil de l'évaluation : 0.5
Critère perception : 0.4 seuillage variance > Fond
Critère perception : 0.4 seuillage source < Fond
Critère perception : 0.2 compacité

# perception autres
Perception précédente utilisée : oui
Remplissage des trous : oui
Adaptation selon info cellule : non
Taille minimale pour conserver : 10

# PARAMÈTRES INTERACTION
Comportement fusion actif : oui

# PARAMÈTRES DIFFÉRENCIATION
Comportement différenciation actif : non

# PARAMÈTRES REPRODUCTION
Comportement reproduction actif : oui

# reproduction évaluation
Seuil de l'évaluation : 0.35
Critère reproduction : 1.0 extréma global source < Fond
Critère reproduction : + Noyau 0.5 distance > 10

# reproduction autres
Reproduction même image fin : Cytoplasme x 7
Reproduction même image fin : Crête x 12
Reproduction même image fin : Pseudopode x 12
Reproduction même image fin : Mouvement x 15

```

Agent Région : Cytoplasme

PARAMÈTRES GLOBAUX

Composante groupe : oui
Segmentation intermédiaire : non
Priorité de l'agent : 1

Suivi de l'agent : non
Affichage sur l'image source : non
Affichage du germe : non
Tracé du périmètre : non

PARAMÈTRES PERCEPTION

Comportement perception actif : oui

perception évaluation

Seuil de l'évaluation : 0.5
Critère perception : 0.3 homogénéité source 8
Critère perception : 0.1 inclusion Noyau
Critère perception : 0.5 inclusion
Critère perception : 0.2 compacité

perception autres

Perception précédente utilisée : non
Remplissage des trous : oui
Adaptation selon info cellule : non
Taille minimale pour conserver : 10

PARAMÈTRES INTERACTION

Comportement fusion actif : oui

PARAMÈTRES DIFFÉRENCIATION

Comportement différenciation actif : non

PARAMÈTRES REPRODUCTION

Comportement reproduction actif : oui

reproduction évaluation

Seuil de l'évaluation : 0.6
Critère reproduction : 0.5 inclusion
Critère reproduction : 0.5 distance 0.5

reproduction autres

Reproduction même image fin : Pseudopode x 12
Reproduction même image fin : Mouvement x 15

Agent Région : Mouvement

PARAMÈTRES GLOBAUX

Composante groupe : non
Segmentation intermédiaire : oui
Priorité de l'agent : 1

Suivi de l'agent : non
Affichage sur l'image source : non
Affichage du germe : non
Tracé du périmètre : non

PARAMÈTRES PERCEPTION

Comportement perception actif : oui

perception évaluation

Seuil de l'évaluation : 0.6
Critère perception : 0.5 homogénéité diffsigne 0.1
Critère perception : 0.5 seuillage diffnorme 0.5

perception autres

Perception précédente utilisée : non
Remplissage des trous : oui
Adaptation selon info cellule : non
Taille minimale pour conserver : 10

PARAMÈTRES INTERACTION

Comportement fusion actif : oui

PARAMÈTRES DIFFÉRENCIATION

Comportement différenciation actif : oui

différenciation évaluation

Poids des critères : 0.3 0.5 0.2

différenciation autres

Identités possibles : Fond HaloBlanc
Identités possibles : Pseudopode Cytoplasme
Redémarrage de la perception : oui

PARAMÈTRES REPRODUCTION

Comportement reproduction actif : oui

reproduction évaluation

Seuil de l'évaluation : 0.5
Critère reproduction : 0.5 extréma global > diffnorme
Critère reproduction : 0.5 extréma local > diffnorme

reproduction autres

Reproduction même image fin : Mouvement x 10

Agent Contour : Crête

PARAMÈTRES GLOBAUX

Composante groupe : non
Segmentation intermédiaire : oui
Priorité de l'agent : 1

Suivi de l'agent : non
Affichage sur l'image source : non
Affichage du germe : non

PARAMÈTRES PERCEPTION

Comportement perception actif : oui

perception évaluation

Seuil de l'évaluation : 0.75
Critère perception : 0.25 extréma local < gradnorme
Critère perception : 0.35 extréma local > source
Critère perception : 0.3 opposition local > graddir
Critère perception : 0.1 taille 10

perception autres

Taille minimale pour conserver : 10

PARAMÈTRES INTERACTION

Comportement fusion actif : oui

fusion autres

Distance maximale d'exploration : 20

PARAMÈTRES REPRODUCTION

Comportement reproduction actif : oui

reproduction évaluation

Seuil reproduction : 0.4
Critère reproduction : 0.35 extréma global > source
Critère reproduction : 0.35 extréma local > source
Critère reproduction : 0.15 extréma local < variance
Critère reproduction : 0.15 opposition global > graddir

reproduction autres

Reproduction même image fin : HaloBlanc x 10

Bibliographie

- [AFIA 95] A. Cornuéjols, M. Moulet, and R. Vincent. Dossier apprentissage. *Bulletin de l'AFIA*, (22), pp. 14–45, July 1995.
- [Aloimonos 87] Y. Aloimonos, I. Weiss, and A. Bandyopadhyay. Active Vision. *Int. J. Comput. Vision*, 1(4), pp. 333–356, January 1987.
- [Aloimonos 90] Y. Aloimonos. Purposive and Qualitative Active Vision. In *IEEE Int. Conf. on Pattern Recognition*, vol. 1, 346–360, 1990.
- [Aloimonos 94] Y. Aloimonos. What I have learned. *CVGIP: Image Understanding*, 60(1), pp. 74–85, July 1994.
- [Bajcsy 88] R. Bajcsy. Active Perception. *Proc. of IEEE*, 76(8), pp. 996–1005, August 1988.
- [Ballard 89] D.H. Ballard. Reference Frames for Animate Vision. In *Int. Joint Conf. on Artificial Intelligence*, vol. 2, 1635–1641, 1989.
- [Ballard 92] D.H. Ballard and C.M. Brown. Principles of Animate Vision. *CVGIP: Image Understanding*, 56(1), pp. 3–21, July 1992.
- [Barron 94] J.L. Barron, D.J. Fleet, and S.S. Beauchemin. Performance of optical flow techniques. *International Journal on Computer Vision*, 12, pp. 43–77, 1994.
- [Bellet 95] F. Bellet, J.M. Salotti, and C. Garbay. Une approche opportuniste et coopérative pour la vision de bas niveau. *Traitement du Signal*, 12(5), pp. 479–494, 1995.
- [Bellet 98] F. Bellet. *Une approche incrémentale, coopérative et adaptative pour la segmentation des images en niveaux de gris*. Thèse de doctorat, Institut National Polytechnique de Grenoble, France, June 1998.
- [Bhanu 94a] B. Bhanu, J. Ming, and S. Lee. *Genetic Learning for Adaptive Image Segmentation*. Kluwer Academic, 1994.
- [Bhanu 94b] Special Section on learning in computer vision. Une approche opportuniste et coopérative pour la vision de bas niveau. *IEEE Trans. Patt. Anal. Mach. Int.*, 16(9), pp. 865–919, 1994.
- [Boissier 92] O. Boissier and Y. Demazeau. A Distributed Artificial Intelligence View on General Purpose Vision Systems. In *Decentralized AI*, vol. 3, 311–330. Elsevier Science, 1992.
- [Boissier 93] O. Boissier. *Problème du contrôle dans un système intégré de vision*. Thèse de doctorat, Institut National Polytechnique de Grenoble, France, January 1993.
- [Boucher 96a] A. Boucher and C. Garbay. A Multi-Agent System to Segment Living Cells. In *IEEE Int. Conf. on Pattern Recognition*, vol. 3, 558–562, 1996. Vienna (Austria).

- [Boucher 96b] A. Boucher and C. Garbay. Segmentation de séquences d'images cytologiques par un système multi-agents. In *Journées Francophones IAD SMA*, 125–135. Hermès, 1996. Port Camargue (France).
- [Boucher 97] A. Boucher, A. Doisy, X. Ronot, and C. Garbay. A Society of Goal-Oriented Agents for the Analysis of Living Cells. In E. Keravnou, C. Garbay, R. Baud, and J. Wyatt, editors, *Proc. 6th AIME*, vol. 1211 of *LNAI*, 429–440. Springer-Verlag, 1997. Grenoble (France).
- [Boucher 98a] A. Boucher, A. Doisy, X. Ronot, and C. Garbay. Cell Migration Analysis After In Vitro Wounding Injury With a Multi-Agent Approach. *Artif. Intell. Rev.*, 12(1), February 1998.
- [Boucher 98b] A. Boucher, A. Doisy, X. Ronot, and C. Garbay. A Society of Goal-Oriented Agents for the Analysis of Living Cells. *Artif. Intell. Med.*, 14, pp. 183–199, 1998.
- [Boucher 98c] A. Boucher and C. Garbay. Des agents spécialisés pour la compréhension de séquences d'images. In *Actes 11ème RFIA*, vol. 2, 275–284, 1998. Clermont-Ferrand (France).
- [Bouthemy 95] P. Bouthemy. Interactions entre segmentation et mesure dans l'analyse du mouvement 2D : orientations récentes et perspectives. In *École thématique du GDR: Traitement du signal et images*, 1–11, 1995. Dourdan (France).
- [Brazdil 91] P. Brazdil, M. Gams, S. Sian, L. Torgo, and W. van de Velde. Learning in distributed systems and multi-agent environment. In *Proc. of European Workshop Session on Machine Learning*, 424–439, March 1991. Porto (Portugal).
- [Brooks 86] R.A. Brooks. Achieving Artificial Intelligence Through Building Robots. Technical Report TR 899, MIT, 1986.
- [Brooks 90] R.A. Brooks. Elephants Don't Play Chess. *Robotics and Autonomous Systems*, 6, pp. 3–15, 1990.
- [Brooks 91a] R.A. Brooks. Intelligence Without Reason. In *Int. Joint Conf. on Artificial Intelligence*, 569–595, August 1991. Sydney (Australia).
- [Brooks 91b] R.A. Brooks. Intelligence without representation. *Artif. Intell.*, 47, pp. 139–159, 1991.
- [Brown 94] C.M. Brown. Vision, Learning and Development. Technical Report Technical Report 492, University of Rochester, Computer Science Department, May 1994.
- [Castelfranchi 90] C. Castelfranchi. Social Power: A Point Missed in Multi-Agent, DAI and HCI. In *Decentralized AI*, vol. 1, 33–46. Elsevier Science, 1990.
- [Charroux 96] B. Charroux. *Analyse d'images : coopération d'opérateurs de segmentation guidée par l'interprétation*. Thèse de doctorat, Université Paris XI Orsay, France, January 1996.
- [Clément 93a] V. Clément, G. Giraudon, S. Houzelle, and F. Sandakly. Interpretation of remotely sensed images in a context of multi-sensor fusion using a multispecialist architecture. *IEEE Trans. Geosci. and Remote Sensing*, 31(4), pp. 779–791, 1993.
- [Clément 93b] V. Clément and M. Thonnat. Integration of image processing procedures, ocap: a knowledge-based approach. *CVGIP: Image Understanding*, 57(2), March 1993.

- [Cloppet-Oliva 96a] F. Cloppet-Oliva. *Analyse d'images de cultures cellulaires obtenues par microscope optique : application à des images de neuroblastomes de souris*. Thèse de doctorat, Université René Descartes - Paris V, France, 1996.
- [Cloppet-Oliva 96b] F. Cloppet-Oliva and G. Stamon. Segmentation Coopérative Région/Contour pour une Analyse Automatique d'Images de Cellules en Culture. In *Proc. 10ème RFIA*, vol. 2, 1063–1072, 1996.
- [Clouard 94] R. Clouard. *Planification incrémentale et opportuniste appliquée à la construction de graphes d'opérateurs de Traitement d'Images*. Thèse de doctorat, Université de Caen, France, February 1994.
- [Clouard 95a] R. Clouard, A. Elmoataz, C. Porquet, and M. Revenu. BORG: a Knowledge-based system for the automation of Image Segmentation Task. In *IEEE Int. Conf. on Image Processing and its Applications*, 524–528, 1995. Edinburgh (UK).
- [Clouard 95b] R. Clouard, C. Porquet, and A. Elmoataz and M. Revenu. Why building Knowledge-Based Image Segmentation is so difficult. In *Int. Conf. on Knowledge-Based Systems for Reusing Programs*, 137–148, 1995. Sophia-Antipolis (France).
- [Crevier 97] D. Crevier and R. Lepage. Knowledge-Based Image Understanding Systems: A Survey. *Computer Vision and Image Understanding*, 67(2), pp. 161–185, August 1997.
- [Crowley 93] J. Crowley and H. Christensen, editors. *Vision as Process*, Basic Research Series. Springer-Verlag, 1993.
- [CVGIP 94] M.J. Tarr and M.J. Black. Dialogue: A Computational and Evolutionary Perspective on the Role of Representation in Vision. *CVGIP: Image Understanding*, 60(1), July 1994.
- [Dalle 98] P. Dalle and P. Dejean. Planification en traitement d'image: Approche basée sur les données. In *Actes 11ème RFIA*, vol. 2, 75–84, 1998. Clermont-Ferrand (France).
- [Doisy 89] A. Doisy, X. Ronot, and P. Tracqui. Quantification des déformations et de la migration cellulaires: de la mitose à la cicatrisation in vitro. In X. Ronot and D. Schoëvaërt, editors, *Dynamique de la cellule vivante*. Ed. INSERM, 1989. sous presse.
- [Doisy 96] A. Doisy, S. Paillason, P. Tracqui, F. Germain, F. Leitner, M. Robert-Nicoud, and X. Ronot. In Vitro Dynamics of Chromatin Organization and Migration. *Cell Biol. Toxicol.*, 12, pp. 363–366, 1996.
- [Draper 89] B.A. Draper, R.T. Collins, J. Brolio, and A.R. Hanson. The Schema system. *Int. J. Comput. Vision*, 2, pp. 209–250, 1989.
- [Draper 96] B.A. Draper. *Learning Control Strategies for Object Recognition*. Oxford University Press, 1996.
- [Drogoul 92] A. Drogoul and J. Ferber. From Tom Thumb to the Dockers: some experiments with foraging robots. In J.A. Meyer, H. Roitblat, and S. Wilson, editors, *From Animals to Animats: Second Conference on Simulation of Adaptive Behaviour*. MIT Press, 1992.
- [Dubuisson 90] B. Dubuisson. *Diagnostic et reconnaissance des formes*, chapter 4. Traité des nouvelles technologies. Hermès, 1990.

- [Dugnolle 98] P. Dugnolle, C. Garbay, and P. Tracqui. A mechanical model to simulate cell reorganisation during in-vitro wound healing. In *Proc European Simulation Multiconference*, 343–347. SCS Europe, 1998. Manchester (UK).
- [Durfee 87] E.H. Durfee, V.R. Lesser, and D.D. Corkill. Coherent Cooperation Among Communicating Problem Solvers. *IEEE Trans. on Computers*, 36(11), pp. 1275–1291, 1987.
- [Fan 96] J. Fan, R. Wang, L. Zhang, D. Xing, and F. Gan. Image Sequence Segmentation Based on 2D Temporal Entropic Thresholding. *Pattern Recognition Lett.*, 17, pp. 1101–1107, 1996.
- [Fenet 98] S. Fenet and S. Hassas. Une approche multi-agents de résolution de problèmes par interaction : cas de l'équilibrage dynamique multi-critères. In *Journées Francophones IAD SMA*, 115–130. Hermès, 1998. Pont-à-Mousson (France).
- [Ferber 95] J. Ferber. *Les systèmes multi-agents : vers une intelligence collective*. Inter-Editions, 1995.
- [Fu 81] K.S. Fu and J.K. Mui. A survey on image segmentation. *Patt. Recog.*, 13(1), pp. 3–16, 1981.
- [Galliers 90] J.R. Galliers. The Positive Role of Conflict in Cooperative Multi-Agent Systems. In *Decentralized AI*, vol. 1, 33–46. Elsevier Science, 1990.
- [Garbay 98] C. Garbay, F. Bellet, and A. Boucher. Des agent situés pour l'interprétation de scènes. *Revue d'Intelligence Artificielle*, 12(1), pp. 11–36, 1998.
- [Garnesson 91] P. Garnesson. *MESSI : un système d'analyse de scène, application à l'imagerie aérienne*. Thèse de doctorat, Université de Nice - Sophia Antipolis, France, 1991.
- [Germain 98] F. Germain, A. Doisy, X. Ronot, and P. Tracqui. Characterization of cell migration using a parametric estimation of image motion. *IEEE Trans. Bio. Eng.*, 1998. sous presse.
- [Germond 97] L. Germond, C. Garbay, C. Taylor, and S. Solloway. Coopération entre processus guidés par les données et processus guidés par les modèles pour la segmentation. In *Actes GRETSI*, vol. 2, 371–374, 1997. Grenoble (France).
- [Guarda 98] A. Guarda. *Apprentissage génétique de règles de reconnaissance visuelle : Application à la reconnaissance d'éléments du visage*. Thèse de doctorat, Institut National Polytechnique de Grenoble, France, October 1998.
- [Guessoum 94] Z. Guessoum. Systèmes asynchrones de production. In *Journées Francophones IAD SMA*, 169–180, 1994. Voiron (France).
- [Guessoum 96a] Z. Guessoum. *Un environnement opérationnel de conception et de réalisation de systèmes multi-agents*. Thèse de doctorat, Université Pierre et Marie Curie, France, 1996.
- [Guessoum 96b] Z. Guessoum and M. Dojat. A Real-Time Agent Model in a Asynchronous Object Environment. In *Agents Breaking Away*, LNAI, 1990–2003, January 1996. Eindhoven (Netherlands).
- [Guessoum 96c] Z. Guessoum and R. Durand. Un système multi-agents pour modéliser l'évolution économique. In *Journées Francophones IAD SMA*, 47–57. Hermès, 1996. Port Camargue (France).
- [Gwydir 94] S.H. Gwydir, H.M. Buettner, and S.M. Dunn. Non-rigid motion analysis of the growth cone using continuity splines. *Innov. Tech. Biol. Med.*, 15(3), pp. 309–321, 1994.

- [Hanson 78] A. Hanson and E. Riseman. *VISIONS: A Computer System for Interpreting Scenes*, 303–333. Academic Press, 1978.
- [Hayes-Roth 95] B. Hayes-Roth. An Architecture for Adaptive Intelligent Systems. *Artif. Intell.*, 72, pp. 329–365, 1995.
- [Hennebert 96] C. Hennebert, V. Rebuffel, and P. Bouthemy. Structuration Spatio-temporelle d’une Scène Texturée : Segmentation au Sens du Mouvement et de la Profondeur. In *Proc. 10ème RFIA*, vol. 2, 838–847, 1996.
- [Horn 80] B.K.P. Horn and B. Schunck. Determining Optical Flow. *Artif. Intell.*, 17, pp. 185–203, 1980.
- [Jain 81] R. Jain. Extraction of motion information from peripheral processes. *IEEE Trans. Patt. Anal. Mach. Int.*, 3(5), pp. 489–503, 1981.
- [Jolion 94] J-M. Jolion. Computer Vision Methodologies. *CVGIP: Image Understanding*, 59(1), pp. 53–71, January 1994.
- [Kara Falah 94] R. Kara Falah, P. Bolon, and J.P. Cocquerez. A Region-Region and Region-Edge Cooperative Approach of Image Segmentation. In *IEEE Int. Conf. on Image Processing*, vol. 3, 470–474, 1994.
- [Kermad 97] C.D. Kermad. *Segmentation d’images : Recherche d’une mise en oeuvre automatique par coopération de méthodes*. Thèse de doctorat, Université de Rennes I, France, July 1997.
- [Le Huynh 98] L.H. Le Huynh. *Apprentissage pour la paramétrisation d’agents de segmentation*. Rapport de DEA, Institut National Polytechnique de Grenoble, France, June 1998.
- [Lefèvre 94] V. Lefèvre. *Système multi-agents d’aide à la photo-interprétation sur architecture multi-processeurs*. Thèse de doctorat, Université Paris Sud, France, November 1994.
- [Leitner 95] F. Leitner, S. Paillason, X. Ronot, and J. Demongeot. Dynamic functional and structural analysis of living cells : new tools for vital staining of nuclear DNA and for characterization of cell motion. *Acta Biotheor.*, (43), pp. 299–317, 1995.
- [Leung 95] M. K. Leung and Y.H. Yang. First Sight: A Human Body Outline Labeling System. *IEEE Trans. Patt. Anal. Mach. Int.*, 17(4), pp. 359–377, 1995.
- [Leymarie 93] F. Leymarie and M.D. Levine. Tracking Deformable Objects in the Plane Using an Active Contour Model. *IEEE Trans. Patt. Anal. Mach. Int.*, 15(6), pp. 617–634, 1993.
- [Liedtke 87] C.E. Liedtke, T. Gahm, F. Kappei, and B. Aeikens. Segmentation of Microscopic Cell Scenes. *Analyt. Quant. Cytol. Histol.*, 9, pp. 197–211, 1987.
- [Lockett 94] S.J. Lockett and B. Herman. Automatic Detection of Clustered, Fluorescent-Stained Nuclei by Digital Image-Based Cytometry. *Cytometry*, 17, pp. 1–12, September 1994.
- [Maes 89] P. Maes. How to do the right thing. *Connection Sci. J.*, 1(3), 1989.
- [Maes 94] P. Maes. Modeling Adaptive Autonomous Agents. *Artificial Life Journal*, 1-2(1-2), April 1994.
- [Marr 82] D. Marr. *Vision*. W.H. Freeman, 1982.
- [Matsuyama 90] T. Matsuyama and V. Hwang. *SIGMA: a Knowledge-based Aerial Image Understanding System*. Plenum, 1990.

- [Mayet 96] F. Mayet, J-C. Pinoli, and M. Jourlin. Justifications physiques et applications du modèle LIP pour le traitement des images obtenues en lumière transmise. *Traitement du Signal*, 13(3), pp. 251–262, 1996.
- [Nazif 84] A.M. Nazif and M.D. Levine. Low level Image Segmentation: An Expert System. *IEEE Trans. Patt. Anal. Mach. Int.*, 6(5), pp. 555–577, March 1984.
- [Pal 93] N.R. Pal and S.K. Pal. A review on image segmentation. *Patt. Recog.*, 26(9), pp. 1277–1294, 1993.
- [Salotti 94] J.M. Salotti. *La gestion des informations dans les premières étapes de la vision par ordinateur*. Thèse de doctorat, Institut National Polytechnique de Grenoble, France, January 1994.
- [Sandakly 95] F. Sandakly. *Contribution à la mise en oeuvre d'une architecture à base de connaissances pour l'interprétation de scènes 2D et 3D*. Thèse de doctorat, Université de Nice - Sophia Antipolis, France, July 1995.
- [Sibarita 96] J.B. Sibarita. *Formation et restauration des images en microscopie à rayons X & application à l'observation d'échantillons biologiques*. Thèse de doctorat, Université Joseph Fourier - Grenoble 1, France, March 1996.
- [Siegert 94] F. Siegert, C.J. Weijer, A. Nomura, and H. Miike. A gradient method for the quantitative analysis of cell movement and tissue flow and its application to the analysis of multicellular Dictyostelium development. *J. Cell Sci.*, (107), pp. 97–104, 1994.
- [Spinu 97] C. Spinu. *Une approche multi-agents pour la segmentation d'images associant estimation et évaluation*. Thèse de doctorat, Université Joseph Fourier - Grenoble, France, April 1997.
- [Thompson 80] W.B. Thompson. Combining Motion and Contrast for Segmentation. *IEEE Trans. Patt. Anal. Mach. Int.*, 2(6), pp. 543–549, 1980.
- [Thonnat 93] M. Thonnat, V. Clément, and J. Van den Elst. Supervision of perception tasks for autonomous systems: the OCAPAPI approach, June 1993. Rapport de Recherche INRIA No 2000.
- [Touboul 98] E. Touboul. *Modèle d'architecture pour la mise en oeuvre d'un système de vision générique*. Thèse CNAM, December 1998.
- [Trivedi 89] M.M. Trivedi and A. Rosenfeld. On making computers "see". *IEEE Trans. Syst. Man Cybern.*, 19(6), pp. 1333–1335, 1989.
- [Vercouter 98] L. Vercouter, P. Beaune, and C. Sayettat. Apprentissages dans les SMA. In *Journées Francophones IAD SMA*, 341–354. Hermès, 1998. Pont-à-Mousson (France).
- [Von Martial 90] F. Von Martial. Interactions Among Autonomous Planning Agents. In *Decentralized AI*, vol. 1, 105–119. Elsevier Science, 1990.
- [Werner 90] E. Werner. Distributed Cooperation Algorithms. In *Decentralized AI*, vol. 1, 17–32. Elsevier Science, 1990.
- [Wooldridge 95] M. Wooldridge and N.R. Jennings. Intelligent Agents: Theory and Parctice. *Knowledge Engineering Review*, 10(2), pp. 115–152, 1995.
- [Wu 95] K. Wu, D. Gauthier, and M.D. Levine. Live Cell Image Segmentation. *IEEE Trans. Bio. Eng.*, 42(1), pp. 1–12, 1995.
- [Zhou 97] P. Zhou and D. Pycock. Robust Statistical-Models for Cell Image Interpretation. *Image Vision Comput.*, 15(4), pp. 307–316, April 1997.

Résumé

Cette thèse propose une nouvelle approche de la vision par ordinateur, insistant sur le rôle joué par les différentes informations et connaissances présentes dans un système intégré de vision. Ces connaissances peuvent être vues comme un but à atteindre dans la chaîne de traitement ou comme un instrument pour guider ces différents traitements. Ce double rôle est mis en évidence en insistant sur la gestion des informations disponibles à tout moment pour effectuer la tâche de reconnaissance. Une approche distribuée est proposée, sous la forme d'un système multi-agents, où chaque agent est spécialisé pour la reconnaissance d'un concept de l'image. Le modèle générique d'agent est composé de quatre comportements de base : *perception* (croissance de région, suivi de contour), *interaction* (fusion de primitives), *différenciation* (interprétation de primitive) et *reproduction* (stratégie de focalisation des agents). Ces comportements prennent des décisions à l'aide d'évaluations multi-critères, faisant référence à plusieurs critères spécialisés. Des capacités d'adaptation et de coopération sont nécessaires aux agents pour mener à bien leurs tâches, et ce tant au niveau local qu'au niveau des groupes d'agents (travaillant sur un même groupement perceptuel). Ce système est appliqué pour la reconnaissance des cellules en mouvement. Chaque cellule est divisée en plusieurs composantes (noyau, pseudopode, cytoplasme, ...) qui servent de concepts de base pour la spécialisation des agents.

Mots-clés : vision par ordinateur, intelligence artificielle distribuée, système multi-agents, information et connaissance, adaptation, coopération, cytologie