

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de L'Enseignement Supérieur et de la Recherche Scientifique
UNIVERSITE MOHAMED KHIDER
BISKRA

Faculté des Sciences et des Sciences de l'Ingénieur
Département d'Informatique

Mémoire de Magister
Option : Intelligence Artificielle et Images

Résolution de problèmes par écosystèmes :
Application au traitement d'images

Présenté par :
M^{me}. Boukhrouf Djemaa

Soutenu le 17/11/2005 devant le jury composé de :

Président	Dr. Khouadi Mohamed Khireddine, Docteur à l'université de Constantine.
Rapporteur	Pr. Batouche Med Chaouki, Professeur à l'Université de Constantine.
Examineurs	Dr. Benmohammed Mohamed, Maître de Conférence à l'Université de Constantine.
	Dr. Saidouni Djamel Eddine, Maître de Conférence à l'Université de Constantine

Chapitre**01****Systèmes complexes****Introduction**

L'étude des systèmes complexes s'est considérablement développée au cours des quinze dernières années, tant du point de vue des méthodes que de la formalisation et des outils.

Dans ce chapitre, nous effectuerons initialement un survol sur les concepts de systèmes complexes. La simulation de ces dernières est facilitée par l'approche multi-agent. Nous réservons alors une partie du chapitre pour exposer les concepts liés aux systèmes multi-agents et aux automates cellulaires. Ce chapitre aborde également les notions de vie artificielle, dont les approches sont parfois similaires aux simulations des systèmes complexes. Enfin, quelques techniques inspirées de systèmes naturels seront exposées.

1.1 Et les systèmes complexes ?

Dans cette section, nous reprenons quelques définitions trouvées dans la littérature scientifique.

Von Bertalanffy [Moise 01] définit des complexes d'«*éléments*». Il dit que l'on doit tenir compte de trois critères de distinction: le nombre d'éléments, leur espèce et les relations entre ces éléments. Pour lui, un complexe est la somme des éléments et des relations entre ceux-ci.

Jean-Louis Le Moigne [Moise 01] (et Clergue) fait la distinction entre systèmes complexes et systèmes compliqués. Les systèmes compliqués sont des systèmes que l'on peut réduire en éléments plus simples que l'on peut analyser séparément pour comprendre le système global. Dans le cas des systèmes complexes, la somme des éléments fait émerger de nouvelles propriétés qui ne sont pas dans les éléments eux-même.

D'après Sloot [Moise 01], un système complexe est une population d'un élément unique avec des attributs bien définis. De plus, si ces éléments ont des interactions non-linéaires, il se peut que l'on assiste à l'émergence d'un comportement global.

Donc, un système complexe est composé de nombreux éléments autonomes différenciés interagissant entre eux de manière non triviale. Il se caractérise également par l'émergence au niveau global de propriétés nouvelles, non observables au niveau des éléments constitutifs et par une dynamique de fonctionnement global difficilement prédictible à partir de l'observation et de l'analyse des constituants et de leurs interactions élémentaires. Cela implique en particulier qu'un système complexe ne peut être analysé par une décomposition en sous-systèmes plus simples.

1.2 Caractéristiques des systèmes complexes

On peut résumer les caractéristique des systèmes complexes comme suit [Bertelle 03,+]:

- Le caractère hétérogène des constituants de nombreux systèmes naturels (biologiques, économiques, ...)
- Un système complexe se caractérise également par l'émergence au niveau global de propriétés nouvelles, non observables au niveau des éléments constitutifs et par une

dynamique de fonctionnement global difficilement prédictible à partir de l'observation et de l'analyse des constituants et de leurs interactions élémentaires.

- Principe d'organisation en vue d'une fonction collective (survie, adaptation, reproduction) :
 - comportement téléologique, c'est à dire guidé par ses buts.
 - le concept d'instrumentation d'un tel comportement est le "feed-back" informationnel (issu de la notion de contrôle en cybernétique)
- Organisation hiérarchique de systèmes et sous-systèmes assurant sa stabilité.
- Non-linéarité.
- Stratégies de développement :
 - adaptatives : réorganisations structurales face à des fluctuations.
 - paradoxales: coexistence d'éléments contradictoires, complémentaires, concurrents ou antagonistes. (Ces deux stratégies ne sont pas exclusives ...)
- Irréductibilité.

1.3 Concepts de base

1.3.1 Agents et systèmes multi agents [Kévin 04,+] [Kévin 04,+] [Ferber 95]

Les Systèmes Multi-Agents ou SMA représentent une nouvelle façon d'analyser, de concevoir et d'implémenter des systèmes informatiques complexes. Les agents sont utilisés dans une variété d'applications sans cesse croissante. La classe des applications concernés par les SMA s'élargit de plus en plus.

Dans ce qui suit, on va donner quelques notions sur les agents et les Systèmes Multi-Agents (SMA).

1.3.1.1 Qu'est-ce qu'un agent ?

D'un point de vue étymologique, le mot *agent* vient du latin "*agere*" qui signifie agir. Littéralement, l'agent est donc celui qui agit.

Il existe, à l'heure actuelle, encore plusieurs définitions de ce qu'est un agent et aucune d'entre-elles n'est totalement admise. Ferber définit un agent comme une entité physique ou virtuelle [Ferber 95]:

- Qui est capable d'agir sur son environnement et sur les objets présents dans cet environnement (cela inclut les objets ressources ainsi que d'autres agents) ;
- Qui peut communiquer et interagir avec d'autres agents ;

- Qui est mue par un ensemble de tendances (sous la forme d'objectifs individuels ou d'une fonction de satisfaction, voire de survie, qu'elle cherche à optimiser) ;
- Qui possède des ressources propres ;
- Qui est capable de percevoir(mais de manière limitée) son environnement ;
- Qui ne dispose que d'une représentation partielle de cet environnement (et éventuellement aucune) ;
- Qui possède des compétences et offre des services ;
- Qui peut éventuellement se reproduire ;
- Dont le comportement est la conséquence de ses objectifs, de sa perception, de ses représentations, de ses compétences et des communications qu'elle peut avoir avec les autres agents.

A part les définitions du terme agent, il y a certaines propriétés qui caractérisent un agent. Ce sont ces caractéristiques particulières qui différencient les agents des autres concepts informatiques qui s'en rapprochent, tels que les « objets », les « acteurs », les « modules logiciels » ou bien les « processus parallèles ».

Un agent est [Kévin 04,+]:

- ↳ situé : il évolue dans un environnement
- ↳ autonome : il a la capacité d'agir sans aucune intervention extérieure
- ↳ actif : il est
 - *réactif* : il réagit aux changements de son environnement dans un temps limité
 - *proactif* : il est capable de prendre des initiatives sans qu'on le sollicite
 - *social* : il communique avec le monde qui l'entoure (êtres humains ou autres agents)

Il y a certains cas d'applications où des chercheurs attribuent aux agents des caractéristiques particulières, notamment des caractéristiques semblables à celles que les humains possèdent, pour modéliser le comportement des agents. Des caractéristiques comme la connaissance, les croyances, l'intention et l'obligation sont des exemples de notions mentales normalement constatées chez l'homme qui sont éventuellement utilisées dans la conception d'agents.

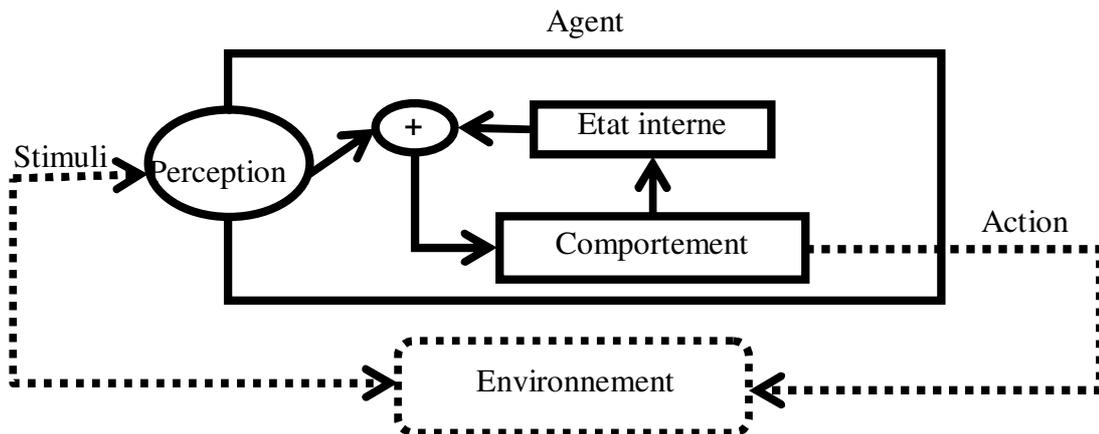


Figure 1.1 Schéma générale d'un agent, d'après [Ramat 98]

1.3.1.2 Différents types d'agents

Il y a deux grandes écoles de pensée dans la communauté multi-agents, chacune concevant les agents de manière différente :

- La première, l'école cognitive, conçoit les agents comme des entités déjà « intelligentes », c'est-à-dire capable de résoudre certains problèmes par eux-mêmes.
- La deuxième école, la réactive, conçoit les agents comme des entités très simples réagissant directement aux modifications de l'environnement.

a. Agents cognitifs

L'origine de l'approche cognitive se trouve dans la volonté de faire communiquer et coopérer des systèmes experts classiques. Dans les systèmes multi-agents cognitifs chaque agent est considéré comme un système expert d'où la cognition, synonyme de connaissance rationnelle, sert de guide à ses actions. Dans ce cadre, le système multi-agent est composé d'un petit nombre d'agents, où chaque agent dispose d'une base de connaissance comprenant l'ensemble des informations et le savoir-faire nécessaires à la réalisation de ses tâches et à la gestion des interactions avec les autres agents et avec l'environnement. Les agents cognitifs, en raison de leur sophistication et de leur capacité de raisonnement, peuvent travailler indépendamment sur certaines tâches. Ils peuvent ainsi résoudre des problèmes de manière relativement individuelle. Cette capacité offre une grande souplesse dans l'expression du comportement des agents, par contre ses limites sont mises en évidence par les problèmes rencontrés pour résoudre des conflits entre les différents buts des agents du système.

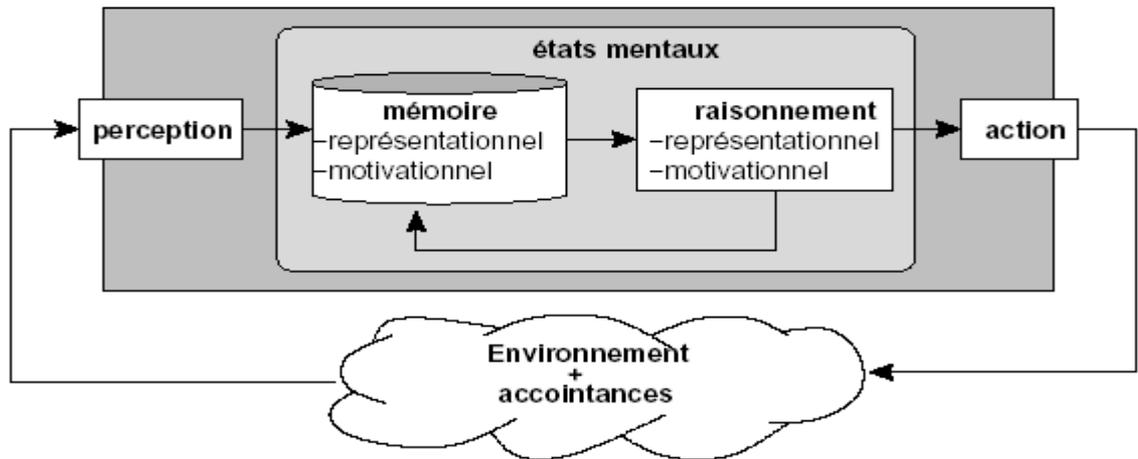


Figure 1.2 Agent cognitif, d'après [Duchesnay 01,a]

b. Agents réactifs

Les agents réactifs, contrairement aux cognitifs, n'ont pas une intelligence individuelle. Cette branche avance qu'il n'est pas nécessaire que les agents soient intelligents pour que le système ait un comportement global intelligent. Les agents réactifs possèdent des mécanismes simples de réaction aux événements. Ils ont parfois un objectif, mais ils n'ont aucune planification de la manière dont on peut arriver aux objectifs. Son mode de fonctionnement est basé sur une fonction du type $f(\text{stimulus}) \rightarrow \text{réponse}$, c'est-à-dire, ils n'agissent qu'à travers les stimulus aperçus dans l'environnement ou créés à partir d'autres agents.

En effet, la caractéristique principale des systèmes d'agents réactifs est le comportement émergent du système global qui n'est souvent pas prévu. Ainsi, une approche typique des systèmes multi-agents réactifs débute par la création d'éléments simples, gouvernés par des lois simples, qui interagissent pour développer une structure globale sans contrôleur responsable du comportement de chaque élément.

En effet, la division cognitif / réactif est parfois trop simpliste. Ferber nous présente une classification plus détaillée en citant d'autres termes utilisés par la communauté multi-agent pour plus de détails consulter [Ferber 95].

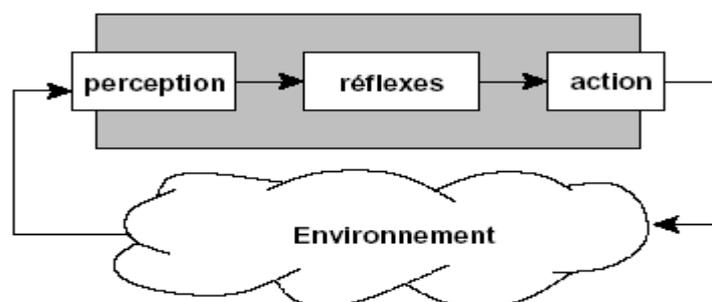


Figure 1.3 Agent réactif d'après [Duchesnay 01,a]

1.3.1.3 Systèmes Multi-Agents (SMA)

Les Systèmes Multi-Agents sont nés au cours des années 80 de la rencontre entre les approches d'algorithmiques parallèles d'une part et d'intelligence artificielle d'autre part. Un SMA est défini comme étant un système artificiel composé d'une population d'agents autonomes interactifs qui coopèrent pour aboutir à un but commun [Kévin 04,+]. L'entité de base qui est l'agent est une entité réelle ou abstraite qui poursuit un objectif individuel et qui est capable d'agir sur elle-même et sur le monde environnant en général peuplé d'autres agents. Chaque entité dispose d'une représentation partielle de cet environnement et peut communiquer avec d'autres agents. Son comportement est la conséquence de ses observations, de son savoir et de ses interactions avec le monde et les autres agents. Elle est également capable d'apprendre et donc d'améliorer ses connaissances.

Un SMA est aussi un système informatique distribué composé d'entités informatiques dites « intelligentes » qui interagissent entre elles [Kévin 04, +].

Jacques Feber [Ferber 95] donne une excellente définition d'un système multi-agents. Il est composé des éléments suivants :

- Un environnement E , dans notre cas, c'est l'espace où évoluent les agents.
- Un ensemble d'objets O . Ces objets sont situés, c'est-à-dire que pour tout objet, il est possible, à un moment donné, d'associer une position dans E .
- Un ensemble A d'agents qui sont des objets particuliers ($A \subseteq O$), lesquels représentent les entités actives du système.
- Un ensemble de relations R qui unissent des objets (et donc des agents) entre eux.
- Un ensemble d'opérations Op permettant aux agents de A de percevoir, produire, consommer, transformer, et manipuler des objets de O . Cela correspond à la capacité des agents de percevoir leur environnement, de manger, etc.
- Des opérateurs chargés de représenter l'application de ces opérations et la réaction du monde à cette tentative de modification, que l'on appellera les lois de l'univers.

L'approche multi-agent facilite la conception de modèles pour les systèmes réels, d'entités vivantes, mais elle ne se limite pas à cet aspect. Les systèmes de vie artificielle en font également l'usage. Il convient de donner quelques notions sur le sujet de la vie artificielle où des nombreuses contributions sont effectuées dans la simulation écologique ; et vice-versa également. Le paragraphe suivant présente ce domaine de recherche.

1.3.2 Vie Artificielle

La vie artificielle, aussi connue sous la dénomination ALife - Artificial Life, est un domaine où l'on cherche à concevoir des systèmes capables de faire apparaître des processus similaires à ceux des systèmes vivants naturels [Campos 00].

C.Langton, l'un des pères de la vie artificielle l'a définie en 1989 comme « l'étude de systèmes construits par l'homme qui présente des comportements caractéristiques des systèmes vivants naturels. »

Des processus comme la reproduction, la mort, la compétition pour des ressources, l'apprentissage, le développement et bien d'autres encore peuvent faire partie d'un système ALife. Ce domaine de recherche complète l'approche traditionnelle de la biologie, où le mode de fonctionnement est l'analyse des êtres vivants, tout en essayant de synthétiser des comportements « vivants » sur ordinateur.

Nous pouvons trouver beaucoup d'applications conceptuellement différentes. Les premières réalisations mais non les moins intéressantes, sont les automates cellulaires comme le jeu de la vie de Conway (que nous ne détaillerons pas ici). Cet automate cellulaire est équivalent à une machine universelle de Turing. C'est-à-dire que théoriquement (la mise en pratique est autrement compliqué), le jeu de la vie de Conway peut résoudre les mêmes problèmes qu'un ordinateur.

La recherche sur la vie artificielle, malgré toute notre ignorance par rapport au processus complexe de la vie, peut éclairer des questions fondamentales dans plusieurs domaines de la science. Parmi ces domaines, citons d'abord la biologie et l'informatique, et en deuxième plan la physique, l'écologie, la sociologie, la psychologie, la philosophie [Charton 03],.....

1.3.3 Automates cellulaires

Les automates cellulaires sont des réseaux de cellules à une, deux, trois ou n dimensions. Dans un automate cellulaire, le temps et l'espace sont représentés de manière discrète. Chacune des cellules contient un automate à états finis qui calcule son état en fonction d'un voisinage [Batouche 02,a].

L'histoire des Automates cellulaires (AC) remonte aux années 1940 et au mathématicien Stanislas Ulam. Il s'est intéressé à l'évolution de constructions graphiques engendrées à partir de règles simples [Batouche 02,a] :

- La Base = espace à deux dimensions divisé en cellules.
- Une cellule pouvait avoir deux états: Allumée ou Éteinte.
- A partir d'une configuration donnée => déterminer la génération suivante en fonction des règles de voisinage.

Ce mécanisme permettait de générer des figures esthétiques et complexes. En parallèle, Von Neumann s'intéressait à la théorie des automates autorépliqueurs et travaillait à la conception d'une machine autorépliquatrice. S.Ulam qui a suggéré à V.N d'utiliser un espace cellulaire pour construire cette machine. Sur cette base, il conçut un AC de 200.000 cellules à 29 états contenant un copieur universel, une description de lui-même et une machine de Turing pour la supervision.

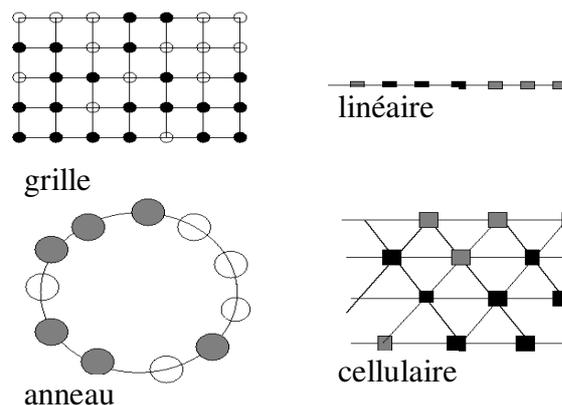


Figure 1.4 Exemples de topologies d'automates cellulaires.

Les AC sont sortis des laboratoires en 1970 avec le fameux Jeu de la Vie (Life Game) de John Horton Conway.

Donc, les automates cellulaires sont des structures abstraites qui permettent d'étudier des univers virtuels dont on maîtrise l'ensemble des lois. Ils contribuent à la connaissance de notre propre Univers : Ainsi, les systèmes physiques et biologiques complexes peuvent-ils reposer sur les mêmes classes universelles que les modèles mathématiques idéaux fournis par les automates cellulaires. La connaissance du comportement des automates cellulaires peut amener à des résultats plus généraux concernant le comportement des systèmes naturels complexes.

Les applications des AC sont nombreuses et diverses. Quelques exemples d'applications d'AC sont:

- Simulation du comportement d'un gaz. Un gaz =un ensemble de molécules dont le comportement est fonction de celui des molécules voisines.
- Les AC peuvent être utilisés comme alternative aux équations différentielles
- Conception d'ordinateurs massivement parallèles
- Simulation et étude du développement urbain.
- Simulation des processus de cristallisation.
- Simulation de la propagation des feux de forêt.

Dans un domaine plus quotidien, les AC peuvent être utilisé comme générateur graphique.

1.4 Quelques techniques inspirées de systèmes naturels

1.4.1 Algorithmes génétiques

Les algorithmes génétiques représentent une branche d'algorithmes puissants inspirée d'évolution par sélection naturelle. Ils ont été inventés dans les années 60. Depuis les années 80, ils sont devenus une technique très adoptée, on les trouve dans de nombreuses applications industrielles, où ils sont utilisés par exemple pour diagnostiquer des pannes complexes, pour traiter des images médicales...

Le principe de ce type d'algorithmes est comme suit [Bertelle 03,+] :

On génère initialement un grand nombre d'individus (ou solutions) et l'algorithme les fait évoluer en 3 phases :

- Reproduction
- Croisement
- Mutation

Le processus de sélection est basé sur une fonction d'évaluation (ou fonction objectif) qui correspond à une *performance* de l'individu. On en déduit une probabilité pour chaque individu de se reproduire ou de générer des clones (cette probabilité s'appelle *fitness*).

Finalement, ce processus contribue à produire une population ayant une meilleure adaptabilité (convergence vers l'optimum).

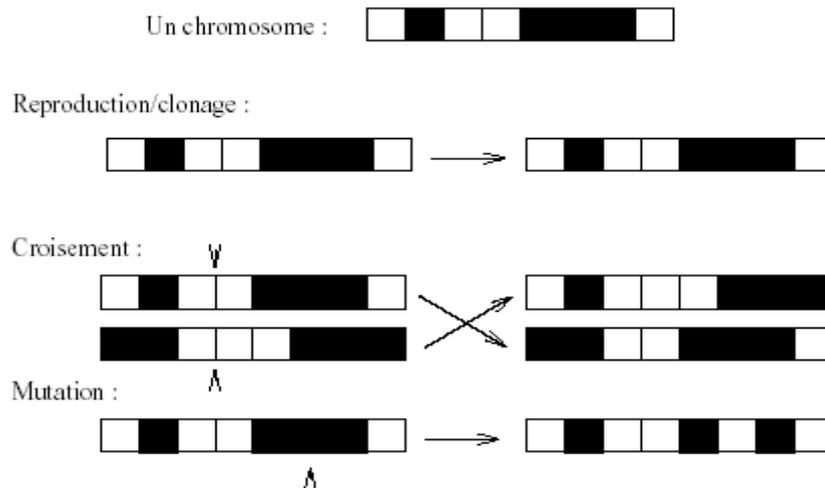


Figure 1.5 Exemple de reproduction, croisement, et mutation [Bertelle 03,+]

Par exemple, pour résoudre le fameux problème du voyageur de commerce on construit une population représentant un ensemble de parcours aléatoires. On sélectionne ensuite les meilleurs (les plus courts) que l'on croise entre eux pour obtenir une nouvelle population de parcours et ceci aussi longtemps qu'on le souhaite pour approcher le parcours optimum.

1.4.2 Réseaux neuronaux artificiels

Le champ des réseaux neuronaux va démarrer par la présentation en 1943 par W. MCCulloch et W. Pitts du neurone formel qui est une abstraction du neurone physiologique.

Le schéma des réseaux neuronaux s'inspire du mode de fonctionnement du système nerveux. Cette représentation a pour origine l'observation directe de la structure du cerveau et de ses ramifications dans l'ensemble des organismes vivants.

Un neurone peut être décrit comme un petit automate de décision. Il se compose de 2 parties (*voir figure 1.6*) ayant des fonctions distinctes [Batouche 02,b]:

- évaluation de la stimulation reçue
- évaluation de son état interne

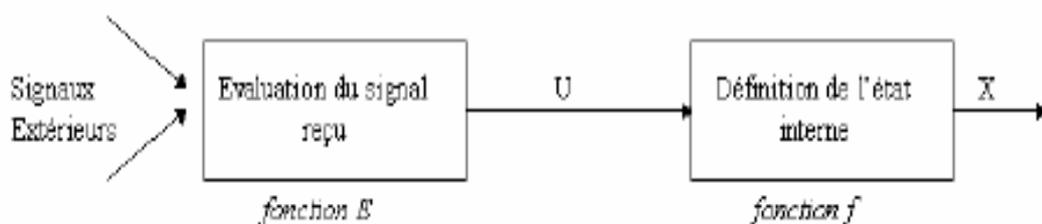


Figure 1.6 Le neurone formel, d'après [Batouche 02,b]

Un neurone est donc caractérisé par:

- son état (X)
- le niveau d'activation qu'il reçoit en entrée (U)
- sa fonction de transition (f)
- sa fonction d'entrée (une somme en général)

La topologie des réseaux de neurones peut être très variée. On peut concevoir plusieurs types de réseaux seulement en modifiant les règles de connexion. Par analogie à la biologie, on peut dire que les connexions représentent l'axone et les différentes dendrites qui relient le neurone aux autres neurones. La particularité de la connexion est d'être pondérée. On simule ainsi la relation entre les neurones, c'est à dire l'influence de l'activation (désactivation) d'un neurone sur un autre.

Dans la majorité des cas, les connexions servent à calculer l'entrée du neurone tandis que les fonctions d'activation détermineront la sortie.

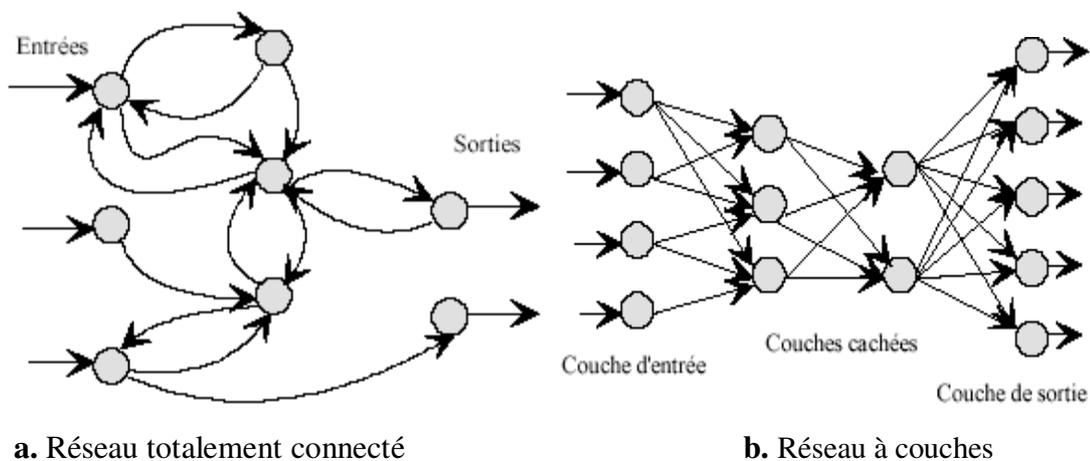


Figure 1.7 Exemple de topologies de réseaux de neurones d'après [Batouche 02,b]

Aujourd'hui, les réseaux neuronaux sont utilisés dans de nombreux domaines (entre autres, vie artificielle et intelligence artificielle) à cause de leur propriété en particulier, leur capacité d'apprentissage, et qu'ils soient des systèmes dynamiques.

1.4.3 Intelligence en essaim [Bonabeau 00,+]

Les recherches sur les comportements collectifs des insectes sociaux fournissent aux informaticiens des méthodes puissantes pour la conception d'algorithmes d'optimisation combinatoire et de routage et de contrôle distribué.

Chez les insectes sociaux, le comportement collectif qui émerge des comportements simples des individus est nommé intelligence en essaim. Ces phénomènes d'intelligence en essaim sont de plus en plus étudiés en informatique et en robotique, où des systèmes de contrôle centralisés gagnent à être remplacés par d'autres, plus autonomes et plus flexibles, fondés sur les interactions d'éléments simples [Jourdan 03].

L'histoire de l'intelligence en essaim remonte à l'étude du comportement de fourmis à la recherche de nourriture au départ de leur nid, par Goss, Deneubourg et leur équipe.

On distingue ici deux notions fondamentales :

a. Pheromone

En se déplaçant du nid à la source de nourriture et vice-versa (ce qui, dans un premier temps, se fait essentiellement d'une façon aléatoire), les fourmis déposent au passage sur le sol une substance odorante appelée phéromone, ce qui a pour effet de créer une piste chimique.

Les fourmis peuvent sentir ces phéromones qui ont un rôle de marqueur de chemin: quand les fourmis choisissent leur chemin, elles ont tendance à choisir la piste qui porte la plus forte concentration de phéromones (*voir figure 1.8*). Cela leur permet de retrouver le chemin vers leur nid lors du retour. D'autre part, les odeurs peuvent être utilisées par d'autres fourmis pour retrouver les sources de nourriture détectées par leurs consœurs.

Il a été démontré expérimentalement que ce comportement permet l'émergence des chemins les plus courts entre le nid et la nourriture, à condition que les pistes de phéromones soient utilisées par une colonie entière de fourmis.

Le système de fourmis (Ants System - AS) est une méthode d'optimisation basée sur ces observations [Jourdan 03]. Le système de fourmis a été employé avec succès sur des nombreux problèmes (voyageur de commerce, affectation quadratique, ...) mais les auteurs ont remarqué que l'AS n'a pas un comportement très exploratoire ce qui a conduit les auteurs à utiliser des hybridations du système de fourmis avec des recherches locales.

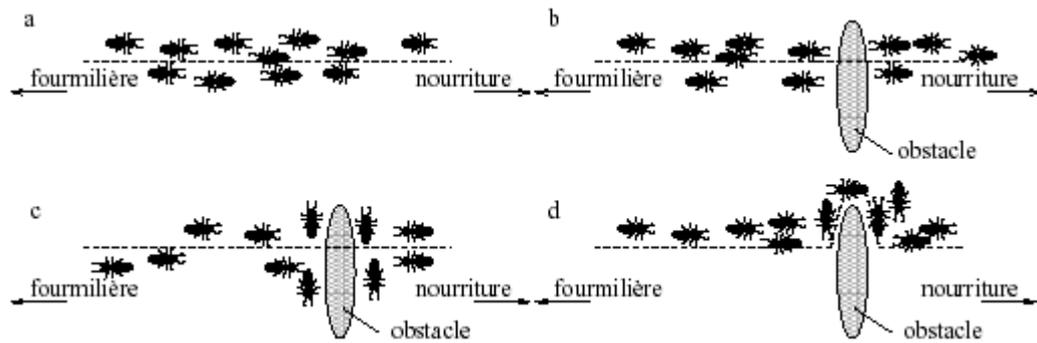


Figure 1.8 (a) Les fourmis suivent un chemin entre la fourmilière et la nourriture. (b) Un obstacle apparaît sur le chemin ; les fourmis choisissent entre prendre à droite et à gauche avec équiprobabilité. (c) La phéromone s'évapore sur le chemin le plus long. (d) Toutes les fourmis choisissent le chemin le plus court. [Jourdan 03]

b. Stigmergie

Le principe de *stigmergie* a été découvert par le biologiste Français *Grassé* dans les années 1950, par ses études sur le comportement d'insectes sociaux. Mais seulement actuellement qu'il devient une question majeure dans la conception des systèmes d'agents.

Le mot « *stigmergie* » est une combinaison des mots grecs “*stigma*” (signe remarquable) et “*ergon*.” (travail). Ce qui indique que quelques activités d'agents sont déclenchées par des signes externes qu'eux-mêmes peuvent être produits par l'activité de l'agent [Brueckner 00].

Dans un processus stigmergique, le comportement d'un individu est influencé par le résultat des actions passées des autres individus [Bourjot 01, +].

Deux formes générales de stigmergie ont été indiquées dans [Brueckner 00] :

- Stigmergie qualitative : qui implique un changement dans les caractéristiques physiques de l'environnement. La construction du nid chez les termites est un exemple de ce type de stigmergie.
- Stigmergie quantitative : ici, une marque est déposée dans l'environnement qui ne fait aucune contribution directe à la tâche qui est accomplie, mais elle a une influence sur la tâche subséquente. Un exemple de ce type stigmergie est le comportement de fourmis quand ils construisent collectivement la plus courte trajectoire d'une source de nourriture au nid.

Conclusion

Ce chapitre a présenté les concepts liés aux systèmes complexes. Il a abordé également les notions de vie artificielle et d'automates cellulaires.

Les systèmes multi-agents sont de plus en plus utilisés, non seulement pour la résolution de problèmes et le développement de logiciels mais aussi pour mieux comprendre les systèmes complexes par la simulation. L'intérêt de réaliser ce genre de simulations est de voir apparaître des comportements ou des structures qui n'étaient pas explicitement programmés.

Les phénomènes collectifs en systèmes naturels sont une source d'inspiration pour proposer des méthodes multi-agents de résolution de problèmes. Ils permettent l'élaboration de comportements individuels simples produisant collectivement des phénomènes complexes. Le chapitre suivant va aborder l'une de ces sources d'inspiration : écosystèmes.



Chapitre**02****Ecosystèmes****Introduction**

Le développement impressionnant des technologies de l'information, l'augmentation de la puissance de calcul et de mémorisation des informations sur ordinateur ont permis aux scientifiques de modéliser et de simuler des systèmes de plus en plus complexes à travers des logiciels informatiques. La conséquence de ce développement est l'augmentation de nos connaissances sur des phénomènes pour lesquels auparavant nous n'avions qu'une vague compréhension. Cela inclut, certes, les systèmes industriels, économiques, sociaux et physiques, mais également les systèmes biologiques et les écosystèmes.

Un écosystème est défini comme une unité de base formée par le milieu et les organismes qui y vivent. Ces organismes ont normalement des formes d'interaction complexes, qui rendent difficile la compréhension de certains phénomènes. Cette complexité impose l'emploi d'outils et de méthodes aptes à faciliter la compréhension de tels phénomènes. Le processus de modélisation et simulation est un de ces outils.

Le domaine de la simulation d'écosystèmes requiert la contribution de plusieurs disciplines, notamment certaines liées à l'informatique et à la biologie.

Dans ce chapitre, nous effectuerons dans une première partie, un survol sur les concepts d'écosystèmes. Dans une deuxième partie, quelques exemples de simulations seront exposés.

2.1 Des systèmes complexes : les écosystèmes

2.1.1 Notions de base

2.1.1.1 Ecologie

Le terme d'écologie vient du grec "*oikos*" : la maison et "*logos*" : la science. Il existe plusieurs définitions de l'écologie. La plus répandue est : « *étude des interactions entre les organismes vivants et leur milieu, et des organismes vivants entre eux dans les conditions naturelles* ». On peut également dire que l'écologie est l'étude des interactions qui déterminent la distribution et l'abondance des organismes, ou encore l'étude des écosystèmes [Gaudin 97].

Une autre définition (Ernst Haeckel, 1866) : science de l'habitat. D'abord études disjointes des espèces puis nécessité de prendre en compte les autres espèces et le milieu environnant avec lesquels chaque espèce interagit [Bertelle 03,+a].

L'écologie moderne a pour objet l'étude des êtres vivants dans leur milieu (y compris l'homme) et pour méthodes non seulement la méthode expérimentale mais aussi les méthodes de nombreuses autres sciences avec lesquelles elle travaille: paléontologie, géographie, économie, philosophie. Elle a une dimension pratique (art) qui en fait aussi une science appliquée qui a donc une dimension politique et sociale.

2.1.1.2 Qu'est-ce qu'un écosystème ?

Le mot *écosystème* est une contraction de deux mots: *écologie* et *système*. «éco» signifie habitat ou maison; «système» se rapporte à l'interdépendance de chaque élément vivant et non vivant d'une communauté naturelle. Les écosystèmes naturels fonctionnent à peu près de la même façon. On y retrouve les parties non vivantes, appelées abiotiques (p. ex. les roches, l'air, le sol et l'eau) et les parties biotiques (comme les plantes et les animaux). Les éléments vivants et non vivants d'un écosystème sont liés par la circulation de l'énergie et le cycle des éléments chimiques [Espaces 02].

Tansley (1935), a défini un écosystème : système d'interaction entre les populations de différentes espèces vivant dans un site et entre ces populations et le milieu physique. Les écosystèmes sont des systèmes ouverts. Ils sont ainsi constamment traversés par des flux de différentes natures [Bertelle 03,+a].

Un écosystème est constitué par l'association dynamique de deux composantes en constante interaction :

- un environnement physico-chimique, géologique, climatique ayant une dimension spatio-temporelle définie : le biotope,
- un ensemble d'êtres vivants caractéristiques : la biocénose.

L'écosystème est une unité fonctionnelle de base en écologie qui évolue en permanence de manière autonome au travers des flux d'énergie.

L'écosystème peut être divisé en trois catégories [Gaudin 97]:

- **les producteurs** : ce sont tous les végétaux, algues et plantes, et en particulier les algues microscopiques du phytoplancton. Ils utilisent la lumière solaire pour fabriquer des matières organiques par photosynthèse. Les plantes aquatiques consomment le gaz carbonique contenu dans l'eau, ainsi que des sels minéraux et des nutriments comme l'azote, le phosphore et la silice, et elles rejettent de l'oxygène.
- **les consommateurs** : ce sont des herbivores, des omnivores ou des carnivores. On y trouve notamment le zooplancton, les invertébrés et mollusques, les poissons, ainsi que certains oiseaux et petits mammifères. Certains se nourrissent de plantes (consommateurs primaires) ou d'autres animaux (consommateurs secondaires et tertiaires) pour obtenir l'énergie et les éléments nutritifs dont ils ont besoin. Ces animaux respirent en consommant l'oxygène produit par les plantes et en rejetant du gaz carbonique.
- **les décomposeurs** : ce sont les micro-organismes tels les bactéries ou les champignons, qui décomposent les matières et cellules mortes. Ces organismes utilisent également l'oxygène produit par les plantes et remettent dans le circuit des sels minéraux qui vont être à leur tour assimilés par la flore pour une nouvelle photosynthèse. Ils permettent aussi d'épurer les écosystèmes. Leur rôle est donc prépondérant et ils bouclent ainsi la chaîne alimentaire.

2.1.2 Propriétés caractéristiques

Les écosystèmes sont caractérisés par [Bertelle 03,+ ,a]:

- Structuration en une grande variété d'échelles d'espace-temps (d'une souche d'arbre en décomposition à l'organisation écologique planétaire).
- Flux d'énergie les traversant (rayonnement solaire, énergie de mise en mouvement de matière telle que l'eau ou l'air) permettant la mise en relation d'entités qui interagissent ...source du développement de structures organisées;
- Interactions multiples et non réductibles.

2.1.3 Exemples de relations biotiques dans les écosystèmes

2.1.3.1 Compétition [Gaudin 97]

Il y a *compétition* quand deux ou plusieurs organismes (ou populations) :

- utilisent des ressources communes présentes en quantité limitée,
- utilisent des ressources communes présentes en grande quantité,

mais dont la recherche implique une nuisance entre les organismes.

Par ressource, on entend des ressources :

- **trophiques** (eau, aliments, éléments minéraux),
- **d'ordre spatial** (site de ponte ou de nidification, abris),
- **d'ordre reproductif** (partenaires sexuels).

On distingue :

- compétition *intra-spécifique* : il y a alors compétition entre individus de la même espèce et,
- compétition *inter-spécifique* : compétition entre individus d'espèces différentes.

D'une manière ou d'une autre, la compétition a un effet aboutit le plus souvent à une diminution de la contribution des individus à la génération suivante. Ainsi, la concurrence entre prédateurs pour une proie qui se fait rare diminue le nombre de portées et la chance de survie des jeunes.

On admet habituellement que l'intensité de cet effet dépend de la densité. Ainsi, quand la compétition devient trop forte (dépassement d'un certain seuil, la *capacité biotique*, notée K), il y a régulation de la densité (diminution du nombre des individus). Bien entendu, lorsqu'on est en dessous de ce seuil, il y a augmentation de la population. Les effets de la surdensité sur la qualité et les performances des individus sont multiples.

On peut noter :

- un affaiblissement physiologique,
- un ralentissement de la croissance individuelle,
- une diminution de la longévité et de la fécondité,
- des modifications du comportement.

2.1.3.2 Symbiose (ou mutualisme) [Gaudin 97]

On parle de *symbiose* quand il y a une association profitable entre deux espèces. On réserve généralement le terme de mutualisme aux associations qui ne sont pas obligatoires et celui de symbiose aux associations obligatoires. Dans le premier cas, les deux espèces peuvent vivre l'une sans l'autre, dans le second, c'est impossible.

On peut donner un exemple forestier de symbiose : Lors de la reproduction chez les Angiospermes, on peut avoir diverses formes de symbiose. Par exemple la *pollinisation*, qui est très souvent assurée par les insectes. Il y a dans ce cas symbiose car la plante fournit des substances nutritives à l'insecte (nectar) et celui-ci allant de plante en plante transporte le pollen pour la plante. On a ainsi une pollinisation croisée très efficace. On constate même des phénomènes de co-évolution et de spécialisation de plantes et d'insectes (beaucoup d'insectes ne pollinisent qu'une espèce de plante dans une période de l'année donnée).

2.1.3.3 Parasitisme [Gaudin 97]

Les parasites vivent au moins une partie de leur vie aux dépens d'un organisme vivant (hôte). Toutefois leur présence ne signifie pas toujours la mort de l'hôte. En effet, un parasite peut infester un hôte et s'y développer sans qu'il y ait nécessairement mort de l'hôte.

Les parasites se caractérisent par une extrême *spécialisation* dans l'exploitation des ressources. Beaucoup d'espèces de parasites dépendent en effet d'une seule espèce-hôte, contrairement à la majorité des prédateurs et des herbivores non parasites qui apparaissent comme des consommateurs généralistes. La spécialisation n'implique pas seulement, lorsqu'elle est extrême, une dépendance totale du parasite vis-à-vis de l'hôte. Elle signifie aussi que l'espèce consommatrice a développé des *adaptations* morphologiques, biochimiques, physiologiques, éthologiques et démographiques qui interviennent de manière essentielle dans la dynamique et l'évolution de la relation parasite-hôte.

Un parasite se développe le plus souvent quand l'hôte est en état de faiblesse. Par exemple, les insectes « nuisibles » pour la forêt se développent à la suite de stress divers

(tempêtes, sécheresse) ou bien lorsque les essences ne sont pas bien adaptées à la station (altitude, sol, pluviosité...).

Cet état de faiblesse peut également résulter d'une surdensité de l'hôte (dépassement de la capacité biotique du milieu). La population renferme alors une proportion de plus en plus importante d'individus physiologiquement déficients.

2.1.3.4 Prédation [Gaudin 97]

On appelle prédateur tout organisme libre qui se nourrit aux dépens d'un autre. Cette définition permet de considérer les animaux herbivores comme des prédateurs de végétaux. Elle élimine également les parasites qui vivent au moins une partie de leur vie aux dépens d'un hôte (que ce soit en surface ou à l'intérieur de l'hôte) et ne sont donc pas libres.

2.1.4 Coevolution

Le terme de coevolution est attribué à l'étude des biologistes Paul Ehrlich et Peter Raven sur les papillons et les plantes, qui ont publié un papier sur la coevolution : "Butterflies and plants: A study in coevolution" en 1965. La coevolution peut être définie comme l'évolution d'une espèce qui affecte l'évolution des espèces avec lesquelles elle interagit, par exemple par des relations de type chaîne alimentaire, prédateur proies, symbiose, parasitisme ...etc [coevolution].

Des exemples incluent la pollinisation d'orchidées par les colibris. Ces espèces coevolvent puisque les oiseaux sont dépendant sur les fleurs pour nectar et les fleurs sont dépendant sur les oiseaux pour propager leur pollen donc ils peuvent reproduire. La coevolution est aussi utilisée pour faire référence aux interactions évolutives entre molécules dans le champ d'évolution moléculaire (par exemple, entre hormones et récepteurs).

Ces observations ont été exploitées dans ces dernières années, pour créer une sous-classe d'algorithmes évolutifs artificiels plus robustes appelés : *algorithmes coevolutionnaires*. Dans ce type d'algorithmes les espèces peuvent *coevoluer* par compétition ou bien par coopération [Zaritsky 03] [Kirley 02].

2.1.4.1 Coevolution compétitive

Les relations prédateur-proie sont les exemples de coevolution les plus célèbres. Il y a une forte pression évolutionnaire des proies pour défendre mieux (par exemple, en courant plus vite, etc.) en réponse que les générations futures de prédateurs doivent développer de mieux capacités pour attaquer (e.g, meilleure vue, courir plus vite, etc.). Dans telles relations, le succès pour un coté représente l'échec à l'autre côté. Ce processus de coevolution peut résulter en amélioration considérable de la proie et le prédateur.

Les algorithmes coevolutionnaires compétitifs sont inspirés des relations de prédateur-proie. L'aptitude (*fitness*) d'un individu d'une espèce est basé sur la compétition directe avec les individus d'une autre espèce. Donc, l'augmentation d'aptitude d'une espèce, implique une diminution de l'aptitude de l'autre espèce. Cette pression évolutionnaire a tendance à produire de nouvelles stratégies dans les populations impliquées comme maintenir leurs chances de survie.

2.1.4.1 Coevolution coopérative

Le terme *coevolution coopérative* a été inventé par Potter et De Jong [Potter 00,+]. Leur architecture exige que plusieurs individus travaillent ensemble pour résoudre un problème donné.

Les algorithmes coevolutionnaires coopératifs impliquent plusieurs espèces évoluant indépendamment, et qui forment des structures complexes, bien assorti pour résoudre ensemble un problème. L'idée est d'utiliser plusieurs populations maintenus indépendamment (espèces), chacun s'est spécialisé à accomplir une niche du problème, avec l'aptitude d'un individu selon sa capacité de collaborer avec les individus d'autre espèce pour construire une solution globale (aucun individu d'une seule espèce comprend une solution au problème à condition que toutes les espèces doivent coopérer).

2.1.5 Etapes d'évolution d'écosystèmes

Les étapes d'évolution d'écosystèmes sont comme le suivant [Bertelle 03,+a] :

- ↳ état juvénile : croissances importantes mais instables ;
- ↳ état de maturité: stabilisation par complexification des espèces permettant l'adaptabilité et la mise en place de structures de survie ;

- ↳ état de vieillissement: prédominance de certaines espèces aux dépens des autres et contribuant à la simplification globale ;
- ↳ Plus des perturbations générant des stress et remettant en cause l'organisation en provoquant la disparition d'espèces et favorisant le développement rapide d'autres espèces ...
- ↳ rajeunissement.

2.2 Exemples de simulations et de plates formes

On nomme simulation la démarche scientifique qui consiste à réaliser une reproduction artificielle, appelée *modèle*, d'un phénomène réel que l'on désire étudier, à observer le comportement de cette reproduction lorsqu'on en fait varier certains paramètres, et à en induire ce qui se passerait dans la réalité sous l'influence de variations analogues. La démarche de simulation passe donc par trois étapes distinctes: l'étape de modélisation, qui consiste à construire le modèle du phénomène à étudier, l'étape d'expérimentation, qui consiste à soumettre ce modèle à un certain type de variations, et l'étape de validation, qui consiste à confronter les données expérimentales obtenues avec le modèle à la réalité.

La nouvelle technologie des agents et systèmes multi-agents connaît un vif succès chez les industriels pour qui elle promet des outils de conception et d'implémentation flexibles et adaptatifs. Afin de réaliser une opération alisation plus accessible des systèmes multi-agents, des travaux ont tenté de réutiliser des architectures et des langages existants pour construire des environnements de développement de ces systèmes. Les environnements de développement ou les plates-formes multi-agents sont nécessaires pour renforcer le succès de la technologie multi-agents. Les plates-formes multi-agents permettent aux développeurs de concevoir et réaliser leurs applications sans perdre de temps à réaliser des fonctions de base pour la création et l'interaction entre agents et éliminent, dans la plupart des cas, la nécessité d'être familier avec les différents concepts théoriques des systèmes multi-agents.

Il existe un nombre important d'environnements de développement des applications orientées agents : il y a aussi bien des produits commerciaux que des logiciels dans le domaine public.

Dans cette section, on va présenter quelques plates formes et exemples de simulations multi-agents de systèmes complexes. Il faut noter que ces plates-formes ne sont pas unique, et qu'il y a aussi d'autres plates-formes qui ont été utilisées avec beaucoup de succès pour bâtir diverses applications.

2.2.1 SIMDELTA

Le simulateur SimDelta a été utilisé pour synthétiser les connaissances d'un ensemble de spécialistes (halieutes, écologistes, pédologues, anthropologues, etc.) ayant travaillé pendant plusieurs années sur l'étude du système de pêche du delta central du Niger au Mali. Il s'agissait de pouvoir modéliser des informations tant quantitatives (concernant l'évolution des crues du Niger par exemples) que qualitatives (telles que les techniques de pêche).

Ce simulateur, réalisé par F.Bousquet et C.Cambier (Cambier et al 1992), permet de simuler en même temps la dynamique de population des poissons, en tenant compte des nombreux facteurs biologiques et topologiques qui peuvent affecter son évolution, et la prise de décision des pêcheurs. Les agents sont ici les bancs de poissons et les pêcheurs. La technique employée met en présence trois types d'agents (figure 2.1): les biotopes qui représentent des portions d'environnements, les poissons qui ont un comportement plutôt réactif et les pêcheurs qui se comportent comme des agents cognitifs [Cambier 94].

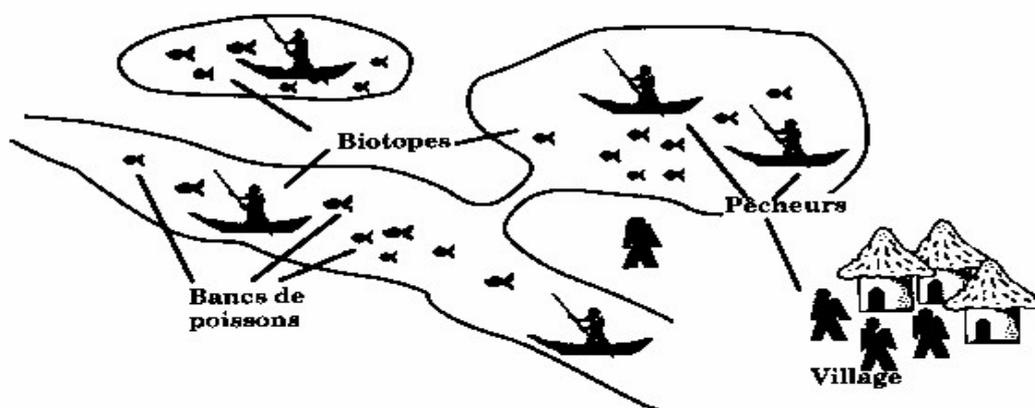


Figure 2.1 *Le monde artificiel de SimDelta, d'après [Ferber 95]*

L'environnement est constitué de biotopes dont les connexions peuvent être modifiées dynamiquement lorsque le niveau de l'eau change. Pour chaque biotope, une fonction de ressource indique la quantité de la nourriture disponible pour une population de poissons en fonction du temps. Les bancs de poissons sont des agents qui représentent des ensembles de poissons et dont les paramètres (taille et nombre d'œufs, processus de migration, régime, etc.) caractérisent leur stratégie d'adaptation.

Chaque pêcheur est représenté sous la forme d'un agent cognitif et son comportement est décrit par un système à base de connaissances composé d'une base de données qui contient ses croyances et sa mémoire de pêcheur et d'un système de règles qui représente sa stratégie cognitive pour exploiter les biotopes.

Deux séries d'expérimentations ont été réalisées. La première a porté sur l'étude de la dynamique de population des poissons en fonction d'un effort de pêche de plus en plus grand. La simulation de la dynamique de ces poissons est fondée sur le comportement des poissons d'eau douce sahéliens et, en particulier, sur les connaissances des biologistes concernant la reproduction, la croissance, la migration et la mortalité de ces poissons. Cette simulation a permis de reproduire la courbe en trois phases caractéristiques de l'évolution d'un système de pêche limité et surtout de faire apparaître des variations portant sur la composition spécifique et la taille des poissons.

La seconde a eu pour objet de modéliser des pêcheurs qui prennent des décisions et agissent sur la réserve renouvelable. Cette modélisation a montré l'importance du mécanisme de décision sur la dynamique des poissons.

2.2.2 MANTA

Le système MANTA est caractéristique d'une approche réactive dans laquelle l'informatique et les sciences de la vie s'apportent mutuellement de leurs apports. En effet ce projet, qui porte sur la modélisation par systèmes multi agents d'une colonie de fourmis *Ectatomma ruidum* au cours de son évolution, est né de la rencontre de deux domaines de recherches : l'éthologie et les systèmes multi agents. Les expérimentations ont porté notamment sur sa sociogenèse, c'est-à-dire sur la constitution d'une colonie mature à partir d'une (ou plusieurs) reine(s), sur l'adaptation d'une colonie à son environnement et, en particulier, sur la division des tâches (polyéthisme) et la spécialisation des ouvrières, et en fin sur la polygynie. Il s'agissait en particulier de tester une hypothèse éthologique concernant l'aspect distribué de la prise de décision dans une colonie de fourmis. Le comportement des individus suffit-il à expliquer la génération et la stabilité des formes sociales observées (division du travail, polyéthisme d'âge, dynamique de la fondation de la société, etc.)? Peut-on montrer qu'une société d'agents peut survivre sans faire appel à un système de contrôle centralisé ni à une quelconque organisation hiérarchique? C'est pour répondre à ce type de question que le système MANTA fut réalisé. Il mit en évidence que l'organisation de travail dans une colonie de fourmis peut résulter d'un ensemble d'interactions et de contrôles locaux, sans qu'il soit nécessaire de faire intervenir une quelconque entité centrale de régulation, les performances

adaptatives de la société étant le résultat des comportements, forcément élémentaires, de chacun de ses membres. Dans un système de simulation multi agent tel que MANTA, il s'agit de modéliser les différentes entités qui interviennent dans un nid de fourmis sous forme d'agents réactifs sans faire appel à une organisation prédéfinie ni à des spécialisation préétablies.

Une colonie de fourmis peut être divisée en trois groupes d'agents :

- Les "assistés": qui regroupent tous les membres du couvain, c'est-à-dire les œufs, les larves à leurs différents stades de maturation et les cocons.
- Les "assistants": qui comprennent la reine, les ouvrières et les mâles.
- Les autres: où sont regroupés les aliments, sous quelque forme qu'ils soient, les déchets et déjections en tout genre et les cadavres de fourmis.

Le lecteur intéressé pourra se référer à [Drogoul 93] qui présente la problématique, le système et les expériences avec toute la précision nécessaire. La figure (2.2) montre l'écran servant à la définition du nid et à l'observation directe de la fourmilière.

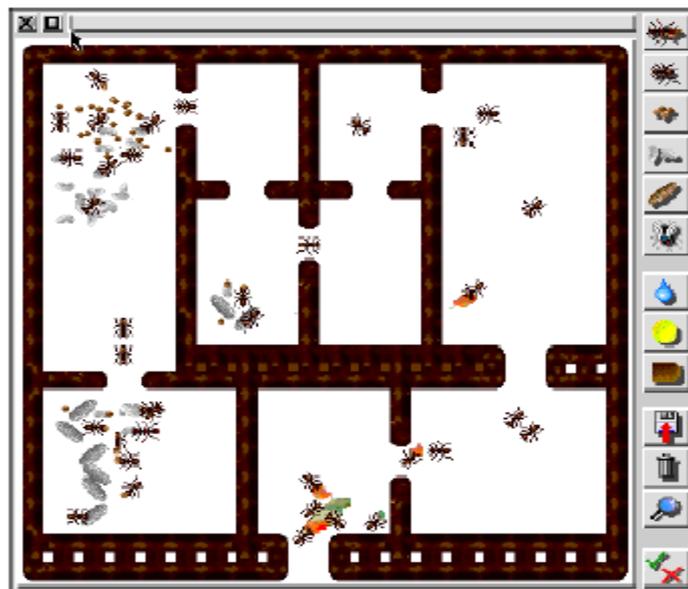


Figure 2.2 Un nid de fourmis tel qu'il est visualisé par MANTA, d'après [Drogoul 93]

2.2.3 PENG1: Modélisation d'un écosystème abstrait

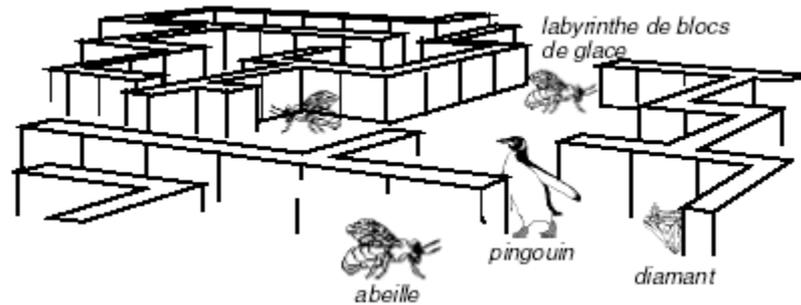


Figure 2.3. Une représentation en perspective de PENG1, du point de vue du pingouin, d'après [Drogoul 93]

PENG1 est dérivé d'un jeu vidéo nommé PENGO. Ce jeu se déroule dans un micro-monde peuplé d'un pingouin, d'abeilles, de diamants et de blocs de glace (voir Figure). Le pingouin, contrôlé par le joueur, se déplace dans un labyrinthe dont les parois sont faites des blocs de glace qui peuvent être poussés afin de libérer un chemin ou d'écraser les individus qui se trouvent de l'autre côté. Le but du jeu est de faire récolter par le pingouin un certain nombre de diamants, tout en lui évitant d'être piqué par les abeilles, ou écrasé par un bloc de glace poussé par l'une d'entre elles. Celles-ci ont un déplacement semi-aléatoire, ce qui signifie qu'elles suivent le pingouin quand elles en sont assez proches, mais qu'elles peuvent tout à coup décider de l'abandonner, la probabilité pour qu'elles le fassent étant fixée une fois pour toutes au commencement de la partie. Un bloc de glace ne peut être poussé que dans les directions où il n'est pas immédiatement adjacent à un autre bloc. Après avoir été poussé, il glisse dans la direction choisie jusqu'à ce qu'il heurte un autre bloc de glace (ou les bords de l'écran). Tout animal, abeille ou pingouin, qui croise son chemin est irrémédiablement écrasé. La stratégie que doit utiliser le joueur s'appuie donc sur deux points:

- Il doit être capable de faire réagir le pingouin à toute attaque d'abeilles en le faisant fuir ou contre-attaquer s'il en a la possibilité. Il doit ainsi faire attention à ne pas le laisser se faire enfermer par un essaim d'abeilles, dont il a peu de chance de sortir vivant.
- Il doit en même temps veiller à lui fournir un chemin correct dans le labyrinthe de manière à ce qu'il ramasse les diamants le plus vite possible. Dans la version originale du jeu, le temps imparti est en effet strictement limité.

Les types d'agents présents dans PENG1 sont au nombre de quatre. Les diamants, les cubes de glace, les abeilles et enfin le pingouin. De manière générale, voici comment a été

conçu le système. Le pingouin est guidé dans sa recherche de satisfaction par un gradient émis par les diamants et propagé dans l'environnement. Il émet lui même un gradient, de rayon plus limité, qui a pour effet d'attirer les abeilles quand elles le sentent (leur recherche de satisfaction est entachée d'une erreur aléatoire). A leur tour, les abeilles propagent un stimulus qui déclenche chez le pingouin son comportement de fuite. Enfin, les cubes de glace ne sont sensibles qu'au fait d'être poussés par les abeilles ou le pingouin, ce qui déclenche leur comportement de fuite. Lors de cette fuite, ils propagent un stimulus activant le comportement de fuite du pingouin. Selon le paradigme de résolution collective de problèmes, il s'avère que les deux buts informels que doit atteindre le pingouin sont, d'une part de supprimer tous les diamants de l'environnement, d'autre part de rester en vie le plus longtemps possible. Et ces buts à atteindre sont bien sûr transcrits dans l'environnement sous la forme d'une entropie globale (le gradient créé par les diamants) et de plusieurs températures locales (celles générées par les abeilles et les cubes de glace en translation), dont la perception par le pingouin déclenche ses comportements téléonomiques respectifs de satisfaction et de fuite, qui doivent avoir des propriétés néguentropiques (disparition ou évitement des gradients mentionnés).

2.2.4 Swarm

Le projet *SWARM* [Cardon 03+] [Baptiste 03] a débuté en 1994 par Chris Langton, puis son développement a été repris par Dave Hiebeler, de l'institut de Santa F. La plate-forme Swarm est devenue, en l'espace de quelques années, la référence en termes d'environnement de développement de simulations multi-agents. Elle propose en effet un noyau applicatif éprouvé et très bien conçu, consacré à la modélisation, à la simulation et à l'exploration de systèmes complexes, des bibliothèques (de visualisation, de calcul) complètes, et bénéficie d'une maintenance technique régulière et du support d'une communauté extrêmement active de modélisateurs. La démarche est comme suit :

- Création du monde virtuel : environnement artificiel spatial et temporel où évoluent des entités ;
- Création d'agents d'observation qui sondent le monde virtuel précédent ;
- Fonctionnement de l'ensemble (monde virtuel + observateurs) dans une simulation discrétisée en temps avec des horloges synchronisées.

Possibilités de hiérarchies et de structures récursives. De nombreuses applications développées, notamment à caractère écologique.

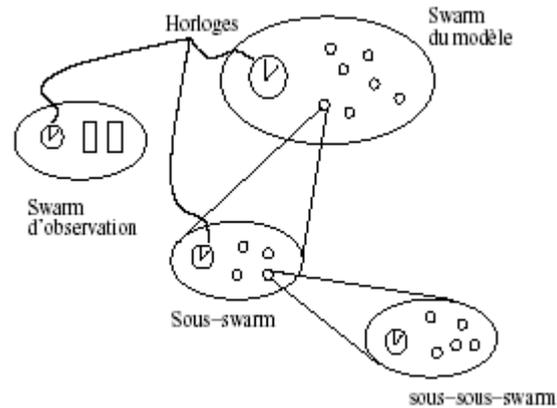


Figure 2.4. *Swarm - structures récursives* [Cardon 03,+]

2.2.5 Cormas

CORMAS [Bousquet 98,+] [Baptiste 03] est un environnement développé au CIRAD, pour la programmation de modèles multi-agents. CORMAS, spécialisé dans la gestion de ressources renouvelables, offre un cadre de développement de modèles de simulation, des modes de coordination entre des individus et des groupes qui exploitent ces ressources. Le cadre de CORMAS se structure en trois modules :

- un module permettant de définir les entités du système (agents) et leurs interactions,
- un module permettant de contrôler la dynamique de la totalité du modèle,
- un module permettant d'observer la simulation depuis différents points de vue (affichages de différentes propriétés du modèle en cours de simulation).

La plupart des éléments composants ces trois modules sont prédéfinis et disponibles dans des classes SmallTalk. L'interface de développement de CORMAS est l'environnement de développement intégré de VisualWorks.

2.2.6 Jardins des Hasards

Le Jardin des Hasards [Hutzler 95] [Jardin 04] trouve son origine dans le projet artistique imaginé par le peintre Bernard Gortais, dont le but est de proposer aux spectateurs une représentation graphique poétique de l'ambiance climatique d'un lieu réel, par l'intermédiaire de la présentation d'un monde virtuel où vivent et coexistent, se déplacent et interagissent des entités de formes et de couleur différentes.

Les "Jardins des hasards" sont des jardins virtuels dont l'évolution est fonction de données numériques reçues par modem en temps réel. Ils sont composés de plusieurs familles de formes qui naissent, grandissent, meurent et interagissent entre elles suivant des comportements inspirés de la vie. Ils constituent des écosystèmes de vie artificielle.

Le Jardin des Hasards est un espace de création en même temps qu'un espace de vie artificielle, dans lequel chaque spectateur est appelé à participer à la définition d'un jardin particulier. Le Jardin du Caplan est l'un de ces jardins.

Dans les Jardins des Hasards, des données numériques prélevées de manière continue dans le monde réel constituent la "météorologie" d'un jardin virtuel de formes colorées. Chacune de ces formes va bénéficier de manière individuelle de la présence ou de l'absence de certaines données pour se déplacer et interagir avec d'autres formes, croître, se reproduire si les conditions sont favorables ou disparaître lorsque les ressources s'épuisent.

Chaque jardin est ainsi un écosystème virtuel imaginaire dont l'évolution est soumise à l'influence d'un système complexe réel. Cet écosystème est peuplé de formes graphiques qui sont autant de métaphores d'organismes vivants, essayant de survivre dans un environnement dont le perpétuel renouvellement traduit les évolutions des données décrivant le système complexe réel. De ce point de vue, un jardin est une fenêtre virtuelle offrant un nouveau regard sur l'évolution d'un système complexe réel.

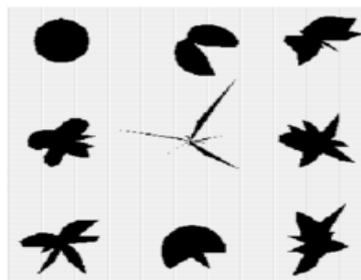


Figure 2.5 Exemples de formes générées [Hutzler 95,+]

Le système peut être décrit de manière générique comme un ensemble d'entités indépendantes (des agents) qui évoluent dans un espace continu (l'environnement). Dans le "Jardin des hasards", les formes de chaque famille ont un comportement autonome dans l'espace de l'image, sous l'influence des données météo du "Caplan".

A chaque forme sont associés des paramètres qui décrivent les ressources dont elle dispose, ainsi que des signaux qu'elle peut propager autour d'elle. Par ailleurs, l'environnement est décrit par des paramètres locaux indiquant les ressources disponibles à un endroit donné ainsi que par les données météorologiques reçues par le système.

Les formes interagissent entre elles en étant mutuellement sensibles aux signaux qu'elles émettent. Les formes réagissent également aux conditions environnementales qu'elles rencontrent en se déplaçant dans l'image.

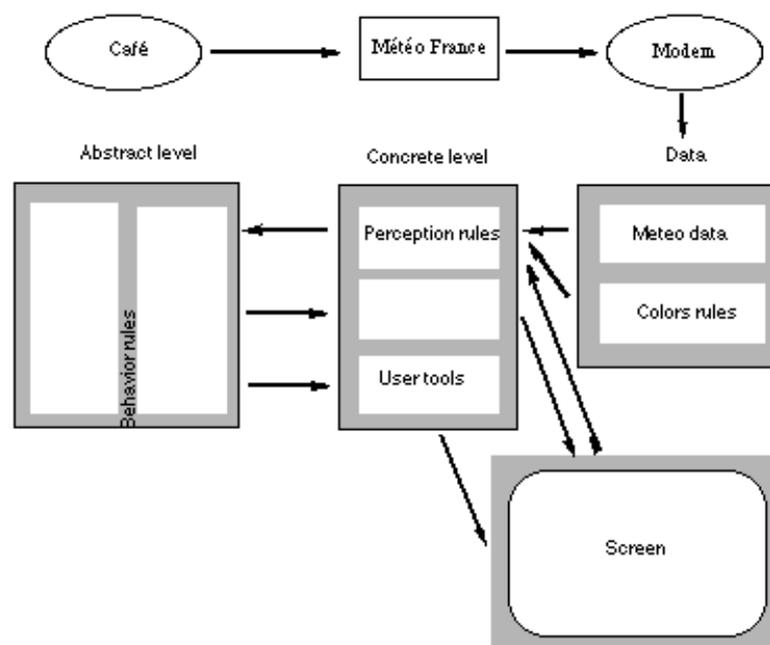


Figure 2.6 . Structure du système ``Jardin des Hasards'' [Jardin 04]

2.2.7 ECLPSS

Eclpss (Ecological Component Library for Parallel Spatial Simulation) [Baptiste 03] est un environnement basé sur le langage Java destiné aux écologues. Cet outil permet d'élaborer facilement des simulations d'écosystèmes dans de nombreuses échelles de temps et d'espace.

Un modèle Eclpss est composé de trois entités : un ensemble de variables, un ensemble de composants qui modifient ces états et un ensemble de simulateurs pour ces composants.

Les composants Eclpss sont développés en code java et stockés dans une bibliothèque pour réutilisation. L'intérêt principal de cet environnement est sa capacité à effectuer des simulations sur des machines parallèles.

Conclusion

Ce chapitre a présenté les concepts liés aux écosystèmes. Il a abordé également le concept de coevolution où l'évolution d'une espèce affecte l'évolution des espèces avec lesquelles elle interagit.

Le domaine de la simulation d'écosystèmes, en particulier, requiert la contribution de plusieurs disciplines, notamment certaines liées à l'informatique et à la biologie. Dans ce chapitre nous avons présenté quelques exemples de ces simulations.



Chapitre

03**Généralités sur le traitement d'images****Introduction**

Le traitement d'images est un domaine très vaste qui a connu, et qui connaît encore, un développement important depuis quelques dizaines d'années.

On désigne par *traitement d'images numériques* l'ensemble des techniques permettant de modifier une image numérique afin d'améliorer ou d'en extraire des informations.

De ce fait, le traitement d'images est l'ensemble des méthodes et techniques opérant sur celles-ci, dans le but de rendre cette opération possible, plus simple, plus efficace et plus agréable, d'améliorer l'aspect visuel de l'image et d'en extraire des informations jugées pertinentes.

Dans ce chapitre, nous abordons les notions de base nécessaires à la compréhension des techniques de traitement d'images. Ensuite, nous allons donner un aperçu sur les différents techniques connues dans ce domaine.

3.1 Définition de l' image

Une image est plutôt difficile à décrire d'une façon générale. Une image est une représentation du monde. En traitement d'image, la majorité du temps, on considère qu'il s'agit d'une fonction mathématique de $R \times R$ dans R où le couplet d'entrée est considéré comme une position spatiale, le singleton de sortie comme l'intensité (couleur ou niveaux de gris) du phénomène physique. Il arrive cependant que l'image soit dite "3D" donc la fonction est de $R \times R \times R$ dans R . Les images couleurs peuvent être représentées soit par trois images représentant les trois couleurs fondamentales, soit par une image de $R \times R$ dans $R \times R \times R$ [wikipedia 04].

L'image numérique est l'image dont la surface est divisée en éléments de tailles fixes appelés cellules ou pixels, ayant chacun comme caractéristique un niveau de gris ou de couleurs prélevé à l'emplacement correspondant dans l'image réelle, ou calculé à partir d'une description interne de la scène à représenter[chakib 99].

3.2 Acquisition d'une image

L'acquisition d'images constitue un des maillons essentiels de toute chaîne de conception et de production d'images. Pour pouvoir manipuler une image sur un système informatique, il est avant tout nécessaire de lui faire subir une transformation qui la rendra lisible et manipulable par ce système. Le passage de cet objet externe (l'image d'origine) à sa représentation interne (dans l'unité de traitement) se fait grâce à une procédure de numérisation. Ces systèmes de saisie, dénommés optiques, peuvent être classés en deux catégories principales [chakib 99] [Bénallal 03]:

- les caméras numériques ,
- et les scanners.

A ce niveau, notons que le principe utilisé par le scanner est de plus en plus adapté aux domaines professionnels utilisant le traitement de l'image comme la télédétection, les arts graphiques, la médecine, etc. Le développement technologique a permis l'apparition de nouveaux périphériques d'acquisition appelés cartes d'acquisition, qui fonctionnent à l'instar des caméras vidéo, grâce à un capteur *C.C.D.* (Charge Coupled Device). La carte d'acquisition reçoit les images de la camera, de la T.V. ou du scanner afin de les convertir en

informations binaires qui seront stockées dans un fichier[chakib 99]. L'une des caractéristiques intéressantes de ces capteurs est la taille du plus petit élément (*pixel*), mais aussi l'intercorrélation de deux éléments voisins : plus cette intercorrélation est faible, meilleure est l'image[wikipedia 04].

3.3 Caractéristiques d'une image numérique

L'image est un ensemble structuré d'informations caractérisé par les paramètres suivants:

3.3.1 Dimension

C'est la taille de l'image. Cette dernière se présente sous forme de matrice dont les éléments sont des valeurs numériques représentatives des intensités lumineuses (pixels). Le nombre de lignes de cette matrice multiplié par le nombre de colonnes nous donne le nombre total de pixels dans une image[chakib 99].

3.3.2 Résolution

C'est la clarté ou la finesse de détails atteinte par un moniteur ou une imprimante dans la production d'images. Sur les moniteurs d'ordinateurs, la résolution est exprimée en nombre de pixels par unité de mesure (pouce ou centimètre). On utilise aussi le mot résolution pour désigner le nombre total de pixels affichables horizontalement ou verticalement sur un moniteur; plus grand est ce nombre, meilleure est la résolution [chakib 99].

3.3.3 Bruit

Un bruit (parasite) dans une image est considéré comme un phénomène de brusque variation de l'intensité d'un pixel par rapport à ses voisins, il provient de l'éclairage des dispositifs optiques et électroniques du capteur [chakib 99].

3.3.4 Histogramme

L'histogramme des niveaux de gris ou des couleurs d'une image est une fonction qui donne la fréquence d'apparition de chaque niveau de gris (couleur) dans l'image. Il permet de donner un grand nombre d'information sur la distribution des niveaux de gris (couleur) et de voir entre quelles bornes est répartie la majorité des niveaux de gris (couleur) dans le cas d'une image trop claire ou d'une image trop foncée.

Il peut être utilisé pour améliorer la qualité d'une image (Rehaussement d'image) en introduisant quelques modifications, pour pouvoir extraire les informations utiles de celle-ci.

Pour diminuer l'erreur de quantification, pour comparer deux images obtenues sous des éclairages différents, ou encore pour mesurer certaines propriétés sur une image, on modifie souvent l'histogramme correspondant [chakib 99] [Pillou 04].

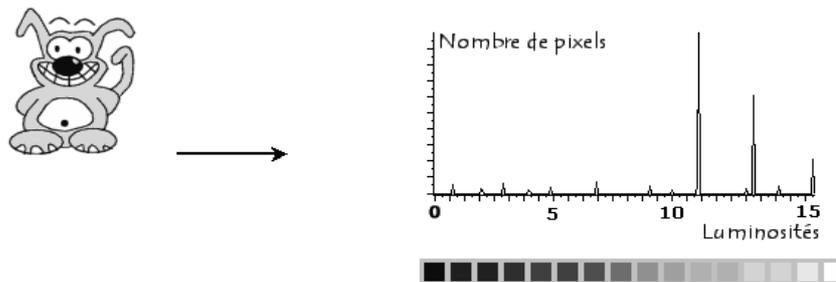


Figure 3.1 Histogramme et palette associés à une image [Pillou 04]

3.3.5 Luminance

C'est le degré de luminosité des points de l'image. Elle est définie aussi comme étant le quotient de l'intensité lumineuse d'une surface par l'aire apparente de cette surface, pour un observateur lointain, le mot luminance est substitué au mot brillance, qui correspond à l'éclat d'un objet. Une bonne luminance se caractérise par [chakib 99] [Nicolas 01] :

- Des images lumineuses (brillantes);
- Un bon contraste : il faut éviter les images où la gamme de contraste tend vers le blanc ou le noir; ces images entraînent des pertes de détails dans les zones sombres ou lumineuses.
- L'absence de parasites.

3.3.6 Contraste

C'est l'opposition marquée entre deux régions d'une image, plus précisément entre les régions sombres et les régions claires de cette image. Le contraste est défini en fonction des luminances de deux zones d'images. Si L_1 et L_2 sont les degrés de luminosité respectivement de deux zones voisines A_1 et A_2 d'une image, le contraste C est défini par le rapport [chakib 99] :

$$C = \frac{L_1 - L_2}{L_1 + L_2}$$

3.3.7 Images à niveaux de gris

Le niveau de gris est la valeur de l'intensité lumineuse en un point. La couleur du pixel peut prendre des valeurs allant du noir au blanc en passant par un nombre fini de niveaux intermédiaires. Donc pour représenter les images à niveaux de gris, on peut attribuer à chaque pixel de l'image une valeur correspondant à la quantité de lumière renvoyée. Cette valeur peut être comprise par exemple entre 0 et 255. Chaque pixel n'est donc plus représenté par un bit, mais par un octet. Pour cela, il faut que le matériel utilisé pour afficher l'image soit capable de produire les différents niveaux de gris correspondant.

Le nombre de niveaux de gris dépend du nombre de bits utilisés pour décrire la "couleur" de chaque pixel de l'image. Plus ce nombre est important, plus les niveaux possibles sont nombreux [chakib 99].

3.3.8 Images en couleurs

Même s'il est parfois utile de pouvoir représenter des images en noir et blanc, les applications multimédias utilisent le plus souvent des images en couleurs. La représentation des couleurs s'effectue de la même manière que les images monochromes avec cependant quelques particularités. En effet, il faut tout d'abord choisir un modèle de représentation. On peut représenter les couleurs à l'aide de leurs composantes primaires. Les systèmes émettant de la lumière (écrans d'ordinateurs,...) sont basés sur le principe de la synthèse additive : les couleurs sont composées d'un mélange de rouge, vert et bleu (modèle R.V.B.) [chakib 99].

3.4 Système de traitement d'images

Un système de traitement numérique d'images est composé de :

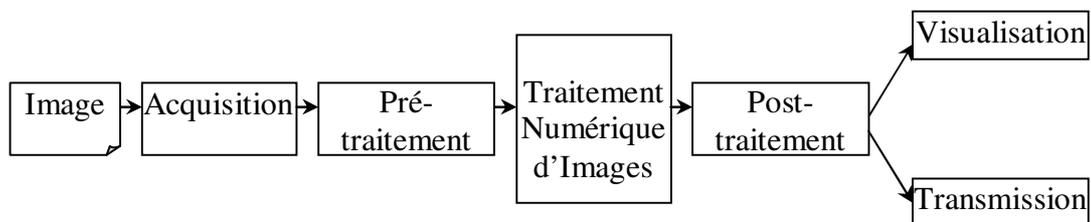


Figure3.2 Schéma d'un système de traitement d'images[chakib 99]

3.5 Pré-traitement d'images

Cette phase a lieu juste après l'acquisition des images et a pour objectif d'améliorer la qualité de l'image en vue de sa segmentation.

Le temps de traitement (temps CPU) est très important. C'est un facteur décisif, il doit être le plus petit possible. Ceci implique que les opérateurs doivent être locaux, c'est-à-dire qu'ils doivent intervenir sur un nombre limité de pixels et essentiellement sur des pixels dans le voisinage proche du pixel courant.

Les techniques de pré-traitements les plus courantes qu'on va présenter sont [Bensrhair02,a] [Cocquerez 95,+]:

- ↳ La modification d'histogramme,
- ↳ La réduction du bruit par filtrage,

3.5.1 Modification d'histogramme

On cherche à améliorer l'image en appliquant une transformation ponctuelle d'intensité; à tout pixel $a_s(x,y)$ on associe une intensité par la transformation T croissante de façon à conserver des contrastes entre régions (claire sur fond sombre, apparaît plus claire après transformation). Le pixel après transformation est le pixel $a'_s(x,y)$:

$$a'_s(x,y) = T(a_s(x,y))$$

Dans ce qui suit, on va présenter deux types de modification d'histogramme:

- Expansion de dynamique
- Egalisation d'histogramme

Expansion de dynamique

Cette méthode consiste à utiliser au mieux l'échelle des niveaux de gris disponible sur le système d'acquisition d'images.

Soit $A[i,j]$ l'image de départ, $A'[i,j]$ image après transformation et $[a_0, a_1]$ l'intervalle des intensités présentes dans l'image et soit $[a_{\min}, a_{\max}]$ l'intervalle disponible.

L'expansion de dynamique correspond à la transformation: $a'_s = \alpha + \beta a_s$ telle que:

$$\forall a \in [a_0, a_1] \quad a_s \xrightarrow{T} a'_s \in [a_{\min}, a_{\max}]$$

$$\alpha = \frac{a_{\min} \cdot a_1 - a_{\max} \cdot a_0}{a_1 - a_0}, \quad \beta = \frac{a_{\max} - a_{\min}}{a_1 - a_0}$$

Egalisation d'histogramme

Cette transformation consiste à rendre le plus plat possible l'histogramme de niveaux de gris de l'image. L'image $A[i,j]$ est considérée comme un ensemble de réalisation d'une variable aléatoire A , admettant une densité de probabilité $f(a)$ et une fonction de répartition $F(a)$, f représente l'histogramme normalisé de $A[i,j]$.

On cherche une transformation T continue dérivable par morceaux et strictement croissante telle que la variable aléatoire $B=T(A)$ soit uniformément répartie dans l'intervalle $[b_{\min}, b_{\max}]$.

Soit $g(b)$ la densité de probabilité de B , et T' la transformation dérivée de T :

$$g(b) = \begin{cases} f(a) \frac{1}{T'(a)} = \frac{1}{b_{\max} - b_{\min}} & \text{si } b_{\min} < b < b_{\max} \\ 0 & \text{ailleurs} \end{cases}$$

$$T'(a) = (b_{\max} - b_{\min}) \cdot f(a) \quad , \quad b = T(a) \quad , \quad a \in [a_{\min}, a_{\max}]$$

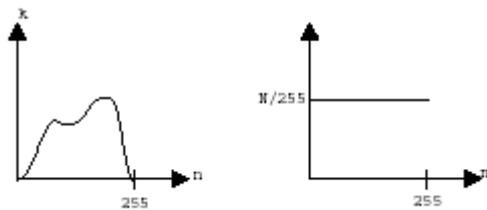


Figure 3.3 Egalisation d'histogramme [Bensrhair 02,a]

3.5.2 Réduction du bruit par filtrage

Le bruit peut provenir de différentes causes :

- Environnement lors de l'acquisition,
- Qualité du capteur,
- Qualité de l'échantillonnage.

En général, le bruit d'images est considéré comme un champ aléatoire caractérisé par sa densité de probabilité f et sa fonction de répartition F . La nature plus ou moins impulsionnelle du bruit peut être décrite à l'aide de densité de probabilités de la forme [Cocquerez 95]:

$$f(a) = C \cdot \exp(-K|a|^\alpha)$$

Plusieurs filtres sont utilisés pour la réduction de bruit, ils sont divisés en deux catégories :

- Filtres linéaires stationnaires ,
- Filtres non linéaires stationnaires.

La première catégorie comprend tous les opérateurs pouvant exprimer leur résultat comme une combinaison linéaire des niveaux de gris d'un voisinage de l'image. Ces filtres possèdent des caractéristiques spectrales, on parle ainsi de filtre passe-bas (l'image devient floue) ou de filtre passe-haut (les contours ressortent).

Voici quelques exemples de filtres réducteurs de bruit :

☞ *Filtre moyenneur*

☞ *Filtre gaussien*

☞ *Filtre exponentiel de Shen*

Le principale inconvénient des filtres linéaire est que la réduction de bruit s'accompagne d'un étalement des transitions entre régions. Ce problème peut être surmonté par l'utilisation des filtres non linéaires.

Le domaine de filtrage non linéaire est extrêmement vaste. Les opérateurs les plus courants en pré-traitement d'images sont :

- Les filtres d'ordre,
- Les filtres homomorphique,
- Les filtres morphologique.

3.6 Extraction de contours [Cocquerez 95,+] [Monga 95,+] [UFRIMA 03][Girard 99,+]

La détection de contour est une étape préliminaire à de nombreuses applications de l'analyse d'images. Les contours constituent en effet des indices riches, au même titre que les points d'intérêts, pour toute interprétation ultérieure de l'image. Les contours dans une image proviennent des :

- discontinuités de la fonction de réflectance (texture, ombre),
- discontinuités de profondeur (bords de l'objet),

et sont caractérisés par des discontinuités de la fonction d'intensité dans les images.



Figure 3.4 Différents types de contours : marche, toit et pointe [Bensrhair 02,b]

Le principe classique de la détection de contours repose sur l'étude des dérivées de la fonction d'intensité dans l'image : les extréma locaux du gradient de la fonction d'intensité et les passages par zéro du laplacien.

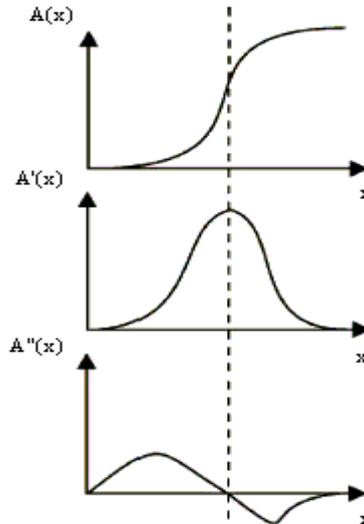


Figure 3.5 Cas bidimensionnel [Bensrhair 02,b]

3.6.1 Approche Gradient [Bensrhair 02,b]

La recherche de la transition peut être réalisée en recherchant les maxima locaux du gradient. Dans le cas bidimensionnel, le vecteur gradient est défini au point $M(x, y)$ par :

$$\vec{\nabla}A(x,y) = \left(\frac{\partial A}{\partial x}, \frac{\partial A}{\partial y} \right)$$

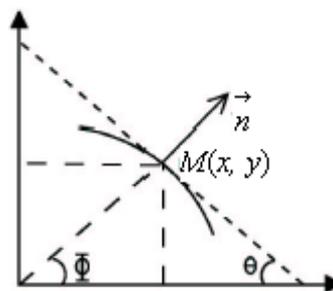


Figure 3.6 Orientation du contour [Bensrhair 02,b]

$\vec{\nabla}A(x,y)$ est normal à la courbe de niveau donnée par $A(x, y) = cste$ qui passe par M .

Au point $M(x, y)$ l'orientation Φ du gradient est donnée par : $\Phi = \arctan \left(\frac{\frac{\partial A}{\partial y}}{\frac{\partial A}{\partial x}} \right)$

et le vecteur unitaire \vec{n} normal au contour : $\vec{n} = \begin{pmatrix} \cos \Phi \\ \sin \Phi \end{pmatrix}$

L'orientation du contour a pour expression : $\theta = \frac{\pi}{2} - \Phi$

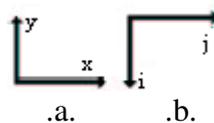


Figure 3.7 Orientation des axes. a) cas continu. b) cas discret

Pour une image numérique, on a :

$$\frac{\partial A}{\partial y} = \frac{\Delta A}{\Delta i} = A_i[i, j] = A_i[i+1, j] - A_i[i, j]$$

$$\frac{\partial A}{\partial x} = \frac{\Delta A}{\Delta j} = A_j[i, j] = A_j[i, j+1] - A_j[i, j]$$

La norme du gradient est :

$$|\nabla A[i, j]| = \sqrt{A_j^2[i, j] + A_i^2[i, j]}$$

ou :

$$|\nabla A[i, j]| = \max\{A_j[i, j], A_i[i, j]\}$$

ou bien

$$|\nabla A[i, j]| = |A_j[j, j]| + |A_i[i, j]|$$

Opérateurs Prewitt et Sobel

$$\begin{aligned}
 A_j[i, j] &= h_j * A[i, j] & h_j &= \begin{pmatrix} 1 & 0 & -1 \\ c & 0 & -c \\ 1 & 0 & -1 \end{pmatrix} \\
 A_i[i, j] &= h_i * A[i, j] & h_i &= \begin{pmatrix} 1 & c & 1 \\ 0 & 0 & 0 \\ -1 & -c & -1 \end{pmatrix}
 \end{aligned}$$

Les matrices h_j et h_i sont appelées masques. Elles sont les noyaux de convolution de filtres à réponse impulsionnelle finie. Pour Prewitt, on prend $c= 1$. Pour Sobel, on prend $c= 2$.

Opérateur de Kirsh

C'est un opérateur à 8 masques correspondant chacun à une direction préférentielle et obtenue par rotation de $\frac{\pi}{4}$ de l'opérateur de base h_0 . (la numérotation des masques est faite dans l'ordre des directions de Freeman en 8-connextité).

$$h_0 = \begin{pmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{pmatrix}$$

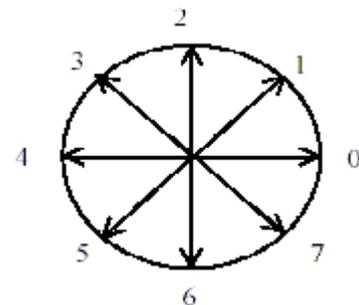


Figure3.8 Les directions de Freeman d'après[Cocquerez 95]

3.6.2 Approche Laplacien

La recherche des contours dans l'image $A(x, y)$ peut aussi être réalisée en recherchant les passages par zéro du laplacien. On dispose donc, de la valeur du laplacien en chaque point de l'image soit la fonction Δ .

$$\Delta A(x, y) = \frac{\partial^2 A}{\partial x^2} + \frac{\partial^2 A}{\partial y^2}$$

On considère que les points de contours sont localisés aux passages par zéro de laplacien. Si le calcul du laplacien était exact il suffirait de sélectionner les points M tels que $\Delta(M)=0$.

3.6.3 Approche par filtrage optimal : Approche de Canny

Filtre optimal de réponse impulsionnelle $h(x)$ qui satisfait les trois contraintes suivantes :

- ↪ Bonne détection
- ↪ Bonne localisation
- ↪ Réponse unique

Soit $A(x)$ un signal mono-dimensionnel représentant un saut d'amplitude U_0 noyé dans un bruit blanc stationnaire $N(x)$ de moyenne nulle et de densité spectrale N_0^2 .

$$A(x) = U_0 U(x) + N(x)$$

Le signal de sortie est :

$$C(x) = A * h(x) = \int_{-\infty}^{\infty} A(t)h(x-t)dt$$

Bonne détection : Le rapport RSB (Rapport Signal/Bruit) doit être maximisé.

$$RSB = \frac{U_0 \int_0^{\infty} h(x-t)dt}{\left[E \left(\left| \int_{-\infty}^{\infty} N(t)h(x-t)dt \right|^2 \right) \right]^{\frac{1}{2}}} = \frac{U_0 \int_0^{\infty} h(x-t)dt}{N_0 \left[\int_{-\infty}^{\infty} h^2(t)dt \right]^{\frac{1}{2}}}$$

Bonne localisation : Les points détectés doivent être aussi près que possible du centre du contour véritable (à maximiser)

$$\frac{U_0}{N_0} \frac{|h'(0)|}{\left[\int_{-\infty}^{\infty} h'^2(t)dt \right]^{\frac{1}{2}}}$$

Unicité de la réponse : On utilise le critère de Canny. On veut minimiser la densité $d0$ des passages par 0 de la réponse du bruit.

Critère généralisé :

$$d_G = \frac{1}{\pi} \left[\frac{\int_{-\infty}^{\infty} h'^2(t) dt}{\int_{-\infty}^{\infty} h^2(t) dt} \right]^{\frac{1}{2}}$$

Critère de Canny :

$$d_G = \frac{1}{\pi} \left[\frac{\int_{-M}^M h'^2(t) dt}{\int_{-M}^M h^2(t) dt} \right]^{\frac{1}{2}}$$

$h(x)$ est un *RIF* défini sur l'intervalle $[-M, M]$ Canny a défini les conditions aux limites suivantes :

- $h(0) = 0$
- $h(M) = 0$
- $h'(0) = S$ (pente à l'origine)
- $h'(M) = 0$

3.7 Segmentation en régions [Baillie 03] [Franck 01] [Duchesnay 01,a]

Fondamentalement, la segmentation est un processus qui consiste à découper une image en régions connexes présentant une homogénéité selon un certain critère, comme par exemple la couleur. L'union de ces régions doit redonner l'image initiale

On regroupe généralement les algorithmes de segmentation en trois grandes classes (voir [Baillie 03]) :

- Segmentation basée sur les pixels
- Segmentation basée sur les régions
- Segmentation basée sur les contours

La première catégorie travaille sur des histogrammes de l'image. Par seuillage, clustering ou clustering flou, l'algorithme construit des classes de couleurs qui sont ensuite projetées sur l'image. La segmentation est implicite puisqu'on suppose que chaque cluster de l'histogramme correspond à une région dans l'image. En pratique, ce n'est pas forcément le cas et il faut séparer les régions de l'image qui sont disjointes bien qu'appartenant au même cluster de couleur. Ces algorithmes sont assez proches des algorithmes de réduction de couleur.

La deuxième catégorie correspond aux algorithmes d'accroissement ou de découpage de région. L'accroissement de région est une méthode bottom-up : on part d'un ensemble de petites régions uniformes dans l'image (de la taille d'un ou de quelques pixels) et on regroupe les régions adjacentes de même couleur jusqu'à ce qu'aucun regroupement ne soit plus

possible. Le découpage de région est le pendant top-down des méthodes d'accroissement : on part de l'image entière que l'on va subdiviser récursivement en plus petites régions tant que ces régions ne seront pas suffisamment homogènes.

La troisième catégorie s'intéresse aux contours des objets dans l'image. La plupart de ces algorithmes sont locaux, c'est à dire fonctionnent au niveau du pixel.

Les algorithmes que nous allons présenter sont :

- Accroissement de région fonctionnant au niveau du pixel.
- *Split and merge* classique.

Des études détaillées de ces algorithmes et d'autres sont données dans [Baillie 03].

3.7.1 Accroissement de région

Les méthodes d'accroissement de régions sont les méthodes de segmentation les plus simples. Le principe est basé sur une approche bottom-up : l'algorithme part de petits éléments de l'image qu'il va tenter de regrouper en éléments plus importants.

Nous présentons ici la version de base de l'algorithme d'accroissement de région qui fonctionne en agrégeant des pixels. Supposons une région de couleur homogène R . Initialement, $R = 1 \text{ pixel}$. On va étendre la région R en incluant les pixels situés sur la frontière et dont la couleur est proche de celle de R (la variation de couleur est inférieure à un seuil δ caractéristique de ce type d'algorithmes). En répétant cette procédure jusqu'à ce qu'il n'y ait plus de pixels de couleur assez proche sur la frontière, on obtient une région de couleur homogène maximale autour du pixel de départ. La région initiale "gonfle" en absorbant des pixels de la frontière, jusqu'à stabilité par rapport à la propriété d'homogénéité. Afin d'assurer un critère de connexité, on utilise les voisins en 4-connexités pour l'accroissement de région et la 8-connexités pour l'accroissement de contour (cas de détection de contours).

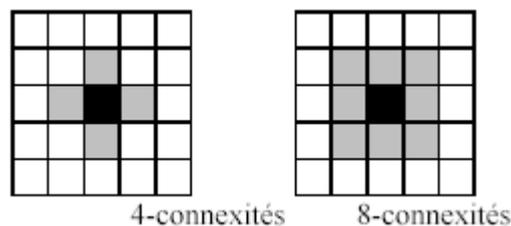


Figure 3.9 Voisinage d'un pixel [Achard 03]

Cette méthode présente deux limitations:

- Les régions obtenues dépendent fortement des pixels d'amorçage choisis et de l'ordre dans lequel les pixels de la frontière sont examinés.
- Le résultat final est très sensible à la valeur du seuil δ .

Cet algorithme fait parti de la classe d'algorithmes de segmentation dit "locaux". L'opération élémentaire consiste à manipuler des pixels adjacents et l'algorithme n'a aucune vision globale du résultat qu'il obtient.

3.7.2 Méthodes du type "split and merge"[Baillie 03] [Duchesnay 01,a]

L'algorithme "split and merge" a été présenté la première fois en 1974 par Pavlidis et Horowitz [Baillie 03]. Cet algorithme s'apparente dans son principe à l'algorithme d'accroissement de région que nous venons de présenter. La différence principale provient de la nature des régions élémentaires agrégées. Dans l'algorithme "split and merge", les régions agrégées proviennent d'une première phase (split) de traitement de l'image qui construit de manière récursive des régions carrées de taille variable mais homogènes.

☞ *Split*

La méthode de découpage de l'image utilisée dans cet algorithme est basée sur la notion de "quadtree". Cette structure de données est un arbre quaternaire qui permet de stocker l'image à plusieurs niveaux de résolution. On part d'une région initiale qui est l'image tout entière. Si cette image vérifie un critère d'homogénéité de couleur, l'algorithme s'arrête. Sinon, on découpe cette région en quatre parties de même taille et on lance la procédure récursivement dans ces quatre parties. La région initiale va être stockée comme un nœud dans un graphe et les sous parties comme des fils de ce nœud. La figure 3.10 montre une image en noir et blanc 8x8 et le découpage correspondant à chaque niveau, ainsi que la structure d'arbre associée à ce découpage.

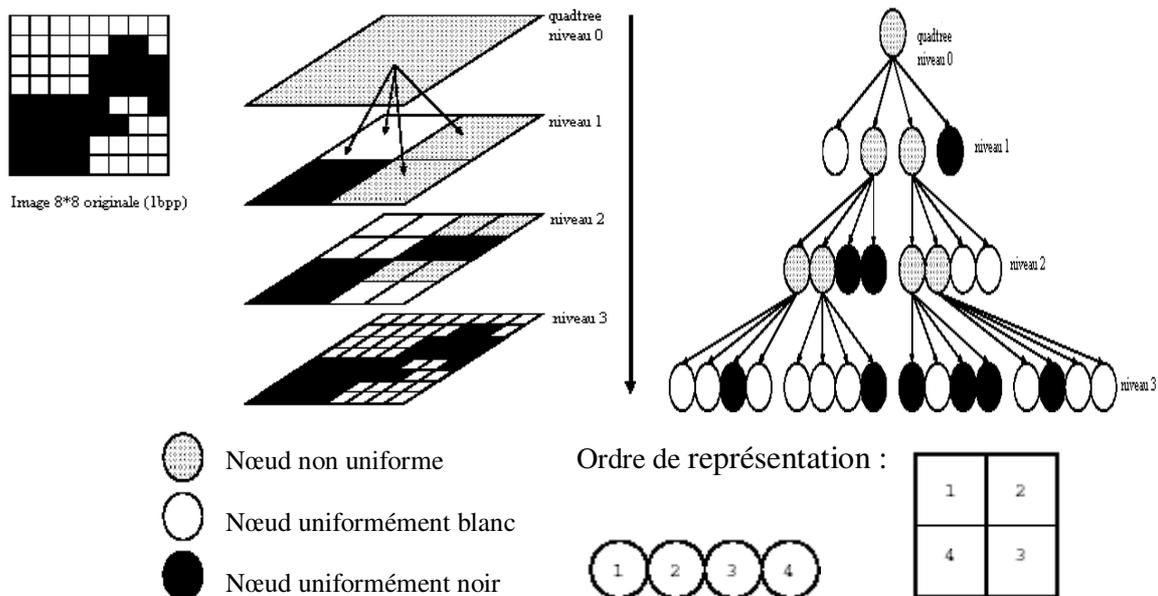


Figure 3.10 *Découpage par quadtree d'une image 8x8. Traitement avec seuil=100% [Baillie 03]*

☞ Merge

La procédure de découpage décrite précédemment aboutit à un nombre de régions trop élevé. La cause fondamentale de cette sur-segmentation est que l'algorithme découpe les régions de manière arbitraire. Il se peut qu'il coupe de cette façon une zone homogène en deux ou quatre parties (Figure 3.11).

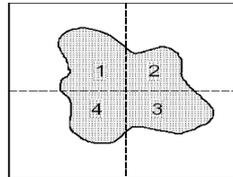


Figure 3.11 *Problème de découpage arbitraire de régions dans la phase de "split"*[Baillie 03]

Les parties 1,2,3 et 4 de la (Figure 3.11) par exemple, appartiennent à des branches différentes du quadtree. Elles sont donc considérées comme des régions différentes bien que leur couleur soit identique.

La solution, qui correspond à la phase "merge" de l'algorithme, est de procéder à une fusion de régions après le découpage. L'implémentation la plus simple de cette fusion cherche tous les couples de régions adjacentes dans l'arbre issu du découpage et cherche à les fusionner si leur couleur est assez proche.

3.8 Approches coopératives en traitement d'images

La résolution collective de problème se préoccupe de développer des systèmes d'agents *simples* capables de résoudre collectivement un problème complexe ; par exemple construire un chemin reliant tous les nœuds d'un graphe et dont la longueur est minimale (problème dit du voyageur de commerce ou TSP).

La question principale qui se pose alors est de déterminer le système de représentation de l'environnement, les comportements individuels, et la dynamique qui va régir le fonctionnement du système afin qu'il produise une réponse au problème qui lui est posé.

Plusieurs modèles inspirés des insectes sociaux ont déjà été élaborés et utilisés pour effectuer de la résolution de problème. L'une des applications de ces modèles est le traitement d'images.

Dans ce type d'approches, l'image y est vue comme un environnement ou territoire analysé par une société d'agents (ou processus) distribués spatialement. Cette approche est novatrice car elle va à l'encontre de la conception classique centralisée et planifiée des tâches d'analyse d'images. Ces architectures mettent en relief la distribution de calcul, l'adaptation

locale des traitements et la richesse des coopérations possibles entre les aspects contour et région.

Dans ce qui suit, on va présenter quelques approches appliquées aux traitement d'images.

3.8.1 Système de Bellet

Bellet [Bellet 98] a développé un système coopératif pour la segmentation de bas niveau des images. 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(figure 3.12).(le lecteur intéressé peut se référé à [Bellet98]).

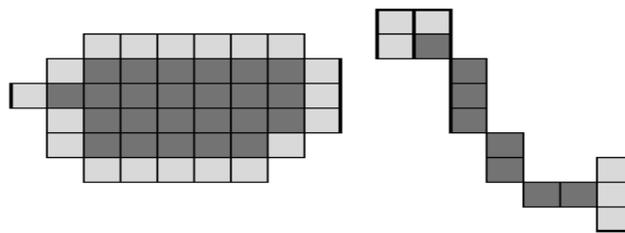


Figure3.12 Principes de la croissance de région et du suivi de contour. Pour faire croître la primitive (pixels foncés), différents candidats sont possibles (pixels clairs). Le pixel ayant la meilleure évaluation sera ajouté à la primitive.

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 fils 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.

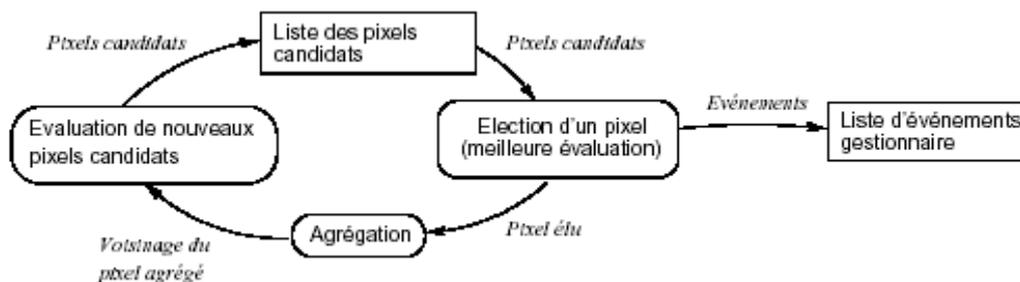


Figure3.13 Boucle de contrôle du comportement de perception. Ce comportement boucle sur une liste de pixels tant que celle-ci n'est pas vide. [Boucher 99]

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.

3.8.2 Approche de Boucher

Alain Boucher [Boucher 99] étend les processus de Bellet [Bellet 98] à des agents permettant de bien identifier les structures internes aux agents et leurs interactions. Les agents sont spécialisés dans la segmentation et l'interprétation d'images cytologiques; un des aspects les plus intéressants concerne le traitement de séquences d'images et la manière de lancer des agents dans l'image suivante de la séquence, en fonction de ce qui est en train de se dérouler dans l'image courante. Les agents disposent de quatre comportements :

- Le comportement de perception explore son environnement et sélectionne des pixels candidats à l'agrégation.
- Le comportement d'interaction gère les fusions et les échanges de pixels entre agents.
- Le comportement de différenciation correspond à une interprétation à valeur sémantique. Les agents représentent au départ des primitives région ou contour ; les informations accumulées sur eux-même et dans un environnement commun vont leur permettre de déterminer la nature sémantique de la primitive traitée.
- Le comportement de reproduction, permet de créer au besoins des agents à certains endroits de l'image. Il conditionne l'exploration de cette image et donc le contrôle de système.

3.8.3 Approche de Duchesnay

Duchesnay [Duchesnay 01] s'appuie sur la structure de pyramide irrégulière (voir la figure 3.14) pour gérer le processus de fusion de régions et assurer la convergence de la segmentation; une coopération région-région assez sophistiquée est mise en œuvre, mais qui ne tire pas suffisamment parti de l'information contour. Un des aspects intéressants de son approche est l'utilisation d'une procédure de décimation (récursive) pour le passage du niveau k au niveau $k+1$. La pyramide se construit en partant de la base qui représente l'image pré segmentée (par exemple avec l'algorithme des quadrees) jusqu'au dernier niveau de la pyramide comportant le minimum d'information. Les niveaux de cette pyramide sont des graphes d'adjacence de régions.

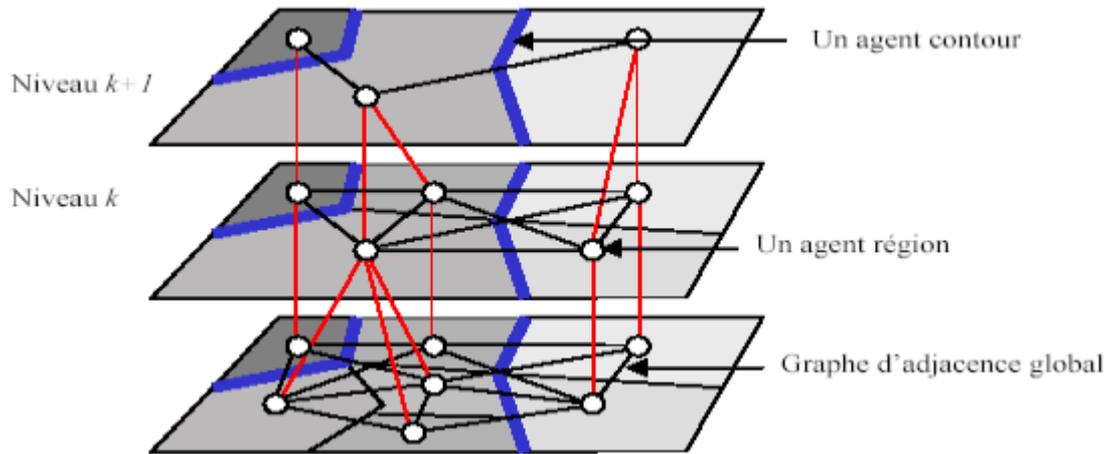


Figure3.14 La pyramide de graphe d'adjacence des régions d'après [Duchesnay 01]

3.8.4 Approche de Liu & Tang : Segmentation par « agents migrants »

Dans cette approche [Liu 99,+], des agents répartis aléatoirement dans l'image vont migrer(se diffuser) de pixel en pixel jusqu'à ce qu'ils trouvent dans leur environnement local un motif recherché. Une fois fixé sur le pixel, l'agent l'étiquette puis se reproduit en initialisant de nouveaux agents dans son voisinage et devient inactif.

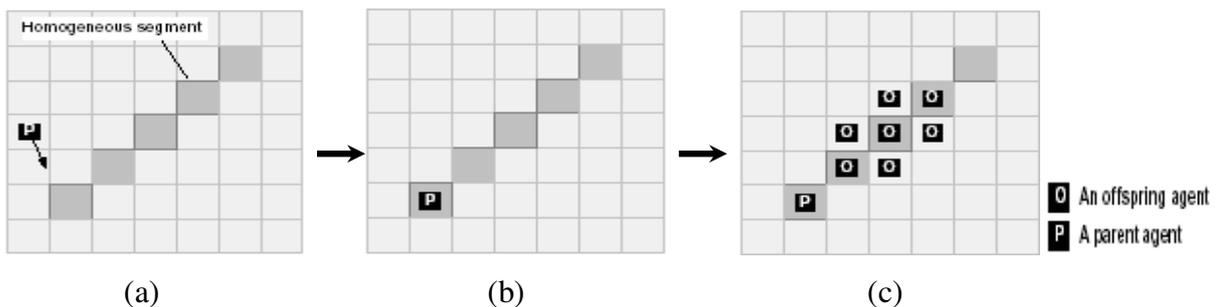


Figure3.15 Comportement de reproduction. (a) Au temps t , un agent, indiqué par le symbole 'P', est prêt à déplacer selon une direction. (b) Au temps $t+1$, l'agent est déplacé à un pixel de segment homogène. (c) Au temps $t+2$, l'agent s'auto-produit un nombre d'agents descendants, indiqués par le symbole 'O', en une direction spécifique calculée à partir de son vecteur propre. [Liu 99,+]

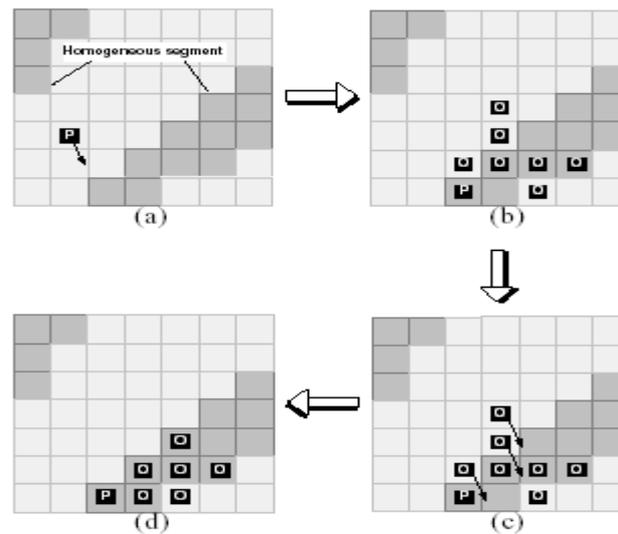


Figure 3.16 Comportement de diffusion (a) Au temps t , un agent 'P', est prêt à se déplacer à un emplacement voisin dans une direction comme indiqué. (b) Dans le temps $t+1$ et $t+2$, l'agent rencontre un pixel de segment homogène et par la suite, il s'auto-reproduit un nombre d'agents descendants 'O'. (c) Les agents descendants 'O' dans des régions non homogènes sont prêts à entreprendre des mouvements de diffusion dans leurs régions voisines, dans une direction comme héritée de leur parent et les agents de mêmes parents. Les agents qui se trouvent aux pixels de segment homogène, produiront plus d'agents descendants (bien qu'il ne soit pas montré dans la figure). (d) Au temps suivant, certains des agents se répandant comme indiqué dans (c) rencontreront plus de pixels de segment homogène. [Liu 99 ,+]

Au cours de sa migration, l'agent va vieillir et peut mourir si le nombre de cycles de recherche excède son temps de vie. Les auteurs ont aussi introduit la notion d'évolution associée aux directions de diffusion et de reproduction. En effet, en cas de succès d'un des comportements dans une direction, la pondération associée à cette direction sera renforcée. Ainsi les comportements de migration et reproduction de ses descendants favoriseront cette direction.

Les auteurs appliquent leur approche à la détection de segment de contour ; les agents migrent jusqu'à trouver un contour, ils s'y fixent, l'étiquettent, et initialisent de nouveaux agents qui auront de fortes chances de trouver un contour recherché dans un voisinage proche.

3.8.5 Une plate-forme pour la segmentation des images IRM

L'objectif de cette plate-forme [Settache 02] multi-agent est la segmentation d'images appliquée aux IRM cérébrales 2D. Dans ce travail, le parallélisme de segmentation permet d'envisager de réaliser la fusion simultanée de couples de régions voisines quelle que soit leur position dans l'image, et quels que soient les critères de candidature à la fusion utilisés.

Dans [Settache 02], Il existe différents protocoles de coopération :

- Coopération région-région pour la fusion de régions.
- Coopération contour-région pour la fusion de régions : il ne doit pas exister de contour entre les deux régions qui souhaitent fusionner
- Coopération contour-région pour la division de région : un contour qui traverse une région peut inciter cette région à se diviser.
- Coopération contour-contour pour le raccordement de contours.
- Coopération région-contour pour le raccordement de contours.
- Coopération région-contour pour le prolongement de contours : on peut prolonger un contour à condition que les régions voisines ne s'y opposent pas. Par exemple, si le contour suit un morceau de la frontière entre deux régions, on pourra le prolonger le long de la frontière.

La figure 3.17 illustre la coopération région-région et région contour pour la fusion de régions. Au départ toutes les régions calculent leur désir de fusion avec chacun de leur voisins et ce désir sont communiqués au meilleur voisin par un message (par exemple une flèche sur la figure 3.17a entre R10 et R110). Une opération de fusion est exécutée quand le désir de fusion est mutuel (figure 3.17.b entre R160 et R122) et la région se met alors dans un état d'attente jusqu'à exécution de la fusion de son meilleur voisin (par exemple la région R110 dans la figure 3.17.b). Comme on peut voir sur la figure 3.17.d la fusion des deux régions R210-R130 est stoppée par l'agent contour qui se situe aux frontières de ces deux régions.

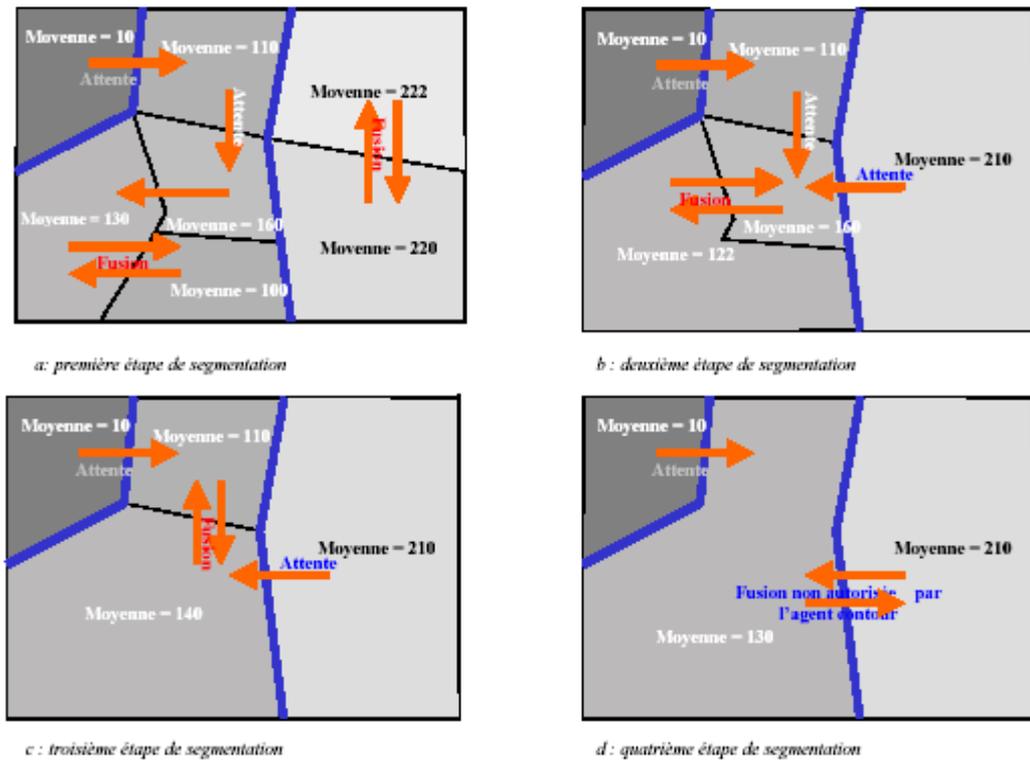


Figure 3.17 Exemple de coopération région-contour et région-région pour la fusion des régions [Settache 02]

3.8.6 Détection de régions dans des images à niveaux de gris : l'inspiration des araignées sociales

Ce travail [Bourjot 01,+] a exposé une démarche de transposition pour passer d'un modèle de simulation à un modèle de résolution. Le cadre biologique initial concerne un processus de construction collective de toile chez les araignées sociales. Sa transposition concerne la détection de régions dans des images à niveau de gris.

L'environnement est un carré de $N \times N$ piquets dans lequel évoluent les araignées et correspond à l'image à analyser (figure 3.18.a). Un piquet correspond à un ou plusieurs pixels de l'image d'origine, selon l'échantillonnage réalisé. La hauteur d'un piquet correspond au niveau de gris du ou des pixels correspondants (figure 3.18.b). Il peut donc y avoir 256 hauteurs différentes. Initialement, il n'y a pas de fils dans l'environnement, ils apparaîtront au fur et à mesure du déplacement des araignées.

Les items comportementaux d'une araignée sont le *déplacement*, la *pose* d'un fil et, en plus, un comportement de *retour* sur la toile. L'araignée est dotée d'un état interne pour permettre une pose contextuelle.

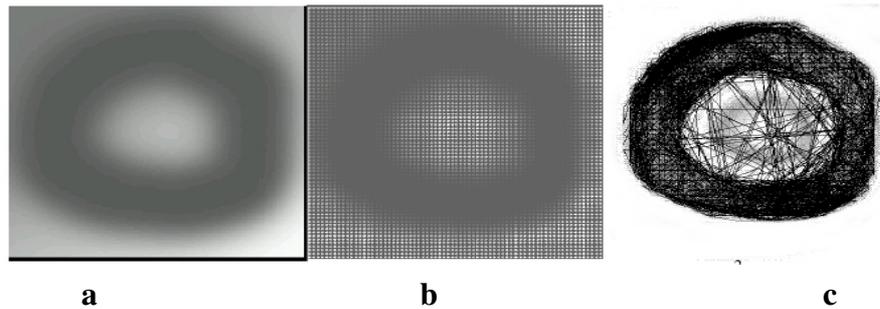


Figure 3.18 Exemple de résultat, d'après [Bourjot01+]

La dynamique du système est la même que dans le cas de la simulation et est de type stigmergique. Ce processus construit des toiles qui sont interprétées comme des régions de l'image.

3.8.7 Traitement d'images par vie artificielle

Carden [Carden 02] propose une approche de détection des contours des régions dans une image simple en niveaux de gris par vie artificielle. Ce travail est basé sur les travaux de Reynolds et MacGill [MacGill 00] où les agents représentent les « géoboids », qui permettent de calculer le gradient.

Les agents ont différents comportements de direction trois inspirés de Boids de Reynolds : (voir figure 3.19) cohésion, alignement et séparation plus deux autres inspirés de Géoboids de MacGill : chercher et fuir.

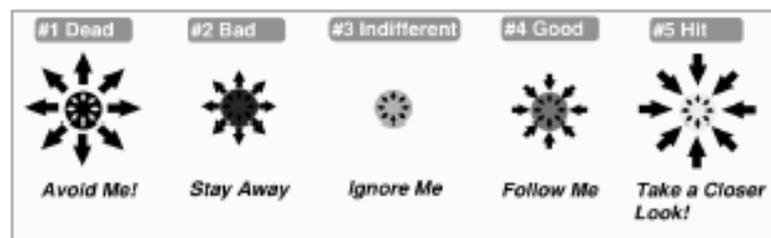


Figure 3.19 Cinq états de geoBoid, utilisés pour les boids de traitement d'images [MacGill 00]

Ces deux derniers comportements visent à chercher : se diriger vers des agents satisfaits (qui sont dans une position proche du contour), et à fuir : s'éloigner des agents insatisfaits (position plus loin du contour).

Un résultat d'application de cette méthode est indiqué dans la (figure 3.20).

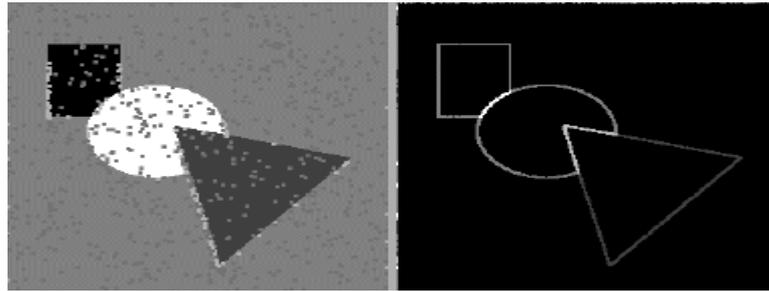


Figure 3.20 Exemple de résultat (image de synthèse) d'après [Carden 02]

Conclusion

Nous avons introduit dans ce rapport les notions de base qui servent de fondement à la compréhension de différentes techniques de traitement d'images. Plusieurs méthodes classiques de traitement ont été proposées dans la littérature, nous avons présenté quelques unes qui nous semblent les plus courantes dans le processus du traitement et analyse d'image.

Les pré-traitements d'images permettent d'améliorer la qualité de l'image en vue de traitements ultérieurs. Deux grandes approches peuvent être envisagées pour extraire les zones pertinentes des images :

On cherche les discontinuités dans la scène, c'est l'approche contour.

On cherche des zones de niveaux de gris homogènes, c'est l'approche région

De nouvelles recherches s'intéressent aux systèmes complexes et leurs caractéristiques ainsi l'application de ces approches sur le traitement d'images.

Les phénomènes collectifs naturels sont une source d'inspiration pour proposer des méthodes multi-agents de résolution de problèmes. Ils permettent l'élaboration de comportements individuels simples produisant collectivement des phénomènes complexes.

A la fin de ce chapitre, nous avons présenté les différentes approches qui ont été élaborées pour le traitement d'images en niveaux de gris soit par agents simples situés dans l'image, ou bien, par inspiration des systèmes naturels.



Chapitre 04

Traitement d'images par écosystèmes

Introduction

Le domaine de la simulation d'écosystèmes requiert la contribution de plusieurs disciplines notamment certaines liées à la biologie et à l'informatique. Le but de ce type de simulation est de comprendre les formes complexes d'interactions qui caractérisent les organismes d'écosystèmes. L'inspiration la plus reconnue de ces interactions est la coevolution (*voir §2.1.4*), où l'évolution d'une espèce affecte l'évolution des espèces avec lesquelles elle interagit, par exemple par des relations de type chaîne alimentaire, prédateur proies, symbiose, parasitisme ...

Dans ce chapitre, nous présentons une approche de segmentation par *coevolution* d'images en niveaux de gris. Pour cela on a utilisé deux types hétérogènes d'agents réactifs : agent région et agent contour, qui contribuent à la segmentation de l'image qui est considérée comme leur environnement d'évolution.

Afin de détailler cette méthode, ce chapitre est organisé de la façon suivante :

Premièrement, les différentes approches concernant l'application de coevolution ainsi des définitions de base seront présentés. Le paragraphe suivant s'attardera sur la description de notre approche suivi des détails d'évolution des différents agents. Les comportements des agents seront abordés à la fin de ce chapitre. Ce dernier sera achevé par une conclusion.

4.1 Travaux récents dans les algorithmes coevolutionnaires

Dans ces dernières années, plusieurs recherches qui s'intéressent à l'application de la coevolution ont été publiées dans la littérature. Ces recherches utilisent, un type d'algorithmes qui constitue une extension des *algorithmes evolutionnaires* est reconnu par : *Algorithmes coevolutionnaires* [Kirley 02].

Dans [Drezewski 03], un système multi agent coévolutionnaire (*co-evolutionary multi-agent system* : CoEMAS) a été introduit ainsi un modèle formel de CoEMAS est présenté. Dans un tel système, deux ou plusieurs espèces d'agents coévolvent afin de résoudre un problème donné. L'application de CoEMAS à l'Optimisation de la Fonction Multi – Modale a été présentée dans ce travail.

Une autre application qui utilise les algorithmes coevolutionnaires est présentée dans [Zaritsky 03]. Cette application concerne la recherche de la plus courte ficelle qui contient toutes les ficelles d'un ensemble donné. Ce problème est reconnu sous l'appellation (SCS: Shortest Common Superstring). Trouver le plus court ficelle commun a des candidatures dans la compression de données. SCS a aussi des candidatures importantes dans la biologie computationnelle (computational biology) concernant le problème de séquençement d'ADN...

Dans le domaine de traitement d'images, le travail de [Veenman 03,+], présente un modèle pour le problème de segmentation, qui n'a pas besoin de connaissance au sujet du nombre de groupes a priori. Le système proposé est basé sur un algorithme du coevolution cellulaire pour l'optimisation du modèle. Dans ce modèle des agents sont placés dans une grille 2D régulière qui représente l'image et qui impose des relations avoisinantes sur eux. Les agents, d'une manière coopérative, considèrent la migration de pixel d'un agent à l'autre pour améliorer l'homogénéité de l'ensemble des régions de l'image qu'ils représentent. Si l'union des régions d'agents avoisinantes est homogène alors les agents forment des alliances. De l'autre côté, si un agent découvre un sujet déviant, il isole le sujet.

4.2 Définitions de base

4.2.1 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]. Nous avons choisi une méthode reposant sur l'utilisation de gradient discret (voir §4.5). La norme du gradient est calculée en utilisant les directions représentées par le code de Freeman.

4.2.2 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. 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 [Boucher 99]. Dans notre méthode, le seuil n'est utilisé que s'il est nécessaire (C'est à dire, lorsque l'agent région rencontre un agent contour).

4.3 Description de l'approche

L'approche proposée dans ce mémoire concerne la segmentation d'images en niveaux de gris par coevolution. Pour cela, deux types d'agents réactifs sont définis : *Agents contour* et *Agent Région*. Initialement, les agents sont initialisés aléatoirement comme des germes dans l'image, qui représente l'environnement d'évolution de ces agents.

Nous avons utilisé la construction incrémentale (pour le contour et la région) qui permet de prendre en compte les configurations locales de l'environnement, tant en termes photométriques qu'en termes de primitives préexistantes dans le système. L'incrémentalité dans un système de vision apporte donc une richesse de fonctionnement à travers une

évolution temporelle, des possibilités de prises de décision dynamiques, et plus globalement une adaptabilité des méthodes.

Après initialisation, les agents régions et contours se déplacent et marquent leur environnement suivant le principe de l'incrémentalité.

Dans ce qui suit, on va détailler le processus d'évolution de chaque type d'agents.

4.4 Evolution d'agent région

A partir d'un germe (pixel) initial, l'agent région commence à se déplacer d'un pixel à un autre afin de marquer une région homogène. Le déplacement de l'agent se fait aléatoirement à l'un de ses 4-voisins qui représentent les candidats à ajouter à la région. L'illustration est montrée dans la (figure 4.1).

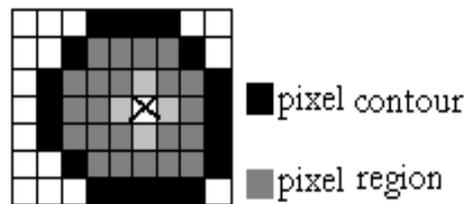


Figure 4.1 Evolution d'un agent region à partir d'un germe initial.

Considérons une méthode de croissance incrémentale de régions utilisant le schéma générique décrit au chapitre 3, dont le but est de fournir des régions correspondant à des zones de luminosité homogène de l'image. L'algorithme d'évaluation des pixels devra donc logiquement privilégier les points candidats dont l'intensité des niveaux de gris est la plus proche du niveau de gris moyen de la région courante. Définissons ainsi $P(x,y)$ le pixel candidat, ayant une intensité $I(x,y)$. La région R courante contient n pixels référencés $P(x_0,y_0), P(x_1,y_1), \dots, P(x_{n-1}, y_{n-1})$, et la moyenne des niveaux de gris de R est définie par :

$$\bar{I}_R = \frac{1}{n} \sum_{i=0}^{n-1} I(x_i, y_i)$$

L'évaluation E_1 du pixel P par rapport à la région R se définit donc par :

$$E_1(x, y) = 1 - \min\left(1, \frac{|I(x, y) - \bar{I}_R|}{k}\right)$$

où k est une constante

L'évaluation E_1 est maximum pour les points dont le niveau de gris correspond exactement à celui moyen de la région, et décroît lorsque cette différence devient plus importante. Le seuil k permet de normaliser l'expression dans l'intervalle $[0..1]$.

L'application de ce critère unique fournira certainement un résultat adéquat, mais au prix d'une forte sensibilité au bruit et aux éléments texturés de l'image. Dans de telles conditions, l'algorithme de croissance génère une région à la frontière déchiquetée, et peu exploitable en l'état. En effet, le pixel recherché est le plus homogène, il peut donc être située n'importe où à la frontière de la région. Aucune contrainte de forme ne vient forcer la région à conserver un aspect régulier. D'autres considérations et contraintes d'implantation peuvent suggérer de construire des régions aux formes plus régulières, sans pour autant renoncer au critère d'homogénéité initial.

L'idée est d'introduire une évaluation supplémentaire, notée E_2 , dont le rôle est d'évaluer le degré de compacité du pixel $P(x,y)$, comparé à la région R : Définissons des relations de voisinage $N_8(P(x_1,y_1),P(x_2,y_2)) = 1$ si et seulement si $\max(|x_2 - x_1|, |y_2 - y_1|) = 1$, ensuite l'évaluation :

$$\begin{aligned} E_2(x, y) &= \frac{1}{8} \sum_{p(x_i, y_i) \in R} N_8(P(x_i, y_i), P(x, y)) \\ &= \frac{1}{8} (\text{Nombre de pixels du voisinage immédiat de } P) \end{aligned}$$

Cette seconde évaluation E_2 fournira de fortes valeurs pour les pixels les plus entourés par une région.

Chaque évaluation considérée séparément correspond à un critère élémentaire. E_1 est le critère *d'homogénéité* tandis que E_2 reflète ce que l'on peut qualifier de critère de *compacité*, dont le comportement seul sera de combler les petits trous et aspérités au cours de la croissance de la région.

4.5 Evolution d'agent contour

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

La construction du contour suit le même principe de construction de région basé sur le modèle incrémental de croissance. La *figure 4.2*, illustre le principe d'évolution d'un contour. Après initialisation aléatoire dans l'image, l'agent contour commence à se déplacer d'un pixel

C'est à dire que la distance de P_1 à P_2 vaut 1 pour la norme ∞ . Le déplacement (dx,dy) entre les pixels P_2 et P_1 se définit alors par $(dx,dy) = (x_2-x_1,y_2-y_1)$, avec comme contrainte : $dx,dy \in \{-1,0,1\}$.

On définit alors la direction de Freeman entre les pixels P_2 et P_1 par : $D = F(dx,dy)$, avec :

$$\begin{array}{lll} F(-1,-1)=3 & F(0,-1)=2 & F(1,-1)=1 \\ F(-1,0)=4 & F(0,0) \text{ indéfini} & F(1,0)=0 \\ F(-1,1)=5 & F(0,1)=6 & F(1,1)=7 \end{array}$$

Dans le cas où la direction de Freeman F est paire (figure 4.4, cas 2a), la sélection des trois pixels grises est effectuée, correspondant aux directions F , $(F + 1) \bmod 8$ et $(F + 7) \bmod 8$. Dans le cas où la direction de Freeman est impaire (figure 4.4, cas 2b et 2c), il convient alors de sélectionner en plus les pixels correspondant aux directions $(F + 2) \bmod 8$ et $(F + 6) \bmod 8$.

La principale évaluation relative à un pixel contour candidat sera la norme du gradient, les autres types d'évaluation ne serviront qu'à pondérer cette évaluation principale. Considérons le pixel candidat $P(x,y)$ pour le contour C dans l'image P . Le gradient est calculé à partir des valeurs d'intensité dans le voisinage $\overrightarrow{G(x,y)}$, sous la forme de sa norme $G(x,y) = \|\overrightarrow{G(x,y)}\|$, et de sa direction G_{d8} , échantillonnée suivant les huit directions de Freeman $G_{d8} \in \{0..7\}$.

- La première évaluation repose sur une mesure locale du gradient :

$$E_1 = \frac{G(x, y)}{\max(x_i, y_i)_{i \in P} G(x_i, y_i)}$$

Cette évaluation privilégie les points avec les plus fortes valeurs de la norme du gradient.

- $E_2 = 1$, la norme du gradient est localement maximum dans la direction du gradient.

$E_2 = 0$ sinon.

$$E_2 = 1 \quad - \quad \left\{ \begin{array}{ll} G(x-1, y) < G(x, y) & \text{et } G(x+1, y) < G(x, y) \\ & \text{si } G_{d8} \in \{0, 4\} \\ G(x, y-1) < G(x, y) & \text{et } G(x, y+1) < G(x, y) \\ & \text{si } G_{d8} \in \{2, 6\} \\ G(x-1, y+1) < G(x, y) & \text{et } G(x+1, y-1) < G(x, y) \\ & \text{si } G_{d8} \in \{1, 5\} \\ G(x+1, y+1) < G(x, y) & \text{et } G(x-1, y-1) < G(x, y) \\ & \text{si } G_{d8} \in \{3, 7\} \end{array} \right.$$

4.6 Coevolution d'agents pour la segmentation d'images

On a présenté dans le chapitre 2 (§4), le principe de coevolution ainsi que ses formes (coevolution coopérative et coevolution compétitive). Dans notre travail, nous avons exploité le principe de coevolution compétitive pour la segmentation d'images en niveaux de gris en s'inspirant des travaux de [Veenman 03,+]. Pour cela on a défini deux types d'agents : agent contour et agent région présentés ci-dessus. Ces agents contribuent à la segmentation de l'image qui constitue leur environnement. Le principe de ce processus est illustré dans la figure ci-dessous.

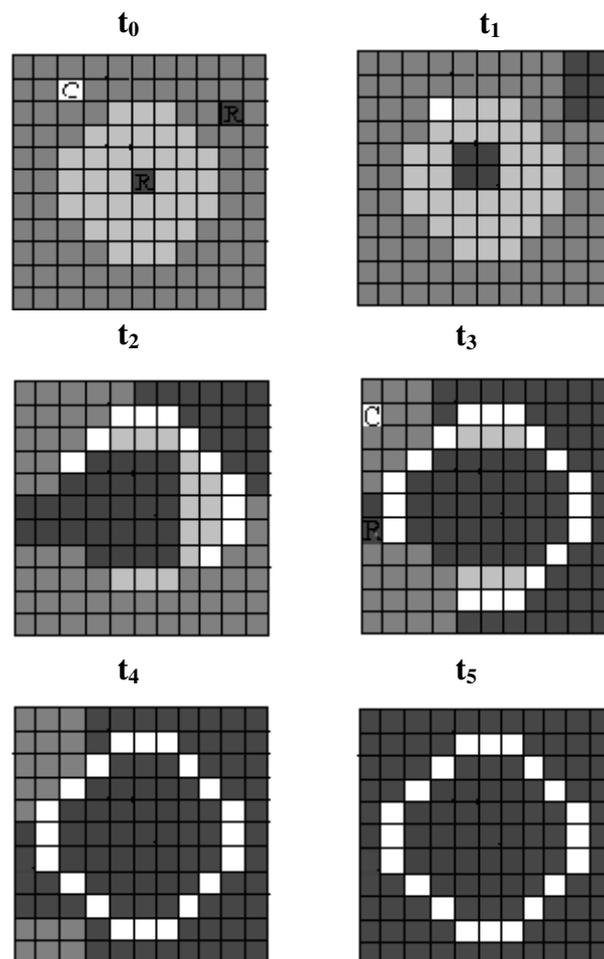


Figure 4.4 Principe de segmentation d'une image par coevolution entre agents région et agents contour. A l'instant t_0 , trois agents sont initialisés dans l'image, agent contour (case marquée par la couleur blanche et avec une lettre C) et deux agents région (cases grises foncées avec une lettre R). à l'instant t_1 , lancement des trois agents ; l'agent contour se déplace à l'un des pixels voisins il se trouve à un pixel qui représente le germe initial ; les deux agents région commencent à marquer leurs régions suivant le principe incrémental de région. À l'instant t_2 , il y a une compétition entre agent contour et agent région, l'agent région dépasse le contour avant l'arrivée d'agent contour. A l'instant t_3 , l'agent contour arrête l'agent région de se dépasser sa région et reproduction de deux agents : agent région qui continue l'évolution de la nouvelle région et un agent contour pour évoluer un contour s'il existe. A l'instant t_4 , la mort d'un agent région ainsi les agents contour et la fusion des deux agents région. A l'instant t_5 , image résultat, définition de deux régions.

4.7 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. Chaque comportement correspond à une étape nécessaire du processus de segmentation. Trois comportements sont définis dans notre méthode concernant les deux types d'agents (région et contour):

- **perception** : Ce comportement 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.
- **reproduction** : Sert à créer des agents en certains lieux de l'image. Ce comportement permet à un agent d'explorer son environnement local et de lancer des germes d'agents qui continueront le travail de segmentation.
- **interaction** : Le comportement d'interaction gère tous les processus d'interactions entre deux agents. Il s'agit d'actions qui résultent en des communications directes entre deux agents concernés par une situation précise comme la fusion de primitives décrivant la même entité. 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, 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.

Le principe de base dans l'action de ces comportements est l'évaluation de l'information afin de prendre des décisions. La coevolution entre les agents est le résultat de ces actions, où l'évolution d'un agent affecte l'évolution d'un autre agent. Ceci est évident pour le cas d'une évolution d'un agent région par exemple, et qui dépasse les limites de celui-ci, l'agent contour va arrêter l'évolution de l'agent région sans utilisation d'un seuil sur l'ensemble de la région.

Le schéma général du système est représenté dans la figure (4.5) :

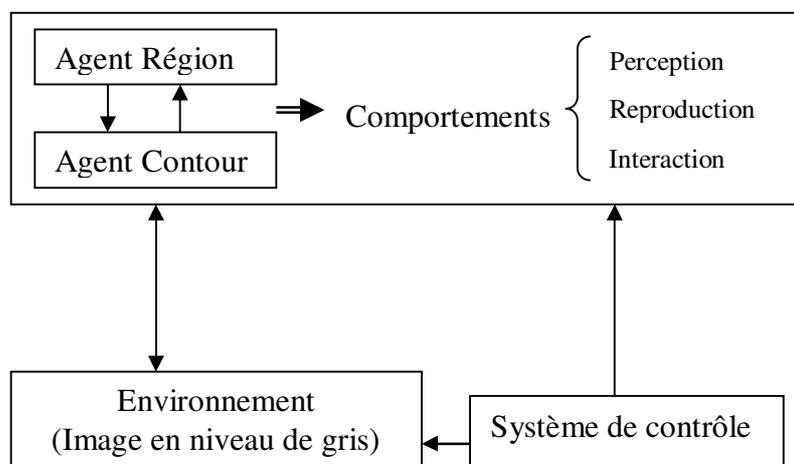


Figure 4.5 Schéma général du système

Conclusion

Dans ce chapitre, on a présenté une description détaillée d'une nouvelle méthode pour la segmentation d'images en niveaux de gris en se basant sur le principe de coevolution. On a vu que deux types d'agents (région et contour) contribuent à la segmentation d'une image par une coevolution compétitive. Les résultats obtenus par application de cette méthode seront abordés dans le chapitre suivant.



Chapitre 05

Résultats

Introduction

Afin de mettre en valeur, ou de mieux expliquer, les aspects importants de notre système, différents résultats illustrant le système de segmentation d'images de synthèse et d'images médicales sont montrés dans ce chapitre. L'aboutissement à ces résultats est facilité par l'utilisation d'un *PC PIII 500MH* d'une *RAM 128 Mo*, ainsi par l'utilisation du *Builder C++*. Ainsi l'application de la méthode sous NetLogo, qui a donnée des résultats encourageants.

5.1 Initialisation du système

Avant de démarrer le système, un ensemble de paramètres doivent être initialisés. Ces paramètres présentent :

- *Nombre d'agents contour et d'agents région (N_{region} , $N_{contour}$)* : ces paramètres sont définis avant de placer les agents dans l'image originale. Ils représentent respectivement le nombre d'agents contour qui démarrent la croissance du contour, et le nombre d'agents région qui démarrent la croissance de régions. Ces deux paramètres varient au fur et à mesure de l'évolution des agents, ceci est expliqué par le comportement de reproduction.
- *Seuil région* : Il est nécessaire seulement dans le cas où une région est arrêtée par un contour, il permet de déterminer les candidats (les 4-voisins) à agréger au région.

5.2 Application aux images de synthèse

Afin de valider notre approche, nous avons l'appliquée sur des images de synthèse (voir l'exemple de la figure 5.1).

Dans l'exemple ci-dessous, nous avons utilisé une image de synthèse en niveau de gris de taille 256*256. Initialement, nous avons défini $N_{region}=05$ (nombre de germes région) et $N_{contour}=02$ (nombre de germes contour).

A $t=t_0$, les agents région cherchent les candidats en 4-voisins et les marquent comme appartenant à la région homogène trouvée. Les agents contours, n'apparaissent pas, puisque ils cherchent les germes initiaux qui démarrent la croissance de contour en se basant sur les 8-voisins du germe courant.

A $t=t_1$ les agents région et contour en compétition pour segmenter l'image ; les agents contour évoluent de telle façon à interdire les agents région de dépasser les limites des régions homogènes. Dans ce cas, le comportement de reproduction est apparaît. Il est évident aussi, que deux agents de même type permettent une fusion.

A $t=t_2$, deux agents région meurent. Les autres agents continuent la segmentation. L'image résultat est donnée dans la figure ci-dessous.

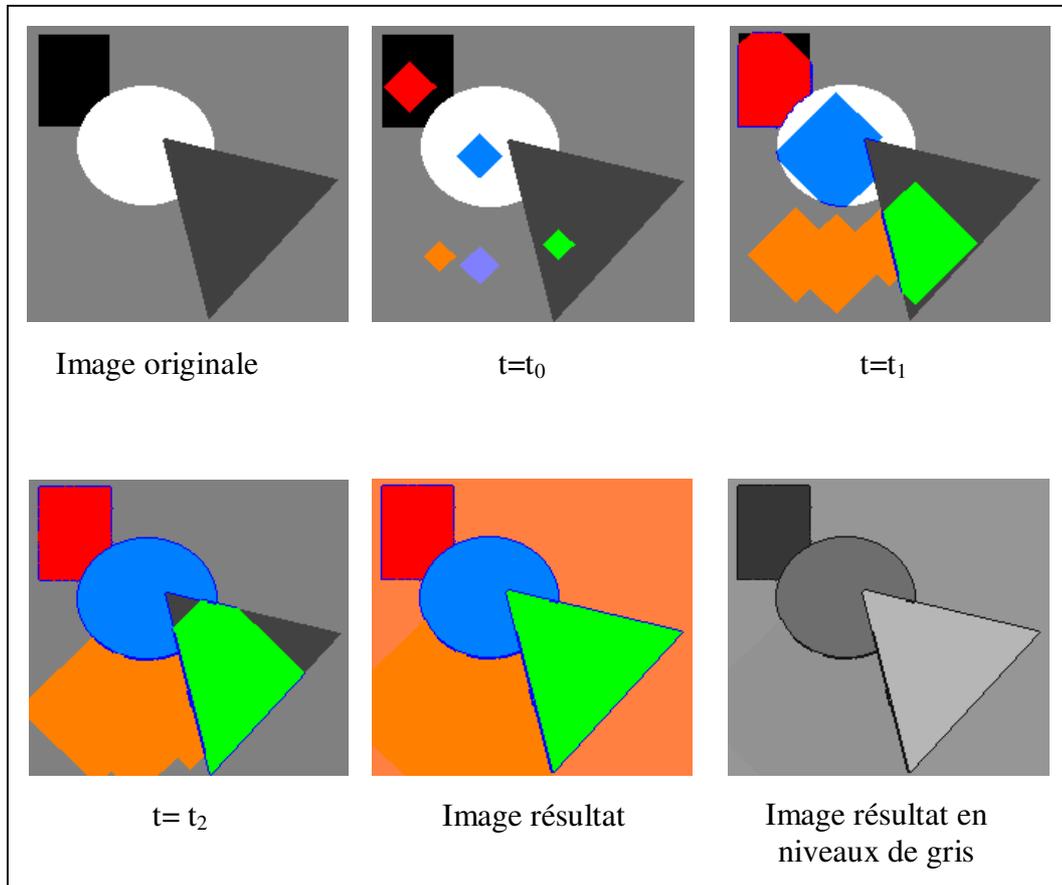


Figure 5.1 *Processus de coevolution d'agents (contour et région) pour la segmentation d'une image de synthèse (256*256)*

Durant l'évolution de processus de segmentation, on a acquérir le nombre de pixels représentant les régions marquées ainsi le nombre de pixels des contours détectés. La figure 5.2, représente la croissance du nombre de pixels régions et contours dans le temps. La courbe bleue représente le nombre de pixels des régions, et la courbe mauve représente le nombre de pixels contours récoltés par les agents au fur et à mesure de processus de coevolution.

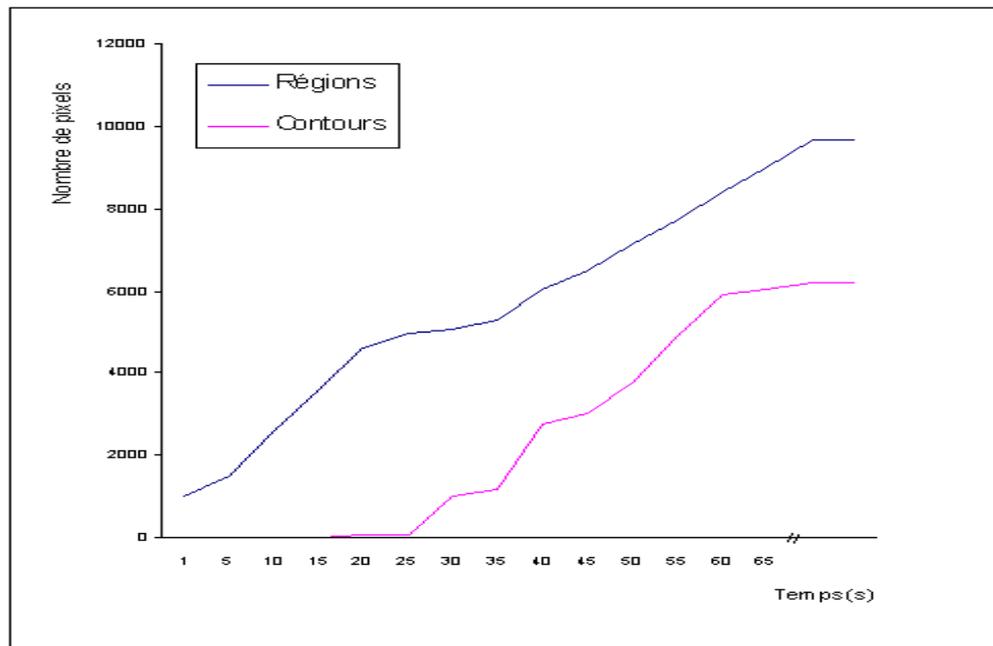


Figure 5.2 Evolution du nombre de pixels régions et contours en fonction du temps

Nous avons remarqué, qu'il y'a une stabilisation après un certain temps du nombre de pixels détectés soit par les agents région ou bien par les agents contour. Ceci est expliqué par le fait que les agents terminent leurs travaux et qu'il ne reste pas de candidats à agréger aux primitives.

5.3 Application aux images médicales

Notre système est appliqué sur des images médicales (IRM), un des exemples est présenté dans la figure 5.2.

Dans cet exemple, nous avons utilisé une image cérébrale 140*170. Le système est initialisé par $N_{\text{région}}=8$ et $N_{\text{contour}}=5$. Durant le processus de coevolution compétitive d'agents, il y'a une reproduction d'agents qui continuent la recherche des régions à segmenter.

A $t=t_0$, les agents commencent leur évolution. A $t=t_1$, reproduction de nouveaux agents qui cherchent des régions à segmenter.

A $t=t_2$, les régions qui segmentent une même composante sont fusionnées. Un seul des deux agents continuera le travail, l'autre se terminant pour permettre de diminuer la charge du système.

A $t=t_3$, le résultat est présenté sous formes de régions colorées par les couleurs d'agents. Le résultat en niveaux de gris est montré dans la figure ci dessous.

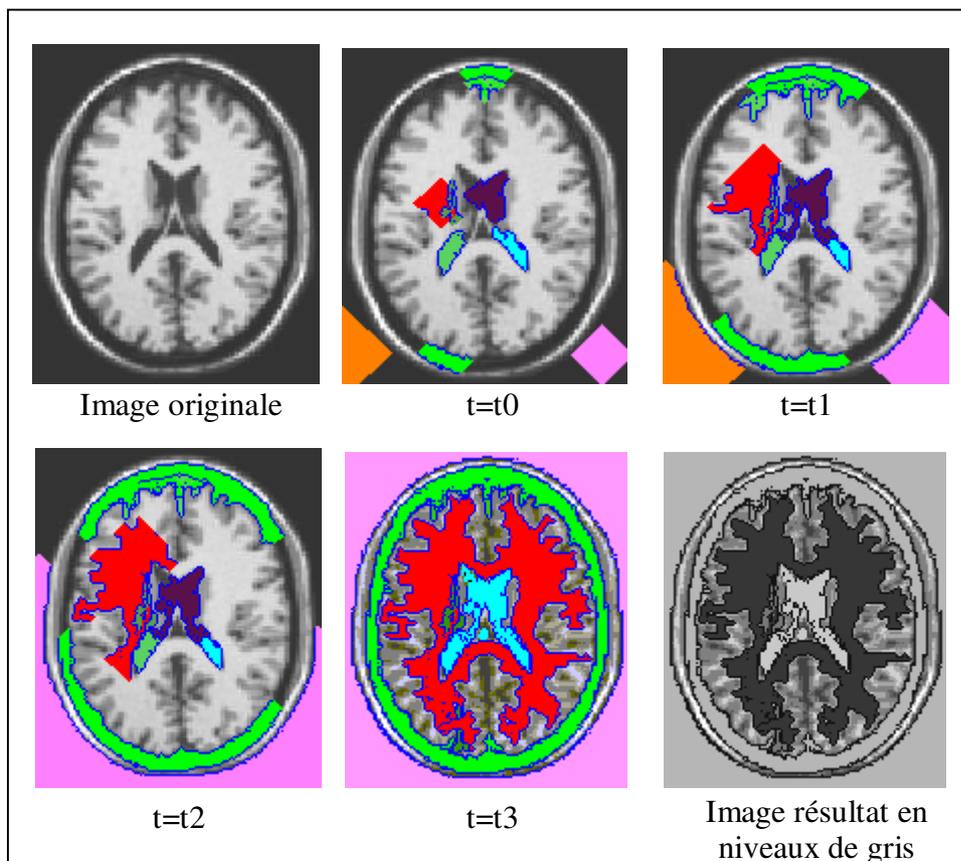


Figure 5.3 *Processus de segmentation par coévolution d'une image cérébrale (140*170)*

5.4 Utilisation du NetLogo (1)

NetLogo est un environnement de programmation multi agents écrit en Java, il a plusieurs versions. Vu les grands avantages (parallélisme...) qu'offre NetLogo, nous avons l'exploiter pour l'application de notre méthode sur différents types d'images. Des résultats satisfaisants sont démontrés dans la figure (5.4).

NetLogo permet de simuler des phénomènes naturels et sociaux. Il est bien convenu pour modeler des systèmes complexes qui se développent avec le temps en particulier. Les modeleurs peuvent donner des directives à centaines ou milliers d'agents " *indépendants* " qui fonctionnent en concurrence.

NetLogo est doté de trois types d'agent : tortues, patchs, et observateur. Les tortues sont des agents qui peuvent prendre différentes formes et qui installent sur le monde. Le monde est une grille de deux dimensions divisé en cases représentant les patchs. Ils constituent l'environnement des tortues qui peuvent se déplacer sur ces patchs. L'observateur n'a pas d'emplacement, vous pouvez l'imaginer en dehors du monde de tortues et patchs.

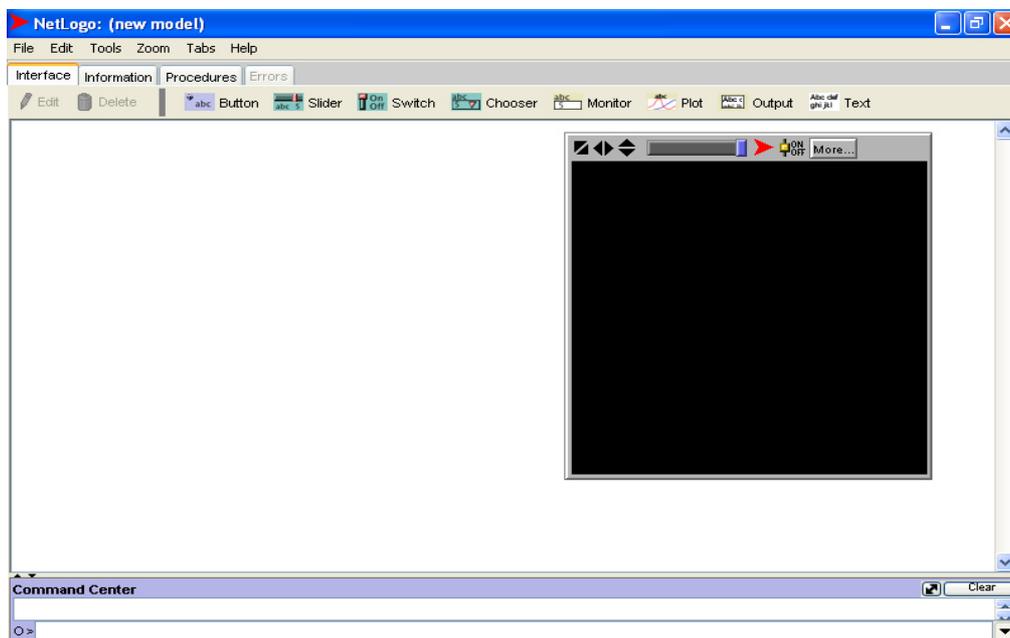


Figure 5.4 Interface du NetLogo (Version 2.1)

(1) <http://ccl.northwestern.edu/netlogo/>

Notre méthode est appliquée sous NetLogo et nous avons abouti aux résultats présentés dans la figure ci-dessous (images de synthèse et réelles).

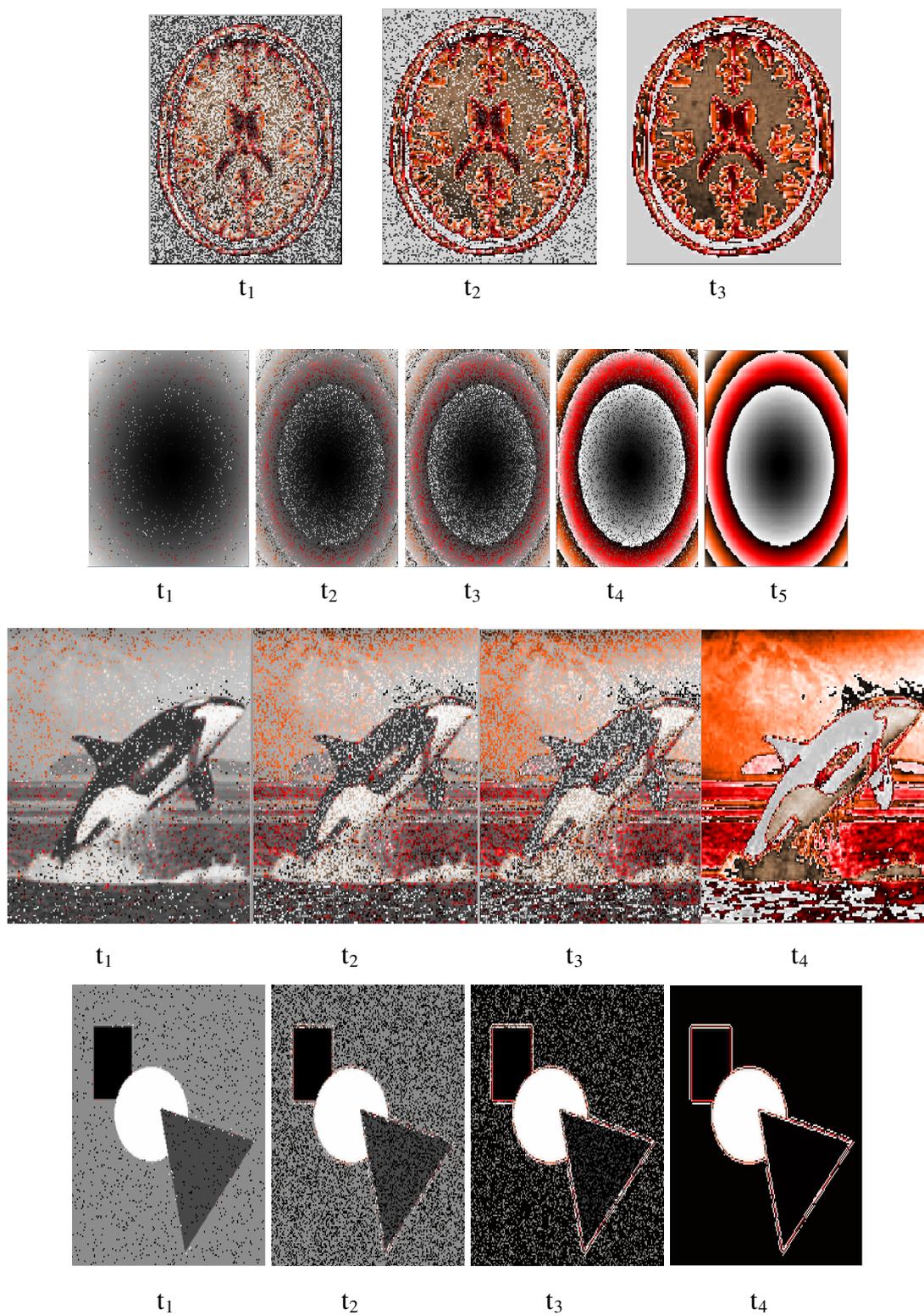


Figure 5.5 Ensemble de résultats obtenus à partir du NetLogo

5.4 Discussion

Les résultats présentés dans ce chapitre, montrent le processus de coevolution entre les deux types d'agents : région et contour pour la segmentation d'images de différents types. Nous avons remarqué que, plus le nombre de germes contours et région est élevé, plus que la segmentation est converge.

Le nombre important des agents dans l'image, mène par la suite, à des fusions entre ceux qui travaillent sur la même composante. En effet, tel qu'explique avec le comportement de reproduction.

Un agent 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.

Notre approche suit le modèle de croissance de primitives (région et contour) dans leur évolution de même que le travail de [Djerou 02] qui présente une approche de segmentation par émergence. Concernant les interactions entre agents, notre système, est inspiré des travaux mentionnés dans le chapitre 04.

Au niveau des critères d'évaluation du contour, il est important de tenir en compte en plus les fermetures ainsi que l'utilisation des méthodes plus puissantes pour vérifier et choisir les maxima locaux du contour.

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 résultats obtenus sont encourageants puisque nous avons appliqué notre approche sur différents types d'images et nous avons remarqué que les régions détectées sont satisfaisantes.



Conclusion générale

Dans ce mémoire, nous avons présenté une nouvelle méthode de segmentation basée sur les écosystèmes. Plusieurs notions et concepts ont été abordées : systèmes complexes, écosystèmes, techniques de traitement d'images, vie artificielle....etc. Ainsi, nous avons s'attardé sur les travaux récents dans le traitement d'images par les algorithmes coevolutionnaire. La méthode proposée, est inspirée de ces derniers, elle montre comment à partir des agents réactifs simple utilisant le principe de coevolution compétitive, on peut segmenter une image en niveau de gris.

L'approche adoptée dans ce mémoire est fondée sur les particularités suivantes :

- ✓ Deux types d'agents réactifs existent dans le système qui évoluent dans un environnement commun : l'image, ce qui mène à une compétition entre eux.
- ✓ Les régions marquées et les contours détectés suivent dans leur évolution le principe de l'incrémentalité.
- ✓ Trois types de comportements sont définis pour les agents : perception, reproduction et interaction.
- ✓ Les agents ont une perception locale de leur environnement. (selon leurs voisinages : 8-voisins pour le contour et 4-voisins pour une région).
- ✓ La croissance des régions et contours départ des germes initialisés dans l'image, les agents vont mourir lorsqu'il n'existe pas de primitive à agréger ou bien ils décident de fusionner leurs primitives, donc un seul agent va continuer la segmentation.

Le système est généraliste et ne bénéficie d'aucune connaissance sur les images qu'il segmente. Il est appliqué sur des images de synthèse et d'autres médicales, les résultats obtenus sont satisfaisants et encourageants.

Conclusion générale

Au niveau de l'évaluation des candidats contours, il existe un problème de contours parasites : des contours de petites longueurs qui perturbent le travail des autres agents.

Afin d'améliorer le système, plusieurs perspectives sont envisagées :

- L'utilisation des techniques plus performant pour extraire les maxima locaux du gradient. Ainsi il faut éliminer les contours parasites.
- Amélioration de la méthode pour permettre la segmentation des images couleurs.
- Intégration des algorithmes génétiques afin de profiter des opérations de mutation, croisement...dans la coevolution des agents.



Table des figures

1.1 Schéma générale d'un agent	08
1.2 Agent cognitif	09
1.3 Agent réactif	09
1.4 Exemples de topologies d'automates cellulaires	12
1.5 Exemple de reproduction, croisement, et mutation	14
1.6 Le neurone formel	14
1.7 Exemple de topologies de réseaux de neurones	15
1.8 Emergence du chemin le plus court entre le nid et la nourriture chez les fourmis	17
2.1 Le monde artificiel de SimDelta	27
2.2 Un nid de fourmis tel qu'il est visualisé par MANTA	29
2.3 Une représentation en perspective de PENGI	30
2.4 Swarm - structures récursives	32
2.5 Exemples de formes générées des Jardins des Hasards	33
2.6 Structure du système ``Jardin des Hasards''	34
3.1 Histogramme et palette associés à une image	39
3.2 Schéma d'un système de traitement d'images	40
3.3 Egalisation d'histogramme	42
3.4 Différents types de contours : marche, toit et pointe	43
3.5 gradient et laplacien- cas bidimensionnel-	44
3.6 Orientation du contour	44
3.7 Orientation des axes cas continu et cas discret	44
3.8 Les directions de Freeman	45
3.9 Voisinage d'un pixel	48
3.10 Découpage par quadtree d'une image 8x8	49
3.11 Problème de découpage arbitraire de régions dans la phase de "split"	50
3.12 Principes de la croissance de région et du suivi de contour	51
3.13 Boucle de contrôle du comportement de perception	51
3.14 La pyramide de graphe d'adjacence des régions	53
3.15 Comportement de reproduction	53
3.16 Comportement de diffusion	54
3.17 Exemple de coopération région-contour et région-région pour la fusion des régions	56

3.18 <i>Exemple de résultat</i> pour la détection de régions (l'inspiration des araignées sociales)	57
3.19 <i>Cinq états de geoBoid, utilisés pour les boids de traitement d'images</i>	57
3.20 <i>Traitement d'images par vie artificielle.Exemple de résultat</i>	58
4.1 <i>Evolution d'un agent region à partir d'un germe initial</i>	62
4.2 <i>Evolution d'un agent contour à partir d'un germe initial.</i>	63
4.3 <i>Le mode de sélection des pixels candidats pour la croissance de contour</i>	64
4.4 <i>Principe de segmentation d'une image par coevolution entre agents région et agents contour</i>	66
4.5 <i>Schéma général du système</i>	68
5.1 <i>Processus de coevolution d'agents (contour et région) pour la segmentation d'une image de synthèse</i>	71
5.2 <i>Evolution du nombre de pixels régions et contours en fonction du temps</i>	72
5.3 <i>Processus de segmentation par coevolution d'une image cérébrale</i>	73
5.4 <i>Interface du NetLogo</i>	74
5.5 <i>Ensemble de résultats obtenus à partir du NetLogo</i>	75

Table des matières

Introduction générale	1
Problématique	1
Objectif.....	2
Plan.....	2
Chapitre 01 : Systèmes complexes	4
Introduction	4
1.1 Et les systèmes complexes ?.....	5
1.2 Caractéristiques des systèmes complexes.....	5
1.3 Concepts de base.....	6
1.3.1 Agents et systèmes multi agents.....	6
1.3.1.1 Qu'est-ce qu'un agent ?.....	6
1.3.1.2 Différents types d'agents	8
1.3.1.3 Systèmes Multi-Agents (SMA).....	10
1.3.2 Vie Artificielle.....	11
1.3.3 Automates cellulaires.....	11
1.4 Quelques techniques inspirées de systèmes naturels.....	13
1.4.1 Algorithmes génétiques.....	13
1.4.2 Réseaux neuronaux artificiels.....	14
1.4.3 Intelligence en essaim.....	16
Conclusion	18
Chapitre 02 : Ecosystèmes	19
Introduction	19
2.1 Des systèmes complexes : les écosystèmes.....	20
2.1.1 Notions de base.....	20
2.1.1.1 Ecologie.....	20
2.1.1.2 Qu'est-ce qu'un écosystème ?.....	20
2.1.2 Propriétés caractéristiques.....	22
2.1.3 Exemples de relations biotiques dans les écosystèmes.....	22
2.1.3.1 Compétition.....	22
2.1.3.2 Symbiose (ou mutualisme).....	23
2.1.3.3 Parasitisme.....	23
2.1.3.4 Prédation.....	24
2.1.4 Coevolution.....	24
2.1.4.1 Coevolution compétitive.....	25
2.1.4.2 Coevolution coopérative.....	25
2.1.5 Etapes d'évolution d'écosystèmes.....	25
2.2 Exemples de simulations et de plates formes.....	26
2.2.1 SIMDELTA.....	27
2.2.2 MANTA.....	28
2.2.3 PENGI: Modélisation d'un écosystème abstrait.....	30
2.2.4 Swarm.....	31
2.2.5 Cormas.....	32
2.2.6 Jardins des Hasards	32
2.2.7 ECLPSS.....	34

Conclusion	35
Chapitre 03 : Généralités sur le traitement d'images	36
Introduction	36
3.1 Définition d'une image.....	37
3.2 Acquisition d'une image.....	37
3.3 Caractéristiques d'une image numérique.....	38
3.4.1 Dimension.....	38
3.4.2 Résolution.....	38
3.4.3 Bruit.....	38
3.4.4 Histogramme.....	38
3.4.5 Luminance.....	39
3.4.6 Contraste.....	39
3.4.7 Images à niveaux de gris.....	40
3.4.8 Images en couleurs.....	40
3.4 Système de traitement d'images.....	40
3.5 Pré-traitement d'images.....	41
3.5.1 Modification d'histogramme.....	41
3.5.2 Réduction du bruit par filtrage.....	42
3.6 Extraction de contours.....	43
3.6.1 Approche Gradient.....	44
3.6.2 Approche Laplacien.....	46
3.6.3 Approche par filtrage optimal : Approche de Canny.....	46
3.7 Segmentation en régions.....	47
3.7.1 Accroissement de région.....	48
3.7.2 Méthodes du type "split and merge".....	49
3.8 Approches coopératives en traitement d'images.....	50
3.8.1 Système de Bellet.....	51
3.8.2 Approche de Boucher	52
3.8.3 Approche de Duchesnay	52
3.8.4 Approche de Liu & Tang : Segmentation par « agents migrants ».....	53
3.8.5 Une plate-forme pour la segmentation des images IRM	54
3.8.6 Détection de régions dans des images à niveaux de gris : l'inspiration des araignées sociales	56
3.8.7 Traitement d'images par vie artificielle	57
Conclusion	58
Chapitre 04 : Traitement d'images par écosystèmes	59
Introduction	59
4.1 Travaux récents dans les algorithmes coevolutionnaires.....	60
4.2 Définitions de base.....	61
4.2.1 Gradient.....	61
4.2.2 Seuillage	61
4.3 Description de l'approche.....	61
4.4 Evolution d'agent région.....	62
4.5 Evolution d'agent contour.....	63
4.6 Coevolution d'agents pour la segmentation d'images.....	66
4.7 Comportements de l'agent.....	67

Conclusion	68
Chapitre 05 : Résultats	69
Introduction	69
5.1 Initialisation du système.....	70
5.2 Application aux images de synthèse.....	70
5.3 Application aux images médicales	72
5.4 Utilisation du NetLogo.....	74
5.5 Discussion.....	76
Conclusion	77
Conclusion générale	78
Références	80