

# Segmentation d'images :

## vers une méthodologie

Image segmentation : towards a methodology



### Olivier MONGA

INRIA Sophia-Antipolis, avenue Émile-Hugues, 06565 VALBONNE CEDEX.

Olivier Monga est chargé de recherche à l'INRIA depuis novembre 1986. Il est diplômé de la Maîtrise et du DEA d'informatique de l'Université Scientifique et Médicale de Grenoble. Après une année passée dans le laboratoire de traitement d'image de l'ETCA à Arcueil, il rejoint l'INRIA Rocquencourt, puis l'INRIA Sophia-Antipolis. Il poursuit des recherches en vision par ordinateur plus particulièrement sur la segmentation des images et ses applications.



### Brigitte WROBEL

CRIN LORIA, BP n° 239, 54506 VANDŒUVRE-LES-NANCY CEDEX.

Après une Maîtrise, DESS et DEA d'Informatique obtenus à l'Université de Nancy-I, Brigitte Wrobel est actuellement Assistante à l'Université de Nancy-II. Elle effectue sa recherche au CRIN, au sein de l'équipe Reconnaissance des Formes et Intelligence Artificielle, dans le cadre d'un projet de vision par ordinateur, et s'intéresse plus particulièrement à la vision stéréoscopique appliquée à la navigation d'un robot mobile.

## RÉSUMÉ

Nous présentons une méthode de segmentation d'images pouvant s'appliquer à des problèmes variés tels que : segmentation d'images naturelles monochromatiques ou couleur, segmentation d'images tridimensionnelles sismiques ou scanner. Notre algorithme procède par croissance de régions. Son originalité repose sur l'optimisation d'une suite de critères emboîtés. Nous séparons les critères de segmentation spécifiques à chaque problème particulier de leur stratégie d'utilisation que nous définissons de manière générale. Cette déconnection entre « l'outil mathématique » et « l'outil algorithmique » confère à notre méthode sa généralité. Des structures de données adaptées permettent d'obtenir pour notre méthode une complexité algorithmique faible. Des résultats expérimentaux sont présentés.

### MOTS CLÉS

Segmentation en régions, croissance de régions, graphe valué, structures de données, optimisation de critères.

## SUMMARY

*We present an image segmentation method which may be applied to various tasks such as natural segmentation of monochromatic or color, three dimensional seismic or scanner images. Our algorithm is based on the region growing principle. Its originality lies on optimising the use of a sequence of criteria. We separate the common strategy of using segmentation criteria from the task specific definition of those criteria. This separation between algorithm and mathematical aspects of our method provides for its generality. Experiments results are shown.*

### KEY WORDS

*Region segmentation, region growing, attributed relational graph, data structures, optimal use of criteria.*

## Introduction

Une étape fondamentale dans la plupart des systèmes de vision par ordinateur est d'engendrer une description compacte d'une image, plus exploitable que l'ensemble des pixels.

Une grande variété de techniques dites de « segmentation d'images » permettent d'atteindre cet objectif. Elles sont généralement fondées sur la recherche des discontinuités locales (détection de contours) [18 à 21] ou sur la détection de zones de l'image présentant des caractéristiques d'homogénéité (extraction de régions) [1 à 16]. Ces deux approches sont duales en ce sens qu'une région définit une ligne par son contour et qu'une ligne fermée définit une région. Elles amènent cependant à des algorithmes différents et ne fournissent pas les mêmes résultats.

Nous nous sommes attachés à définir une méthodologie pour la segmentation d'images en régions. L'idée sous-tendue par ce type d'approche est de décrire une image par une partition de son ensemble de pixels possédant certaines propriétés.

Malgré le nombre important de travaux consacrés à ce domaine de la segmentation, peu d'algorithmes vont dans le sens d'une approche générale du problème. En effet, la plupart des algorithmes de segmentation sont conçus pour résoudre un problème spécifique; et le plus souvent, ils ne déconnectent pas les propriétés spécifiques au problème traité des stratégies d'utilisation de ces propriétés.

Nous proposons une méthode de segmentation s'appliquant à des problèmes variés : segmentation d'images naturelles monochromatiques ou couleur, segmentation d'images tridimensionnelles sismiques ou scanner, ... [10 à 15].

Dans un premier temps, nous donnons une définition formelle d'une segmentation : la définition classique utilise un prédicat d'homogénéité auquel nous ajoutons un critère d'optimisation d'une fonction caractérisant la qualité globale d'une segmentation. Cette définition nous conduit à un problème NP complet.

Nous proposons un algorithme résolvant de manière sous-optimale ce problème par optimisation locale : la fonction de qualité globale est comprise comme une fonction monotone et symétrique des fonctions de qualité locale de chacune des régions.

Cet algorithme procède par « croissance de régions » [1 à 4]. Cette technique consiste à regrouper des points voisins (ou des ensembles de points) pour former des régions plus importantes en utilisant des conditions dépendant de propriétés d'homogénéité. Notre algorithme regroupe itérativement des couples de régions adjacentes; à chaque étape, il sélectionne le couple de régions optimisant un critère de similarité.

Les propriétés souhaitées par les régions se décomposent souvent de manière hiérarchique. Cette remarque nous conduit naturellement à utiliser, non pas un seul critère comme dans beaucoup de méthodes, mais sur une suite de critères emboîtés. Chacun de ces critères est associé à une fonction mesurant la qualité locale d'une région. L'optimisation de ces fonctions locales

revient alors à optimiser la fonction de qualité globale de l'image.

La détermination, à chaque étape, des meilleurs couples de régions à fusionner, sous-tend une complexité algorithmique importante. Une suite de conditions sur les propriétés utilisées et une structure de données adaptée permettent cependant d'aboutir à une implantation de coût réduit. Nous déconnectons la stratégie d'utilisation des critères de fusion de la définition de ces critères. Cette séparation entre l'**outil algorithmique** et l'**outil mathématique** permet de résoudre des problèmes de segmentation différents avec un effort de programmation minimale.

La méthode proposée fournit un moyen efficace de spécification d'un problème de segmentation. On définit un tel problème par la donnée de ses critères de fusion.

Cette méthode a été appliquée à des problèmes variés. Nous présentons les critères choisis pour segmenter des images naturelles monochromatiques ou couleur. Les critères sont issus de deux études menées indépendamment au CRIN par Brigitte Wrobel [10, 11] et à l'INRIA par Olivier Monga [12 à 15]. Ils permettent l'obtention de résultats de segmentation satisfaisants en ce sens que les régions correspondent souvent à la projection d'objets ou de parties d'objets présents dans la scène d'où est issue l'image. La qualité des résultats illustre la nécessité d'utiliser une suite de critères de manière optimisée.

Afin d'attester de la valeur des résultats obtenus, nous présentons brièvement une application à la vision stéréoscopique fondée sur l'appariement des régions issues d'une paire d'images [10, 12, 17].

Dans ce papier, nous présentons au chapitre 1 une définition formelle de la segmentation, puis, au chapitre 2, un algorithme sous-optimal, les conditions permettant son implantation efficace et l'implantation réalisée. Nous présentons les différents critères choisis pour les segmentations d'images effectuées au chapitre 3. Nous illustrons l'efficacité de notre méthode pour la vision stéréoscopique et concluons sur les limitations, les avantages et les perspectives du travail réalisé.

## 1. Définition de la segmentation

Le but de cette section est de donner une définition formelle de la segmentation et de soulever les problèmes inhérents à cette définition.

### 1.1. INTRODUCTION

La segmentation a pour objectif l'extraction d'indices visuels dans une image. Elle permet de partitionner une image en ensembles de points appelés régions, homogènes pour une ou plusieurs caractéristiques (intensité lumineuse, texture, ...), et différentes d'au moins une caractéristique des régions voisines.

Elle n'est généralement pas considérée comme un but en soi, mais plutôt comme un moyen efficace de

réduction des données permettant leur analyse par un processus de niveau supérieur. Par exemple, la segmentation est une étape dans le processus d'interprétation de scènes ou dans celui de la vision stéréoscopique.

Que devrait être une bonne segmentation? Il n'existe pas, en fait, de réponse absolue à cette question. La qualité du résultat d'une segmentation est généralement jugée par l'homme en fonction de critères sémantiques difficilement implantables. De plus cette qualité est souvent fonction du traitement ultérieur choisi. Néanmoins, le but généralement poursuivi est de décrire l'image en régions qui correspondent à la projection d'objets ou de parties d'objets présents dans la scène d'où est issue l'image.

Il est cependant souvent possible de donner une définition formelle de la segmentation et de déterminer la fonction « qualité » à optimiser afin d'obtenir une partition satisfaisante.

**1. 2. DÉFINITION FORMELLE**

Une segmentation d'une image I utilisant le prédicat P est généralement définie comme une partition  $S = \{R_1, R_2, \dots, R_n\}$  de I telle que [4] :

- (a)  $I = \bigcup_{i=1, n} R_i$ ;
- (b)  $R_i$  est connexe,  $\forall i \in [1, n]$ ;
- (c)  $P(i) = \text{vrai}$ ,  $\forall i \in [1, n]$ ;
- (d)  $P(R_i \cup R_j) = \text{faux}$ ,  $\forall i \neq j$ ,  $R_i$  et  $R_j$  étant deux ensembles connexes.

La première condition demande simplement que la segmentation soit complète : tout pixel de l'image doit appartenir à une région et une seule. Tout algorithme ne doit pas s'arrêter avant d'avoir traité tous les points de l'image.

La seconde condition demande que les régions soient des ensembles de points connexes. C'est la raison pour laquelle les algorithmes prennent généralement en compte le voisinage des points.

Les critères qui définissent la segmentation sont introduits à la troisième condition.

La quatrième condition exprime la maximalité de chaque région.

Généralement, les techniques de croissance de régions [1 à 3] fusionnent des régions adjacentes vérifiant les conditions (a), (b) et (c) jusqu'à vérifier également la condition (d).

La vérification de ces quatre conditions est une condition nécessaire et suffisante pour qu'une partition d'une image I soit une segmentation. Rien, toutefois, n'implique l'unicité de cette segmentation. En particulier, les résultats de la segmentation dépendent de l'ordre dans lequel les régions sont fusionnées et non uniquement du contenu de l'image comme cela devrait être le cas. Par exemple, une segmentation effectuée sur la transposée d'une image ne donne pas, en général, la transposée de la segmentation de l'image initiale.

Il est cependant possible de diminuer cette indétermination en ajoutant l'optimisation d'une fonction C caractérisant la **qualité d'une segmentation**.

Soit Q une fonction caractérisant la qualité locale d'un élément R de la partition S de l'image I. Par exemple, Q peut être la variance des points de la région R. La fonction C de la qualité globale est souvent une fonction monotone et symétrique de chacune des fonctions de qualité des éléments R de la partition S :

$$C(S) = C(Q(R_1), Q(R_2), \dots, Q(R_n))$$

Par exemple, C peut être la moyenne arithmétique de la qualité de chacune des régions.

Nous ajoutons alors aux conditions (a), (b), (c) et (d) la condition (e) suivante :

(e) De toutes les segmentations S possibles qui vérifient les quatre conditions (a), (b), (c) et (d), nous cherchons la (ou une) segmentation  $S^*$  qui optimise la fonction de qualité C, c'est-à-dire :

$$C(S^*) \leq C(S), \quad \forall S \in S_p(I)$$

où  $S_p(I)$  est l'ensemble de toutes les partitions possibles de I et C une fonction positive.

La fonction C est très souvent additive. Dans le cas d'images 2D, C peut être par exemple choisie comme suit :

$$C(S) = \sum_{i=1, n} \sum_{(k, l) \in R_i} (I(k, l) - M_i)^2$$

avec  $n$  = nombre de régions de la partition

$$M_i = \sum_{(k, l) \in R_i} I(i, j) / \text{Card}(R_i)$$

= moyenne des intensités lumineuses  
des points de la Région  $R_i$

$\text{Card}(R_i)$  = nombre de points de la région  $R_i$ .

Nous pouvons alors remarquer que les propriétés requises pour les régions peuvent généralement être décomposées de façon hiérarchique de manière à simplifier leur optimisation. Une séquence de couples (Prédicat, Qualité) :  $(P_1, Q_1), \dots, (P_n, Q_n)$  apparaît alors plus adaptée pour décrire un processus de segmentation qu'un seul couple (P, Q). Une telle décomposition permet de diviser le problème et de réduire ainsi sa complexité.

Ceci peut être illustré par l'exemple suivant :

Soit un ensemble de pixels dans lequel nous cherchons des angles droits. Nous pouvons, dans un premier temps, utiliser un couple  $(P_1, Q_1)$  permettant de trouver des lignes droites, puis dans un second temps, utiliser un couple  $(P_2, Q_2)$  pour former des angles droits à partir de ces lignes.

Il est clair que la condition (e) n'est pas suffisante pour obtenir une segmentation unique puisque plusieurs segmentations de même valeur minimale pour la fonction de qualité C peuvent exister. Cependant, cette condition nous fournit une définition plus précise de la segmentation.

Il est possible de montrer que le problème d'optimisation est NP complet. Néanmoins, la recherche d'algorithmes sous-optimaux est intéressante.

## 2. Outil algorithmique

Cette section est consacrée à la définition de l'outil algorithmique de la segmentation d'images. Nous présentons la structure générale de notre algorithme de segmentation, puis les conditions à vérifier par les critères de segmentation afin de permettre une implantation algorithmique efficace. L'implantation est présentée en troisième partie où nous décrivons également des structures de données adéquates pour cet algorithme.

### 2.1. STRUCTURE DE L'ALGORITHME

L'idée de base de notre algorithme est d'optimiser la qualité globale  $C$  de la segmentation par fusions successives de régions vérifiant le prédicat  $P$ . Parmi toutes les fusions permises à une étape, seule la meilleure va être réalisée, c'est-à-dire la fusion produisant la région de meilleure qualité locale  $Q$ .

L'algorithme a alors la structure suivante :

$S=I$   
**tant** qu'il existe un couple de régions adjacentes dont la réunion vérifie le prédicat  $P(P(R_i \cup R_j)=\text{vrai})$  **faire**  
 choisir parmi tous les couples  $(R_i, R_j)$  dont la réunion vérifie le prédicat  $P$ , celui pour lequel la qualité  $Q(R_i \cup R_j)$  soit optimale  
 mettre à jour la partition  $S$   
**tantantque**

Cet algorithme est en fait très naturel et permet de s'affranchir de la dépendance du sens d'examen des régions candidates à une fusion. Nous ne définissons pas de sens *a priori* de traitement des données, notre algorithme est essentiellement guidé par les données.

De plus, cette heuristique de regroupement de régions permet d'optimiser la qualité globale de la segmentation  $C(Q(R_1), \dots, Q(R_n))$  par optimisation locale à l'aide de la fonction  $Q$ .

Pour utiliser une séquence de critères  $(P_1, Q_1), \dots, (P_n, Q_n)$  au lieu d'un seul, il suffit d'itérer l'algorithme écrit précédemment.

### 2.2. PROPRIÉTÉS DES CRITÈRES DE SEGMENTATION

Afin de permettre une implantation rapide et efficace de cet algorithme, certaines propriétés doivent être vérifiées par les critères de segmentation. Nous donnons ci-après quelques définitions avant de faire des hypothèses sur les paires (Prédicat, Qualité).

Soit  $E = \{e_1, e_2, \dots, e_1\}$  l'ensemble initial des éléments. Cet ensemble peut être une image ou une segmentation initiale. Nous parlerons plus loin des différentes manières d'obtenir une telle segmentation.

Soit  $\mathcal{P}(E)$  l'ensemble des sous-ensembles de  $E$ .

Soit  $\mathbb{R}$  l'ensemble des réels.

Soit  $k$  le nombre de mesures attachées à un sous-ensemble (donc à une région).

Soit  $A$  une fonction de  $\mathcal{P}(E)$  dans  $\mathbb{R}^k$  qui associe un vecteur de caractéristiques à un sous-ensemble de  $E$ . Par exemple : la surface, la variance, la moyenne, le minimum et le maximum de l'intensité lumineuse sont des caractéristiques attachées à une région de  $E$ .

Soit  $m$  le nombre de mesures attachées à un couple de sous-ensembles de  $E$  (c'est-à-dire à un couple de régions).

Soit  $G$  la fonction de  $\mathcal{P}(E) \times \mathcal{P}(E)$  dans  $\mathbb{R}^m$  qui associe un vecteur de caractéristiques à un couple de sous-ensembles de  $E$ . Par exemple : la valeur moyenne du gradient à la frontière, l'orientation de la frontière, la longueur de la frontière, ...

Soit  $S$  une partition de  $E$ ,  $S = \{E_1, E_2, \dots, E_p\}$ .

Il est possible d'associer à  $S$  l'ensemble des caractéristiques de ses éléments et l'ensemble des caractéristiques de ses paires d'éléments appelés respectivement  $V(S)$  et  $W(S)$  et définis par :

$$V(S) = \{A(E_1), \dots, A(E_p)\}$$

$$W(S) = \{G(E_r, E_s) \text{ où } r, s \in [1, p]\}.$$

Tout triplet  $(S, V(S), W(S))$  définit et caractérise une segmentation de  $E$ .

Définissons à présent la fusion de deux éléments de  $S: E_i$  et  $E_j$ . Cette opération consiste à définir un nouveau triplet :

$$(S, V(S), W(S))_{(i,j)} = (T, V(T), W(T))$$

tel que :

$$T = \{E_1, \dots, E_{i-1}, E_i \cup E_j, E_{i+1}, \dots, E_{j-1}, E_j \cup E_i, \dots, E_{p-1}\}$$

= l'ancien ensemble  $S$  auquel on enlève

les deux sous-ensembles  $E_i$  et  $E_j$

et auquel on ajoute

le nouveau sous-ensemble  $E_i \cup E_j$ .

$$V(T) = \{A(E_1), \dots, A(E_{i-1}), A(E_i \cup E_j), \dots, A(E_{p-1})\}$$

$$W(T) = \{G(E_r, E_s) \text{ où } r \neq i, r \neq j, r \in [1, p], G(E_r, E_i \cup E_j) \dots\}$$

Nous posons les deux conditions suivantes :

$(B_1)$  : La séquence de couples (Prédicat, Qualité) :  $(P_1, Q_1), \dots, (P_n, Q_n)$  est telle que :

$\forall E_i, E_j \in S$ , nous avons :

$$P_i(E_i \cup E_j) = P_i(A(E_i), A(E_j), G(E_i, E_j))$$

$$Q_i(E_i \cup E_j) = Q_i(A(E_i), A(E_j), G(E_i, E_j))$$

Ceci signifie que le prédicat d'homogénéité et la fonction de qualité d'un couple de régions peuvent être calculés directement en fonction des caractéristiques des deux régions concernées.

$(B_2)$  : Nous pouvons trouver deux fonctions  $H$  et  $J$  définies comme suit :

$H$  est une fonction de  $\mathbb{R}^k \times \mathbb{R}^k$  dans  $\mathbb{R}^k$  telle que :

$$\forall S_1, S_2 \in \mathcal{P}(E),$$

$$A(S_1 \cup S_2) = H(A(S_1), A(S_2))$$

Ceci signifie que les caractéristiques de l'union de deux régions peuvent être directement calculées à partir des caractéristiques des deux régions concernées.

$J$  est une fonction de  $\mathbb{R}^m \times \mathbb{R}^m$  dans  $\mathbb{R}^m$  telle que :

$$\forall S_1, S_2 \in \mathcal{P}(E),$$

$$G(S_1, S_2 \cup S_3) = J(G(S_1, S_2), G(S_1, S_3))$$

Ceci signifie que les caractéristiques attachées à la paire  $(S_1, S_2 \cup S_3)$  peuvent être calculée directement à partir de celles des deux paires  $(S_1, S_2)$  et  $(S_1, S_3)$ . Les deux fonctions  $H$  et  $J$  permettent alors de mettre à jour facilement le triplet  $(S, V(S), W(S))$  après une fusion. Donc, si le triplet initial  $(S, V(S), W(S))$  et les fonctions  $H$  et  $J$  sont connus, nous pouvons réaliser une succession de fusions par simple mis à jour du triplet courant.

Les conditions ci-avant citées permettent de réaliser une implantation de faible complexité de notre algorithme de fusion.

Dans beaucoup de problèmes de segmentation, nous pouvons trouver des critères de segmentation qui vérifient ces conditions. Il est généralement aisé de trouver un ensemble de caractéristiques attachées aux prédicats et aux fonctions de qualité qui vérifie la condition  $(B_1)$ . Mais ces caractéristiques ne s'accordent pas toujours avec la seconde condition  $(B_2)$  qui implique l'existence des deux fonctions de mise à jour  $H$  et  $J$  précédemment citées.

La fonction  $H$  permet aux caractéristiques d'une région  $R$  d'être calculées à partir de celles attachées aux régions formant une partition quelconque de la région  $R$  traitée. La fonction  $J$  possède la même propriété relativement aux caractéristiques attachées aux couples de régions.

Pratiquement, lorsqu'une caractéristique ne vérifie pas la condition  $(B_2)$ , nous cherchons à la décomposer en plusieurs caractéristiques qui, elles, la vérifient. D'un point de vue théorique, une telle décomposition peut toujours être trouvée. Dans le pire des cas, on arrive au niveau des points de la région. Nous nous intéressons évidemment à l'utilisation de caractéristiques qui peuvent être calculées à partir d'un nombre réduit de caractéristique vérifiant la condition  $(B_2)$ .

Illustrons ce cas de décomposition pour la caractéristique « variance ». La variance du niveau de gris de la fusion de deux régions  $R_1$  et  $R_2$  ne peut pas être calculée en fonction de la variance de chacune des deux régions concernées. Par contre, elle peut être calculée à partir de la surface, de la somme des niveaux de gris et de la somme des carrés de niveau de gris des deux régions  $R_1$  et  $R_2$ . Le vecteur de caractéristiques de ces trois valeurs est attaché à la fonction  $H$  définie comme suit :

$$H((n_1, s_1, c_1), (n_2, s_2, c_2)) = (n_1 + n_2, s_1 + s_2, c_1 + c_2)$$

De la même façon, par exemple, la matrice de covariance d'un ensemble de points 2D ou 3D peut être facilement calculée à partir des sommes des produits des coordonnées.

Nous pouvons faire les mêmes remarques au sujet des caractéristiques attachées aux couples de régions. Par exemple, la valeur du gradient moyen se décompose en deux caractéristiques : le nombre de points le long de la frontière et la somme des valeurs du gradient.

## 2.3. IMPLANTATION DE L'ALGORITHME

### 2.3.1. Graphe image

Nous venons de présenter les conditions à vérifier par les caractéristiques choisies pour définir la qualité d'une segmentation et permettre une implantation algorithmique efficace. Celle-ci peut être effectuée en représentant un triplet  $(S, V(S), W(S))$  caractérisant une partition de l'image, par un graphe relationnel valué.

A chaque nœud de ce graphe nous associons un élément de la partition  $S$  et son vecteur de caractéristiques, c'est-à-dire une région et ses attributs.

A chaque arc du graphe, nous associons deux éléments de la partition  $S$  et le vecteur d'attributs qui caractérise les relations existant entre les deux nœuds concernés.

En pratique, nous ne conservons que les arcs entre les paires de nœuds qui vérifient certaines propriétés comme la connexité ou le recouvrement [15]. Pour les images naturelles 2D, nous travaillons avec le graphe d'adjacence de la partition. Ceci réduit fortement le nombre de mesures à calculer sur les arcs entre deux nœuds.

### 2.3.2. Structures de données

Il existe différentes structures de données pour représenter un graphe d'adjacence valué. Certaines sont fortement adaptées à ce problème et permettent à l'algorithme d'être efficace.

A chacune des étapes de fusion du processus de segmentation, nous choisissons le meilleur arc par rapport la paire (Prédicat  $P_i$ , Qualité  $Q_i$ ) courante, c'est-à-dire, choisir parmi tous les couples de régions vérifiant le prédicat  $P_i$ , celui qui minimise la qualité  $Q_i$ .

De plus, lors de la réalisation de la fusion de deux nœuds, nous devons mettre à jour tous les attributs des nœuds et des arcs modifiés par cette fusion.

Les structures de données choisies doivent alors essentiellement favoriser les deux traitements suivants :

- accès rapide au meilleur arc;
- mise à jour facile des attributs des nœuds et des arcs modifiés par la fusion.
- accès au meilleur arc

L'accès à un arc peut s'effectuer rapidement en utilisant par exemple un tas [22 à 24] ou un arbre binaire de recherche, comme cela a été effectivement implanté dans nos deux algorithmes de segmentation. Ces deux structures de données sont décrites en Annexe C.

Dans les deux cas, nous plaçons chaque pointeur d'arc vérifiant le prédicat courant  $P_i$ , soit dans le tas, soit dans l'arbre, selon les valeurs croissantes de la qualité  $Q_i$ . Au sommet du tas, ou à la feuille d'une branche située d'un côté de l'arbre, se trouve alors le pointeur sur le couple de régions à fusionner.

A chaque fusion, de nouveaux arcs sont créés; ceux qui vérifient le prédicat courant sont alors ajoutés dans la structure de données utilisée. Nous conservons pour chaque arc, la valeur correcte du coût de la

fonction de qualité locale  $Q_i$ . Nous avons alors le choix, lors de la fusion de deux nœuds  $N_1$  et  $N_2$  :

— soit de lui donner comme résultat le nœud  $N_1$  et de modifier alors le coût de tous les arcs reliés au nouveau nœud  $N_1$ ;

— soit de créer un nouveau nœud  $N_3$  et de créer autant de nouveaux arcs que de nœuds adjacents aux deux anciens nœuds  $N_1$  et  $N_2$ .

Dans les deux cas, la recherche d'un arc de coût minimal dans le tas ou dans l'arbre se poursuit jusqu'à trouver un arc entre deux nœuds qui existent encore.

— mise à jour des attributs

Pour effectuer une mise à jour facile et rapide des attributs des nœuds et des arcs après une fusion, chaque nœud du graphe est associé à un pointeur vers la tête d'une double liste chaînée des arcs à laquelle le nœud est lié.

Chaque arc entre deux nœuds  $N_1$  et  $N_2$  est alors inclut dans les deux listes chaînées issues respectivement des nœuds  $N_1$  et  $N_2$ . Chaque liste d'arcs pour un nœud donné est doublement chaînée : un élément dans cette liste a un pointeur sur l'élément précédent et un pointeur sur l'élément suivant dans la liste.

Ceci signifie que chaque arc ( $N_1, N_2$ ) est relié à quatre pointeurs : deux de ces pointeurs concernent la liste issue de  $N_1$ , les deux autres concernent la liste issue de  $N_2$ .

A chaque nœud, respectivement à chaque arc, est attaché un pointeur sur la liste des attributs du nœud (exemple : le nombre de points de la région, la somme des niveaux de gris, ...), respectivement, la liste des attributs de l'arc (exemple : la longueur de la frontière, ...).

Cette façon de représenter le graphe d'adjacence permet d'effectuer la fusion de deux nœuds  $N_1$  et  $N_2$  simplement en parcourant les deux listes d'arcs, l'une attachée au nœud  $N_1$ , l'autre au nœud  $N_2$ .

La mise à jour des attributs des nœuds et des arcs du graphe est alors réalisée par l'utilisation des deux fonctions  $H$  et  $J$ .

Remarquons que l'utilisation d'une double liste chaînée n'est nécessaire que dans le cas où nous utilisons une stratégie de division de régions, comme cela est réalisé dans [15]. Dans le cas d'une méthode par fusion de régions, la double liste chaînée augmente la place mémoire occupée, mais améliore le temps de traitement.

### 2.3.3. Structure de l'algorithme

Les principales étapes de notre algorithme peuvent être résumées de la façon suivante :

- construire la partition initiale et initialiser les structures de données utilisées.
- lire les fonctions  $H$  et  $J$ .
- lire la séquence de paires (Prédicat, Qualité) :  $(P_1, Q_1), \dots, (P_n, Q_n)$ .
- pour chacune de ces  $n$  paires  $(P_i, Q_i)$  faire
  - mettre dans la structure de données choisie (tas ou arbre) tous les couples de régions adjacentes  $(R_k, R_l)$  tels que  $(P_i(R_k \cup R_l) = \text{vrai})$  et en fonction de la valeur de la qualité locale de la nouvelle région union de  $R_k$  et  $R_l$ , soit  $Q_i(R_k \cup R_l)$ .

- tantque la structure (tas ou arbre) est non vide faire
  - prendre le meilleur arc dans cette structure jusqu'à en trouver un valide (c'est-à-dire situé entre deux nœuds qui existent encore).
  - effectuer la fusion par une mise à jour des nœuds et des arcs à l'aide de deux fonctions  $H$  et  $J$ .

fin tantque  
fin pour

Le résultat de la segmentation se trouve dans les différentes structures de données dont l'exploitation fournit l'image de régions.

## 2.4. COMPLEXITÉ ALGORITHMIQUE

Étudions brièvement la complexité de l'algorithme proposé :

Soit  $N$  le nombre de nœuds du graphe initial.

Soit  $A$  le nombre d'arcs le nombre d'arcs du graphe initial.

Chaque nœud du graphe est connecté à une moyenne de  $V = 2A/N$  nœuds. Cette valeur reste sensiblement identique pendant toute la durée du processus de segmentation.

Soit  $F$  le nombre de fusions réalisées pendant le traitement.

Soit  $T_n$  le nombre d'opérations que doit effectuer la fonction  $H$  de mise à jour des attributs des nœuds.

Soit  $T_a$  le nombre d'opérations que doit effectuer la fonction  $J$  de mise à jour des attributs des arcs.

Soit  $T_x$  le nombre d'opérations demandées pour réaliser l'évaluation d'un prédicat et de la fonction de qualité locale qui lui est associée.

Soit  $M$  le nombre moyen de nœuds adjacents à chacun des nœuds qui vérifient le prédicat donné; c'est-à-dire le nombre de fusions potentielles de chaque nœud avec l'un de ses voisins.

Le temps de calcul nécessaire pour mettre à jour la structure de données des attributs des nœuds et des arcs du graphe est :

$$F * T_n + F * T_a * V$$

Après la fusion de deux nœuds, il nous faut mettre à jour la structure de données contenant tous les couples de nœuds candidats à une fusion; c'est-à-dire, après une fusion, nous insérons dans cette structure une moyenne de  $V$  nouveaux arcs et calculons donc  $V$  nouvelles valeurs du prédicat courant et de la fonction de qualité locale. Le nombre maximal d'éléments dans cette structure est d'environ  $V * M$ .

Ceci nous permet alors de calculer la complexité du traitement de mise à jour des structures de données de la façon suivante :

$$F * T_x * V + M * O(V * M * \text{LOG}(V * M))$$

Nous en déduisons alors la complexité de tout l'algorithme :

$$O(F * T_n + F * T_a * V + F * T_x * V + M^2 * O(V * \text{LOG}(V * M)))$$

L'algorithme de segmentation présenté est guidé par les données, c'est-à-dire par la structure même de

l'image. Cette méthode sous-tend une complexité algorithmique très importante, mais nous venons de montrer que cette complexité est en réalité très faible sous condition d'utiliser des structures de données adaptées au problème considéré.

## 2.5. CONCLUSION

Nous venons de définir une stratégie de segmentation comme :

- un graphe initial issu d'une partition initiale de l'image et les caractéristiques correspondantes attachées aux nœuds et aux arcs de ce graphe.
- deux fonctions de mise à jour des attributs des nœuds et des attributs des arcs;
- une séquence de paires (Prédicat de fusion, Qualité de fusion) qui définit ainsi la stratégie de fusion.

La complexité de cet algorithme ne dépend pas de la taille globale du graphe initial mais de la « densité d'arc » de ce graphe. De ce fait, nous nous intéressons à limiter le nombre de relations entre les nœuds du graphe. De ce fait, nous limitons le nombre de relations entre les nœuds du graphe utilisés pour la segmentation.

Il est coûteux d'utiliser cet algorithme sur l'image originale : le graphe est alors composé de tous les points de l'image comme nœuds et de tous les arcs entre les points en relation (par exemple connexité ou recouvrement [13, 15]).

Il est donc préférable d'effectuer une partition initiale de l'image originale. Ceci peut être réalisé par la fonction Merge de Pavlidis [5], ou par un traitement procédant par balayage séquentiel de l'image. Le but est d'avoir des critères de regroupement très peu tolérants afin de construire beaucoup de petites régions mais les plus homogènes possible.

L'outil algorithmique étant ainsi défini, nous pouvons passer à l'étude de la segmentation d'images naturelles et définir, pour chaque type d'image, les outils mathématiques à utiliser dans l'algorithme.

## 3. Outils mathématiques pour la segmentation d'images naturelles

Nous présentons dans cette section les différents critères mathématiques choisis selon le type d'image traitée : image 2D monochrome, image 2D. Ces critères sont ensuite utilisés par l'algorithme décrit au chapitre précédent pour effectuer les segmentations demandées.

### 3.1. INTRODUCTION

Beaucoup d'études ont été menées ces dernières années dans le domaine de la segmentation d'images naturelles 2D [1 à 16] afin d'en comprendre automatiquement le contenu (analyse de scènes par exemple). Une revue de ces différentes techniques peut être trouvée dans [1].

L'objectif de la segmentation est alors de déterminer la projection des différents objets qui composent la scène observée. Une image naturelle dépend des diverses caractéristiques d'une scène 3D, telles que la forme des objets, les matériaux des objets, l'éclairage, l'orientation de la caméra, ... La conjonction de tous ces paramètres fournit cependant une image ne contenant qu'un seul type d'information : le **pixel**, c'est-à-dire une position dans l'image à laquelle est associée une **intensité lumineuse**. Il est évidemment assez difficile de séparer l'influence de ces différents facteurs dans l'image, dans le but de reconnaître les projections des objets.

La qualité de la segmentation est maximale si chaque région correspond à un objet et réciproquement.

La propriété d'homogénéité d'un objet n'est généralement pas connue. Par exemple, il est difficile de caractériser le critère d'homogénéité d'un mur sur lequel l'éclairage est dégradé (l'intensité lumineuse varie du blanc au noir), ou celui du feuillage d'un arbre.

La segmentation d'images fournit de bons résultats sur des images où les objets visualisés sont simples ou quand la lumière peut être contrôlée (vrac industriel par exemple). Cependant, les résultats obtenus à partir d'images naturelles sont généralement peu satisfaisants; les algorithmes mis en œuvre ne s'appuient souvent que sur une seule propriété d'homogénéité sans en optimiser l'utilisation.

Une solution pour obtenir de meilleurs résultats est d'utiliser de façon optimale **plusieurs propriétés d'homogénéité**. Nous proposons une méthode pour résoudre ce problème en appliquant l'algorithme sous-optimal présenté au chapitre précédent.

Pour être capable de segmenter une image, nous devons tout d'abord décomposer de **manière hiérarchique** les propriétés que doivent vérifier les régions pour obtenir une partition de bonne qualité.

Puis nous devons associer à chacune de ces propriétés d'homogénéité un **prédicat**, une fonction caractérisant la **qualité globale** de la partition de l'image, de laquelle est déduite une fonction estimant la **qualité locale** de chaque région. Nous déterminons également les **attributs** des régions et des couples de régions avec lesquels sont calculés le prédicat d'homogénéité et les fonctions de qualité.

Sous ces conditions, nous pouvons appliquer l'algorithme décrit au chapitre précédent.

Ci-après nous nous intéressons à la segmentation d'images monochromatiques, puis étendons notre approche aux images 2D couleur. Cette méthode peut aussi être appliquée pour des images 3D [15].

### 3.2. SEGMENTATION D'IMAGES 2D MONOCHROMATIQUES

#### 3.2.1. Propriétés des régions

Nous supposons qu'une région est un ensemble de points connexes tel que :

(a) La somme des gradients calculée sur les points de la région est faible, c'est-à-dire qu'aucun contour ne traverse la région.

(b) La région peut être décomposée en petites zones de faible variance de niveau de gris.

(c) Chacune de ces zones de faible variance peut être décomposée en zones de points de niveau de gris quasi identiques.

Ces trois propriétés vont servir à définir la séquence de paires (Prédicat de fusion, Qualité de fusion) à utiliser dans l'algorithme.

3.2.2. Séquence de (Prédicat de fusion, Qualité de fusion) à utiliser

Nous devons trouver une séquence de paires (Prédicat, Qualité) :  $(P_1, Q_1), \dots, (P_n, Q_n)$  caractérisant la segmentation attendue. Il est évident que le premier prédicat  $P_1$  de la séquence travaille au niveau de l'image initiale, et donc crée les « zones » les plus petites de l'image. Le second prédicat  $P_1$  travaille sur les régions obtenues à l'étape précédente par le prédicat  $P_2$ , etc.

Ceci conduit naturellement à définir le premier couple  $(P_1, Q_1)$  en fonction de la propriété d'homogénéité que doivent vérifier les plus petites « zones » de l'image, c'est-à-dire la condition (c).

Pour les mêmes raisons, le couple  $(P_2, Q_2)$  est défini en fonction de la condition (b), et le couple  $(P_3, Q_3)$  en fonction de la condition (a).

Ce qui donne la séquence de paires  $(P_i, Q_i)$  suivante : Soit  $I$  une image, et  $I(k, l)$  un point de cette image.

Soit  $S = \{R_1, R_2, \dots, R_n\}$  une partition de  $I$ .

**Prédicat  $P_1$ , fonction globale  $C_1$ , fonction locale  $Q_1$**  [respect de la condition (c)]

Prédicat :

$$P_1(R_i) = [|\text{Max}_i - \text{Min}_i| < s_1]$$

où  $s_1$  est un seuil,  $\text{Max}_i$  et  $\text{Min}_i$  sont respectivement le niveau de gris maximal et minimal des points appartenant à la région  $R_i$ .

Qualité globale :

$$C_1(S) = \sum_{i=1, n} \sum_{k, l \in R_i} ((I(k, l) - \text{Max}_i)^2 - (I(k, l) - \text{Min}_i)^2)$$

Qualité locale :

$$Q_1(R_i) = \text{Max}_i - \text{Min}_i$$

**Prédicat  $P_2$ , fonction globale  $C_2$ , fonction locale  $Q_2$**  [respect de la condition (b)]

Prédicat :

$$P_2(R_i) = [V(R_i) < S_2]$$

où  $S_2$  est un seuil,  $V(R_i)$  est la variance de niveau de gris des points de la région  $R_i$ .

Qualité globale :

$$C_2(S) = \sum_{k, l \in I} (I(k, l) - M(k, l))^2$$

où  $M(k, l)$  est la moyenne des niveaux de gris des points de la région à laquelle appartient le point  $(k, l)$ .

Qualité locale :

$$Q_2(R_i) = V(R_i)$$

**Prédicat  $P_3$  fonction globale  $C_3$ , fonction locale  $Q_3$**  [respect de la condition (a)]

Prédicat :

Soient  $R_1$  et  $R_2$  deux ensembles adjacents de points connexes.

Soit  $F(R_1, R_2)$  l'ensemble des points de deux régions de part et d'autre de la frontière comme l'illustre la figure 1.

1																			
1	1																		
1	1	1	1	1	1	1	2	2	2	2									
1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2
1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2
1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2
1	1		1	1	1	2	2	2											
1					1	2													
1																			

Fig. 1. - Les paires des points extrémités des petits segments définissent la frontière entre la région « 1 » et la région « 2 », soit  $F(\langle 1 \rangle, \langle 2 \rangle)$ .

Soit  $D(R_1, R_2)$  la valeur moyenne du gradient entre les deux régions  $R_1$  et  $R_2$ , c'est-à-dire :

$$D(R_1, R_2) = \left( \sum_{((i, j), (k, l)) \in F(R_1, R_2)} |I(i, j) - I(k, l)| \right) / \text{Card}(F(R_1, R_2))$$

où  $\text{Card}(F(R_1, R_2))$  représente le nombre de points de cet ensemble.

Pendant le processus de regroupement de régions :

$$P_3(R_1 \cup R_2) = [D(R_1, R_2) < S_3]$$

où  $S_3$  est un seuil.

Qualité globale :

$$C_3(S) = \sum_{((i, j), (k, l)) \in X_s(I)} |I(i, j) - I(k, l)|$$

où  $X_s(I)$  est l'ensemble des couples de points connexes de  $I$  appartenant à une même région et ce, pour toutes les régions de la partition  $S$ .

Qualité locale :

$$Q_3(R_1 \cup R_2) = D(R_1, R_2)$$

3.2.3. Vérification des conditions pour une implantation efficace.

Nous avons défini au paragraphe 2.2 les conditions  $(B_1)$  et  $(B_2)$  que doivent vérifier les prédicats de segmentation afin de permettre une implantation efficace et de pouvoir utiliser ainsi notre algorithme sous-optimal.

Rappelons que la condition  $(B_1)$  demande que le prédicat d'homogénéité et la fonction de qualité d'un couple de régions soient calculés directement en fonction des attributs des deux régions concernées. La condition  $(B_2)$  implique l'existence de deux fonctions  $H$  et  $J$  qui, après une fusion, effectuent la mise à jour des attributs des nœuds et des arcs en fonction seulement des attributs des nœuds et des arcs concernés.

Montrons à présent que les trois paires  $(P_i, Q_i)$  précédemment définies vérifient ces deux conditions.

**Condition  $(B_1)$**

Soit  $A(R_i) = (f_{1i}, f_{2i}, f_{3i}, f_{4i}, f_{5i})$  le vecteur d'attributs de la région  $R_i$  de la partition  $S$ .

Les attributs d'une région sont définis de la manière suivante :

- $f_{1i}$  = nombre de points de la région;
- $f_{2i} = \text{MAX}_{(k, l) \in R_i} (I(k, l))$  : niveau de gris maximal;
- $f_{3i} = \text{MIN}_{(k, l) \in R_i} (I(k, l))$  : niveau de gris minimal;
- $f_{4i} = \sum_{(k, l) \in R_i} (I(k, l))$  : somme des niveaux de gris des points;
- $f_{5i} = \sum_{(k, l) \in R_i} (I(k, l))^2$  : somme des carrés des niveaux de gris des points.

Soit  $G(R_1, R_2) = (a_{1ij}, a_{2ij})$  le vecteur d'attributs de l'arc entre les deux régions  $R_1$  et  $R_2$ .

Les attributs d'un arc sont définis comme suit :  
 $a_{1ij} = \text{Card}(F(R_1, R_2))$  : nombre de paires de points le long de la frontière (i. e. longueur de la frontière);

$$a_{2ij} = \sum_{((i, j), (k, l)) \in F(R_1, R_2)} |I(i, j) - I(k, l)|$$

Les prédicats d'homogénéité et les fonctions de qualité locale s'expriment alors par les équations suivantes :

$$\begin{aligned} P_1(R_i) &= [(f_{2i} - f_{3i}) < S_1] \\ Q_1(R_i) &= f_{2i} - f_{3i} \\ P_2(R_i) &= [(f_{5i}/f_{1i}) - (f_{4i}/f_{1i})^2 < S_2] \\ Q_2(R_i) &= (f_{5i}/f_{1i}) - (f_{4i}/f_{1i})^2 \\ P_3(R_i) &= [(a_{2ij}/a_{1ij}) < S_3] \\ Q_3(R_i) &= a_{2ij}/a_{1ij} \end{aligned}$$

Les prédicats et les fonctions de qualité locale se calculent donc uniquement à partir des attributs des nœuds et des arcs comme le demande la condition  $(B_1)$ .

**Condition  $(B_2)$**

Soit  $H$  la fonction de  $\mathbb{R}^5 \times \mathbb{R}^5$  dans  $\mathbb{R}^5$  définie par :

$$\begin{aligned} H((f_1, f_2, f_3, f_4, f_5)(g_1, g_2, g_3, g_4, g_5)) \\ = (f_1 + g_1, \text{MAX}(f_2, g_2), \\ \text{MIN}(f_3, g_3), f_4 + g_4, f_5 + g_5) \end{aligned}$$

Il est alors aisé de vérifier que :

$$A(R_i \cup R_j) = H(A(R_i), A(R_j))$$

Soit  $J$  la fonction de  $\mathbb{R}^2 \times \mathbb{R}^2$  dans  $\mathbb{R}^2$  définie par :

$$J((a_1, a_2)(b_1, b_2)) = (a_1 + b_1, a_2 + b_2)$$

Et nous obtenons alors facilement :

$$G(R_1, R_2 \cup R_3) = J(G(R_1, R_2), G(R_2, R_3))$$

Les deux fonctions  $H$  et  $J$  de mise à jour s'écrivent uniquement à partir des attributs des nœuds et des arcs ainsi que le demande la condition  $(B_2)$ .

**3.2.4. Partition initiale**

Nous avons montré, paragraphe 2.5, qu'il est coûteux d'appliquer notre algorithme de segmentation sur l'image originale; il s'avère préférable d'effectuer une segmentation initiale.

Cette première partition est réalisée par un algorithme classique de regroupement de régions comme la procédure Merge de Pavlidis [5] ou par un balayage séquentiel de l'image. Ces deux procédures sont décrites en Annexe D. L'objectif est de construire une partition initiale et non une segmentation finale, c'est-à-dire d'avoir des critères de regroupement très peu tolérants afin d'obtenir beaucoup de petites régions mais les plus homogènes possible.

Le résultat de l'algorithme utilisé dépend du sens de parcours des données. Le risque est alors d'obtenir une partition initiale dont les régions sont fortement influencées par ce sens d'examen. En réduisant la taille des régions souhaitées, nous minimisons ces effets et permettons à notre algorithme sous-optimal de fusion, uniquement guidé par les données, d'être efficace.

Les critères de segmentation utilisés portent essentiellement sur les attributs des régions tels que : la moyenne des niveaux de gris, la valeur maximale et minimale des niveaux de gris des points des régions en cours de construction.

**3.2.5. Coopération régions-contours**

Les résultats obtenus par l'utilisation des critères de segmentation définis au paragraphe 3.2.2, peuvent être parfois améliorés par l'utilisation d'une image de points de contraste provenant d'un extracteur de contours [11]. Le rôle de ces points de contraste est de contrôler le processus de fusion.

Ceci se traduit par une autre définition de la fonction  $D$  caractérisant la force d'une frontière entre deux régions, utilisée dans l'évaluation du prédicat  $P_3$ . Cette fonction  $D$  est calculée en fonction du nombre de points de contraste, de la longueur de la frontière et de la différence de niveau de gris des couples de points situés de chaque côté de la frontière. Ce qui s'écrit de la façon suivante :

$$\begin{aligned} D(R_1, R_2) \\ = \text{soit} \left( \frac{\sum_{((i, j), (k, l)) \in F(R_1, R_2)} |I(i, j) - I(k, l)|}{(N_f(R_1, R_2) - N_{bc}(R_1, R_2))} \right) \\ \text{si } N_f(R_1, R_2) \neq N_{bc}(R_1, R_2) \\ = \text{soit } \text{INFINI} \quad \text{si } N_f(R_1, R_2) = N_{bc}(R_1, R_2) \end{aligned}$$

où  $R_1$  et  $R_2$  sont deux ensembles adjacents de points connexes.

$F(R_1, R_2)$  est l'ensemble des points de deux régions de part et d'autre de la frontière.

$N_f(R_1, R_2)$  représente le nombre de points le long de la frontière entre  $R_1$  et  $R_2$ .

$N_{bc}(R_1, R_2)$  représente le nombre de points de contraste précédemment détectés le long de la frontière entre  $R_1$  et  $R_2$ .

Dans le cas où le nombre de points frontières est égal au nombre de points de contraste détectés, cela signifie que le contour ainsi trouvé a une très forte « force » et ne doit être supprimé en aucun cas. La valeur de la fonction  $D$  vaut alors théoriquement  $+\infty$ . Pratiquement, nous la fixons à une constante de très grande valeur, appelée ici **INFINI**.

Le prédicat  $P_3$  ne change pas :

$$P_3(R_1 \cup R_2) = [D(R_1, R_2) < S_3]$$

La qualité globale a pour valeur :

$$C_3(S) = \sum_{\substack{(i,j), (k,l) \in X_s(I) \\ \text{avec } (i,j) \notin PC \\ \text{et } (k,l) \notin PC}} |I(i,j) - I(k,l)| + \sum_{\substack{(i,j), (k,l) \in X_s(I) \\ \text{avec } (i,j) \in PC \\ \text{INFINI}}} |I(i,j) - I(k,l)|$$

où  $PC$  est l'ensemble des points de contraste dans l'image  $I$ .

La qualité locale s'écrit :

$$Q_3(R_1 \cup R_2) = D(R_1, R_2)$$

### 3.2.6. Conclusion

Nous venons de spécifier notre problème de segmentation par les trois couples (Prédicat, Qualité) :  $(P_1, Q_1)$ ,  $(P_2, Q_2)$  et  $(P_3, Q_3)$ . Nous avons ensuite déterminé les attributs associés aux régions : nombre de points, niveaux de gris maximal et minimal, somme des niveaux de gris, somme des carrés des niveaux de gris, et les attributs associés aux arcs : longueur de la frontière, somme des gradients le long de la frontière et, éventuellement, nombre de points de contraste situés sur la frontière.

Nous effectuons la partition de l'image par un algorithme classique de regroupement de régions. Nous créons ensuite un graphe d'adjacence afin d'appliquer la stratégie de fusion définie par les trois couples  $(P_1, Q_1)$ ,  $(P_2, Q_2)$  et  $(P_3, Q_3)$ .

La première étape du processus fournit une partition de l'image en régions de même luminance. La seconde construit des régions de luminance similaire mais en tenant compte de l'influence du bruit. La troisième étape autorise la fusion de régions de luminance quelque peu différentes si elles ont une frontière de faible gradient.

Nous ajoutons une dernière étape de suppression des trop petites régions dont les points sont alors absorbés par les régions voisines.

Les résultats de ce type de segmentation sont présentés en Annexe A.

## 3.3. SEGMENTATION D'IMAGES 2D COULEUR

### 3.3.1. Propriétés des régions

La segmentation d'images couleur est une autre application possible de notre algorithme. Pour ce faire, nous allons tout d'abord définir les propriétés globales que doit posséder une telle segmentation, en déduire les fonctions de qualité globale, puis la succession de couples (Prédicat de fusion, qualité locale de fusion) à utiliser, et enfin déterminer les attributs à calculer sur les nœuds et sur les arcs ainsi que les deux fonctions  $H$  et  $J$  de mise à jour de ces attributs.

Nous montrons ci-après l'extension de notre approche de la segmentation d'images monochromatiques à la segmentation d'images couleur.

La couleur d'un point d'une image couleur est généralement donnée sous forme de trois valeurs correspondant respectivement à l'intensité lumineuse rouge (R), verte (V) et bleue (B). Par des transformations linéaires et non linéaires, nous pouvons extraire de ces trois valeurs d'autres caractéristiques telles que l'intensité (D), la saturation (S) et la teinte (H).

Le problème est de trouver les caractéristiques adaptées à la segmentation d'images. A ce sujet, Ohta [7] a proposé d'appliquer un algorithme récursif de seuillage, selon l'algorithme de Ohlander [8], et d'utiliser la transformée de Karhunen-Loeve (K-L) sur les caractéristiques des images couleur; mais, du fait du coût en temps machine de ce type d'algorithme, Ohta a préféré travailler avec des caractéristiques sous-optimales dérivées de la transformée de Karhunen-Loeve.

Notre propos est alors d'appliquer notre méthode de segmentation pour utiliser au mieux ce critère.

Les régions d'une image couleur doivent posséder les mêmes qualités (a), (b) et (c) que celles des régions d'une image monochrome (voir § 3.2.1) généralisées au traitement de la couleur, auxquelles nous ajoutons une qualité supplémentaire caractérisant « l'homogénéité de la couleur » obtenue par la transformée de Karhunen-Loeve.

Pour estimer cette qualité, nous déterminons la plus grande valeur propre de la matrice de covariance centrée (normalisée par la surface de la région). Le vecteur propre associé à cette plus grande valeur propre définit la direction la plus discriminante des points de la région concernée.

La valeur propre est la somme des carrés des projections de tous les points (R, V, B) de la région. Si nous divisons cette valeur propre par le nombre de points de la région (la surface), nous obtenons la variance des projections de tous les points selon la direction la plus discriminante.

Cette qualité nous permet de définir une quatrième étape lors de la fusion des régions, et de fusionner ainsi les régions de couleur similaire.

### 3.3.2. Séquence de (Prédicat de fusion, Qualité de fusion) à utiliser

Nous venons de définir les quatre propriétés que doivent vérifier les régions d'une segmentation d'une image couleur. Nous pouvons à présent déterminer la séquence des quatre couples (Prédicat, Qualité) à utiliser dans notre algorithme.

Soit  $I$  une image et  $I_Y(k, l)$  la luminance d'un point de cette image pour la composante couleur  $Y$ ,  $Y$  a pour valeur rouge (R), vert (V) ou bleu (B).

Soit  $S = \{R_1, \dots, R_n\}$  une partition de l'image  $I$ .

$$C_1(S) = \sum_{i=1, n} \sum_{(k, l) \in R_i} \sum_{Y=R, V, B} ((I_Y(k, l) - MA_Y(R_i))^2 + (I_Y(k, l) - MI_Y(R_i))^2)$$

**Qualité globale  $C_1$ , qualité locale  $Q_1$ , prédicat  $P_1$**  pour définir des régions où les trois composantes rouge, verte et bleue sont quasi identiques.

où  $MA_Y(R_i)$  et  $MI_Y(R_i)$  sont respectivement la valeur maximale et minimale des points de la région  $R_i$  pour la couleur  $Y$ .

$$Q_1(R_i) = \text{MAX}_{Y=R, V, B} (MA_Y(R_i) - MI_Y(R_i))$$

$$P_1(R_i) = [Q_1(R_i) < s_1]$$

où  $s_1$  est un seuil.

**Qualité globale  $C_2$ , qualité locale  $Q_2$ , prédicat  $P_2$**  pour définir des régions où la variance sur les trois composantes rouge, verte et bleue est faible.

$$C_2(S) = \sum_{(i, j) \in I} \sum_{Y=R, V, B} (I_Y(i, j) - M_Y(i, j))^2$$

où  $M_Y(i, j)$  est la moyenne des composantes couleur  $Y$  des points de la région à laquelle le point  $(i, j)$  appartient.

$$Q_2(R_i) = \sum_{Y=R, V, B} V_Y(R_i)$$

où  $V_Y(R_i)$  est la variance de la région  $R_i$  pour la composante couleur  $Y$ .

$$P_2(R_i) = [Q_2(R_i) < S_2]$$

où  $S_2$  est un seuil.

**Qualité globale  $C_3$ , qualité locale  $Q_3$ , prédicat  $P_3$**  pour définir des régions ne contenant aucune ligne de fort gradient, le gradient est calculé pour chacune des trois composantes couleur.

$$C_3(S) = \sum_{(i, j), (k, l) \in X_S(I)} \sum_{Y=R, V, B} |I_Y(i, j) - I_Y(k, l)|$$

où  $X_S$  est l'ensemble des points connexes de  $I$  appartenant à une même région, et ce, pour toutes les régions de la partition  $S$ .

$$Q_3(R_i \cup R_j) = \sum_{Y=R, V, B} D_Y(R_i, R_j)$$

où  $D_Y(R_i, R_j)$  représente la valeur moyenne du gradient entre deux régions  $R_i$  et  $R_j$  pour la composante couleur  $Y$ .

$$P_3(R_i \cup R_j) = [Q_3(R_i \cup R_j) < S_3]$$

où  $S_3$  est un seuil.

**Qualité globale  $C_4$ , qualité locale  $Q_4$ , prédicat  $P_4$**  pour définir des régions de couleur identique.

$$C_4(S) = \sum_{i=1, n} Q_4(R_i) - A(R_i)$$

où  $A(R_i)$  représente le nombre de points de la région  $R_i$ .

$$Q_4(R_i) = \text{MAX}(eg_1, eg_2, eg_3) / A(R_i)$$

où  $eg_1, eg_2$  et  $eg_3$  sont les trois valeurs propres de la matrice  $3 \times 3$  ( $a_{ij}$ ) définie comme suit :

$$a_{ij} = \sum_{(k, l) \in R_i} (I_i(k, l) * I_j(k, l)) - \left( \sum_{(k, l) \in R_i} I_i(k, l) \right) * \left( \sum_{(k, l) \in R_i} I_j(k, l) \right) / A(R_i)$$

avec la convention suivante :

$$I_1 = I_R, \quad I_2 = I_V \quad \text{et} \quad I_3 = I_B$$

$$a_{ij} = S_{ij}(R_i) - S_i(R_i) * S_j(R_i) / A(R_i)$$

avec

$$S_1 = S_R, \quad S_2 = S_V$$

et

$$S_3 = S_B, \quad S_{12} = S_{RV}, \quad \dots$$

et

$$S_R(R_i) = \sum_{(i, j) \in R_i} I_R(i, j)$$

$$S_{RV}(R_i) = \sum_{(i, j) \in R_i} I_R(i, j) * \sum_{(i, j) \in R_i} I_V(i, j)$$

$S_V, S_B, S_{RR}, S_{VV}, S_{BB}, S_{VB}$  et  $S_{RB}$  sont définis de la même façon.

$$P_4(R_i) = [Q_4(R_i) < s_4]$$

où  $s_4$  est un seuil.

Ceci permet de caractériser la segmentation d'une image couleur. Nous sommes à présent en mesure de déterminer les attributs des nœuds et des arcs représentant le graphe image.

### 3.3.3. Attributs des nœuds et des arcs du graphe image

Notons  $I_R, I_V$  et  $I_B$  les images correspondant respectivement aux images de luminance rouge (R), verte (V) et bleue (B).

#### Attributs de la région $R_i$

- le nombre de points de la région  $R_i$  :  $A(R_i)$ ;
- la moyenne des luminances des points pour chacune des trois composantes couleur :  $L_R(R_i), L_V(R_i)$  et  $L_B(R_i)$ ;
- la variance de la luminance des points pour chacune des trois composantes couleur :  $V_R(R_i), V_V(R_i)$  et  $V_B(R_i)$ ;

- la luminance maximale des points pour chacune des composantes couleur :  $MA_R(R_i)$ ,  $MA_V(R_i)$  et  $MA_B(R_i)$ ;
- la luminance minimale des points pour chacune des composantes couleur :  $MI_R(R_i)$ ,  $MI_V(R_i)$  et  $MI_B(R_i)$ ;
- les sommes nous permettant de calculer la matrice de covariance, et ainsi d'en déterminer les valeurs propres :

$$S_R(R_i) = \sum_{(i,j) \in R_i} I_R(i,j)$$

$$S_{RV}(R_i) = \sum_{(i,j) \in R_i} I_R(i,j) * \sum_{(i,j) \in R_i} I_V(i,j)$$

où  $S_V$ ,  $S_B$ ,  $S_{RR}$ ,  $S_{VV}$ ,  $S_{BB}$ ,  $S_{VB}$  et  $S_{RB}$  sont définis de la même façon.

*Attributs d'un arc entre deux régions  $R_1$  et  $R_2$  connexes*

- le nombre de couples de points définissant la frontière :  $F(R_1, R_2)$ ;
- les valeurs moyennes du gradient entre les deux régions pour chacune des composantes couleur :  $D_R(R_1, R_2)$ ,  $D_V(R_1, R_2)$  et  $D_B(R_1, R_2)$ .

La démonstration pour montrer que la séquence  $(P_1, Q_1), \dots, (P_4, Q_4)$  vérifient les deux conditions  $(B_1)$  et  $(B_2)$  est similaire à celle présentée au paragraphe 3.2.3 dans le cas de segmentation d'images monochromatiques. Les deux fonctions  $H$  et  $J$  de mise à jour des attributs des nœuds et des arcs du graphe image après chaque fusion sont évidentes.

### 3.3.4. Conclusion

Les critères de segmentation d'images monochromatiques se généralisent facilement pour le traitement d'images couleur. De plus, nous avons montré comment utiliser la transformée de Karhunen-Loeve de manière optimale.

La partition initiale de l'image couleur est obtenue, elle aussi, par une généralisation des critères utilisés dans le cas d'images monochromatiques.

Les résultats de ce type de segmentation sont présentés en Annexe A.

## 3.4. APPLICATION A LA VISION STÉRÉOSCOPIQUE

### 3.4.1. Introduction

La vision stéréoscopique est un moyen efficace de déterminer la géométrie tridimensionnelle d'une scène avec l'avantage de ne pas interférer avec le milieu comme le font certaines approches dites actives : utilisation de laser ou projection de plans lumineux.

Le principe de base de la stéréovision est de comparer deux images d'une même scène prises sous deux angles légèrement différents. La première étape consiste à apparier des entités visuelles homologues des deux images, représentant les projections de la même entité physique de la scène. L'étape suivante est la reconstruction de la scène : après calibration du banc stéréoscopique [31], on calcule pour chaque paire de primitives homologues, les coordonnées tridimensionnelles de l'entité physique correspondante.

Il existe de nombreuses approches pour résoudre le problème de l'appariement, les primitives généralement manipulées sont des points invariants (points de Moravec [29] ou points de contraste [30] ou des segments de contours [26 à 28]). Notre idée est d'utiliser des primitives plus riches afin d'obtenir une carte tridimensionnelle plus dense, et nous avons choisi la primitive **région**, contenant une information de surface non présente dans les autres primitives.

### 3.4.2. Mise en correspondance des régions

Le résultat de la segmentation de deux images stéréoscopiques par notre méthode est représentée par deux graphes images respectivement associés aux deux partitions en régions. Il est à noter que les deux segmentations sont indépendantes l'une de l'autre.

L'idée de base de notre méthode de stéréovision est d'apparier ces deux graphes.

Soient deux graphes valués  $G_1$  et  $G_2$ .

$$G_1 = [X_1, U_1, AB_1]$$

où  $X_1$  est l'ensemble des nœuds du graphe,  $U_1$  est l'ensemble des couples de nœuds définissant les arcs,  $A_1$  est la fonction définie de  $X_1$  vers  $\mathbb{R}^k$  associant à chaque nœud son vecteur d'attributs,  $B_1$  est la fonction définie de  $U_1$  vers  $\mathbb{R}^p$  associant à chaque arc son vecteur d'attributs.

De la même façon :

$$G_2 = [X_2, U_2, A_2, B_2]$$

Supposons  $\text{Card}(X_1) \leq \text{Card}(X_2)$ .

Soit  $C$  une injection entre les deux ensembles de nœuds  $X_1$  et  $X_2$ .

Soit  $D$  une fonction définie de  $A_1(X_1) * \mathbb{R}$  qui définit une mesure de similarité entre deux vecteurs d'attributs associés respectivement à un nœud de  $G_1$  et à un nœud de  $G_2$  où  $A_1(X_1)$ ,  $A_2(X_2)$  respectivement, est l'ensemble de tous les attributs des nœuds de  $G_1$ , respectivement de  $G_2$ .

Une mesure de l'adéquation de la mise en correspondance déterminée par  $C$  par rapport à la valuation du graphe peut être, par exemple :

$$F_D(G_1, G_2) = \sum_{i=1, \text{Card}(X_1)} D(A_1(n_1), A_2(n_2))$$

où  $X_1 = \{n_1, \dots, n_2\}$ , l'ensemble des nœuds du graphe  $G_1$ .

On dit que  $C$  définit une isomorphie entre  $G_1$  et  $G_2$  si :

- $C$  est une **bijection**;
- il existe une bijection  $H$  de  $U_1$  vers  $U_2$  telle que :

$$H(n_1, n_2) = (C(n_1), C(n_2))$$

où  $(n_1, n_2)$  est un arc quelconque de  $G_1$ .

Notre objectif est de déterminer une injection qui optimise la similitude des attributs des nœuds mis en correspondance, similitude calculée par la fonction  $F_D(G_1, G_2)$ , ainsi que l'isomorphie des deux graphes.

Étant donné que dans la plupart des cas, une isomorphie complète ne peut être déterminée, nous parlerons d'«  $\epsilon$ -isomorphie ».

La mise en correspondance des deux graphes images peut alors s'effectuer en deux étapes successives : dans un premier temps : mettre en correspondance les régions en utilisant simplement leurs attributs; dans un second temps : optimiser l' $\varepsilon$ -isomorphie des graphes.

#### a. Mise en correspondance des régions

Nous recherchons un ensemble de régions homologues en comparant les valeurs de leurs attributs. Cette stratégie revient à optimiser la fonction de qualité de correspondance  $F_D(G_1, G_2)$  sans se préoccuper de l'existence d'une fonction  $H$  de mise en correspondance des arcs du graphe.

Nous utilisons des attributs statistiques et géométriques tels que : le niveau de gris moyen, les valeurs minimale et maximale du niveau de gris des points, la variance de la luminance, le nombre de points, le centre de gravité, les axes principaux d'inertie, le coefficient de compacité, ...

Les mises en correspondance sont effectuées grâce à une stratégie similaire à celle utilisée lors du processus de fusion pour la segmentation d'images en régions. Pour chaque couple de régions, nous définissons un prédicat de mise en correspondance et un coût d'appariement calculé par la fonction  $D$ , précédemment définie, dépendant uniquement des attributs des deux régions concernées. De la même façon, nous rangeons tous les couples de régions qui vérifient le prédicat de mise en correspondance dans une structure de donnée (tas ou arbre binaire de recherche) selon la valeur du coût d'appariement.

Nous apparions ensuite itérativement le couple de régions homologues de meilleur coût d'appariement à condition qu'aucune de ces deux régions n'ait été précédemment mise en correspondance (respect de la condition d'unicité de Marr [25]).

L'utilisation du coût d'appariement permet de résoudre certains conflits : les couples de régions sont de coûts différents.

Si les attributs des régions sont discriminants et stables d'une image de la paire stéréoscopique à l'autre, cette méthode donne des résultats satisfaisants. Cela est souvent le cas quand les points de vue associés aux deux images sont très proches.

Il est possible d'améliorer cet algorithme en prenant en compte la contrainte épipolaire [10], bien que ceci nuise à la généralité de l'approche : application au mouvement par exemple, et augmente le temps d'exécution. Cependant, l'ensemble des couples potentiels est réduit puisque deux régions situées sur des épipolaires trop éloignées ne satisfont pas le prédicat de mise en correspondance. De plus, certaines ambiguïtés peuvent être levées, par exemple dans le cas où l'on doit mettre en correspondance plusieurs régions quasi identiques (carreaux des fenêtres d'une scène de bureau par exemple).

#### b. Optimisation de l' $\varepsilon$ -isomorphie des graphes

En partant d'un ensemble de régions homologues, il est possible de mettre en correspondance les deux graphes valués en cherchant à optimiser l' $\varepsilon$ -isomorphie des graphes. L'idée majeure de J. P. Cocquerez et O. Monga [17] est de propager un ensemble de mises en correspondance initiales à l'aide

de relations de voisinage existant dans les deux graphes; on réalise ainsi une expansion en parallèle des appariements initiaux des régions.

On peut également tenir compte des relations de voisinage existant entre les régions des images par une approche différente : en effectuant une étape d'appariement des frontières des régions. La fonction  $D$  ensuite calculée prend en compte les résultats ainsi obtenus [10].

#### 3.4.3. Calcul de l'information tridimensionnelle

Après l'obtention d'un ensemble de couples de régions homologues, se pose le problème de l'exploitation de ces résultats pour la vision stéréoscopique.

Pour cela, nous supposons que chaque couple de régions appariées est issu de la projection d'un même objet plan. Cette hypothèse est raisonnable dans le cas de scènes d'intérieur où il existe beaucoup de surfaces planes : murs, plafond, ... Nous attachons alors à chaque région les segments qui correspondent à l'approximation polygonale de sa frontière.

Il s'agit ensuite d'apparier les segments se trouvant sur la frontière de deux régions homologues. Ce problème peut être aisément résolu grâce au faible nombre de segments.

La mise en correspondance des segments rend possible le calcul du meilleur plan 3D au sens des moindres carrés, sur lequel la surface, dont la paire de régions est la projection, s'appuie.

Quand un point 3D se déplace dans un plan, les relations entre les coordonnées et ses projections sur les images gauche et droite sont linéaires en coordonnées projectives (homographies dans l'espace cartésien). Ainsi, si nous connaissons au moins quatre lignes non parallèles dans une image et leurs correspondants dans l'autre image, nous pouvons calculer la meilleure matrice de transformation par une technique des moindres carrés quelconque. Pour ce propos, O. D. Faugeras et F. Lustman [32] utilisent un filtre de Kalman étendu. Il est alors possible de déduire de la matrice de transformation les paramètres du plan 3D associé.

Nous attachons donc à chaque région un plan 3D sur lequel se trouve la surface dont elle est la projection. Nous espérons ainsi obtenir des surfaces tridimensionnelles planes existant dans la scène observée.

#### 3.4.4. Conclusion

Nous avons utilisé les résultats obtenus grâce à notre algorithme de segmentation à la vision stéréoscopique. Dans cette approche, nous utilisons l'information d'homogénéité des régions pour définir des couples de régions homologues desquels sont déduits des ensembles de segments homologues issus de la projection d'une même surface plane. Ces couples de segments sont utilisés pour définir le plan 3D sur lequel se trouve la surface dont les segments sont la projection de la frontière.

Une approche différente et complémentaire a été développée par O. D. Faugeras et J. D. Boissonnat [33]. A partir de la mise en correspondance de segments 2D, des segments 3D sont déterminés. Une tétraédrisation de Delaunay est ensuite réalisée sur les points 3D définis par les extrémités des segments. Les

faces des tétraèdres intersectant un triangle formé de deux extrémités d'un segment et d'un centre optique de l'une des deux caméras sont éliminés. Ceci a pour but d'enlever les faces qui cachent des points. On détermine ainsi un ensemble de surfaces planes issues uniquement des propriétés géométriques des points 3D.

Signalons qu'il est possible de réaliser une coopération entre le processus d'extraction des régions et celui d'appariement de ces régions. Ceci permet de réduire les différences entre les deux partitions en régions et d'améliorer ainsi la mise en correspondance. Ceci est particulièrement utile pour prendre en compte l'éloignement des points de vues associés aux deux images et la différence d'éclairage existant entre ces deux images et la différence d'éclairage existant entre ces deux images. Cette méthode de coopération est décrite dans [10].

Des résultats de mise en correspondance stéréoscopique de régions sont présentés en Annexe B.

#### 4. Conclusion. Perspectives

Nous avons présenté une nouvelle méthode de segmentation d'images et son application à divers problèmes. L'originalité de notre approche réside dans les points suivants :

- Tout d'abord, nous déconnectons l'aspect « modélisation mathématique » de l'aspect « algorithmique ». Ainsi nous séparons les **critères de segmentation**, généralement spécifiques à chaque problème particulier, de la **stratégie d'utilisation** de ces critères, que nous définissons alors de manière générale. Notre objectif a été de concevoir une « machine » capable d'utiliser efficacement toute sorte de critères. Cette séparation entre outil algorithmique et outil mathématique confère à notre méthode sa généralité.

- De plus, l'usage des différents critères est **optimisé**. Grâce à cette optimisation, nous ne définissons pas de sens de parcours *a priori* des données comme la plupart des autres algorithmes de segmentation. La stabilité des résultats est ainsi améliorée.

- Au lieu de se servir d'un seul critère, comme dans la majorité des méthodes de segmentation, nous utilisons une **suite de critères**. Cette séquence de critères permet de résoudre des problèmes de segmentation où l'on recherche des régions admettant une description hiérarchique.

- Enfin, une **structure de données** adaptée confère à notre méthode une complexité algorithmique faible. Cette structure permet de passer d'un problème de segmentation à un autre avec un effort minimal de programmation, en changeant simplement les fonctions définissant les critères de segmentation.

Notre algorithme permet donc de spécifier simplement un problème de segmentation par la donnée d'une suite de critères définissant les régions à extraire. Si l'on veut que cette méthode soit la plus générale possible, il suffit de donner des critères très généraux; par contre, si l'on se trouve dans un cas particulier bien défini, il suffit de déterminer les critères adaptés au problème à résoudre. Les propriétés des régions

peuvent être de natures diverses : statistiques, géométriques, analytiques, ...

Cependant, si ce travail a le mérite de décrire et d'appliquer une « mécanique » permettant d'utiliser efficacement et de manière optimisée des critères de segmentation, il ne fournit pas de méthode automatique pour la détermination de ces critères. Le travail de recherche de critères adaptés au problème de segmentation traité reste *ad hoc* et laissé à la libre appréciation de celui qui les détermine.

L'application de notre méthode à des problèmes de segmentation d'images naturelles monochromatiques et couleur donne de bons résultats. La qualité de ces résultats permet d'initialiser une nouvelle approche pour la vision stéréoscopique.

Cet algorithme peut aussi être utilisé pour segmenter des images tridimensionnelles. Une telle application est décrite dans [15].

De nombreuses autres applications peuvent être envisagées. Notre méthode permet d'obtenir de bons résultats dans l'étape fondamentale de vision par ordinateur qu'est la segmentation d'images et ceci, quelque soit le domaine d'application considéré.

#### Remerciements

Les auteurs tiennent à remercier pour leurs suggestions et leur intérêt porté à ces travaux : Roger Mohr, Olivier Faugeras et André Gagalowicz. De plus, nous remercions Djemel Ziou et Rachid Deriche pour l'utilisation des résultats de leurs algorithmes respectifs de détection de contours, Georgio Toscani pour la calibration du système de vision stéréoscopique, et Jean-Pierre Cocquerez pour son travail sur la mise en correspondance des régions.

Manuscrit reçu le 15 janvier 1987.

#### BIBLIOGRAPHIE

- [1] R. M. HARALICK et L. G. SHAPIRO, Survey: Image Segmentation Techniques, *CVGIP*, 29, 1986, p. 100-132.
- [2] A. NAZIF et LEVINE, Low Level Segmentation: An Expert System, *PAMI*, 6, n° 5, September 1984.
- [3] T. PONG *et al.*, Experiment in Segmentation Using a Facet Model Grower, *CGIP*, n° 25, 1984, p. 1-2.
- [4] S. W. ZUCKER, Survey Region Growing: Childhood and Adolescence, *CGIP*, n° 5, 1976, p. 382-399.
- [5] T. PAVLIDIS, *Structural Pattern Recognition*, Springer Verlag, 1977.
- [6] O. D. FAUGERAS, M. HEBERT et E. PAUCHON, Segmentation of Range Data into Planar and Quadratic Patches, *Proc of CVPR*, 1983.
- [7] Y. OHTA, *Knowledge-based Interpretation of Outdoor Natural Color Scenes*, Pitman Publishing Inc., 1985.
- [8] OHLANDER, Analysis of Natural Scene, *Ph. D. Thesis*, Department of Computer Science, Carnegie-Melon University, 1975.
- [9] E. C. FREUDER, A Relative Approach to Region Finding, *CGIP*, 6, 1977, p. 409-433.

- [10] B. WROBEL, Stéréovision : Coopération entre l'extraction et la mise en correspondance symbolique de régions, *Actes du congrès CESTA-MARI*, Paris, mai 1987.
- [11] B. WROBEL et O. MONGA, Segmentation d'images naturelles : Coopération entre un détecteur-contour et un détecteur-région, *Actes du 11<sup>e</sup> Colloque GRETSI*, Nice, juin 1987.
- [12] O. MONGA, A Segmentation Algorithm and its Application to Stereovision, *Proc. of the IEEE International Workshop of Industrial Applications of Machine Vision and Machine Inettligence*, Roppongi, Tokyo, Japon, février 1987.
- [13] A. GAGALOWICZ et O. MONGA, Un algorithme de segmentation hiérachique, *Actes du 5<sup>e</sup> congrès AFCET*, Grenoble, novembre 1985.
- [14] A. GAGALOWICZ et O. MONGA, A New Approach for Image Segmentation, *Proc. of the 8th International Conference on Pattern Recognition*, Paris, October 1986.
- [15] O. MONGA et N. KESKES, A Hierarchical Algorithm for the Segmentation of 3-d Images, *Proc. of the 8th International Conference on Pattern Recognition*, Paris, October 1986.
- [16] J. P. GAMBOTTO et O. MONGA, A Parallel and Hierarchical Algorithm for Region Growing, *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Francisco, juin 1985.
- [17] J. P. COCQUEREZ et O. MONGA, Matching Regions Issued from a Stereoscopic Pair, *Proc. of the 5th Scandinavian Conference on Image Analysis*, Stockholm, juin 1987.
- [18] R. DERICHE, Optimal Edge Detection Using Recursive Filtering, *International Journal of Computer Vision*, 1987 (à paraître).
- [19] O. MONGA et R. DERICHE, A New Three Dimensional Boundary Detection Algorithm, *Proc. of the 8th International Conference on Pattern Recognition*, Paris, October 1986.
- [20] J. F. CANNY, Finding Edges and Lines in Images, MIT AI Lab., TR-720, *Masters Thesis*, MIT, 1983.
- [21] J. SHEN et S. CASTAN, Un nouvel algorithme de segmentation d'images, *Actes du 5<sup>e</sup> Congrès RFIA (AFCET-INRIA)*, Grenoble, novembre 1985, p. 201-213.
- [22] WILLIAMS, Algorithm 232 Heapsort, *Communication ACM*, juin 1984, p. 347-348.
- [23] M. GONDRAN et M. MINOUX, *Graphes et algorithmes*, Eyrolles, Paris, 1979.
- [24] R. W. FLOYD, Algorithm 245: Treesort 3, *Communication ACM*, 7, n° 12, p. 701.
- [25] D. MARR et T. POGGIO, Cooperative Computation of Stereo Disparity, *Science*, 194, 1976, p. 283-287.
- [26] N. AYACHE et B. FAVERJON, Efficient Registration of Stereo Images by Matcgng Graph Descriptions or Edge Segments, *International Journal of Computer Vision*, 1987 (à paraître).
- [27] P. LONG et M. BERTHOD, Passage de primitives de bas-niveau à des primitives de haut-niveau dans le problème de mise en correspondance stéréo, *Actes du 5<sup>e</sup> Congrès RFIA (AFCET-INRIA)*, Grenoble, novembre 1985.
- [28] G. MÉDIONI et R. NEVATIA, Segment Based Stereo Matching., *Proc. of Image Understanding Workshop*, Arlington, Va. (USA), 1983, p. 128-136.
- [29] H. P. MORAVEC, Toward Automatic Visual Obstacle Avoidance, *Proc. of the 5th IJCAI*, Cambridge, MA, USA, août 1977, p. 584.
- [30] W. E. L. GRIMSON, *From Images to Surfaces. A Computational Study of the Human Early Visual System.*, MIT Press, 1981.
- [31] O. D. FAUGERAS et G. TOSCANI, The Calibration Problem for Stereo, *Proc. of CVPR*, Miami, 1986.
- [32] O. D. FAUGERAS et F. LUSTMAN, Let us Suppose that the World is Piece Wise Planar, *Proc. of the 3th International Symposium of Robotics Research*, Gouvieux, octobre 1986.
- [33] O. D. FAUGERAS et J. D. BOISSONNAT, *The Delaunay Triangulation and Passive Stereo*, INRIA, en préparation.
- [34] D. KNUTH, *The Art of Computer Programming*, 3: *Sorting and Searching*, Addison-Wesley, Reading, MA, 1983.

**Annexe A**

**Résultats de la segmentation d'images naturelles**

1. IMAGES MONOCHROMATIQUES

Nous présentons dans cette annexe les résultats de la méthode de segmentation d'images naturelles 2D monochromatiques dont les critères utilisés sont décrits au paragraphe 3.2.2.

Deux études parallèles ont été menées indépendamment l'une au CRIN et l'autre à l'INRIA. Les deux méthodes ainsi créées adhèrent au formalisme pré-

senté dans ce papier mais différent sur les cinq détails suivants :

- choix de l'algorithme de segmentation pour effectuer la partition initiale;
- utilisation ou non d'une image de points de contraste pour guider le processus d'extraction de régions;
- choix des structures de données pour trier les couples de régions en fonction de leur coût de fusion;
- critères de segmentation;
- seuils utilisés par les critères.

Pour être complet, comparons les deux méthodes sur les cinq points précédemment cités :

	INRIA (O. Monga) (1)	CRIN (B. Wrobel) (2)
Partition initiale. . . . .	Procédure MERGE de Pavlidis	Balayage séquentiel de l'image
Image des points de contraste. . . . .	Utilisation optionnelle de l'image des points de contraste [18]	Utilisation obligatoire de l'image des points de contraste [21]
Structure de données. . . . .	Tas	Arbre binaire de recherche
Critères de fusion. . . . .	Min, max Variance Gradient de la frontière Utilisation optionnelle du nombre de points de contraste le long de la frontière (issus d'un détecteur de contours)	Min, max Moyenne Gradient de la frontière Nombre de points de contraste le long de la frontière (issus d'un détecteur de contours)
Seuils	Fixés de manière à obtenir des prédicats emboîtés	Calculés automatiquement à partir de l'image traitée

Puisque nous appliquons par la suite une méthode de stéréovision fondée sur la mise en correspondance des régions des deux images, nous présentons ci-après les résultats obtenus par nos deux algorithmes de segmentation sur un couple d'images

stéréoscopiques d'une scène d'intérieur. Les figures 3 à 10 présentent les différentes étapes et les résultats de l'algorithme (1), les figures 13 et 14 ceux obtenus par l'algorithme (2).

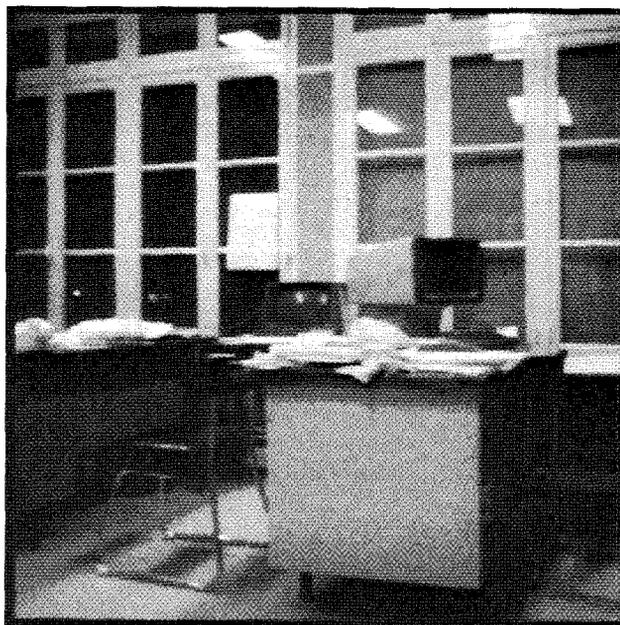


Fig. 2. — Image originale gauche.

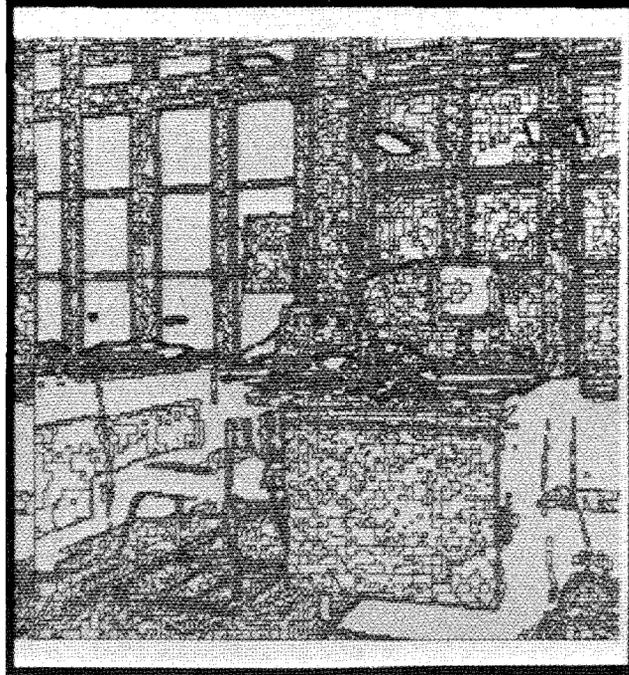


Fig. 3. — Partition initiale.

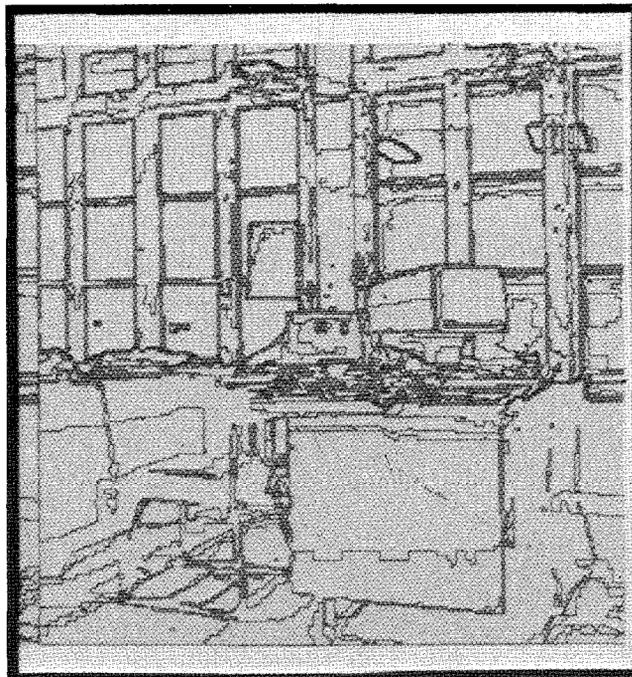


Fig. 4. — Utilisation du critère: max-min.

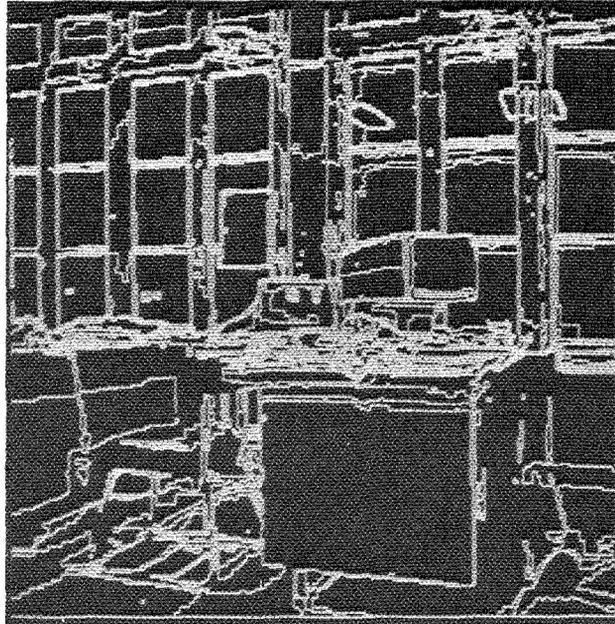


Fig. 5. — Utilisation du critère: variance.

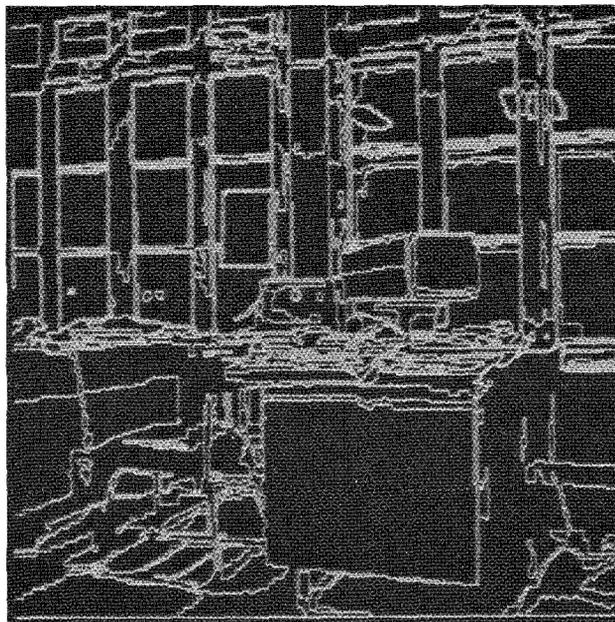


Fig. 6. — Utilisation du critère: gradient sur les frontières (segmentation finale) (algorithme n° 1).

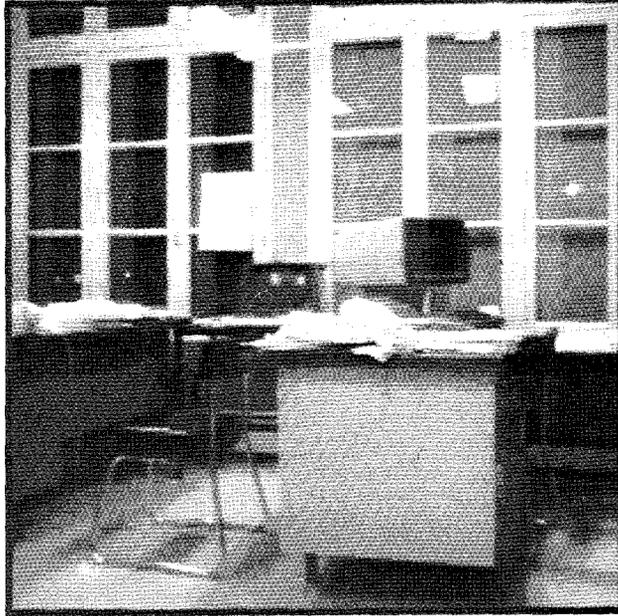


Fig. 7. — Image originale droite.

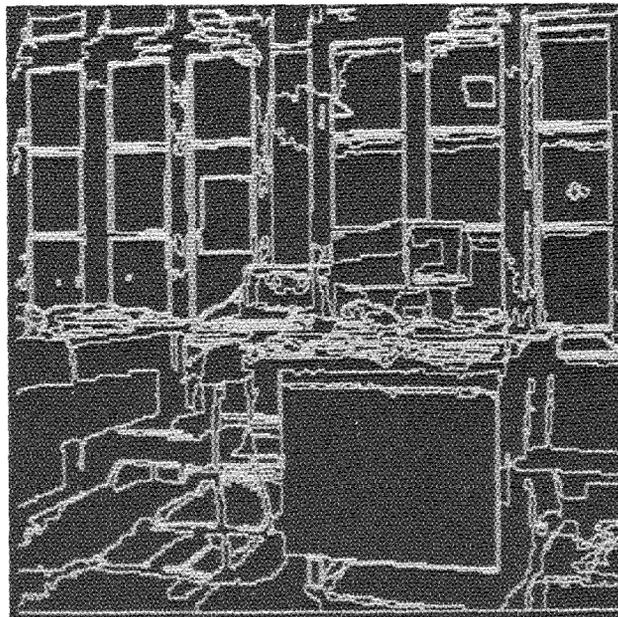


Fig. 8. — Segmentation finale (algorithme n° 1).

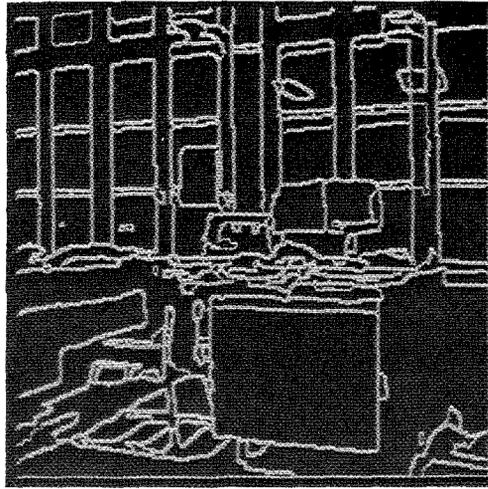


Fig. 9. — Segmentation avec la coopération  
détection contours-détection régions (algorithme n° 1)  
(image gauche).

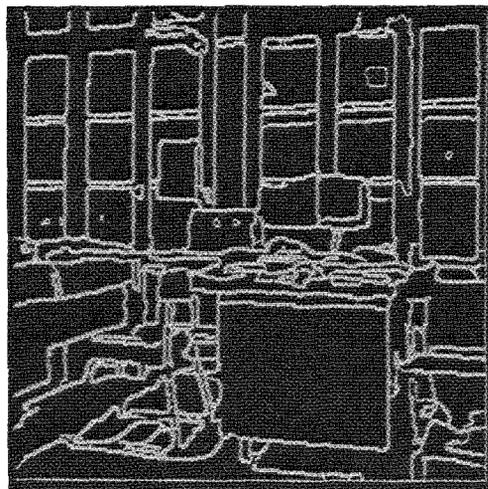


Fig. 10. — Segmentation avec la coopération  
détection contours-détection régions (algorithme n° 1)  
(image droite).

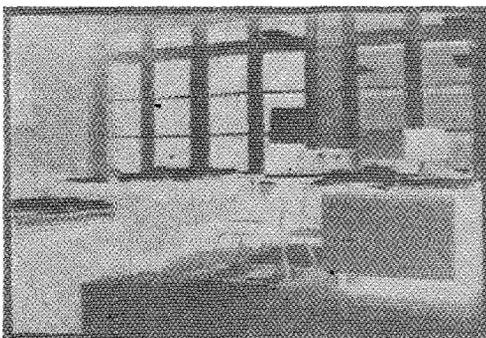


Fig. 11. — Image originale gauche.

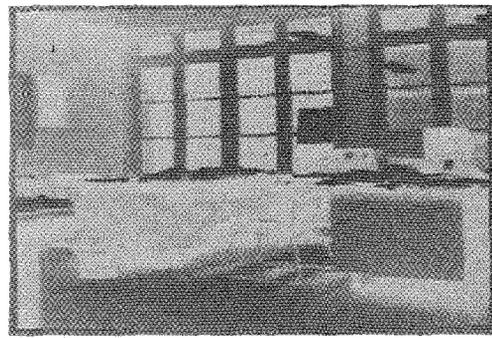


Fig. 12. — Image originale droite.

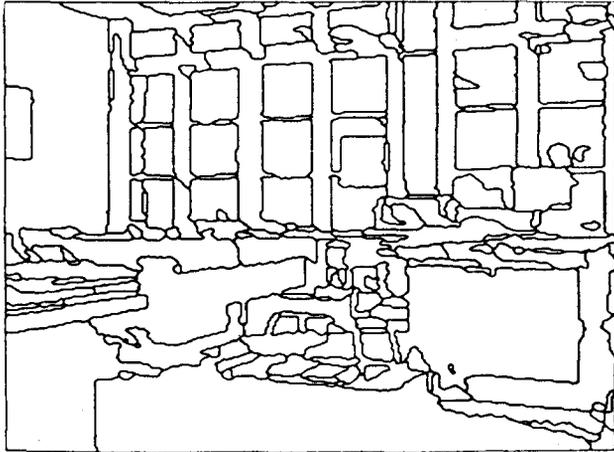


Fig. 13. - Segmentation finale (algorithme n° 2) (image gauche).

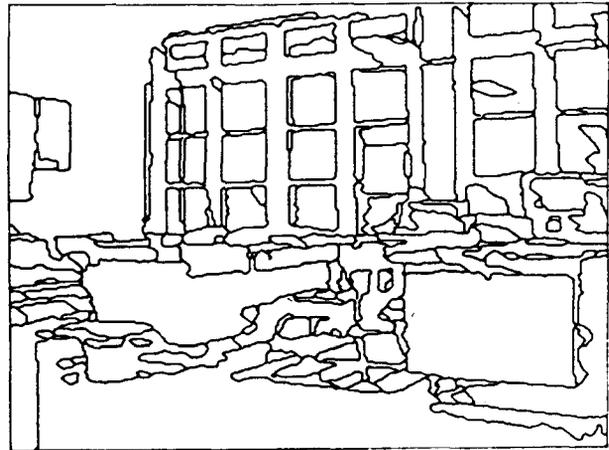


Fig. 14. - Segmentation finale (algorithme n° 2) (image droite).

2. SEGMENTATION D'IMAGES COULEUR

Nous présentons ci-après les résultats obtenus par notre méthode sur une image couleur.

Annexe B

Application à la vision stéréoscopique

Nous présentons dans cette annexe les résultats de la mise en correspondance des régions issues d'une paire d'images stéréoscopiques.

Les figures 17, 18 et 19 présentent les résultats obtenus à l'INRIA, les figures 20, 21 et 22 ceux obtenus au CRIN, ces derniers obtenus avec la coopération des processus d'extraction de d'appariement des régions.

Annexe C

Structures de données 34

Nous avons besoin d'une structure de données efficace nous permettant de stocker des éléments valués et triés, d'accéder au plus petit élément et d'insérer de nouveaux éléments. Pour cela, nous pouvons utiliser un tas ou un arbre binaire de recherche.

1. LE TAS

Un tas est une liste valuée particulière. Chacun de ses éléments est caractérisé par un nom et une valeur numérique. Si nous supposons que le nom est un indice  $i$  variant de 1 à  $n$ , les valeurs numériques associées peuvent être représentées par un vecteur  $P(i)$  (où  $1 \leq i \leq n$ ).

L'ensemble de ces éléments forme un tas si tous les noms des éléments sont dans un vecteur  $T(j)$  où

$1 \leq j \leq n$ .  $T(j)$  devient alors le nom de l'élément placé en position  $j$  et est tel que :

$P(T(j)) \geq P(T(\lfloor j/2 \rfloor))$  pour tout  $j$  où  $\lfloor j/2 \rfloor$  est la notation pour la valeur inférieure à  $j/2$ .

Par exemple : soit

$$P = (9, 8, 8, 2, 7, 11)$$

alors

$$T = (4, 2, 5, 1, 6, 3)$$

La condition précédente peut aussi s'écrire sous la forme suivante :

$$[P(T(j)) \leq P(T(2*j))]$$

et

$$[P(T(j)) \leq P(T(2*j+1))]$$

Il est possible de représenter le tas sous la forme d'un arbre de la manière suivante :

- chaque index  $j$  (où  $1 \leq j \leq n$ ) du vecteur  $T$  représente un nœud  $j$  de l'arbre;
- chaque nœud  $j$  est connecté aux deux nœuds  $2*j$  et  $2*j+1$  si ils existent.

Ces deux conditions demandent que la valeur numérique associée au nœud  $j$  soit inférieure ou égale à celles associées aux nœuds  $2*j$  et  $2*j+1$ .

En reprenant l'exemple précédent, nous obtenons l'arbre suivant (fig. 23) :

Le plus petit élément du tas se trouve alors au sommet de l'arbre, soit en  $T(1)$ .

Il est possible de montrer que le tas vérifie les propriétés suivantes :

- Si  $P(i)$ , où  $1 \leq i \leq n$ , est un vecteur; la construction du tas dont les valeurs numériques sont les vecteurs  $P(i)$  demande  $O(n)$  opérations élémentaires.
- Si  $P(i)$ , où  $1 \leq i \leq n$  est un vecteur; la construction du tas dont les valeurs numériques sont les vecteurs  $P(i)$  demande  $O(n)$  opérations élémentaires.



Fig. 15. — Image couleur.

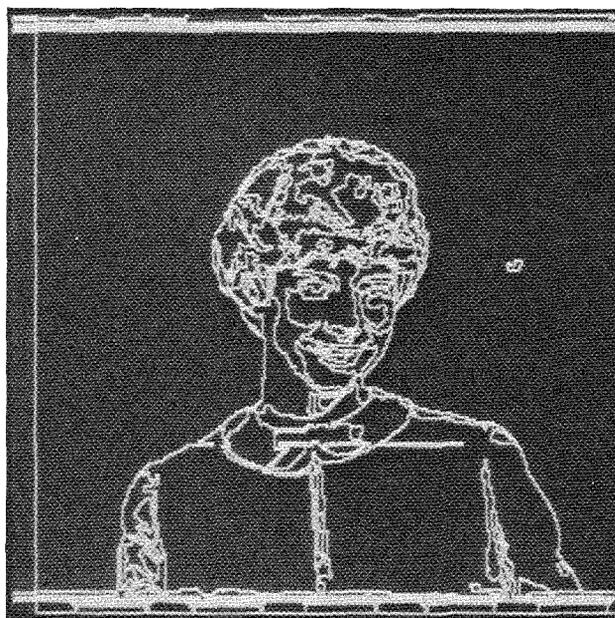


Fig. 16. — Sggmentation de l'image couleur en régions.

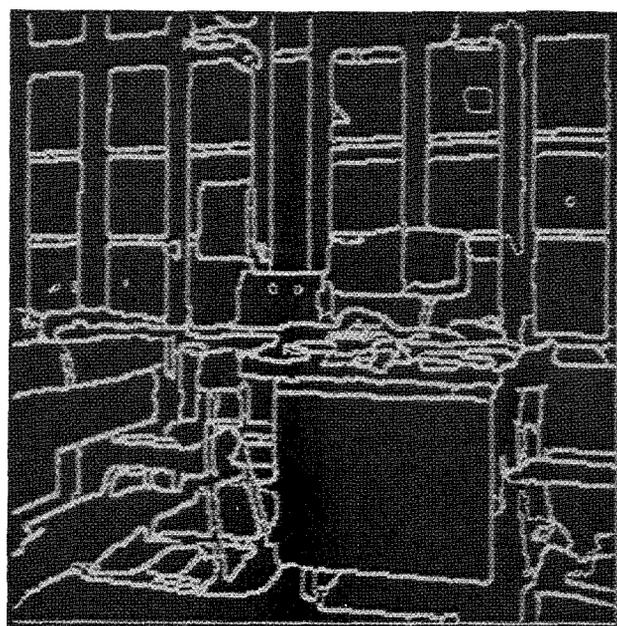
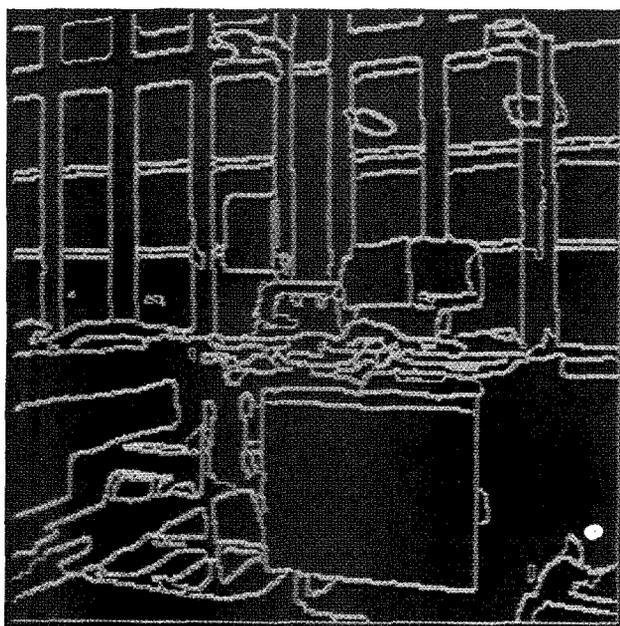


Fig. 17. — Couple d'images segmentées en régions.



Fig. 18. — Appariements réalisés : deux régions homologues possèdent la même couleur.

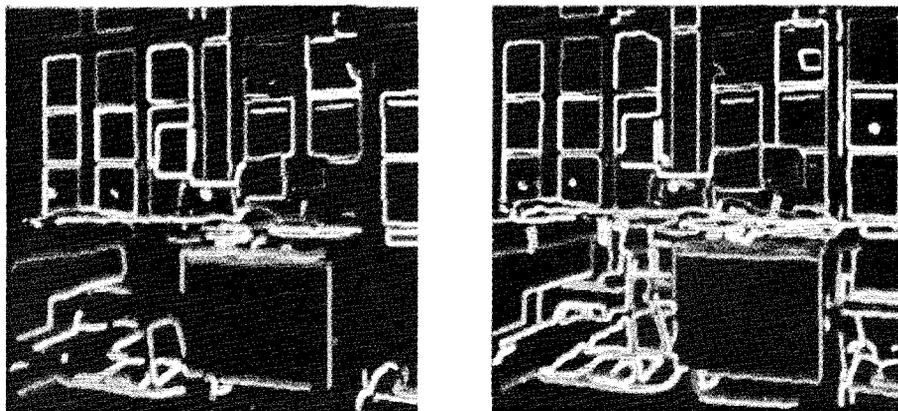


Fig. 19. — Appariements des segments approxinant les frontières des régions.

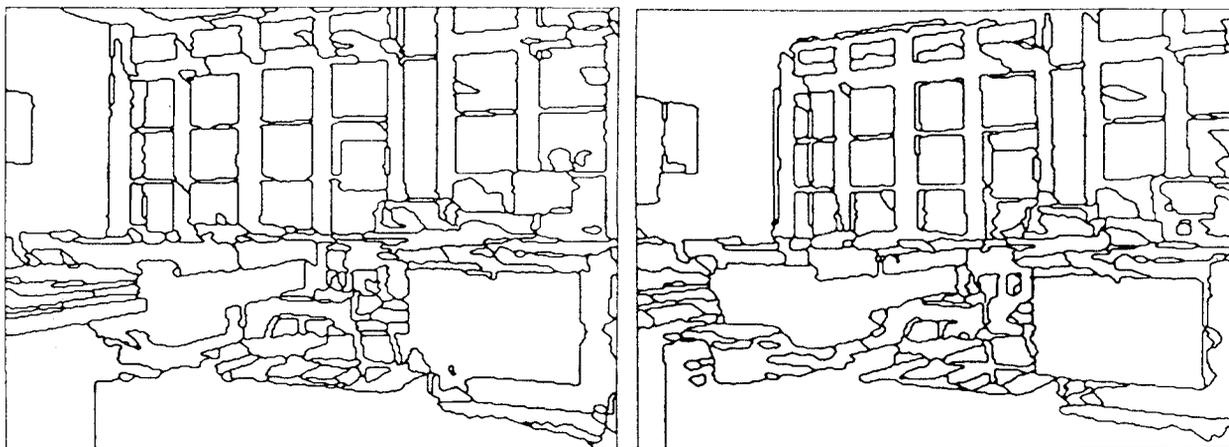


Fig. 20. — Couples d'images segmentées en régions.

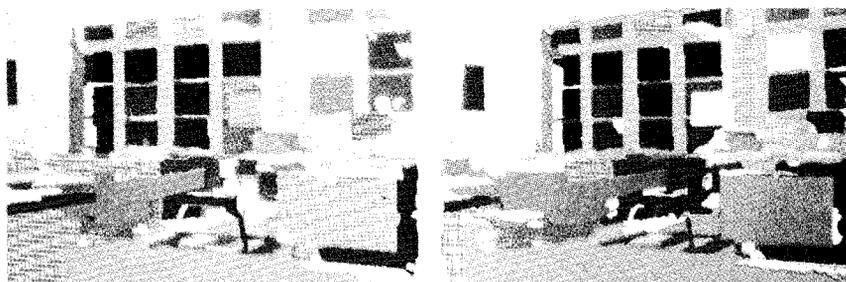


Fig. 21. — Appariements réalisés: deux régions homologues possèdent la même couleur.

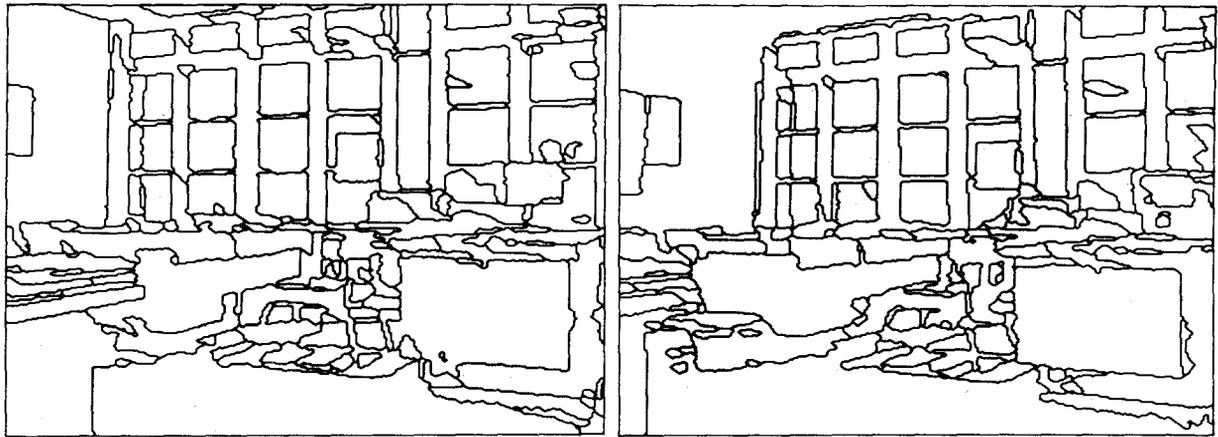


Fig. 22. - Segmentation finale des images en régions avec la coopération extraction-appariement des régions.

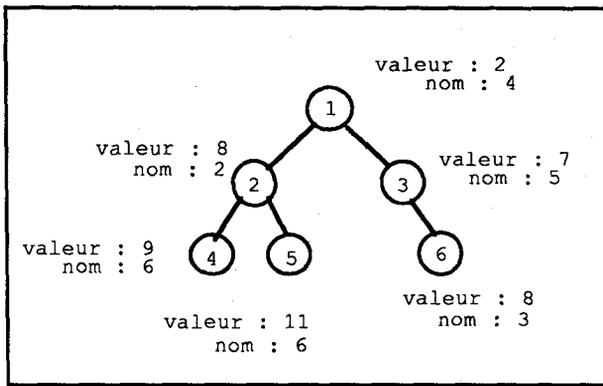


Fig. 23. - Arbre représentant la structure de tas dont les éléments sont dans l'ensemble  $P = \{9, 8, 8, 2, 7, 11\}$ .

- Après l'extraction du plus petit élément  $T(1)$ , la reconstruction du tas demande  $O(\log(n))$  opérations élémentaires.
- Après l'extraction d'un élément du tas en connaissant sa position  $j$ , la reconstruction du tas demande  $O(\log(n))$  opérations élémentaires.
- L'adjonction d'un élément dans le tas demande  $O(\log(n))$  opérations élémentaires.

## 2. L'ARBRE BINAIRE DE RECHERCHE

L'ensemble des éléments valués peut être stocké sous forme d'un arbre binaire de recherche.

La propriété d'être un arbre binaire de recherche peut être définie récursivement de la manière suivante:

A est un arbre binaire de recherche si:

- A est l'arbre vide,

ou

- A est égal à  $\langle A_1, x, A_2 \rangle$  où  $A_1$  et  $A_2$  sont deux arbres binaires de recherche et  $A_1 < x < A_2$  signifie que toutes les valeurs de l'arbre  $A_1$  sont inférieures à la valeur  $x$  de la racine et que toutes les valeurs de l'arbre  $A_2$  sont supérieures à cette même valeur  $x$ .

La plus petite valeur  $x$  se trouve donc à une feuille de la branche la plus « à gauche » de l'arbre.

Par exemple: si  $P = \{9, 8, 8, 2, 7, 11\}$ , l'arbre correspondant est représenté sur la figure 24.

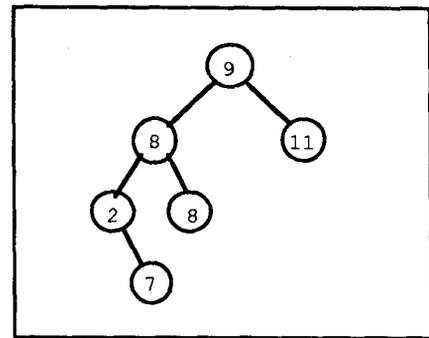


Fig. 24. - Arbre binaire de recherche pour l'ensemble des éléments  $P = \{9, 8, 8, 2, 7, 11\}$ .

Le plus petit élément se trouve à la feuille de la branche la plus « à gauche » de l'arbre.

Il est alors possible de montrer que cet arbre vérifie les propriétés suivantes:

- La construction de l'arbre demande  $O(n \log(n))$  opérations élémentaires.
- L'extraction du plus petit élément se fait en parcourant la branche « extrême gauche », soit en  $O(\log(n))$  opérations élémentaires.
- Après l'extraction du plus petit élément, la mise à jour de l'arbre ne dépend pas du nombre de nœuds présents et demande  $T_1$  opérations élémentaires ( $T_1$  vaut généralement 1).
- Après l'extraction d'un élément quelconque de l'arbre, la mise à jour de l'arbre demande  $O(\log(n) * n/2)$  opérations élémentaires.
- L'adjonction d'un élément dans l'arbre demande  $O(\log(n))$  opérations élémentaires.

## Annexe D

### Partition initiale

#### 1. PROCÉDURE MERGE DE PAVLIDIS

Nous décrivons ici la manière dont nous avons implanté la procédure Merge de Pavlidis [5].

L'image initiale est représentée par une structure pyramidale (quad-tree) obtenu par subdivision récursive. On passe d'un niveau à l'autre du «quad-tree» en divisant en quatre, de manière symétrique, les régions du niveau précédent (fig. 26).

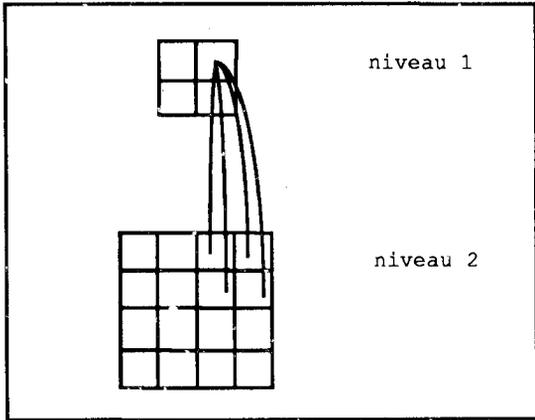


Fig. 25. — Structure de «quad-tree» associé à une image.

Le niveau  $n$  de «quad-tree» définit une image de dimension  $2^n * 2^n$  dont chaque point correspond à quatre points du niveau  $n+1$ . Chaque niveau  $n$  du «quad-tree» décrit l'image à un «niveau de résolution» défini par  $n$ . Le niveau  $N$  tel que  $2^N * 2^N$  soit la dimension de l'image correspond correspond à tous les points de l'image. Chaque point d'un niveau  $n$  correspond donc à  $2^{N-n}$  points de l'image originale.

Notre algorithme comporte deux principales étapes :

- construction d'un «quad-tree» associé à l'image originale en attachant à chaque point d'un niveau  $n$  quelconque le maximum et le minimum des niveaux de gris des  $2^{N-n}$  points de l'image originale auquel il correspond. Ce «quad-tree» est construit par fusion, c'est-à-dire en partant du niveau  $N$  puis en calculant le niveau  $n$  en fonction du niveau  $n+1$ ;
- détermination des points du «quad-tree» tels que les points de la sous-image originale associée vérifient «maximum-minimum < seuil». Cette détermination est effectuée en parcourant le «quad-tree» du niveau 0 vers le niveau  $N$  en mettant à jour deux listes contenant respectivement, lors du passage du niveau  $n$  au niveau  $n+1$ , les points des niveaux précédents validés (c'est-à-dire inférieurs à  $n+1$ ) et les points du niveau  $n$  dont il faut examiner les fils au niveau  $n+1$ .

2. BALAYAGE SÉQUENTIEL DE L'IMAGE

La méthode utilisée pour effectuer une partition initiale est très simple : l'image est examinée séquentiellement point par point, de la gauche vers la droite et de haut en bas. Au fur et à mesure du balayage, chaque point est classé dans une région en fonction de ses voisins déjà étiquetés.

Le voisinage d'un point de l'image est défini comme étant l'ensemble des points qui lui sont directement connectés. Chaque point a soit quatre, soit huit

voisins pour une image représentée en trame carrée, soit 6 voisins dans le cas d'une trame hexagonale. La trame carrée des images utilisées et la nature séquentielle de l'algorithme nous obligent à ne retenir que quatre points dans le cas de huit voisins et deux seulement dans le cas de quatre voisins.

La méthode utilisée n'est présentée que dans le cas d'un voisinage de quatre voisins, mais elle peut être aisément étendue à un autre voisinage.

L'algorithme décrit ci-après est à appliquer pour chacun des points de l'image.

Soit  $X$  le point de l'image que l'on examine. Soient  $A$  et  $B$  respectivement le voisin «supérieur» et le voisin «gauche» de  $X$ , déjà traités et affectés à une région (fig. 26).

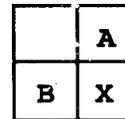


Fig. 26. — Traitement du point  $X$  en fonction des régions auxquelles appartiennent ses deux voisins  $A$  et  $B$ .

```

Début
Si P(X, A)=vrai et P(X, B)=faux alors R(A) ← R(A) ∪ X fsi
Si P(X, A)=faux et P(X, B)=vrai alors R(B) ← R(B) ∪ X fsi
Si P(X, A)=faux et P(X, B)=faux alors Créer(X) fsi
Si P(X, A)=vrai et P(X, B)=vrai
alors Si P(X, A)=vrai alors R(A) ← R(A) ∪ R(B) ∪ X
sinon Si Q(A, X) > Q(B, X)
alors R(A) ← R(A) ∪ X
sinon R(B) ← R(B) ∪ X
fsi
    
```

où  $R(R)$  est la région à laquelle appartient le point  $R$ .  $P(R_1, R_2)$  est le prédicat de fusion des deux régions  $R_1, R_2$ , éventuellement réduites à un point.  $P(R_1, R_2)$  est vrai quand la fusion de  $R_1$  avec  $R_2$  est autorisée. Créer( $X$ ) est une fonction de création d'une nouvelle région à laquelle appartient le point  $X$ .  $Q(R_1, R_2)$  est la fonction calculant la qualité de la fusion entre les deux régions  $R_1, R_2$ .

Rappelons que les prédicats de fusion et les critères ont des seuils très peu tolérants dans le but de construire beaucoup de petites régions mais très homogènes. Les attributs des régions utilisés dans cet algorithme sont les valeurs minimale et maximale du niveau de gris et la moyenne.

Cet algorithme peut être aisément modifié pour tenir compte des résultats obtenus par un détecteur de contours opérant sur l'image initiale. Il procède alors en deux étapes : traitement des points de l'image sur lesquels aucun contraste n'a été détecté, puis absorption de ces points restants par les régions voisines afin que tous les points de l'image appartiennent à une région. C'est de cette façon que cet algorithme est effectivement implanté.