# Analyse de scènes et compression d'images

***1re partie***

M. Benard, M. Kunt, R. Leonardi et P. Volet

***Cet article présente tout d'abord une introduction à la théorie générale de l'analyse de scènes. Les composantes de cette théorie sont définies et illustrées. Les nouvelles techniques de compression d'images sont ensuite développées dans ce contexte. Les derniers résultats de ces techniques sont également présentés.***

***Nach einer Einführung in die allgemeine Theorie der Bildanalyse werden deren Komponenten definiert und dargestellt sowie auf dieser Grundlage die neuen Bildkompressionstechniken entwickelt. Die damit erreichten letzten Resultate werden im Artikel vorgestellt.***

La 2e partie sera présentée dans le numéro 21/1986.

**Adresse des auteurs**

*Michel Benard, Murat Kunt, Riccardo Leonardi* et *Patrick Volet*, Laboratoire de traitement des signaux, Ecole Polytechnique Fédérale de Lausanne, 16, ch. de Bellerive, CH-1007 Lausanne.

## 1. Introduction

Tout système d'acquisition d'images numériques, que ce soit un microdensitomètre à haute résolution ou une caméra de télévision, produit les données en échantillonnant dans l'espace et en quantifiant les luminances d'une scène originale. Une image numérique est ainsi une matrice $N$ par $N$ de points images, nécessitant $N^2 \cdot B$ bits pour sa représentation. Cette matrice est souvent appelée forme canonique d'une image numérique. Le but de la compression ou du codage d'images est de réduire le plus possible le nombre de bits nécessaires pour représenter et reconstituer une réplique fidèle de l'image originale. Les premiers efforts en codage d'images ont été basés sur la théorie de l'information utilisant la statistique de la source d'information. Un grand nombre de méthodes a été développé dans ce cadre pendant ces vingt dernières années. On peut les grouper en trois catégories: prédictives, transformées et hybrides. La compression que l'on peut atteindre, après avoir débuté à 1 au début des années soixante, a atteint un niveau de saturation autour de 10 à 1 il y a quelques temps (Fig. 1a). Ceci ne veut certainement pas dire que la limite supérieure donnée par l'entropie de la source ait été atteinte. D'abord cette entropie n'est pas connue et dépend très étroitement du modèle de la source que l'on utilise. Ensuite, la théorie de l'information ne tient pas compte de ce que l'être humain voit ni de la manière dont il voit.

Les progrès récents dans l'étude du mécanisme de la vision ont ouvert de nouvelles possibilités en codage d'images. La sensibilité directionnelle des neurones dans le système visuel et le traitement séparé des contours et des textures conduisent à des méthodes de codage permettant des compressions allant jusqu'à 150 à 1 [1]. Ces nouvelles méthodes que nous appelons de la deuxième génération font l'objet de cet article. Elles sont présentées dans le contexte de l'analyse de scènes. La figure 1b montre l'évolution du rapport de compression en fonction du temps.

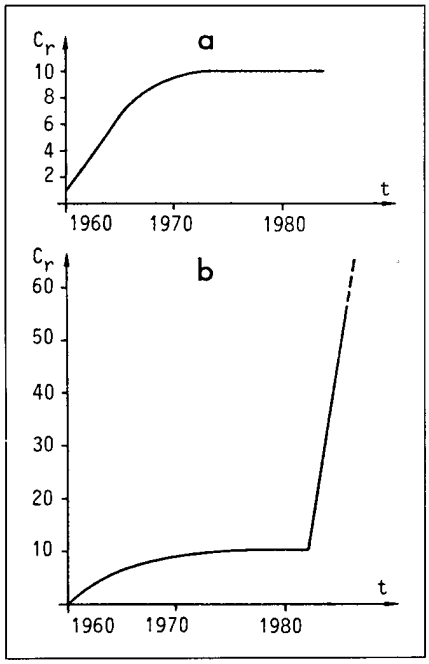


**Fig. 1 Rapport de compression en fonction du temps**
a Méthodes de la première génération
b Méthodes de la deuxième génération

L'analyse de scènes est une opération que nous, êtres humains, effectuons à longueur de journée, la plupart du temps sans même en être conscients. Nous regardons autour de nous et déterminons rapidement un très grand nombre de propriétés des personnes et des objets qui nous entourent. Par exemple, les personnes que nous connaissons par rapport à celle que nous ne connaissons pas, la nature, la forme, la fonction des objets, leurs distances, etc. Nous faisons tout ceci avec aisance et rapidité grâce au plus merveilleux des ordinateurs (c'est

même un ordinateur personnel!) qu'est notre cerveau.

Ce que nous voulons c'est que ces mêmes opérations soient effectuées par une machine en imitant ces facultés humaines. Posé dans sa généralité, le problème est très difficile. Même si nous disposons d'excellents moyens techniques pour transcrire et traiter une image dans un ordinateur, nous ne disposons pas de connaissances suffisantes pour y ajouter notre propre expérience de la vision, forgée depuis notre naissance, ni la façon avec laquelle nous contrôlons ces expériences en fonction des données.

L'analyse de scènes apparaît ainsi comme le chemin qui relie les données provenant de la scène aux modèles préétablis du monde. Pour parcourir ce chemin, nous devons utiliser plusieurs représentations entre les signaux physiques traduisant la scène et les symboles cognitifs représentant les idées et les concepts. Des signaux aux symboles, la densité de l'information augmente au détriment de son aspect local. Si le long de ce chemin on garde constamment la possibilité de reconstituer une replique plus ou moins fidèle des données de départ, on obtient des méthodes de compression très performantes à cause de la représentation très compacte (comprimée) des données.

L'article présente le contexte général du problème de l'analyse de scènes et de la compression d'images. Ensuite, quatres méthodes sont décrites: croissance de région, division et rassemblement, décomposition directionnelle et synthèse de texture. Des résultats correspondants sont commentés.

## 2. Analyse de scènes

### *2.1 Généralités*

L'analyse de scènes constitue une opération quotidienne chez l'être humain. Sans effort, l'homme interprète rapidement un grand nombre de propriétés des personnes et des objets qui l'entourent. Les connaît-il? quelles sont la forme, la couleur, la fonction des objets qu'il contemple? Autant d'opérations qui sont effectuées des milliers de fois par jour par notre cerveau.

Ce que nous voulons c'est faire exécuter les mêmes opérations à une machine en imitant ces facultés humaines. Par exemple, il serait très agréable de disposer d'une machine à laver le linge qui pourrait parcourir la maison, qui

**Fig. 2**
**Représentation d'une image numérique par des hauteurs en perspective**

examinerait tous les linges, qui reconnaîtrait le linge sale et le ramasserait, qui le laverait en faisant le tri des différents tissus, etc. Et si en plus, la même machine s'occupait d'une manière similaire de la cuisine? On peut penser et rêver à d'autres gadgets de ce genre.

Aujourd'hui, dans l'état actuel des travaux de recherche nous sommes très loin de ce résultat. Car, posé dans sa généralité, le problème est très difficile.

Pour illustrer la difficulté de ce problème, on peut essayer de mettre à l'épreuve notre propre ordinateur. Par exemple, l'information représentée à la figure 2 est très difficile à interpréter et à reconnaître. Or il s'agit d'une représentation différente de l'image de la figure 3. Au lieu de représenter le portrait par des niveaux de gris, on l'a représenté par des hauteurs. Il s'agit exactement de la même information représentée différemment. La difficulté que nous avons eue à reconnaître le contenu de la figure 2 découle de notre manque d'expérience de ce genre de représentation. Depuis notre enfance, nous avons l'habitude de voir les photos avec des niveaux de gris ou de la couleur mais pas avec des hauteurs. Maintenant que nous avons fait l'expérience avec cette nouvelle représentation, nous aurons beaucoup moins de difficulté avec la même représentation même pour un objet totalement différent. Les données fournies à l'ordinateur ne sont pas meilleures que celles de la figure 2. En plus, cette machine n'a pas eu le temps de forger une expérience équivalente à la nôtre. Il est normal qu'elle soit en difficulté.

**Fig. 3 Représentation usuelle d'une image**

### *2.2 Composantes du problème*

L'ingénieur a l'habitude de disséquer les problèmes ardus pour essayer de les résoudre par morceaux. Une première division que l'on peut envisager consiste à distinguer les données, l'expérience et le traitement. Ces éléments sont brièvement décrits ci-après.

L'information contenue dans une scène est traduite en signaux électriques par des capteurs (caméra de télévision, caméra CCD, etc.). Ces données brutes doivent ensuite être traitées de manière à ce qu'une représentation symbolique de la scène soit établie explicitement. Il s'agit dans ce cas principalement d'un problème de traitement des signaux, convertissant des signaux en symboles. Ce problème est, dans son ensemble, bien maîtrisé. Des solutions satisfaisantes devraient voir le

jour prochainement.

La deuxième partie concerne la transcription de notre expérience cognitive dans une forme acceptable par une machine. Comme l'exemple précédent le montre, un grand volume d'information à priori doit être disponible. Notre connaissance du monde nécessite donc un modèle qui doit se traduire en une représentation des connaissances. En plus, cette représentation doit être compatible avec la machine de traitement. Malheureusement les travaux ne sont pas suffisamment avancés dans ce domaine pour que l'on dispose d'une base de données de connaissances satisfaisante.

Le dernier problème consiste à traiter les données symboliques de la première étape en tenant compte des connaissances (expériences) de la deuxième pour atteindre le but fixé. La complexité du problème est telle que certains chercheurs ont trouvé refuge dans l'imitation de la vision naturelle [1]. Leur but est de comprendre et de maîtriser le mécanisme de celle-ci pour élaborer la doctrine générale de la vision artificielle. Il ne nous est pas possible de dire s'il s'agit d'une bonne approche. Dans certains cas, l'imitation de la nature si bien faite, a donné d'excellents résultats, alors que dans d'autres cas il fallait éviter cette imitation (personne n'a construit un avion qui vole en battant des ailes comme un oiseau!).

Une autre approche, plus terre à terre, consiste à limiter l'envergure des analyses de scènes souhaitées pour arriver à un système utilisable en pratique. Par exemple, le système Wisard développé par *Aleksander* en Angleterre utilise une architecture imitant le système nerveux et reconnaît les visages des personnes. On peut entraîner cette machine à reconnaître certaines différences subtiles comme par exemple la différence entre un visage souriant et le même visage renfrogné. Elle travaille à une cadence rapide de 25 images par seconde. Toutefois, il est exclu de demander à cette même machine de reconnaître le linge sale!

Revenons à notre première division du problème général en données, expérience et traitement. Dans un tel contexte, l'analyse de scènes apparaît comme le chemin qui relie les données provenant de la scène aux modèles préétablis du monde. Pour parcourir ce chemin, nous devons utiliser plusieurs représentations entre les signaux physiques traduisant la scène et les symboles cognitifs représentant les

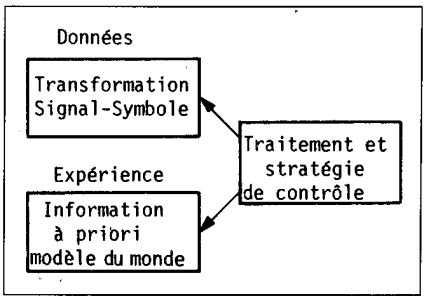


**Fig. 4 Composantes de l'analyse de scènes**

idées et les concepts. On peut les grouper en trois classes ordonnées: représentations générales (iconographiques), représentations segmentées et représentations géométriques. A ceci il faut encore ajouter au moins deux autres représentations: celle des connaissances (expérience) et celle de la stratégie de contrôle (traitement). La figure 4 résume ces différentes composantes. Il est utile de décrire chaque composante avec un peu plus de détail.

### *2.3 Transformation signal-symbole*

Cette transformation correspond à la partie la mieux connue de tout le problème. Elle part des signaux fournis par les capteurs pour engendrer, à la fin de la chaîne, les symboles qui constituent les données finales de l'analyse de scènes. Ces symboles constituent la représentation la plus compacte du contenu informationnel de la scène. A l'autre extrémité, les signaux fournis par les capteurs constituent la représentation la moins compacte, la plus redondante et la plus inefficace. La transformation signal-symbole apparaît ainsi comme une pyramide. A la base, les données brutes constituent la partie la moins dense mais en revanche la plus locale. Chaque donnée ne concerne que la luminance d'une région minuscule (ordre de grandeur millionième) de la scène. La représentation de ce niveau est le tableau des valeurs numériques que l'on appelle image numérique.

Au fur et à mesure que l'on s'élève dans la pyramide, la densité de l'information augmente au détriment de son aspect local. Selon le but fixé et les méthodes utilisées, il est possible que les détails de l'image numérique originale soient perdus en cours de route. Il est également possible de choisir des méthodes qui garantissent un retour en arrière à un niveau quelconque de cette pyramide. Plusieurs représentations jalonnent de bas en haut cette pyramide. Elles sont groupées dans les trois classes mentionnées précédemment et sont ordonnées ci-dessous, du niveau le plus bas vers le niveau le plus haut (fig. 5).

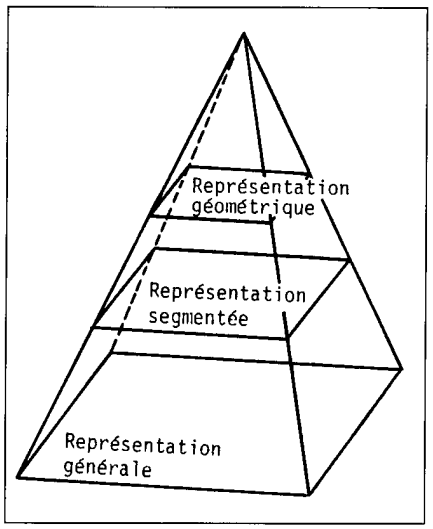


**Fig. 5 Structure hiérarchique pyramidale des représentations**

#### 2.3.1 Représentations générales

Dans ces représentations, les données fournies par les capteurs sont traitées pour les rendre plus utiles aux traitements subséquents, et par là, les alléger. L'image numérique d'entrée subit un traitement qui la transforme en une autre image numérique. Ces traitements sont des traitements des signaux pour extraire les propriétés physiques de la scène. Leur but est de renforcer ou mettre en évidence des caractéristiques nécessaires aux étapes ultérieures. Le filtrage linéaire, l'extraction de contours, la détermination, l'orientation des surfaces, l'extraction d'un objet par rapport à l'arrière-plan sont des exemples de ce genre de traitement. La figure 6 montre une image originale numérique et deux résultats de traitement. Le premier sert à déterminer les régions de luminance plus ou moins uniforme, alors que le second met en évidence les régions où se produit un changement brusque de luminance.

#### 2.3.2 Représentations segmentées

L'image segmentée s'obtient à partir d'une image de la représentation précédente en groupant les points voisins possèdant une même propriété. Les deux segments clefs dans cette représentation sont des segments naturels: les régions et leurs frontières. L'image

segmentée se présente ainsi comme un puzzle. La surface de l'image est décrite en termes de régions juxtaposées. Une des notions importantes dans la segmentation est la définition de la propriété. Les résultats dépendent très fortement de cette définition. Le but est d'obtenir une image décomposée en régions où chaque région correspond à un objet précis de la scène ou à une partie de celui-ci. Si ce résultat idéal est atteint, les frontières des régions s'identifient avec les contours des objets. Malheureusement, il ne peut être atteint que dans des cas de scènes très simples. Dans le cas général, une partie des régions obtenues ne correspond pas à des objets réels d'une scène. Cela résult des imperfections des méthodes de segmentation. Il est également possible de voir plusieurs objets groupés dans une même région, par manque de finesse. On peut les corriger partiellement par des traitements subséquents.

Les régions et leurs frontières (les segments naturels) permettent de définir un modèle très général de scène en termes de contour et de texture. Les contours sont les frontières des régions (des objets de la scène) et la texture représente l'état de surface des objets.

On peut distinguer trois méthodes principales de segmentation:

- extraction de contour,
- croissance de région,
- division et rassemblement.

Les deux dernières méthodes font l'objet des deux sections suivantes. La notion de texture intervient fortement dans les deux dernières méthodes. Pour des scènes dynamiques où l'analyse se fait à l'aide d'une séquence d'images numériques, la notion de mouvement est d'une importance primordiale. Elle sera omise dans ce texte consacré aux scènes statiques.

Les algorithmes d'*extraction de contour* sont très nombreux. Des traitements linéaires (filtrage passe-haut), aux méthodes heuristiques, relaxationnelles, ou optimisées, en passant par des opérateurs locaux, régionaux ou globaux, l'éventail est très grand. Les régions sont définies dans ce contexte comme des zones où il n'y a pas de variations brusques de la luminance (partie texturée). Toutes ces méthodes partagent le même défaut: les contours obtenus sont rarement fermés pour que l'on puisse définir toutes les régions. Des traitements subséquents sont nécessaires pour fermer les contours plus ou moins arbitrairement.

La *croissance de région* nécessite d'abord la définition d'une propriété. Cette propriété engendre ensuite les régions homogènes au sens de cette propriété. Par exemple cette propriété peut être le niveau de gris d'un point image, d'un groupe de points images, une combinaison linéaire de points images, une direction vectorielle, une énergie, une variance, etc. Une fois la propriété définie, il faut examiner, un par un, tous les points de l'image, et grouper dans une même région les points adjacents qui possèdent la même propriété, par exemple le même niveau de gris ou la même variance, la même texture, etc. Une région donnée croît ainsi au fur et à mesure que ses points voisins partagent la même propriété. La croissance s'arrête et une nouvelle région commence là où la propriété n'est plus vérifiée. La procédure est terminée quand toute l'image est analysée. Malheureusement, malgré des propriétés très complexes que l'on peut définir, le résultat n'est jamais exempt d'artefacts. Des méthodes ad hoc sont alors développées pour remédier à cette situation (voir la section 3 pour les détails).

La méthode de *division et rassemblement* procède comme suit (voir la section 4 pour les détails). Une propriété régionale est de nouveau définie en premier comme pour la croissance de région. Elle est testée sur l'image entière. Au cas ou elle n'est pas satisfaite (ce qui sera toujours le cas pour des images naturelles), l'image est divisée en un certain nombre de sous-images. Cette division peut se faire plus ou moins simplement. Par exemple, on peut diviser en 4 carrés égaux, etc. La propriété est testée dans chacune des sous-images ainsi obtenues. Chaque fois qu'elle n'est pas satisfaite, la sous-image correspondante est divisée à son tour en d'autres sous-images de deuxième niveau et ainsi de suite. La division s'arrête lorsque toutes les sous-images des différents niveaux satisfont la propriété. Le rassemblement consiste à réunir deux sous-images voisines qui satisfont la même propriété en une seule région. La propriété peut être la même que celle de la division ou une autre, moins stricte. Le rassemblement est terminé lorsqu'il n'y a plus de sous-images voisines qui partagent la même propriété. Il est clair que le résultat dépend étroitement de la (ou des) propriété(s) utilisée(s). Les frontières des régions finales correspondent plus ou moins bien aux contours des objets de la scène. Des traitements subséquents sont également nécessaires.

### 2.3.3 Représentations géométriques

Les représentations géométriques sont utilisées pour extraire les notions de forme bidimensionnelle ou tridimensionnelle. La description de ces formes doit ensuite être quantifiée. On obtient ainsi un «codage» compact de l'information acquise au niveau précédent. En plus, ces représentations peuvent servir à reconstituer les données de départ. Certaines représentations géométriques sont tellement puissantes qu'elles permettent la simulation des conditions d'éclairage et de mouvement comme c'est le cas, par exemple, dans des images artificielles de simulateurs de vols ou dans des plans d'urbanisme.

Les représentations géométriques peuvent être groupées en trois classes:

- représentation des contours (frontières),
- représentation des régions,
- représentation des formes.

Dans chaque classe, on trouve une multitude de possibilités. Une description exhaustive sortant du cadre de ce texte, ces classes seront illustrées par des exemples.

Les *frontières* des régions sont représentées par des positions dans un plan

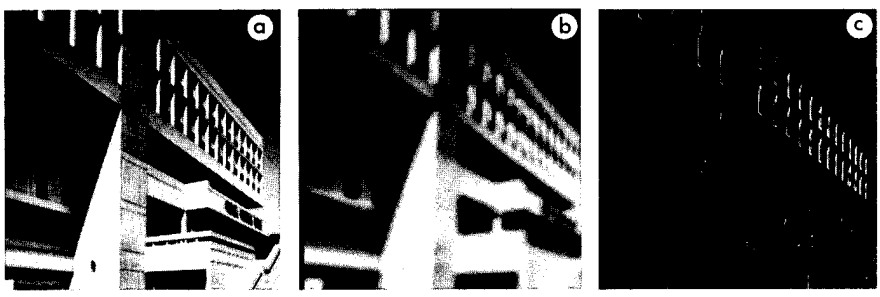


**Fig. 6 Image numérique**
a Image originale b Version filtrée passe-bas c Version filtrée passe-haut

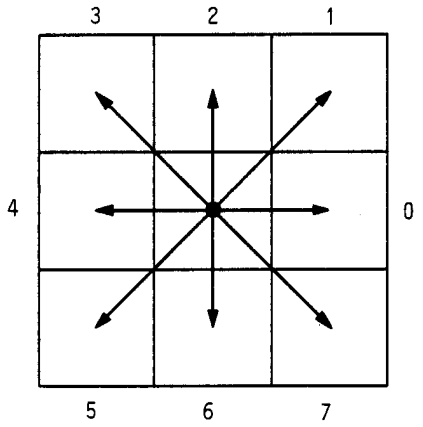


**Fig. 7 Les huit directions possibles entre deux points voisins**

où les coordonnées sont quantifiées. La nature cartésienne de cette quantification fait qu'une position ne possède que huit voisins directs. Pour décrire la suite des positions, il suffit de donner l'orientation déterminée par ces huit directions possibles (fig. 7). Une autre possibilité consiste à approcher la suite des positions par des segments de droites, d'arcs de cercle, de courbes géométriques simples. Vu la sensibilité énorme de notre système visuel à la position des contours, cette approximation, si elle est utilisée, doit être très précise. D'autres fonctions que l'on peut utiliser pour décrire les frontières sont par exemple la courbure en fonction de l'abcisse curviligne, les descripteurs de Fourier, les fonctions spline, etc.

La représentation d'une *région* peut aussi se faire de plusieurs façons. On peut distinguer les méthodes qui préservent l'information initiale de celles qui ne la préservent pas. Dans ce dernier cas, une figure géométrique est utilisée pour représenter la région. On peut citer comme exemples, la surface au sol occupée par la région, les représentations en arbres, le squelette de la région. Pour préserver l'information, on peut approcher les données initiales de la région par des surfaces de formes simples, plans, coniques, polynômes bidimensionnels, etc.

## 3. Compression par croissance de régions

### *3.1 Principes de base*

La croissance de régions [2, ..., 6] offre l'avantage de conduire à des contours fermés. Le résultat de la croissance de région est une image segmentée qui ressemble à un puzzle. Toutefois, si elle est mise en œuvre avec un souci de simplicité, les contours obtenus ne correspondent pas forcément à ceux des objets constituant l'image. Un traitement complémentaire est nécessaire pour éliminer le plus possible les faux contours.

La croissance de régions se fait de la manière suivante. Il faut tout d'abord caractériser les régions que l'on vise par une propriété. Celle-ci peut être, par exemple, le niveau de gris d'un point, l'évolution du niveau de gris ou le contenu énergétique dans une certaine bande de fréquence. Le choix de cette propriété joue un rôle important dans la complexité de la méthode et dans l'exactitude des contours obtenus après segmentation. Ensuite, en commençant par un point quelconque de l'image, on examine ses voisins pour vérifier s'ils partagent la même propriété que le point de départ. Si c'est le cas, le point correspondant est inclus dans la région et la procédure continue en examinant les voisins du nouveau point, et ainsi de suite. Lorsqu'il n'y a plus de points connexes à la région qui possèdent la même propriété, la croissance s'arrête et on examine les autres points pour définir les autres régions. La segmentation est achevée quand tous les points de l'image ont été attribués à une région.

La propriété choisie dans cette méthode est très simple. C'est l'appartenance à un intervalle de niveaux de gris. Cet intervalle est centré autour de la valeur du point de départ. La région est définie par tous les points dont le niveau de gris tombe dans cet intervalle. Pour inclure dans une même région le plus de points possibles, l'intervalle peut se déplacer sur les niveaux de gris, à condition de ne pas perdre un point précédemment inclus dans la région.

Les images originales contiennent des contours et des textures. Dans les parties texturées, il y a beaucoup de faibles variations du niveau de gris. L'application directe de la croissance de région, avec comme propriété l'appartenance à l'intervalle de niveau de gris, conduit à un grand nombre de régions avec beaucoup de faux contours. C'est pourquoi l'image subit d'abord un prétraitement qui cherche à atténuer le plus possible les faibles variations tout en maintenant intacts les contours. Ce traitement peut être réalisé avec plusieurs filtres: filtre gradient inverse, filtre médian, filtre de Nagao [7], etc. Dans notre cas, c'est le filtre gradient inverse qui a été choisi. Les

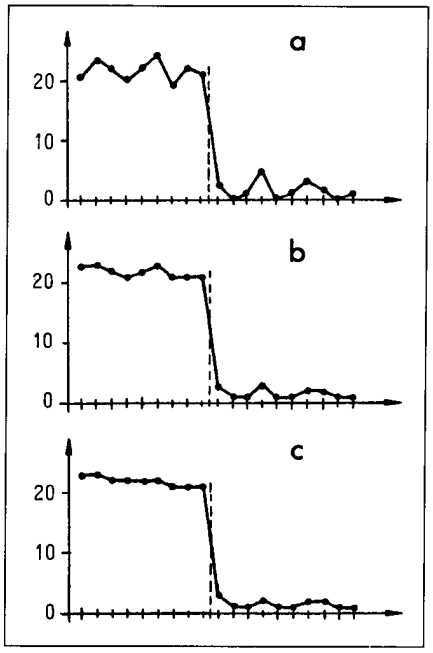


**Fig. 8 Filtrage avec un filtre gradient inverse**
a Signal original
b Résultat après une itération
c Résultat après deux itérations

coefficients de ce filtre sont inversement proportionnels au gradient local de l'image originale. Le résultat obtenu avec ce filtre pour un signal unidimensionnel est montré à la figure 8.

Après ce prétraitement, on peut appliquer la croissance de région aux images originales des figures 9a, b, c. Les résultats obtenus sont montrés aux figures 9 d, e, f. On remarque que la croissance de régions produit, en même temps que les régions, deux types d'artefacts: des contours qui ne séparent pas complètement deux régions et des contours d'épaisseur double (régions sans points intérieurs). Ces artefacts sont illustrés à la figure 10. Un traitement subséquent est

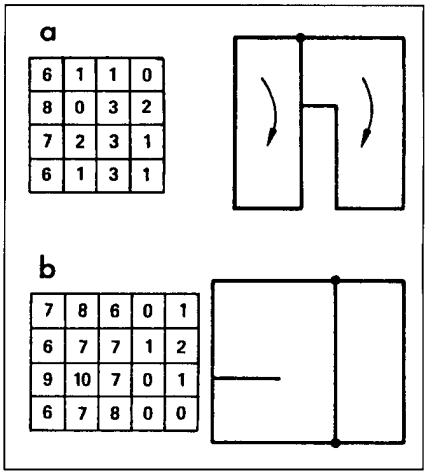


**Fig. 10 Illustration des deux artefacts de la croissance de régions**

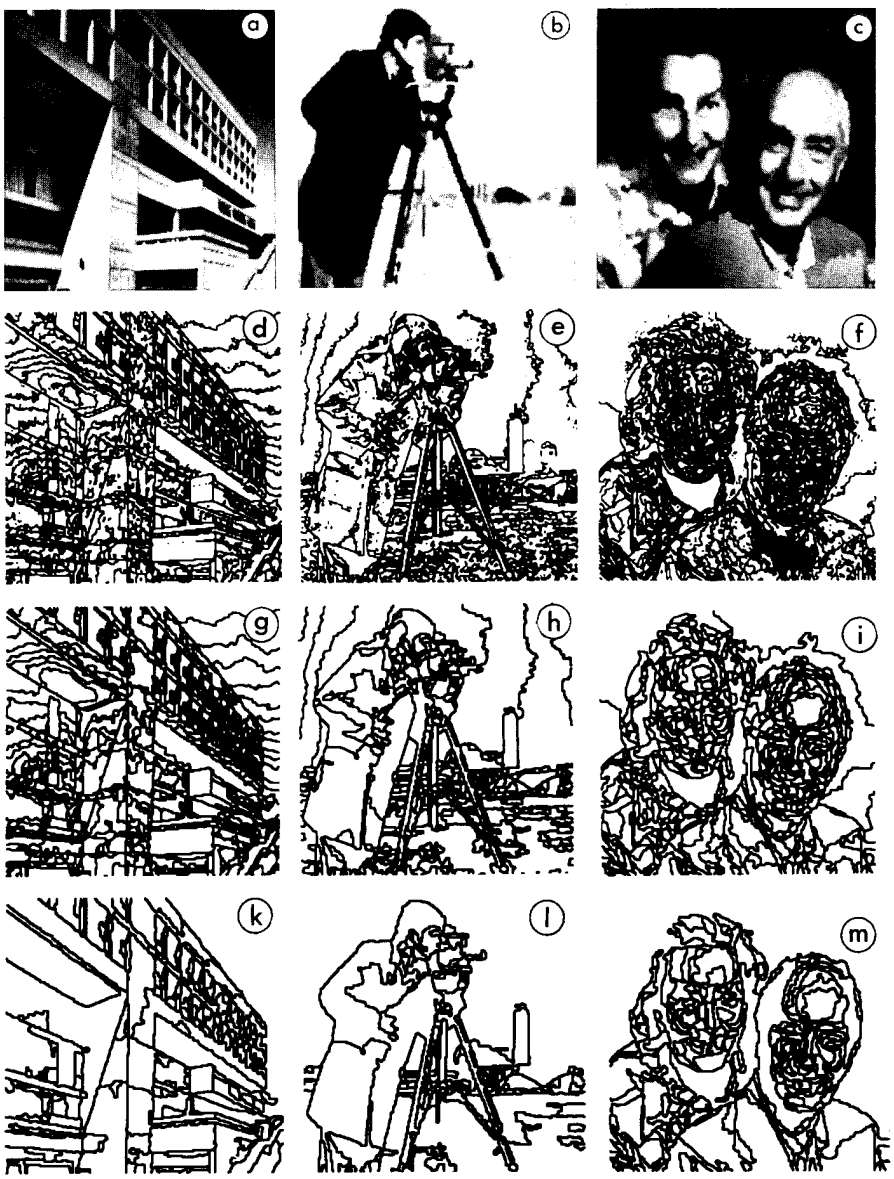

**Fig. 9 Croissance de région**

a, b, c Images originales numériques
points images: 256×256
niveaux de gris: 256

d, e, f Résultat de la croissance de région
g, h, i Résultat obtenu après suppression des artefacts
k, l, m Résultat final de la segmentation

éliminer les faux contours entre deux régions adjacentes, il faut calculer le gradient moyen dans l'image originale le long d'un tel contour. Si l'amplitude de ce gradient moyen est plus faible qu'un seuil, on réunit les deux régions. Ces deux opérations sont illustrées à la figure 11. Elles complètent la procédure de segmentation. Les figures 9 k, l, m montrent les résultats correspondant aux images réelles. Chaque région ainsi obtenue est représentée par un niveau de gris constant qui est la valeur moyenne de la luminance dans cette région. A ce niveau, on dispose d'une image qui ressemble aux dessins peints par des numéros.

### *3.2 Codage des contours*

Les frontières des régions obtenues après segmentation sont les contours cherchés. Elles doivent être codées efficacement. Toutefois, la frontière séparant deux régions adjacentes apparaît comme deux contours, un pour chaque région. Avant le codage, il faut donc réduire encore cette redondance. Les contours des objets naturels sont des courbes à variation plutôt lente. C'est pourquoi une des possibilités de les coder consiste à utiliser la méthode suivante à trois modes:

- approximation par segment de droite
- approximation par arc de cercle
- sans approximation

Selon le coût en bits de chaque mode, on choisit celui qui est le plus économique. Les approximations introduisent une certaine distorsion que l'on maintient à l'intérieur d'une bande centrée sur le contour. La largeur de cette bande détermine la qualité de l'approximation. Elle est en principe

alors appliqué pour les éliminer. Les figures 9 g, h, i montrent le résultat de ce traitement sur les images des figures 9 d, e, f.

A ce niveau on obtient une image segmentée avec des contours fermés d'épaisseur unité. C'est l'image équivalente à un puzzle (fig. 9 g, h, i). Toutefois, dans ces résultats, le nombre de régions de petite taille ou avec des faux contours est très élevé. En décidant d'introduire une certaine distorsion dans l'image décodée que l'on maintiendra à un niveau acceptable, on peut éliminer les régions ne contenant pas plus d'une dizaine de points. Pour

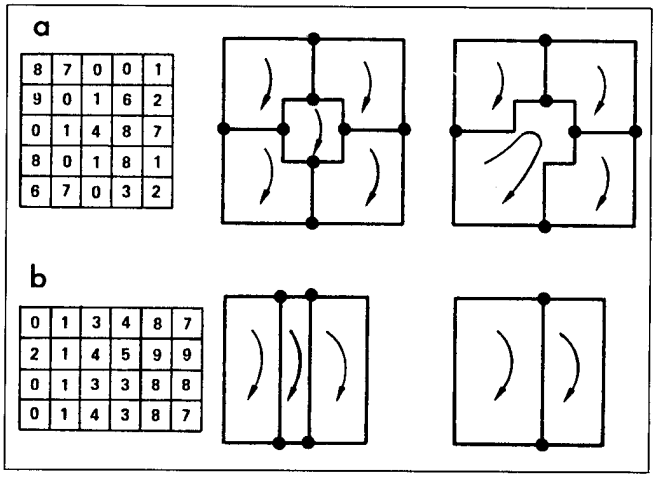


**Fig. 11**
**Suppression des petites régions et réunion des régions à faible gradient moyen**

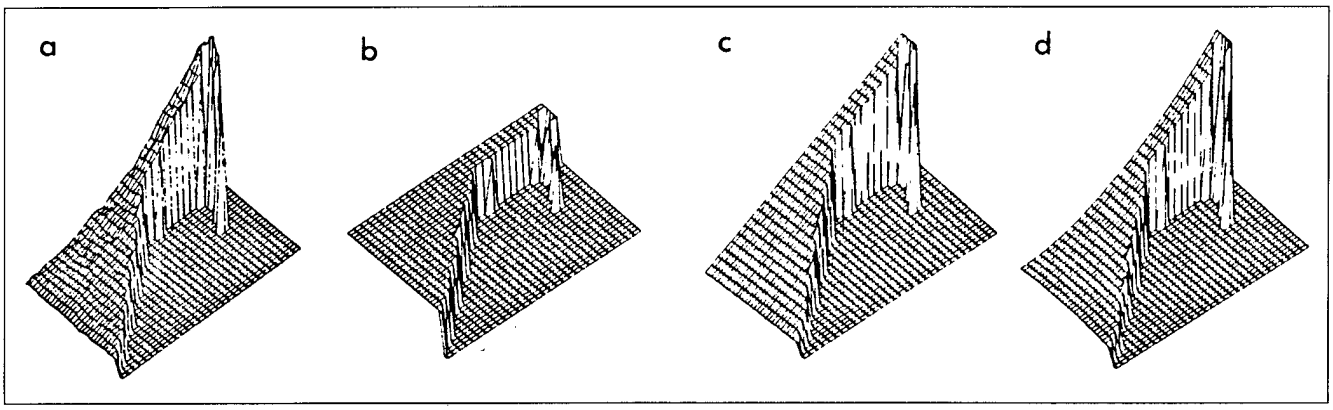


**Fig. 12 Les données de la région du ciel et ses trois approximations**
a Fonction originale b, c, d Approximations

maintenue à un intervalle d'échantillonnage.

Dans les cas où la distorsion introduite par cette approximation n'est pas acceptable, on peut utiliser une technique de codage sans distorsion [8] qui n'utilise que 1,3 bit par point de contour.

### *3.3 Codage de la texture*

La différence entre l'image originale et l'image obtenue après la segmentation est considérée comme une image de texture. Celle-ci possède deux composantes: une composante déterministe qui fournit une description concise de l'évolution globale du niveau de gris et une composante aléatoire qui tient compte de la granularité texturale des régions. Ces deux composantes sont traitées et codées séparément.

La composante déterministe est représentée par un ensemble de fonctions polynômiales d'ordre $n$ à deux variables spatiales. Ce choix se justifie par la segmentation utilisée qui ne laisse pas de discontinuités marquées à l'intérieur des régions et par l'évolution lente et continue de ces fonctions qui s'adaptent bien au problème posé. En limitant le degré des polynômes à $n = 2$, on cherche pour chaque région le polynôme qui approxime le mieux son niveau de gris. Selon le critère de l'erreur quadratique moyenne, cette procédure permet de choisir des polynômes d'ordre 0,1 ou 2. La figure 12 a montre les niveaux de gris de la région du ciel dans l'image du bâtiment et ses trois approximations possibles (fig. 12b, c, d).

La composante aléatoire de la texture représente la granularité de chaque région. Elle est obtenue par la différence entre l'image originale et l'image filtrée par le filtre de prétraitement (filtre gradient inverse). A l'intérieur de chaque région, cette composante est modélisée par une réalisation d'un processus aléatoire de puissance fixée. Il suffit ainsi d'un seul paramètre (puissance) pour la décrire entièrement. Le signal utilisé ici est un signal aléatoire de densité de probabilité gaussienne à valeur moyenne nulle.


**Fig. 13 Résultats de codage par croissance de régions**
Les paramètres sont:
$P$ Nombre de points de contours $N_i$ Nombre de régions d'ordre $i$ $C$ Compression
a P = 21,210, $N_0$ = 182, $N_1$ = 59, $N_2$ = 21, C = 29 à 1
b P = 16,448, $N_0$ = 160, $N_1$ = 38, $N_2$ = 8, C = 38 à 1
c P = 25,352, $N_0$ = 249, $N_1$ = 91, $N_2$ = 41, C = 22 à 1

La méthode de compression basée sur la croissance de région, telle qu'elle est décrite ci-dessus, est une méthode paramétrique. En effet le nombre de régions et la précision de leurs descriptions (contour et texture) peuvent être modifiés par des paramètres. Ainsi, il est possible de modifier le rapport de compression et la qualité de l'image décodée. Les figures 13 et 14 montrent les résultats de codage correspondant à différents rapports de compression.

### *3.4 Remarque*

La méthode décrite ci-dessus n'est pas optimisée. Elle ne conduit pas au même résultat si le point de départ pour la croissance de région est diffé-


**Fig. 14 Résultats de codage par croissance régions**
Les paramètres sont:
$P$ Nombre de points de contours $N_i$ Nombre de régions d'ordre $i$ $C$ Compression
a $P$ = 18,046, $N_0$ = 129, $N_1$ = 40, $N_2$ = 10, $C$ = 38 à 1
b $P$ = 14,590, $N_0$ = 122, $N_1$ = 35, $N_2$ = 5, $C$ = 44 à 1
c $P$ = 22,350, $N_0$ = 179, $N_1$ = 72, $N_2$ = 31, $C$ = 26 à 1

rent. Même si les variations en fonction du point de départ sont supposées être faibles, il est utile de se rendre indépendant de ce point. Pour celà on fait appel à la théorie des graphes. L'image initiale est divisée en petites régions de taille identique 2 par 2 ou 3 par 3. Chaque région ainsi obtenue est représentée par un point dans un graphe. Les lignes reliant ces points représentent les liens entre les régions correspondantes. Deux régions sont unifiées si leur lien est le plus fort. Cette procédure est effectuée jusqu'à ce qu'un seuil dans l'erreur d'approximation soit atteint [9]. La méthode de division et rassemblement de la section suivante utilise aussi cette idée ainsi qu'il est décrit au paragraphe 4.6.

## 4. Compression par division et rassemblement

### *4.1 Généralités*

Le concept de division et rassemblement adaptatif est une autre alternative pour la segmentation. Dans un premier temps, l'image originale est divisée successivement en un ensemble de carrés de tailles différentes [10; 11]. Les données sont approximées à l'intérieur de chaque carré. Le processus de subdivision se termine dès qu'un critère de qualité est atteint. Dans une deuxième phase, les carrés adjacents sont regroupés si le signal approximé de l'ensemble est acceptable.

Cette méthode semble être assez proche du comportement humain dans l'analyse d'une scène naturelle. Elle part d'une approche globale pour aller vers le détail et le particulier. Enfin, différentes zones sont associées sur la base d'une mesure de similarité.

Dans les paragraphes suivants, les buts visés, les aspects liés à l'approximation, le prétraitement des données originales de manière à faciliter l'étape de segmentation sont analysés. On verra également plus en détails, la technique de division, la méthode de rassemblement ainsi que le post-traitement qui peut être imaginé de manière à améliorer la qualité des résultats obtenus. Les différentes étapes de cet algorithme de segmentation sont résumées à la figure 15.

### *4.2 Buts visés et exigences du problème*

Dans ce paragraphe, le problème est défini en termes généraux et les exigences pour la segmentation sont formulées. Dans le contexte général du modèle contour-texture [1; 4], le but est de diviser l'image en une série de régions dont les frontières correspondent aux contours réels des objets contenus dans l'image.

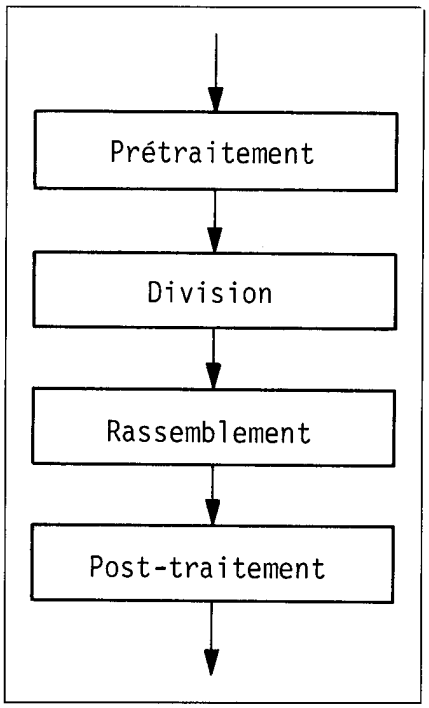


**Fig. 15 Etapes de la méthode**

En relation avec les propriétés du système visuel humain, si l'on admet pouvoir dégrader en partie la scène naturelle observée, il faut néanmoins maintenir une description minutieuse de la position des frontières de chaque région. Dans le cadre d'applications liées au codage d'images, la texture contenue dans chaque partie segmentée peut être reproduite avec une certaine erreur. En effet, une partie importante de l'information sémantique apparaît dans la position des contours.

La méthode de segmentation devra donc remplir le cahier des charges suivant:

1. l'image est divisée en un ensemble de régions correspondant aux zones à variation lente de celle-ci,
2. le nombre de ces régions doit être réduit autant que possible, sans toutefois détruire des contours réels apparaissant dans l'image,
3. l'erreur entre la luminance originale et la luminance approchée doit être maintenue, pour chaque partie segmentée, en dessous d'un seuil limite; ceci pour assurer une qualité visuelle acceptable,
4. la précision nécessaire à décrire les contours des objets doit rester de l'ordre du point image.

### *4.3 Aspects liés à l'approximation*

Dans ce paragraphe, différents points liés à l'approximation sont abordés, tels que le critère d'erreur choisi, le type et le nombre de fonctions approximantes utilisés.

Ce problème est analysé en détail car les résultats de la segmentation sont fortement dépendants du processus d'approximation utilisé pour représenter les données à l'intérieur de chaque région et vice-versa. Considérons un ensemble de $r$ fonctions approximantes $\psi_i(u, v)$. Il existe une façon optimale de segmenter la luminance originale représentée par le signal original avec cet ensemble de fonctions. La meilleure approximation au sens des moindres carrés ($L^2$)

$$\hat{g}(u, v) = \sum_i \alpha_i \psi_i(u, v) \qquad (1)$$

est évaluée pour la région courante. Cette région définit le domaine

$$D = \{(k_i, l_i);\ i = 1,..,N\}$$

contenant les $N$ positions des points images prenant les valeurs $g(k_i,l_i)$. ($N$ peut bien entendu être différent d'une région à l'autre). Les coefficients $\alpha_i$ sont choisis de manière à minimiser l'erreur quadratique sur la région. Ce critère a été choisi pour les raisons suivantes:

- Il n'est pas sensible à des distorsions locales présentes dans les données. Même si de telles altérations peuvent correspondre à des contours réels de l'image, l'utilisation d'une mesure d'erreur le long de ces contours permettra d'obtenir une autre partition de l'image (division ou non-association selon l'étape de segmentation impliquée). Le critère des moindres carrés garantira néanmoins une bonne adéquation entre les données originales et les données approximées.
- Le problème a une solution unique pour autant que les fonctions $\psi_i$ $(u, v)$ soient linéairement indépendantes et que le nombre de points images $N$ soit plus grand ou égal au nombre $r$ de fonctions approximates.
- Il est plus simple à calculer que d'autres types d'approximation.

Considérons maintenant les aspects liés à la sorte et au nombre de fonctions approximantes. La seule restriction est de choisir des fonctions linéairement indépendantes. Différentes études [9; 12] ont déjà montré les avan-

tages à utiliser des fonctions polynômiales pour représenter des surfaces à variation lente. Des considérations qualitatives relatives à la ressemblance entre les polynômes bidimensionnels et des parties de scènes naturelles montrent que la plupart des formes peuvent être fidèlement reproduites avec des polynômes de degré pas trop élevés.

Le choix de fonctions orthogonales n'a pas été considéré jusqu'ici, car il est difficile de les définir sur des domaines de forme quelconque. En outre, il existe de bonnes raisons pour croire que l'approximation présentera des phénomènes d'oscillations très désagréables sur le plan visuel si l'erreur quadratique est non nulle (cf. *Hadamard, Fourier...*).

### *4.4 Prétraitement*

Pour satisfaire la première des exigences formulées par le cahier des charges, il faut dans un premier temps appliquer un préfiltrage aux données afin de faciliter l'opération de segmentation elle-même. Il s'agit donc d'éliminer le bruit de granularité, ou les régions à trop forte agitation tout en préservant la position exacte des contours. La figure 16 montre l'effet du filtre de Nagao sur une image représentant un caméraman. On remarque que même si certains détails sont perdus au niveau de la caméra, le bruit de granularité apparaissant au niveau du gazon est pratiquement éliminé. En outre, les contours forts sont presque tous préservés par cette opération de filtrage.

C'est également dans cette étape qu'est insérée l'extraction d'images de contrôle facilitant l'opération de segmentation. On peut ainsi estimer ces images de manière à garantir l'exactitude de la position des objets de l'image durant la segmentation. On utilise

**Fig. 16 Image filtrée par l'opérateur de Nagao**
après une itération selon la méthode proposée [14]
Nombre de points inchangés: 12 191
Taille de la fenêtre d'analyse: 5×5

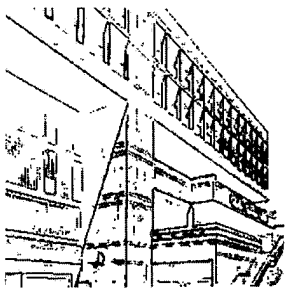

**Fig. 17 Image de contrôle des contours du bâtiment obtenue par filtrage directionnel**

de préférence des opérateurs tenant compte des propriétés du système visuel humain: *Marr-Hildreth* [13], *Canny* [14], filtrage directionnel [15]. La figure 17 montre l'image de contrôle obtenue par filtrage directionnel à partir de l'image du bâtiment.

### *4.5 Processus de division*

Ce paragraphe décrit brièvement l'opération de division. L'attention est portée sur les critères d'erreur utilisés pour contrôler la segmentation. Des résultats sont présentés avec différents types de polynômes.

Pour décider si une certaine partition de l'image est acceptable, on extrait deux indices de qualité. Le premier définit une qualité globale de l'approximation pour la région considérée. Il est basé sur une mesure de l'erreur quadratique moyenne. Le deuxième indice consiste à mesurer l'erreur le long de la position des contours présents à l'intérieur de la région considérée. Dès que l'un ou l'autre de ces deux indices dénote une qualité insuffisante, la région considérée n'est pas approchée correctement. Il faut donc changer la partition initiale de l'image.

Supposons une image carrée de taille $2^q \times 2^q$. Admettons également que l'on désire approximer au sens des moindres carrés le signal image par une certaine fonction bidimensionnelle. Cette approximation est donc dans un premier temps calculée sur toute l'image. Si l'un des deux indices de qualité est jugé insuffisant, l'approximation est calculée sur chacune des sous-images $2^{q-1} \cdot 2^{q-1}$ obtenue par division de l'image originale. Le procédé est répété jusqu'à ce que les mesures de qualité deviennent satisfaisantes, réduisant ainsi l'image à une série de carrés de différentes tailles. Cette étape de la segmentation doit nécessairement se terminer car l'ensemble des fonctions $\psi_i$ va interpoler les points images dès que $r$ devient plus grand que $N$, garantissant ainsi une erreur nulle.

Après cette étape de segmentation, le signal bidimensionnel est donc représenté:

1. par la position des différents carrés de taille $2^i \times 2^i (i = 1,..,q)$
2. par les coefficients d'approximation $\alpha_i$ pour chacune de ces régions.

Analysons maintenant comment peut être codée l'information image mise sous une telle forme (position plus taille plus approximation). En tenant compte de contraintes géométriques, on a pu montrer que la structure du graphe représentant la répartition de ces carrés pouvait être ramenée à un quadtree [16]. Le codage de position et de taille a pu ainsi être réduit à environ un demi-bit par carré. Pour ce qui est des paramètres d'approximation, il est difficile d'estimer la quantification nécessaire des coefficients $\alpha_i$. Pour une estimation de l'efficacité d'une telle représentation du signal original, un code simple a été dérivé pour une approximation d'ordre zéro ($\psi_1 = 1$). Le coefficient $\alpha_i$ doit dans ce cas se trouver entre 0 et 255, le signal original ayant été quantifié à huit bits. Des facteurs de compression allant de 30 à 1 à 60 à 1 ont pu être ainsi atteints selon le type d'images à coder (voir figure 18 d).

Pour illustrer cette méthode, des exemples sont présentés pour deux images différentes. Aux figures 18 a, b, c, d, la segmentation est évaluée sur une image de portrait en utilisant une approximation par niveaux constants, donnant une partition en 5335 carrés. (La plus petite taille de carré a été définie égale à 2×2.) Les figures 18 e, f présentent une partition du portrait avec 4615 carrés où chacun d'entre eux est approximé par un plan, donc trois coefficients d'approximation. Les figures 18 g, h montrent le résultat de l'algorithme de division appliqué à une image de bâtiment lorsqu'on utilise une approximation polynômiale séparable du premier ordre ($\psi_1 = 1$, $\psi_2 = u$, $\psi_3 = v$, $\psi_4 = u\,v$). Quatre coefficients sont nécessaires pour représenter chacune des 6655 régions.

### *4.6 Processus de rassemblement*

Ce paragraphe décrit le regroupement des différentes régions obtenues par l'algorithme de division de manière à obtenir une meilleure segmentation de l'image originale.

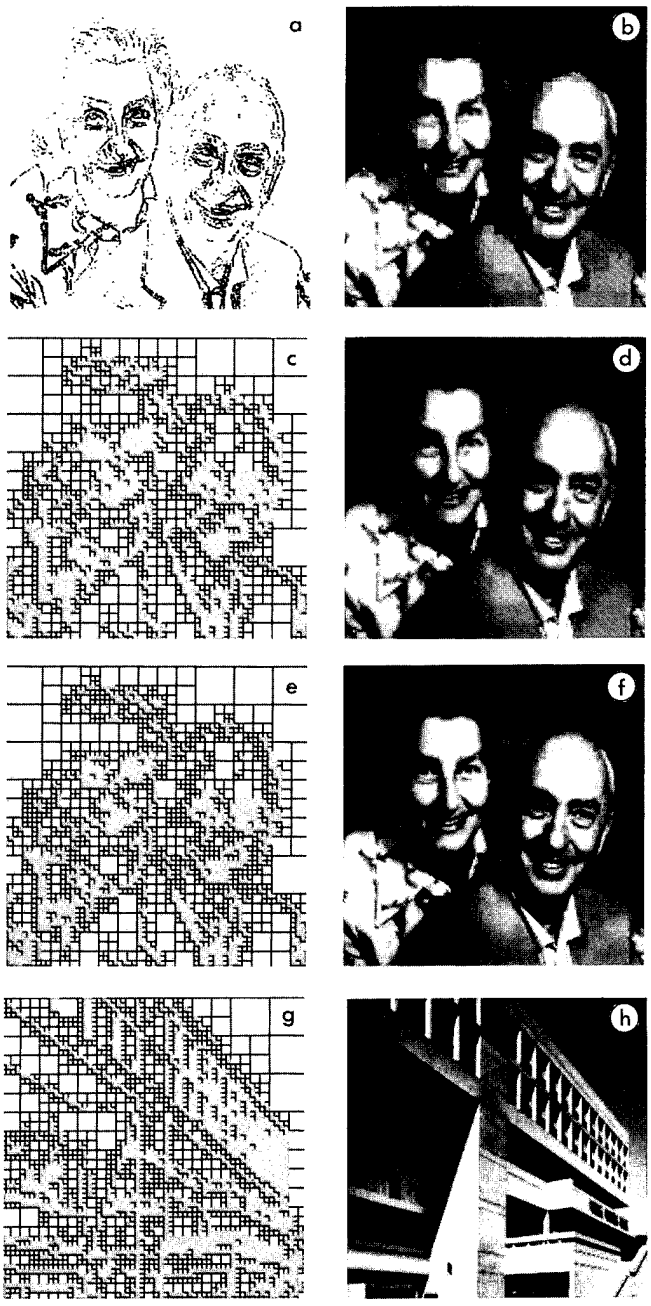


**Fig. 18**
**Processus de division**

a Image de contrôle des contours du portrait
b Image approximée du portrait après l'opération de division, 5335 régions, plus petite taille admise 2×2, approximation par niveaux constants
c Position des carrés correspondant au portrait, les carrés de taille 2×2 sont indiqués en gris moyen
d Image de la figure 18b après codage et post-traitement pour rehausser la qualité, compression 50 à 1
e Position des carrés de la figure 18f, les, carrés de taille 2×2 sont indiqués en gris moyen
f Image du portrait approximée par une série de plans après division, 4615 carrés
g Position des carrés de la figure 18h, les carrés de taille 2×2 sont indiqués en gris moyen
h Image du bâtiment approximée par une série de fonctions polynômiales séparables du premier ordre

Cet algorithme a été obtenu à partir d'une technique adaptative de croissance de région [9]. Quelle que soit la méthode de segmentation choisie, il est nécessaire de définir une structure de données appropriée pour pouvoir accéder ou regrouper facilement les différentes régions. Pour associer les différents carrés résultant de l'opération de division, on a choisi de représenter les données par un graphe de contiguité de régions (GCR). Il s'agit là d'une structure de données classique où chaque nœud représente une région et chaque branche deux régions adjacentes. Seuls des carrés contigus ont été considérés car il faut garantir la connexité des régions obtenues dans la partition finale. La figure 19 montre comment on passe d'une représentation matricielle à une représentation par GCR des données.

Selon ce processus, on commence par associer à chaque branche du GCR une valeur représentant le degré de dissimilarité qui existe entre les deux nœuds. Ce degré de dissimilarité constitue une mesure de qualité de l'approximation. Dans un deuxième temps, la branche présentant la plus faible mesure de dissimilarité est supprimée et les nœuds qu'elle relie sont rassemblés. Les deux régions ont ainsi été regroupées en une seule. La meilleure approximation au sens des moindres carrés est estimée sur cette nouvelle région. Ce processus est répété jusqu'à ce qu'un certain critère soit atteint. Bien entendu, à chaque étape du regroupement, les mesures de dissimilarité qui étaient associées aux branches connectées à l'une des deux régions impliquées, sont réévaluées. On garantit ainsi une segmentation optimale, qui s'adapte au mieux aux configurations intermédiaires. Sur la figure 19, on a illustré une étape de rassemblement. La valeur $E$ indique la mesure de dissimilarité la plus faible.

Deux critères permettent d'arrêter le processus de rassemblement:
- on impose un nombre minimal de régions, donc un certain rapport de compression
- on fixe un seuil sur la qualité qui doit exister en chaque région. Dès qu'il n'existe plus de mesure de dissimilarité en dessous de ce seuil, le processus de segmentation est interrompu. On limite ainsi la dégradation de l'image. Le choix de ce seuil

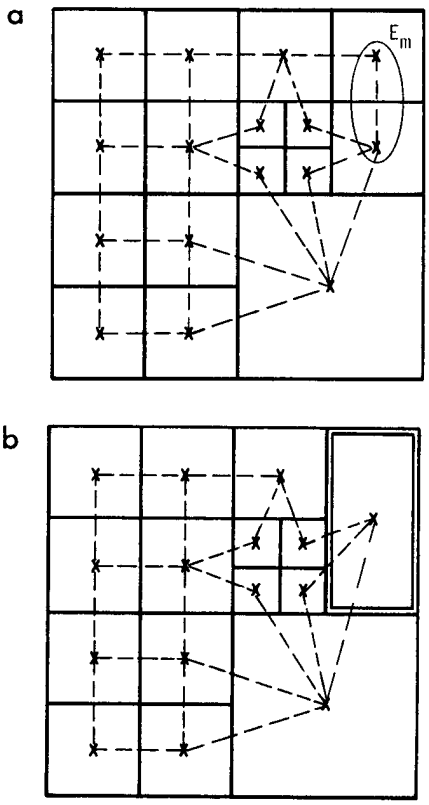


**Fig. 19 Processus de rassemblement à l'aide de graphes de contiguité de régions**

—— Représentation matricielle
---- Représentation par graphe de contiguité de régions (GCR)
a Exemple de représentation matricielle après l'opération de division et représentation par GCR correspondant où $E_m$ est la plus faible des mesures de dissimilarités associées à chaque branche du graphe
b Représentation par GCR et matricielle après la première étape de rassemblement

**Fig. 20 Résultat de la méthode adaptive de division et rassemblement**

a Image du bâtiment approximée par une série de plans après division et rassemblement, 999 régions
b Image du portrait approximée par une série de plans après division et rassemblement, 999 régions
c Image du caméraman approximée par une série de plans après division et rassemblement, 999 régions

peut être fixé de manière similaire à celle proposée dans l'opération de division.

Le choix du GCR initial est essentiel car il permettra d'atteindre la précision requise pour représenter les frontières des régions. En fait, il est impossible d'atteindre le niveau de résolution demandé, si l'on utilise directement la partition résultant du processus de division, excepté pour le cas d'une approximation d'ordre zéro. En effet, il faudrait pouvoir arrêter la segmentation résultant de l'opération de division au niveau du point image. Or cette opération s'arrête en tout cas dès que l'ensemble des fonctions approximantes interpole le signal original. Pour faire correspondre les frontières des parties segmentées avec les contours réels de l'image, on admet de pouvoir associer des zones ne contenant pas de contours avec des parties de régions qui contiennent des contours. Le contrôle de la présence de tels contours est effectué en relation avec les images de contrôle. Une contrainte est inserrée au processus de rassemblement: des régions ne définissant que des points images ne peuvent être associées qu'à des régions contenant au moins $r$ points. A la fin du processus de rassemblement, il est néanmoins possible d'avoir une série de points images isolés. On peut admettre d'insérer ces points images dans des régions avoisinantes sans perte sensible d'information. S'il reste plus de $r$ points images contigus entre eux dans la partition finale, on peut les approximer au moyen des $r$ fonctions approximantes.

Les figures 20a, b, c présentent le résultat de la méthode adaptative de division et rassemblement appliquée aux images originales de la figure 9 lorsqu'on utilise une approximation par plans pour représenter chaque région. Entre l'étape de division et celle de rassemblement, le nombre de régions a été reduit de 6688/4615/4315 respectivement à 999. On a ainsi obtenu une forte réduction de redondance d'information sans perte appréciable de qualité. On peut également comparer ces résultats à ceux de la méthode de croissance de régions lorsqu'on utilise des niveaux constants pour représenter chaque région (voir [4], pp. 103, 106), avec un nombre de régions du même ordre de grandeur. Dans les résultats présentés ci-dessus, on n'a pas tenu compte des régions qui contenaient des points contours. Il est raisonnable d'admettre que les points contours seront le plus souvent associés à des régions voisines, ce qui n'entraînera pas une augmentation sensible du nombre de régions.

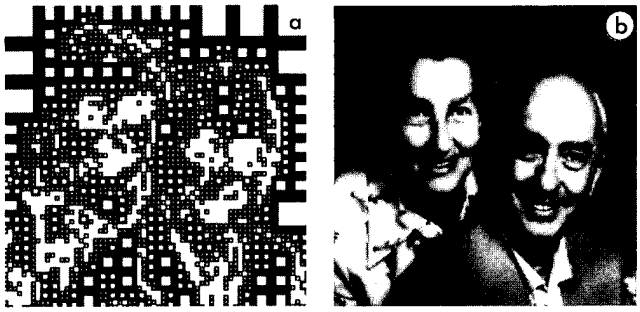

**Fig. 21 Post-traitement**

a Image de contrôle permettant de définir les régions à filtrer pour le rehaussement de l'image du portrait correspondant à la figure 18e
b Image de cette figure rehaussée après application de l'algorithme de post-traitement

### *4.7 Posttraitement*

On présente ici une méthode de rehaussement permettant d'améliorer la qualité de l'approximation lorsqu'on utilise des fonctions polynômiales bidimensionnelles. A cause de la structure de ces fonctions, le signal approximé peut présenter des discontinuités entre régions adjacentes. De faux contours peuvent ainsi apparaître, créant une forte dégradation de la qualité de l'image approximée. Par comparaison avec l'image de contrôle des contours, il est possible de trouver ceux que l'on doit tâcher d'éliminer par cette opération de filtrage. Un algorithme de moyennage ou de relaxation [9] est appliqué de part et d'autre de chacun de ces faux contours. La largeur sur laquelle on applique ce filtrage est rendue dépendante de la taille de la région. Cet algorithme de rehaussement peut bien entendu être appliqué juste après l'algorithme de division. La figure 21 montre le rehaussement de la qualité de l'image correspondant à la figure 18e.

(La suite de cet article sera présentée dans le numéro 21/1986.)

**Literatur**


[1] *M. Kunt, A. Ikonomopoulos* and *M. Kocher:* Second-generations image coding techniques. Proc. IEEE 73(1985)4, p. 549...574.

[2] *M. Kocher* and *M. Kunt:* A contour-texture approach to picture coding. IEEE International Conference on Acoustics, Speech, and Signal Processing, Paris, 1982; vol. 1, p. 436...439.

[3] *M. Kocher* and *M. Kunt:* Image data compression by contour-texture modelling. SPIE International Conference on the Application of Digital Image Processing, Geneva, April 1983, p. 131...139.

[4] *M. Kocher:* Codage d'images à haute compression basé sur un modèle contour-texture. Thèse de l'EPFL N° 476, 1983.

[5] *A. Ikonomopoulos, M. Kocher* and *M. Kunt:* Image coding based on human visual system properties for otimal reduction of redundancy. Proceedings of the third Scandinavian Conference on Image Analysis, Copenhagen/Denmark, July 12...14, 1983, p. 216...222.

[6] *M. Kocher* and *M. Kunt:* A contour-texture aproach to picture coding. Proceedings of the Mediterranean Electrotechnical Conference, Athens/Greece, May 24...26, 1983 (Melecon-83); vol. 2, paper C 2.03.

[7] *M. Nagao* and *T. Matsuyama:* Edge preserving smoothing. Computer Graphics and Image Processing 9(1979)1, p. 394...407.

[8] *M. Eden* and *M. Kocher:* On the performance of a contour coding algorithm in the context of image coding. Signal Processing 8(1985)4, p. 381...386.

[9] *M. Kocher* and *R. Leonardi:* Adaptive region growing technique using polynominal functions for image approximation. Signal Processing 11(1986)1.

[10] *R. Leonardi* and *M. Kunt:* Adaptive split for image coding. IASTED International Symposium on Applied Signal Processing and Digital Filtering, Paris/France, June, 19...21, 1985.

[11] *R. Leonardi* and *M. Kunt:* Adaptive split-and-merge for image analysis and coding. Proceedings of the SPIE-Conference on Image Coding, Cannes/France, December 2...6, 1985.

[12] *M. Eden, M. Unser* and *R. Leonardi:* Polynominal representation of pictures, Signal Processing 10(1986)June.

[13] *D. Marr* and *E. Hildreth:* Theory of edge detection. Proceedings of the Royal Society B 207(1980), p. 187..217.

[14] *J. F. Canny:* Finding edges and lines in images. Technical Report of the Artificial Intelligence Laboratory. Cambridge/USA, Massachusetts Institute of Technology, 1983.

[15] *A. Ikonomopoulos* and *M. Kunt:* High compression image coding via directional filtering. Signal Processing 8(1985)2, p. 179...205.