

Une extension d'un algorithme proposé par J.L. Marroquin pour la restauration d'image de contours incomplets. Application à des images réelles

An Extension of an Algorithm Proposed by J.L. Marroquin for the Restoration of Incomplete Contour Images Application to Real Images

par Sabine URAGO, Marc BERTHOD, Josiane ZERUBIA

INRIA - 2004 Route des Lucioles
BP 93, 06902 Sophia Antipolis Cedex - France
Tél : (33) 93 65 78 57 - Fax : (33) 93 65 76 43.

Résumé

Le propos de cet article est d'étudier une méthode initialement proposée par J.L. Marroquin, permettant la restauration d'images de contours incomplets.

Nous avons modifié cet algorithme afin de l'étendre au traitement de données bruitées et à des scènes réelles. En effet, les résultats donnés par J.L. Marroquin présentent uniquement le traitement d'images synthétiques.

Le principe de la restauration est basé sur une modélisation Markovienne et des distributions de Gibbs.

Certains critères à optimiser sont donc déterminés afin de compléter les parties manquantes.

La mise en œuvre d'une méthode déterministe (« Iterated Conditional Mode ») ou d'une méthode stochastique (échantillonneur de Gibbs) de relaxation permet d'aboutir à une configuration stable dans laquelle les contours sont complétés.

Pour illustrer cette méthode, nous présentons des résultats sur plusieurs types d'images : des images synthétiques bruitées et des images réelles (d'intérieur, industrielles et SPOT).

Mots clés : détection de contours, champs de Markov, distribution de Gibbs.

Abstract

In this paper, we study an algorithm initially proposed by J.L. Marroquin, which allows the restoration of images with incomplete contours. Our study shows ways to modify and extend this method to enable the processing of real and noisy data. The results given by J.L. Marroquin, present only the restoration of synthetic images.

The principle of the restoration is based on a Markovian model and a global measure : a Gibbs distribution.

In order to restore the contours we define some criterions which have to be optimized. A deterministic (« Iterated Conditional Mode ») or a stochastic (Gibbs sampler) relaxation algorithm generates a configuration in which the contours are completed.

To illustrate this algorithm, several examples are given including synthetic, noisy and real images (indoors, industrial and SPOT images).

Key words : Contour detection, Markov Random Fields, Gibbs distribution.

1. Introduction

Les méthodes de détections de contours classiques ne permettant pas toujours des extractions de contours complets (cf [3], [8], [15], [16], [4],[7]), il est intéressant de pouvoir restaurer ces frontières.

Nous avons tout d'abord analysé un **algorithme initialement proposé par J.L. Marroquin** en 1989 [11], [12], permettant **la restauration d'images de contours incomplets**. Puis, nous l'avons modifié et étendu (cf [13] pour plus de détails), afin de pouvoir traiter des **images réelles** (en particulier des images SPOT) pouvant être parfois **bruitées**. En effet, dans les papiers de

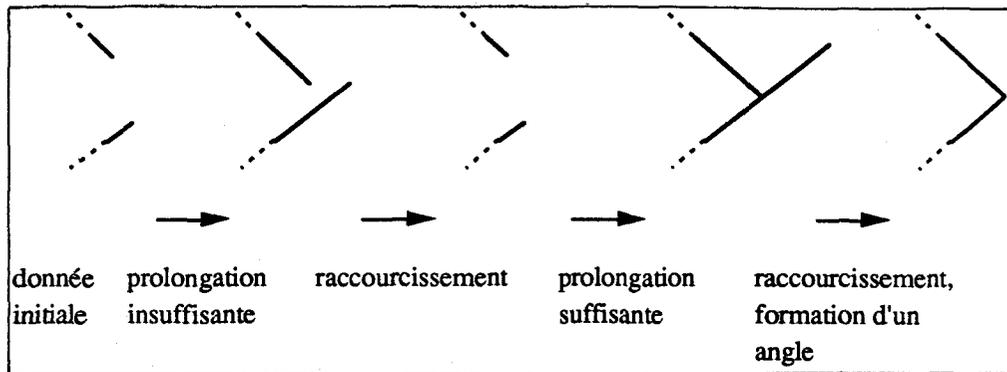


Figure 1. – Exemple de reconstitution de contours.

J.L. Marroquin [11], [12], les résultats concernant cet algorithme présentent uniquement le traitement d'images synthétiques.

Le modèle utilisé pour restaurer ces images de contours incomplets est celui des **champs de Markov** [1] ainsi que les **distributions de Gibbs** qui leur sont associées (cf. théorème de Hammersley-Clifford). L'algorithme implanté met en œuvre un algorithme de relaxation stochastique, fondé sur l'**échantillonneur de Gibbs** [9], [14]. Une des modifications apportées à la méthode proposée par J.L. Marroquin [11], [12] est l'implantation d'un algorithme déterministe : **ICM** (Iterated Conditional Mode) [2]. Ces algorithmes engendrent une configuration stable dans laquelle les contours sont reconstitués.

2. Restauration d'images

Le but est de reconstituer la totalité des frontières des objets d'une image à partir de données incomplètes. Pour ce faire, on effectue successivement des prolongations et des raccourcissements des lignes de contours données initialement, dans le but d'obtenir une figure stable, où des « angles » sont déterminés et empêchent les variations de la longueur des lignes (cf. Fig. 1).

Pour restaurer l'image de contours, on détermine tout d'abord une classification des éléments constituant les frontières, ainsi qu'une caractérisation de l'image restaurée.

2.1. CHAMPS MARKOVIENS D'ÉLÉMENTS DE CONTOURS

Pour chaque site, on mémorise deux informations. Le **champ de Markov** est donc un vecteur qui a pour **première** composante le type d'**élément de contour**, alors que la **seconde** composante correspond à une information plus globale, la **direction macroscopique** de la ligne de contour, direction dans laquelle le contour est sensé être prolongé [11], [12].

2.1.1. Première composante : les états

La donnée initiale, l'image des contours incomplets, est représentée par une matrice binaire (1 si contour, 0 sinon).

La classification des éléments de contours se compose de 21 classes, chacune d'elles correspondant à un état admis. Chaque état est déterminé à partir d'une fenêtre de taille 3*3 de l'image initiale. Pour chaque site de l'image d'entrée, on engendre un état. On construit donc ainsi une matrice d'éléments de contours de même dimension.

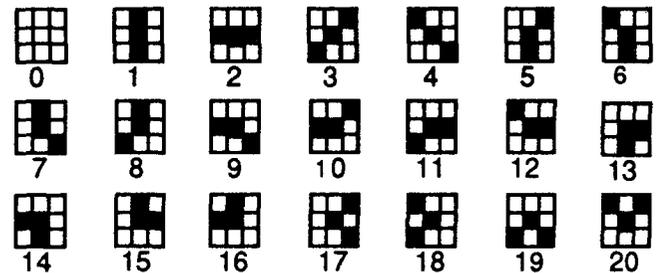


Figure 2 : Description des 21 états prédéfinis.

On établit également la convention suivante : si la fenêtre (3*3) de l'image initiale ne s'identifie à aucun des états prédéfinis, on suppose que l'état du site est nul.

Afin de prolonger les lignes de contours, il est indispensable de connaître les connexions entre deux états voisins. Pour cela, on détermine plusieurs classes correspondant à ces interactions locales (cf. Fig. 3).

Voici l'**interprétation de chaque classe d'interaction**.

Connexion indéterminée :	« o ».
Terminaison de lignes :	« e ».
Jonction en T (type t) :	« t ».
Jonction en T (type t_1) :	« t_1 ».
Virage ayant un angle aigu :	« s ».
Continuation en ligne droite :	« a ».
Virage à droite :	« b », « f », « j », « r ».
Virage à gauche :	« d », « g », « k », « l ».
Virage gauche-droite ou droite-gauche :	« c ».
Connexion non admise :	« - ».

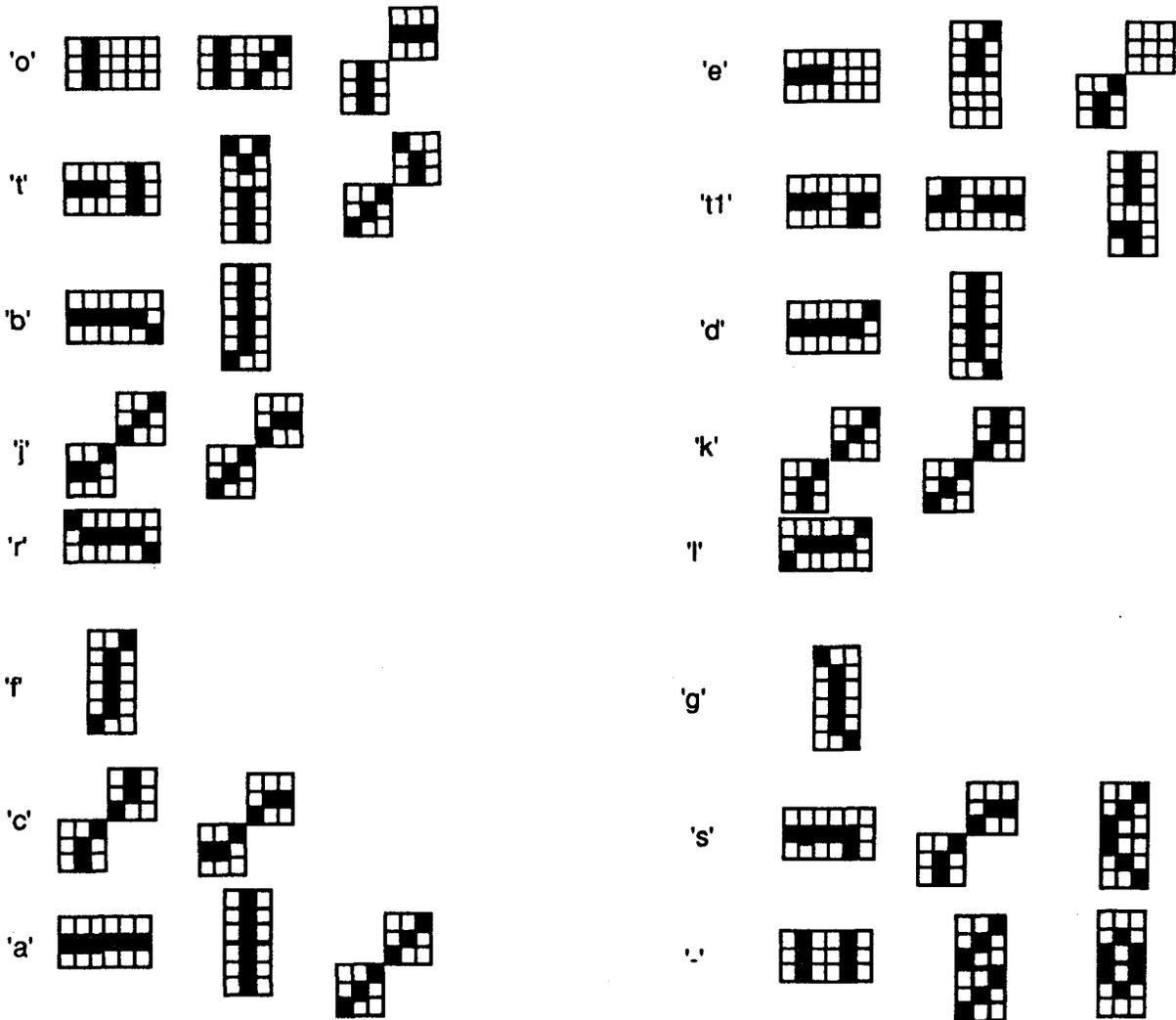


Figure 3. – Exemples représentant chaque type de connexion.

J.L. Marroquin [11], [12] ne définit pas les connexions de type « t_1 », qui sont des T-jonctions, et il considère ces interactions comme étant « interdites ». **On a créé ce nouveau type de connexions dans le but d'obtenir, au bout d'un certain nombre d'itérations, une figure où tous les « angles » sont détectés.**

Voici un exemple où ce type de connexion est indispensable pour créer une jonction entre deux contours :

1	1	1		1	1	1	1
				1			
				1			

Figure 4. – Matrice d'éléments de contour

2	2			13	2	2	2
				1			
				1			

Figure 5. – États initiaux

2	2	2	?	13	2	2	2
				1			
				1			

Figure 6. – États obtenus (? = 0 si pas de t_1 , 2 sinon)

En considérant la connexion 2-13 comme interdite et non comme une connexion « t_1 », la jonction entre les deux contours est impossible. En effet, la prolongation du contour est alors stoppée

durant la phase de prolongation. Par contre, pendant la phase de raccourcissement, les deux contours ne restent pas connectés et un des contours est supprimé. On peut donc, si on ne considère pas la jonction « t_l », obtenir une perte d'information. En fait, l'autorisation de la formation de ce type de connexions est très importante pour reconstruire les contours dans de nombreux cas. Ils sont énumérés dans les tableaux 1 et 2.

Tableau 1 : Matrice des connexions pour des cliques horizontales.

p/q	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
0	o	o	e	o	o	o	o	o	e	o	o	o	e	o	o	e	o	o	o	o	o
1	o	-	t	o	o	-	-	-	t	t	o	o	-	-	-	-	-	-	-	-	-
2	e	t	a	t	t	t	t	t	b	d	t	t	t _l	s	t _l	s	t	-	-	-	-
3	o	o	t	-	s	-	s	-	t	t	-	s	-	-	-	-	-	-	-	-	-
4	o	o	t	s	-	-	-	s	t	t	s	-	-	-	-	-	-	-	-	-	-
5	o	-	t	o	s	-	s	-	t	t	o	s	-	-	-	-	-	-	-	-	-
6	o	-	t	o	o	-	-	-	t	t	o	o	-	-	-	-	-	-	-	-	-
7	o	-	t	s	o	-	-	s	t	t	s	o	-	-	-	-	-	-	-	-	-
8	o	-	t	o	o	-	-	-	t	t	o	o	-	-	-	-	-	-	-	-	-
9	o	o	t	s	o	o	o	s	t	t	s	o	-	-	-	-	-	-	-	-	-
10	o	o	t	o	s	o	s	o	t	t	o	s	-	-	-	-	-	-	-	-	-
11	e	t	d	t	t	t	t	t	s	l	t	t	t _l	s	t _l	s	t	-	-	-	-
12	e	t	b	t	t	t	t	t	r	s	t	t	t _l	s	t _l	s	t	-	-	-	-
13	e	-	s	-	-	-	-	-	s	s	-	-	-	-	-	-	-	-	-	-	-
14	o	-	t _l	-	-	-	-	-	t _l	t _l	-	-	-	-	-	-	-	-	-	-	-
15	e	-	s	-	-	-	-	-	s	s	-	-	-	-	-	-	-	-	-	-	-
16	o	-	t _l	-	-	-	-	-	t _l	t _l	-	-	-	-	-	-	-	-	-	-	-
17	o	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
18	o	-	t	-	-	-	-	-	t	t	-	-	-	-	-	-	-	-	-	-	-
19	o	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
20	o	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Tableau 2 : Matrice des connexions pour des cliques diagonales.

p/q	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
0	o	o	o	e	o	o	o	o	e	o	o	e	o	o	o	o	o	e	e	o	o	
1	o	o	o	t	o	o	o	o	t	o	o	t	o	o	o	o	o	o	o	t	t	o
2	o	o	o	t	o	o	o	o	t	o	o	t	o	o	o	o	o	o	o	t	t	o
3	e	t	a	t	t	t	t	k	t	t	j	t	t	t	t	t	t	s	s	t	o	
4	o	o	o	t	-	o	o	t	o	o	t	o	o	o	o	o	o	o	t	t	o	o
5	e	t	t	k	t	t	t	c	t	t	s	t	t	t	t	t	t	t	s	s	t	o
6	o	o	o	t	o	o	-	o	t	o	o	t	o	o	o	o	o	o	o	t	t	o
7	o	o	o	t	o	o	-	t	o	o	t	o	o	o	o	o	o	o	o	t	t	o
8	o	o	o	t	o	o	o	t	o	o	t	o	o	o	o	o	o	o	o	t	t	o
9	o	o	o	t	o	o	o	t	-	o	t	o	o	o	o	o	o	o	o	t	t	o
10	e	t	t	j	t	t	t	s	t	t	c	t	t	t	t	t	t	t	s	s	t	o
11	o	o	o	t	o	o	o	t	o	o	t	o	o	o	o	o	o	o	o	t	t	o
12	o	o	o	t	o	o	o	t	o	o	t	-	o	o	o	o	o	o	o	t	t	o
13	o	o	o	t	o	o	o	t	o	o	t	o	o	o	o	o	o	o	o	t	t	o
14	o	o	o	t	o	o	o	t	o	o	t	o	o	o	o	o	o	o	o	t	t	o
15	o	o	o	t	o	o	o	t	o	o	t	o	o	o	o	o	o	o	o	t	t	o
16	o	o	o	t	o	o	o	t	o	o	t	o	o	o	o	o	o	o	o	t	t	o
17	e	t	t	s	t	t	t	s	t	t	s	t	t	t	t	t	t	t	s	s	t	o
18	o	o	o	t	o	o	o	t	o	o	t	o	o	o	o	o	o	o	o	t	t	o
19	o	o	o	t	o	o	o	t	o	o	t	o	o	o	o	o	o	o	o	t	t	o
20	e	t	t	s	t	t	t	s	t	t	s	t	t	t	t	t	t	t	s	s	t	o

Toutes les connexions possibles, obtenues à partir de tous les états voisins appartenant à des cliques horizontales ou diagonales sont définies par 4 tables. On donne uniquement deux de ces tableaux (cf tableaux 1 et 2), car on obtient aisément par symétrie les deux autres, correspondant aux cliques symétriques (verticales et diagonales inversées).

Ces différentes connexions et les directions macroscopiques, que nous allons déterminer, apparaissent dans la formulation des potentiels de cliques.

2.1.2. Deuxième composante : la direction macroscopique.

Définition

Lors de la prolongation des lignes de contours, on complète une ligne par un état compatible avec la direction macroscopique de celle-ci. Cette direction est déterminée par un numéro compris entre (-1) et (11).

Soient (X_p, Y_p) et (X_q, Y_q) , les coordonnées des sites ayant pour état respectif « p » et « q » telles que :

Tableau 1 :
$$X_p = X_q - 1 \quad \text{et} \quad Y_p = Y_q \quad (1)$$

Tableau 2 :
$$X_p = X_q - 1 \quad \text{et} \quad Y_p = Y_q + 1 \quad (2)$$

Pour un site donné, le **numéro de direction** peut être **négatif** dans plusieurs cas :

- état nul : absence de contour ou élément de contour indéfini.
- état supérieur à 12 : coin détecté (localisé à l'intersection de deux lignes ayant des orientations différentes).

Subdivision en intervalles homogènes

Nous pouvons remarquer que tous les états ne sont pas compatibles avec toutes les connexions et toutes les directions données.

On effectue une **discrétisation des orientations**, car chaque état possède une pente pouvant être associée à une direction. Discrétiser les différentes orientations est équivalent à **discrétiser les pentes** correspondantes. Pour ce faire, on subdivise donc l'ensemble de définition des pentes $(] - \infty, +\infty[)$ en sous-intervalles, chaque sous-intervalle étant associé à un numéro de direction (numéro variant de 1 à 11). Cette subdivision est créée dans le but suivant : **chaque numéro de direction correspondant à l'un de ces sous-intervalles, doit être compatible avec un unique ensemble d'états et de connexions.**

On définit également une valeur α représentant pour un site donné la **probabilité de tourner sur la probabilité de continuer tout droit**. C'est, en fait, la probabilité d'avoir un état ayant un numéro compris entre 5 et 12 sur celle d'en avoir un compris entre 1 et 4 (ce calcul de α est différent de celui proposé par J.L. Marroquin [11],[12]).

Toutes ces informations sont regroupées dans le tableau 3 : La deuxième colonne correspond à la subdivision de l'ensemble des pentes. A chaque intervalle est associé un numéro de direction (colonne 1) auquel correspond un unique ensemble d'états (colonne 2) et de connexions. Ces interactions sont séparées en trois classes : la première (colonne 4) indique une continuation en ligne droite, la seconde (colonne 5) un tournant et la dernière (colonne 6) un double virage.

Pour une ligne donnée, le numéro de direction, les états et les différentes interactions locales sont compatibles entre eux.

Tableau 3 : Compatibilité des directions, des états et des connexions

N°	Intervalle des pentes	États compatibles	Connexions ligne droite	Connexions virage	Connexions double virage
0	0	2	a		
1	$[0, \frac{1}{2}]$	2, 10, 11	a	d,l	c
2	$[\frac{1}{2}, 1]$	3, 10, 11	a	j,c	l
3	1	3	a		
4	$[1, 2]$	3, 5, 8	a	k,c	f
5	$[2, \infty)$	1, 5, 8	a	b,f	c
6	∞	1	a		
7	$(-\infty, -2]$	1, 6, 7	a	d,g	c
8	$[-2, -1]$	4, 6, 7	a	j,c	g
9	-1	4	a		
10	$[-1, -\frac{1}{2}]$	4, 9, 12	a	k,c	r
11	$[-\frac{1}{2}, 0]$	2, 9, 12	a	b,r	c

2.2. FONCTION D'ÉNERGIE

Des critères doivent être choisis de manière à prolonger les lignes de contours, dans une direction macroscopique donnée, et à créer des angles (ou des T-jonctions) afin d'obtenir des contours fermés. Ils interviennent dans la définition de l'énergie qui doit être minimale lorsque l'image des contours est reconstituée.

La fonction d'énergie est calculée à l'aide de deux fonctions de potentiel $V1$ et $V2$.

Avant de définir ces deux fonctions, il est nécessaire de décrire les notations utilisées.

2.2.1. Notations

L : ensemble des 21 états C représentés par la figure 2.

D : ensemble des numéros de directions macroscopiques définies précédemment ($D = [1, 11] \cup \{-1\}$)

$I_{ij} = I(q_i, q_j)$ correspond à la connexion formée par deux états voisins q_i et q_j

Un numéro de direction en un site i (noté d_i) est compatible avec un unique ensemble de connexions séparé en trois classes :

- $SC_i = SC(d_i)$ Continuation en ligne droite.
- $T_i = T(d_i)$ Tournant.
- $CT_i = CT(d_i)$ Double virage.

$C_i = C(d_i)$ représente l'ensemble des états compatibles avec la direction d_i .

$\alpha_i = \alpha(d_i)$ est la probabilité de tourner sur celle de continuer tout droit.

On a :

$$\alpha_i = \frac{\text{probabilité de sélectionner un état } q \text{ appartenant à } T_i}{\text{probabilité de sélectionner un état } q \text{ appartenant à } SC_i} \quad (3)$$

2.2.2. Fonctions de potentiel

2.2.2.1. Fonction $V1$

La fonction $V1$ est définie pour un site donné sur l'ensemble $(L * D)$. Elle dépend uniquement du vecteur (q_i, d_i) où q_i représente l'état du site et d_i sa direction macroscopique. Quelques remarques sur le choix de cette fonction :

$$\begin{aligned}
 V1(q_i, d_i) &= \infty && \text{si } (d_i \geq 0 \text{ et } q_i \notin C_i) \\
 &= \ln(1 + \alpha_i) && \text{si } (d_i \geq 0 \text{ et } q_i \in [1, 4] \cap C_i) \\
 &= \ln\left(\frac{1 + \alpha_i}{\alpha_i}\right) && \text{si } (d_i \geq 0 \text{ et } q_i \in [5, 12] \cap C_i) \\
 &= 0 && \text{sinon}
 \end{aligned} \quad (4)$$

Quelques remarques sur le choix de cette fonction :

$V1$ est infinie si l'état est incompatible avec la direction. On favorise donc les états ayant une pente compatible avec la direction de la ligne à compléter.

Nous avons affecté, pour chaque élément de contours d'une même ligne, une valeur α correspondant à la probabilité de tourner sur la probabilité de continuer tout droit. Lorsque nous allons compléter les contours, nous allons propager cette information, car elle permet de prolonger les contours dans la bonne direction et avec des éléments adéquats.

En effet, la minimisation de cette fonction de potentiel favorise la prolongation des contours le long d'une ligne de direction d_i par les éléments de contours « droits » lorsque la ligne contient plus d'éléments droits que de tournants et inversement. En fait, lorsque le quotient « α_i » est inférieur à 1, c'est-à-dire lorsque la probabilité de tourner est inférieure à celle de continuer tout droit, on privilégie les états compris entre 1 et 4.

En effet, on a :

$$\ln(1 + \alpha_i) < \ln\left(\frac{1 + \alpha_i}{\alpha_i}\right) \quad (5)$$

De plus, pour un site donné, la direction est négative lorsqu'il n'y a pas de contour ou lorsqu'un coin (ou une « T-jonction ») est détecté(e). $V1$ étant nulle pour les directions négatives, on favorise donc la formation des « T-jonctions ».

2.2.2.2. Fonction V2

La fonction de potentiel V2 est définie pour deux vecteurs (q_i, d_i) et (q_j, d_j) où i et j sont deux sites voisins. V2 est donc une fonction réelle définie sur l'ensemble $(L * D * L * D)$ et fait intervenir quatre constantes réelles VE, VT, VT_1 et VC (cf § 2.2.4).

$$\begin{aligned}
 V2(q_i, d_i, q_j, d_j) &= 0 && \text{si } (I_{i,j} = "o") \\
 &= VE && \text{si } (I_{i,j} = "e") \\
 &= VC && \text{si } (d_i = d_j) \text{ et } (q_i, q_j \in C_i) \\
 &&& \text{et } (I_{i,j} \in SC_i \cup T_i) \\
 &= VC - \ln\left(\frac{1 + \alpha_i}{\alpha_i}\right) && \text{si } (d_i = d_j) \text{ et } (q_i, q_j \in C_i) \\
 &&& \text{et } (I_{i,j} = CT_i) \\
 &= VT && \text{si } (I_{i,j} \notin \{"o", "e", "-"\}) \\
 &&& \text{et } d_k < 0 \text{ et } q_k > 0) \\
 &&& \text{pour au moins un } k \in \{i, j\} \\
 &&& \text{ou} \\
 &&& \text{si } (I_{i,j} = "t") \\
 &= VT_1 && \text{si } I_{i,j} = "t_1" \\
 &= \infty && \text{sinon}
 \end{aligned} \tag{6}$$

Voici quelques caractéristiques de cette fonction :

- Lorsqu'une interaction est indéterminée (« o ») le potentiel V2 est nul.
- V2 a pour valeur VE, pour une connexion de type « e » (fin de ligne).
- Lorsque la connexion de deux états de sites voisins ayant la même direction est une continuation en ligne droite ou un virage, V2 a pour valeur VC, constante réelle positive ou négative. Si cette interaction est une continuation en ligne droite, on a $V2 + V1 = VC$.
- De même, si les deux sites voisins possèdent la même direction et si la connexion est un virage gauche-droite alors $V1 + V2 = VC$.
- Pour les connexions dites « interdites » (« - »), V2 a une valeur infinie, ceci dans le but de rendre certaines configurations inaccessibles.
- VT est affecté à V2 pour toutes les connexions, hormis les connexions de types « o », « e » et « - », en présence d'un angle.
- V2 a également la valeur VT lorsque la connexion entre les deux états est de type « t » (« T-jonction »), et il n'est pas alors nécessaire que l'une des deux directions soit nulle.

- Pour une connexion de type « t₁ », V2 a pour valeur VT₁.

Ces deux derniers points n'apparaissent pas dans les travaux de J.L. Marroquin [11] et [12] et favorisent l'obtention de contours fermés.

2.2.3. Énergie totale

On définit l'énergie totale « U » par la somme sur tous les sites de l'image de la fonction de potentiel V1 plus la somme sur toutes les cliques de la fonction V2 :

$$U = \sum_i V1(q_i, d_i) + \sum_{i,j: ||i-j|| < 2} V2(q_i, d_i, q_j, d_j) \tag{7}$$

i, j représentent des pixels voisins (appartenant à une même clique) de l'image.

D'après le théorème d'Hammersley-Clifford, la probabilité d'avoir au site i un champ de valeur (q_i, d_i) est :

$$P = \exp\left(\frac{-1}{T} \left(\sum_i V1(q_i, d_i) + \sum_{i,j: ||i-j|| < 2} V2(q_i, d_i, q_j, d_j) \right)\right) \tag{8}$$

où T représente la température.

2.2.4. Rôle des paramètres dans la fonction d'énergie

Pour la reconstruction de contours, les paramètres VE, VT, VT_1 et VC doivent prendre des valeurs particulières :

- VE doit être « très grand » et positif pour assurer que les contours soient fermés et éviter la génération de nouvelles lignes.
- VC, VT et VT₁ doivent être fixés de façon à privilégier la continuation en ligne droite par rapport à la formation des « T-jonctions ». Ceci s'obtient en choisissant $|VC| < VT$.
- Le signe de VC détermine la prolongation ou le raccourcissement des lignes;

$$\begin{aligned}
 VC < 0 & : \text{rallongement des lignes} \\
 VC > 0 & : \text{raccourcissement des lignes}
 \end{aligned}$$

Dans la pratique, nous avons choisi les paramètres de telle sorte que :

$$|VC| < VT < VT_1 \ll VE \tag{9}$$

Plus exactement : $VE = 20; VC = \pm 4; VT = 6; VT_1 = 8$

Ces quatre paramètres ont été choisis expérimentalement. Pour l'instant, aucune justification théorique n'est apportée, mais nous essayons actuellement de déterminer ces paramètres en utilisant les boîtes qualitatives [5].

Il est tout de même important de noter que les valeurs choisies pour ces constantes sont conservées tout au long de l'algorithme (mis à part le signe de VC) et pour le traitement de toutes les images, qu'elles soient synthétiques, réelles ou bruitées.

Ces valeurs particulières ne sont pas impératives, mais il est par contre important de toujours satisfaire l'inéquation (9).

2.3. INITIALISATION

Avant d'appliquer l'algorithme d'optimisation d'énergie décrit précédemment, il faut initialiser pour chaque site « i » le vecteur (q_i, d_i) , afin de modéliser l'image initiale.

2.3.1. Initialisation des états

L'ensemble des 21 états prédéfinis est décrit précédemment (cf 2.1.1.).

2.3.2. Initialisation des directions macroscopiques

L'initialisation de la deuxième composante du champ de Markov, la direction, est plus complexe. Elle est décrite de manière très succincte par J.L. Marroquin [11] et [12]. **Des modifications ont été apportées pour l'initialiser, elles sont très importantes pour la mise en œuvre de l'algorithme.**

Cette initialisation s'effectue à partir de la matrice d'états obtenue et permet de connaître, en chaque site où une ligne de contours est détectée, sa direction. Cet algorithme de propagation de directions permet d'obtenir **le maximum de sites voisins ayant un numéro de direction identique, ceci pour permettre de prolonger** une ligne dans la direction la mieux adaptée.

Les états 0 et supérieurs à 12 sont associés au numéro de direction (-1).

Pour les autres états, on définit le numéro de direction de la manière suivante :

Pour un site donné, on essaie tous les numéros de direction (compris entre 0 et 11) compatibles avec l'état du site étudié. On comptabilise alors le nombre d'états (de la fenêtre 3*3 centrée sur ce site) compatibles avec chacun de ces numéros de direction. Le choix d'un numéro définitif doit alors être effectué.

Pour cela, on réitère l'algorithme décrit ci-dessous, implanté de façon parallèle, jusqu'à ce qu'on ne puisse plus affecter de numéro de direction aux sites non initialisés.

Dans cet algorithme, plusieurs cas doivent être pris en compte :

- Si **aucun état voisin n'est compatible avec les numéros de direction essayés**, c'est-à-dire si un état non nul est isolé, le numéro de direction affecté en ce site est (-1).
- Si **une unique direction est compatible avec l'état du site étudié « i » et les états voisins**, cette direction est définitive pour le site « i ».
- Si **plusieurs numéros de direction proposés sont compatibles avec au moins un état voisin**, le numéro sélectionné pour le site « i » est celui qui a le plus grand nombre d'états voisins compatibles, ou le numéro définitif d'un site compatible. Ainsi, on propage la direction de la ligne de contour.
- Si **plusieurs sites voisins ont un numéro de direction définitif**, un tirage aléatoire parmi ceux qui sont compatibles avec l'état de site « i » (« q_i ») est nécessaire.

Après plusieurs itérations, **le traitement des sites n'ayant pas été initialisés nécessite également l'étude de plusieurs cas possibles :**

- **Si trois numéros de direction sont compatibles avec l'état (« q_i ») du site à initialiser**, ceci implique que l'état (« q_i ») est forcément un état compris entre 1 et 4 (voir le tableau 3). Dans ce cas, si l'état du site est l'état 1, le numéro de direction 6 est affecté au site « i » (voir le tableau 3), de même l'état 2 correspond au numéro de direction 0, l'état 3 au numéro 3 et l'état 4 au numéro 9.
- **Si le nombre de numéros de direction compatibles avec l'état (« q_i ») est égal à deux**, un tirage aléatoire est effectué afin de choisir le premier ou le deuxième numéro et de l'affecter à tous les sites de ce type.

2.4. IMPLANTATION PRATIQUE

La distribution de Gibbs ainsi déterminée (cf § 2.2.3) permet de mettre en œuvre une méthode stochastique d'optimisation : l'échantillonneur de Gibbs [9], ou une méthode déterministe : « ICM » [2] dans le but de minimiser l'énergie. Ces méthodes effectuent à un instant donné une mise à jour de la valeur d'un site connaissant les états des sites voisins. Le nouveau vecteur (q_i, d_i) est déterminé par le calcul des probabilités conditionnelles des différents vecteurs (q, d) (où q représente l'état et d la direction) pour le site connaissant les états des sites voisins et par l'algorithme sélectionné.

Voici une mise en œuvre possible :

D'après les propriétés des deux fonctions de potentiel $V1$ et $V2$, uniquement **deux types de vecteurs** (q_i, d_i) (correspondant à une énergie finie) doivent être considérés pour la nouvelle mise à jour. **Les vecteurs pris en compte sont différents de ceux étudiés par J.L. Marroquin [11], [12] :**

1. Les vecteurs $(q_i, -1)$ où q_i est un état « 0 » ou un état supérieur à 12.
 - État « 0 » : aucun contour détecté.
 - État supérieur à 12 : détection de plusieurs directions macroscopiques dans le voisinage du site. La compatibilité de cet état avec une seule orientation est impossible, il est donc compatible uniquement avec la direction (-1).
2. **Deux cas sont alors possibles :**
 - **Si une seule direction « d_j » apparaît dans le voisinage du site « i »**, les seuls états « q_i » devant être pris en compte sont les états compatibles avec la direction « d_j ». On propage la direction de la ligne en affectant à « d_i » la valeur du site voisin « d_j ».
 - Si l'on détecte **plusieurs directions dans les sites avoisinants, ou s'il apparaît plusieurs états voisins non nuls**, la création d'un « coin », de direction (-1) est très probable. Les vecteurs à prendre en compte sont donc formés par :

Les couples (q_i, d_i) où q_i sont tous les états compatibles avec chaque direction détectée et les couples $(q_i, -1)$ formés de tous les états compris entre 1 et 12 avec la direction (-1).

Comment choisir le vecteur (q_i, d_i) le plus adapté au site « i » ?

On sélectionne ensuite, en appliquant l'algorithme de l'échantillonneur de Gibbs [9] (ou l'algorithme « ICM » [2]), un vecteur (q_i, d_i) définitif pour le site « i », parmi les N vecteurs (q_i, d_i) $k = 1 \dots N$ (où q_i détermine l'état et d_i la direction) susceptibles de convenir.

2.5. TRAITEMENTS DES IMAGES BRUITÉS ET RÉELLES

Dans le cas d'une image bruitée, on constate une perte d'information importante lors de la création de la « matrice » correspondant aux états, ainsi que de fausses informations (dues au bruit). En effet, pour chaque site de l'image de départ, on initialise les états en comparant une fenêtre de taille 3*3 centrée en chacun de ces sites avec les états prédéfinis. Lorsque cette fenêtre ne correspond pas à un état prédéfini, on considère l'état comme étant nul.

Prenons l'exemple suivant :

Représentation : « 1 » représente les contours de l'image initiale et « 0 » le bruit.

	1	1	1	1	1
	1	1	1		
	1				
	1				
	1				

Figure 7 : image initiale bruitée.

	0	0	0	0	2
	0	0	0		
	0				
	1				

Figure 8 : initialisation des états.

Certains contours ne sont donc pas détectés, et le fait de les supprimer peut avoir des conséquences sur la reconstruction finale.

En fait, pour traiter une image bruitée, nous devons étudier plusieurs cas :

1. Le bruit peut être isolé (non adjacent à un contour). Pour pallier ce problème, on effectue un traitement préalable sur la matrice

d'états initiale, qui consiste à éliminer les chaînes d'états non nuls formées par un nombre insuffisant d'éléments. Dans ce cas, le traitement est efficace.

2. Un nombre important de pixels adjacents peuvent correspondre à du bruit, auquel cas il est difficile de différencier le bruit d'un contour « réel » initial. L'application du traitement décrit précédemment ne permet pas de supprimer ce genre de bruit, ce qui peut introduire dans les données initiales de fausses informations.

Si ces « faux contours » sont suffisamment éloignés (ou parallèles) des « vrais contours », aucune connexion n'est créée durant la phase de prolongation de contours. L'algorithme de reconstruction des frontières se terminant par une phase de raccourcissement des lignes plus longue que celle de prolongation, ces « faux contours » disparaissent alors.

Par ailleurs, si ces « faux contours » sont proches des « vrais contours », la formation d'une connexion, qui est alors conservée, est probable.

3. Un deuxième traitement préalable est nécessaire car le bruit peut être adjacent aux contours des objets donnés initialement. Ce traitement consiste à initialiser les états en acceptant pour chaque état plusieurs configurations possibles.

Ces configurations permettent une meilleure initialisation de la matrice d'états et ne doivent pas être considérées comme étant une nouvelle définition des états. **Cette modification apportée au travail de J.L. Marroquin [11], [12] est importante car elle permet de mieux transcrire les données initiales** (cf. Fig. 9).

En prenant en compte cette modification, on obtient à partir de la figure 7 les états décrits par la figure 10.

Ainsi, on ne perd pas autant d'information, mais ce résultat n'est pas encore satisfaisant.

On effectue donc un traitement supplémentaire qui consiste à créer à partir de la matrice des états obtenus [figure 10] une matrice binaire de contours. Pour l'obtenir, on définit pour chaque pixel de la matrice des états [figure 10], l'élément de contour (fenêtres 3*3) associé. Lorsqu'on considère deux états voisins de l'image [figure 10], les fenêtres (3*3) vont se chevaucher. On conserve pour le site i la valeur 1 (présence d'un contour) si au moins un pixel des deux fenêtres correspondant à ce site a une valeur 1 et 0 sinon [figure 11]. On considère ensuite cette image comme étant une image d'entrée. On initialise donc à nouveau les états [figure 12].

L'image obtenue est alors considérée comme l'image initiale, et l'initialisation des contours [figure 12] donne cette fois-ci des résultats satisfaisants.

On remarque que dans la majeure partie des cas, ces traitements donnent de bons résultats mais parfois l'information reste incomplète. En voici un exemple (cf. Fig. 13 et Fig. 14) :

Après le traitement permettant d'initialiser les états, un unique site indiquant la présence d'un contour horizontal l'état 2 est conservé. Cependant, il peut être supprimé soit lors du second

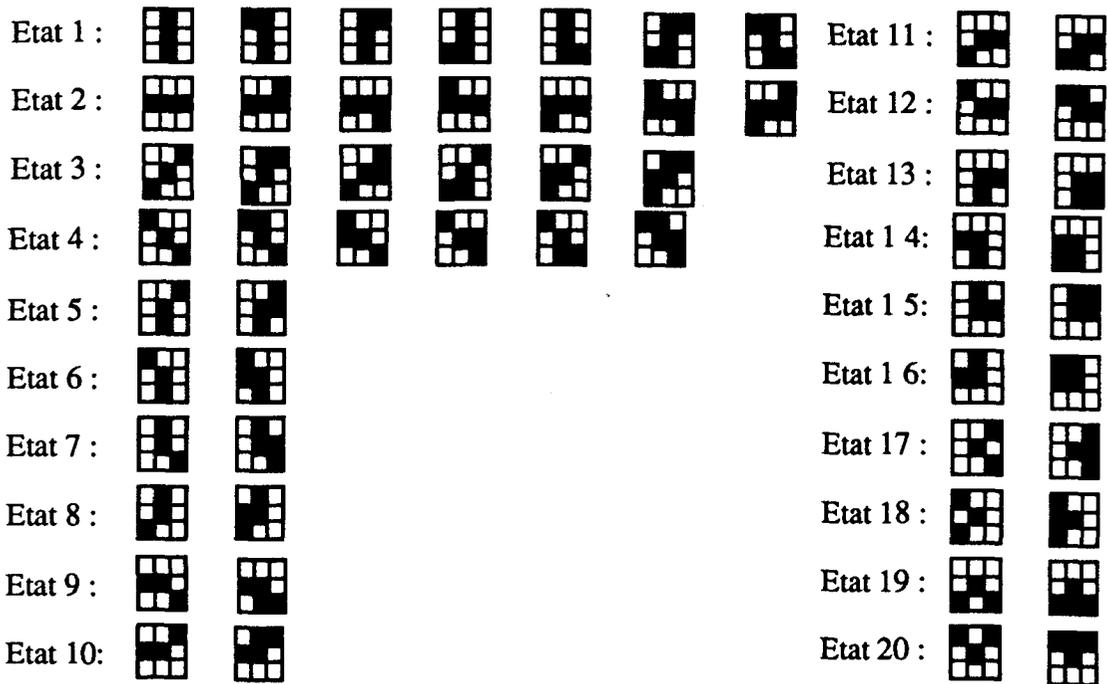


Figure 9. - Les états prédéfinis.

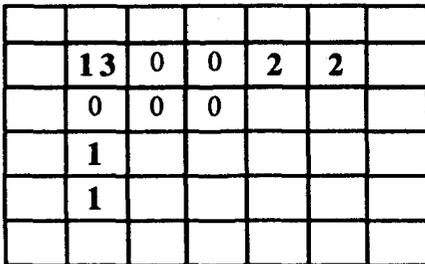


Figure 10. - États obtenus après les traitements décrits précédemment.

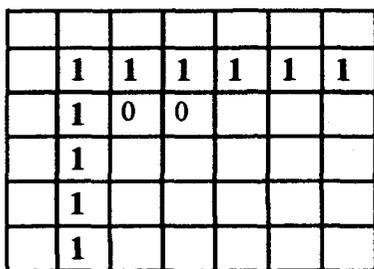


Figure 11. - Matrice d'éléments de contour.

traitement supprimant les petites chaînes détectées, soit au cours de l'algorithme de prolongation de lignes, car la probabilité d'avoir un contour isolé est très faible. Il est donc probable que ceci entraîne une perte d'information et l'angle peut être supprimé.

Les traitements préalables effectués pour traiter les images bruitées s'avèrent efficaces dans la majeure partie des cas et donnent de meilleurs résultats pour compléter toutes les images d'entrée, notamment les images réelles.

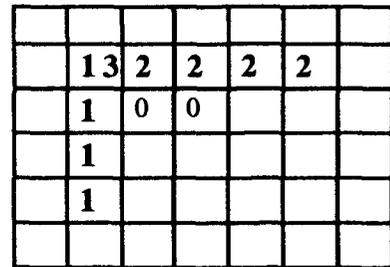


Figure 12. - États obtenus après traitements.

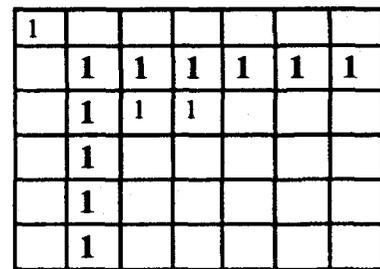


Figure 13. - Matrice d'éléments de contour.

Voici l'exemple (cf. Fig. 15 et Fig. 16) d'une image d'entrée réelle (un contour diagonal), où le contour est correctement détecté alors qu'il aurait été supprimé sans traitement préalable (que des états 0).

En résumé, il est tout d'abord important de noter que ces traitements préalables ainsi que ces configurations des états ne sont pas utilisés uniquement pour les images d'entrée bruitées, mais pour toutes les données, notamment les images réelles.

	0	0	0	2	2
	0	0	0		
	1				
	1				

Figure 14. – États obtenus après traitements.

					1
				1	1
			1	1	
		1	1		
	1	1			

Figure 15. – Matrice d'éléments de contour.

					3
				3	
			3		
		3			

Figure 16. – États obtenus après traitement.

Les configurations des états ont été modifiées, de manière à conserver le maximum d'information, lors du passage de l'image initiale à la matrice contenant les différents états. Étant donné qu'après cette étape on ne prend plus en compte ces données initiales, il est primordial de ne pas les perdre lors de ce transfert. En effet, le fait de supprimer des contours à l'initialisation influence le résultat de la restauration.

De plus, on effectue des traitements préalables qui s'avèrent efficaces, toujours dans le but de ne pas perdre d'information au cours de cette première étape (i.e. l'initialisation des états).

Ces modifications permettent donc une meilleure restauration de contours de l'ensemble des images traitées, qu'elles soient synthétiques, réelles ou bruitées.

2.6. DURÉES DES PHASES DE PROLONGATION ET DE RACCOURCISSEMENT DES LIGNES

Pour compléter les contours, nous effectuons des phases successives de prolongation et de raccourcissement des lignes (cf. Fig. 1). On appelle **période** le nombre d'itérations pendant lesquelles chacune des phases est mise en œuvre. Il faut noter qu'une phase de prolongation est toujours suivie d'une phase de raccourcissement de même période.

Plusieurs méthodes sont envisageables pour définir la période :

- **Méthode de la période stable** : la période des phases successives reste **identique** tout au long de l'algorithme.
- **Méthode de la période variable** : cette méthode fait intervenir une variation dynamique de la période au cours du traitement. **Au départ la période fixée est petite**, ceci de manière à privilégier les connexions des segments les plus proches les uns des autres. Petit à petit, **la période est augmentée** jusqu'à atteindre une valeur qui permette de compléter les plus grandes parties manquantes.

2.7. RÉSULTATS DE SIMULATIONS

2.7.1. Implantation parallèle et mise en œuvre sur une machine à connexions (CM200)

2.7.1.1. La machine à connexions

La méthode de restauration d'image de contours incomplets a été implantée sur **une machine à connexions** (CM200) [10], [16]. Le langage utilisé est le C*, basé sur le C ANSI, il permet la mise en œuvre **d'algorithmes parallèles sur la CM**.

La **machine à connexions** utilisée contient 16K processeurs -1 bit disposant de 32 Koctets de mémoire locale, avec une horloge à 8 MHz. Si les données à traiter sont importantes, l'utilisateur peut s'affranchir de la limite imposée par le nombre de processeurs physiques grâce au mécanisme des processeurs virtuels. Une même opération est en fait effectuée en parallèle, sur une grande masse de données. Ce type de parallélisme est bien adapté au traitement d'image bas niveau puisque les données (l'image) sont de taille importante et que les traitements sont locaux et identiques pour chaque pixel [16].

2.7.1.2. Entrée-sortie d'images

Pour traiter une image, il est nécessaire, dans un premier temps, de la charger dans la mémoire de la machine. Pour cela, il faut configurer la machine suivant une géométrie adaptée aux dimensions de l'image. Généralement, la géométrie choisie est une matrice à deux dimensions dont la taille est égale à la puissance de deux immédiatement supérieure au nombre de pixels. L'image est alors lue sur le calculateur frontal, et chacun de ses pixels peut alors être associé à un processeur virtuel. L'image est représentée dans la machine par une variable parallèle dont chaque élément est un pixel. L'opération d'écriture d'une image depuis sa représentation dans la CM s'effectue par l'opération inverse [16].

2.7.1.3. Implantation parallèle de l'algorithme

Deux méthodes ont été mise en œuvre dans le but de minimiser l'énergie : l'échantillonneur de Gibbs [9], et une méthode déterministe : « ICM » [2]. Ces méthodes effectuent, à chaque itération, la mise à jour du vecteur (q_i, d_i) d'un site i (où q_i représente l'état et

d_i la direction) connaissant les vecteurs des sites voisins. Ces nouvelles valeurs peuvent donc être sélectionnées en parallèle pour des sites non voisins. L'utilisation d'un champ de Markov d'ordre 2 (8-connexité) permet **la mise à jour de tous les sites de l'image en quatre étapes** (tableau de coloriage à quatre couleurs). Une phase de quatre étapes est appelée **itération**. La première étape ne tient compte que des sites d'abscisse et d'ordonnée paires, la seconde d'abscisse impaire et d'ordonnée paire,... etc.

Cette mise en œuvre partiellement parallèle [6] peut être décrite par la figure suivante, dans laquelle est indiqué le numéro de l'étape traitant les pixels (i,j) :

i/j	0	1	2	3	4
0	1	2	1	2	1
1	4	3	4	3	4
2	1	2	1	2	1
3	4	3	4	3	4
4	1	2	1	2	1

Figure 17. – tableau de coloriage à quatre couleurs.

2.7.2. Résultats sur des images synthétiques non bruitées

Cette méthode de restauration d'image de contours incomplets a été testée sur des images synthétiques de taille 128*128 [figures 18 et 19].

Notations :

L'image initiale est notée en gras :

« 1 » : contour présent initialement et inchangé au cours de l'algorithme.

« 0 » : contour initial supprimé par la suite.

« 1 » : contour complété.

« » : pas de contour détecté.

Algorithme utilisé : l'échantillonneur de Gibbs.

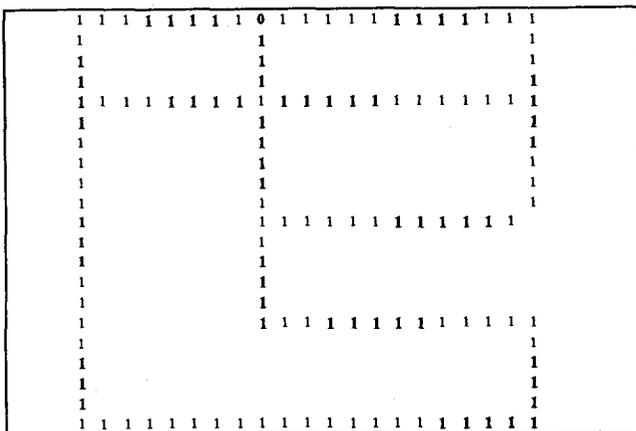


Figure 18. – Reconstitution d'une image de synthèse non bruitée.

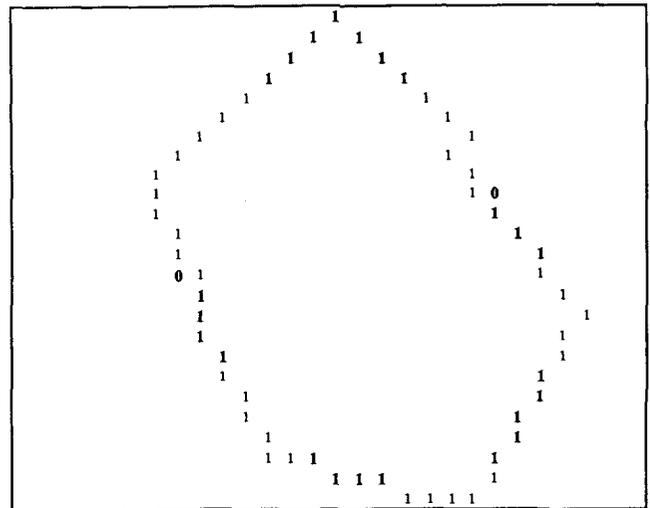


Figure 19. – Reconstitution d'une image de synthèse non bruitée.

• **Méthode de la période variable** (cf. Fig. 18) : La période est initialement égale à 3, puis on l'incrémente de 7 au début de chaque phase de prolongation.

Nombre d'itérations = 2 (2 phases de prolongation et 2 de raccourcissement).

• **Méthode de la période fixe** (cf. Fig. 19) (période égale à 4).

Nombre d'itérations fixé à 1 (une phase de prolongation et une de raccourcissement).

2.7.3. Résultats sur des images synthétiques bruitées

Notations :

0 : représente du bruit détecté dans l'image initiale qui a été supprimé.

1 : bruit non supprimé.

• **Méthode de la période variable** (cf. Fig. 20) : la période est initialement égale à 3, puis on l'incrémente de 7 au début de chaque phase de prolongation.

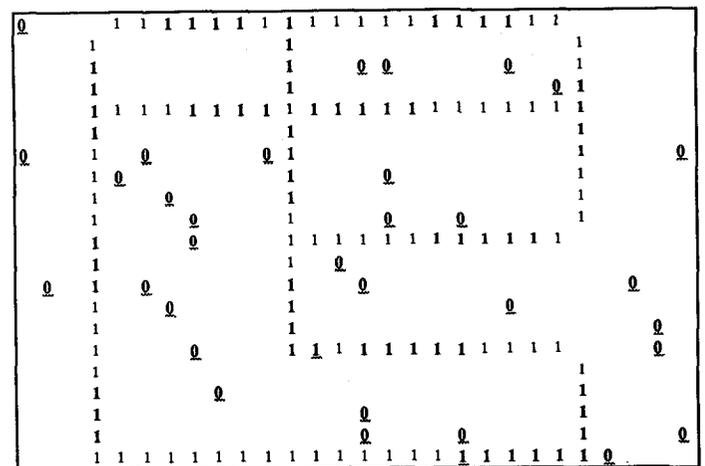


Figure 20 : reconstitution d'une image de synthèse bruitée.

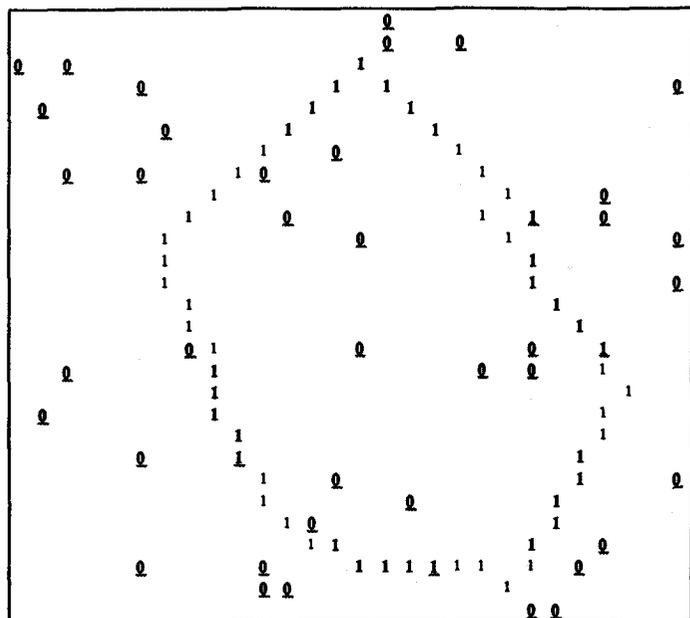


Figure 21 : reconstitution d'une image de synthèse bruitée.

Nombre d'itérations = 2 (2 phases de prolongation et 2 de raccourcissement).

Algorithme utilisé : l'échantillonneur de Gibbs.

• **Période fixe égale à 4** (cf. Fig. 21).

Nombre d'itérations fixé à 1 (une phase de prolongation et une de raccourcissement).

Algorithme mis en œuvre : l'échantillonneur de Gibbs.

2.7.4. Résultats sur des images de scènes réelles

A partir d'une image de scène réelle, on détermine les contours par l'algorithme de Canny-Deriche [4], [7] suivi d'un seuillage des données obtenues (suppression des non-maxima locaux). On applique alors l'algorithme décrit précédemment (échantillonneur de Gibbs), dans le but de compléter les contours détectés.

Des tests ont été effectués sur des images de l'INRIA, de la base de données du GDR TdSI et de l'University of Southern California. Ces images (de taille 256*256), représentent un bureau [figure 22], des images satellites (POT) [figures 25 et 28], une image médicale (muscle) [figure 31] et une image industrielle de bagues [figure 34].

Analyses des résultats :

– L'image du bureau a été correctement reconstituée [figure 24].
 – Pour les deux images satellites [figures 27 et 30], les contours détectés après un filtrage de Canny-Deriche et les suppressions des non-maxima locaux sont dans l'ensemble assez bien complétés.

Par contre, comme après l'initialisation des états, on n'utilise plus l'information contenue dans l'image initiale, on ne détermine pas parfaitement toutes les régions. En effet, la route par

exemple observée dans la figure 25 n'apparaît pas dans l'image restaurée [figure 27].

- Les mêmes remarques restent valables pour le traitement de l'image médicale [figure 33].
- La restauration de l'image représentant les bagues [figure 36] n'est pas satisfaisante sans mise en œuvre de traitements préalables. Ce résultat n'est pas montré ici, mais on peut tout de même observer sur la figure obtenue après la détection des contours [figure 35] un nombre important de diagonales. Elles ne sont pas traitées correctement sans l'application des modifications que nous avons apportées à l'algorithme (cf. figures 15 et 16) initialement proposé par J.L. Marroquin [11], [12].

On peut également souligner le fait que les contours détectés [figure 35] prennent en compte les ombres qui apparaissent sur l'image initiale. Elles sont évidemment conservées lors de la restauration.

2.7.5. Résultats sur des images de scènes réelles bruitées

Des tests ont été effectués sur une image aérienne de taille 128*128 à laquelle a été ajouté un bruit Gaussien.

Les contours de l'image originale [figure 37] détectés après un filtrage de Canny-Deriche sont dans la majorité des cas fermés [figure 38]. Le résultat de la restauration est donné par la figure 39.

On constate que le fait de rajouter du bruit à l'image initiale influence la détection des contours. La figure 40 représente l'image bruitée à 5 dB. Les contours obtenus [figure 41] contiennent des interruptions qui sont assez bien complétées dans la figure 42. On s'aperçoit par contre que la détection d'un faux contour créé par le bruit entraîne sa prolongation. On n'a en effet aucun moyen de vérifier si le contour détecté est véritablement la frontière d'un objet ou s'il est dû au bruit.

Les mêmes remarques restent valables pour la restauration de l'image bruitée à -5 dB [figure 43], les problèmes étant plus apparents dans ce cas du fait du faible rapport signal sur bruit.

2.7.6. Temps d'exécution

Pour chacune des images restaurées, nous indiquons (cf. tableau 6) :

- Le VPR : le nombre de pixels par processeur.
- Le nombre d'itérations effectuées.
- La durée de chaque itération.
- Le temps d'exécution sur la CM [10].
- Le temps total.

Nous avons mis en œuvre cet algorithme uniquement sur la machine à connexions, mais les temps d'exécutions sur une machine plus classique, un SUN par exemple, seraient beaucoup plus importants (de l'ordre de 40 à 50 fois plus longs).

Restauration des concours d'une image d'intérieur

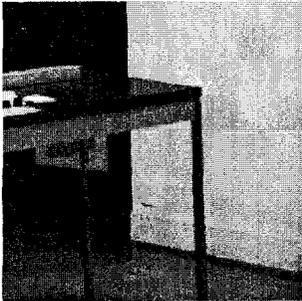


Figure 22. – Image de scène réelle

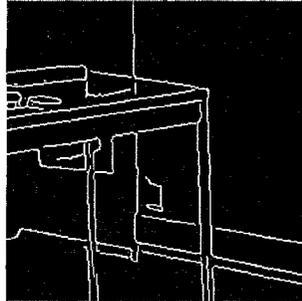


Figure 23. – Contours détectés après filtrage de Canny-Derichet et suppression des non-maxima locaux

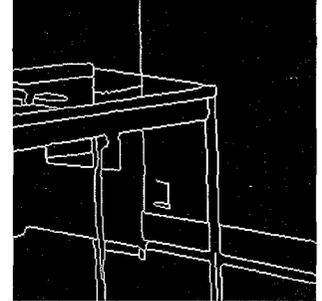


Figure 24. – Restauration des contours incomplets. Période fixe = 12. Nombre d'itérations = 1. Algorithme de l'échantillonneur de Gibbs

Restauration des concours d'une image satellite



Figure 25. – Image de scène réelle

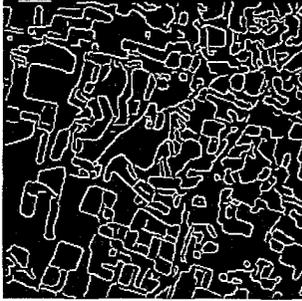


Figure 26. – Contours détectés après filtrage de Canny-Derichet et suppression des non-maxima locaux

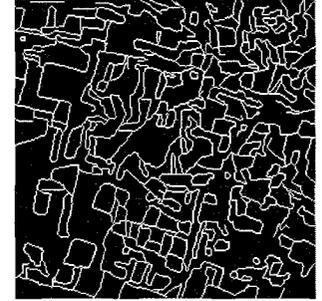


Figure 27. – Restauration des contours incomplets. Période variable = 3 puis 5. Nombre d'itérations = 2. Algorithme de l'échantillonneur de Gibbs

Restauration des concours d'une image satellite



Figure 28. – Image de scène réelle



Figure 29. – Contours détectés après filtrage de Canny-Derichet et suppression des non-maxima locaux

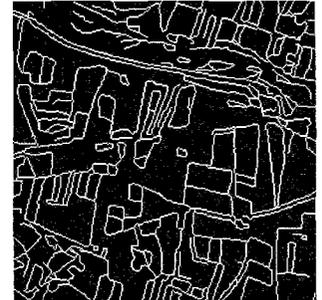


Figure 30. – Restauration des contours incomplets. Période fixe = 3. Nombre d'itérations = 1. Algorithme de l'échantillonneur de Gibbs

Restauration des concours d'une image médicale

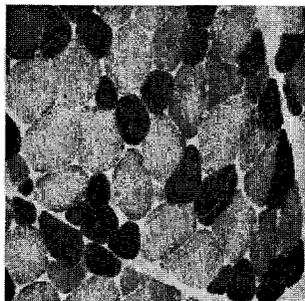


Figure 31. – Image de scène réelle

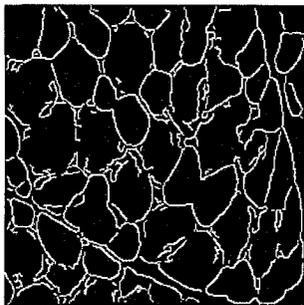


Figure 32. – Contours détectés après filtrage de Canny-Deriche et suppression des non-maxima locaux

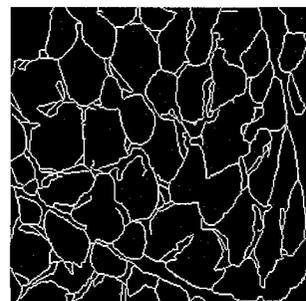


Figure 33. – Restauration des contours incomplets. Suppression des chaînes ayant moins de 3 états. Période fixe = 7. Nombre d'itérations = 1. Algorithme de l'échantillonneur de Gibbs

Restauration des concours d'une image industrielle

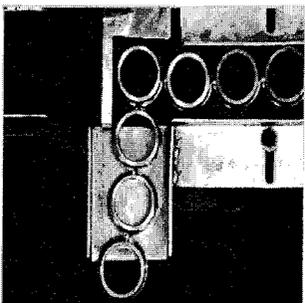


Figure 34. – Image de scène réelle

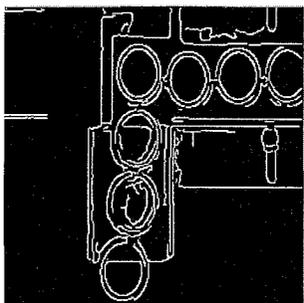


Figure 35. – Contours détectés après filtrage de Canny-Deriche et suppression des non-maxima locaux

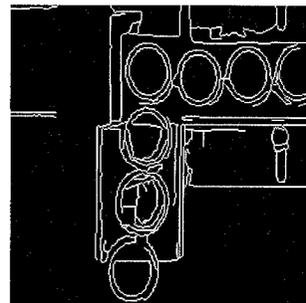


Figure 36. – Restauration des contours incomplets. Période fixe = 4. Nombre d'itérations = 1. Algorithme de l'échantillonneur de Gibbs

Restauration des concours d'une image aérienne bruitée

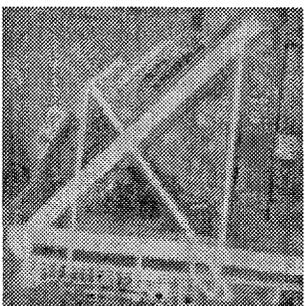


Figure 37. – Image de scène réelle

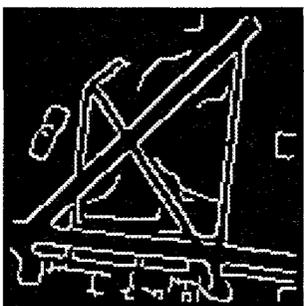


Figure 38. – Contours détectés après filtrage de Canny-Deriche et suppression des non-maxima locaux

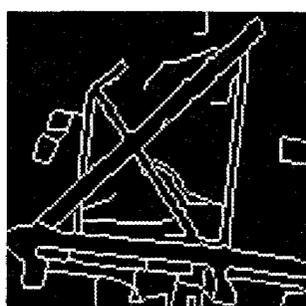


Figure 39. – Restauration des contours incomplets. Période fixe = 5. 1 itération. Echantillonneur de Gibbs

Restauration des concours d'une image aérienne bruitée

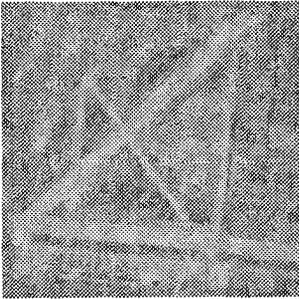


Figure 40. – Image de scène réelle bruitée à 5dB

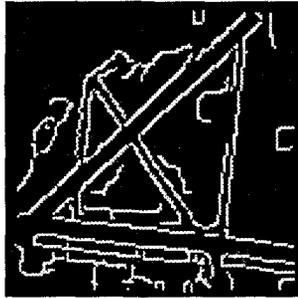


Figure 41. – Contours détectés après filtrage de Canny-Deriche et suppression des non-maxima locaux

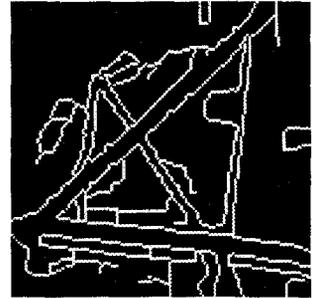


Figure 42. – Restauration des contours incomplets. Période variable = 1 puis 5. 2 itérations. Echantillonneur de Gibbs

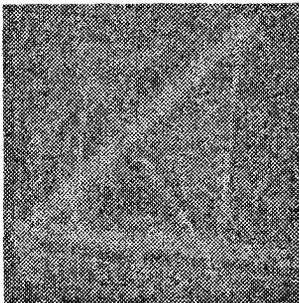


Figure 43. – Image de scène réelle bruitée à -5 dB

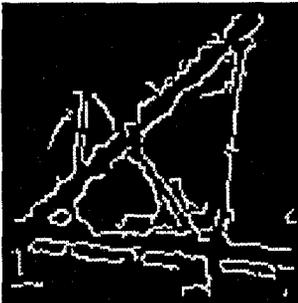


Figure 44. – Contours détectés après filtrage de Canny-Deriche et suppression des non-maxima locaux

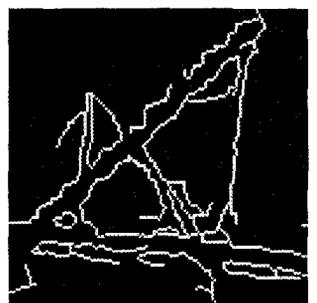


Figure 45. – Restauration des contours incomplets. Période variable = 1 puis 4. 2 itérations. Echantillonneur de Gibbs

Tableau 4 : Temps d'exécution sur la CM.

Figure	VPR	Nombre d'itérations	Durée de chacune des itérations	Temps CM	Temps Total
Figure 18	2	2	3 - 10	1 mn 26 sec	2 mn 01 sec
Figure 19	2	1	4	26 sec 59	41 sec 39
Figure 20	2	2	3 - 10	2 mn 05 sec	2 mn 35 sec
Figure 21	2	1	4	47 sec 15	1 mn 10 sec
Figure 22	8	1	12	4 mn 06 sec	4 mn 36 sec
Figure 25	8	2	3 - 5	2 mn 05 sec	2 mn 36 sec
Figure 28	8	1	3	1 mn 05 sec	1 mn 16 sec
Figure 31	8	1	7	2 mn 15 sec	2 mn 29 sec
Figure 34	8	1	4	1 mn 27 sec	1 mn 48 sec
Figure 37	2	1	5	37 sec 05	54 sec 30
Figure 40	2	2	1 - 5	42 sec 40	58 sec 05
Figure 43	2	2	1 - 4	36 sec 07	48 sec 41

3. Conclusion

La méthode proposée permet la restauration d'images de contours incomplets et a pour intérêt principal de pouvoir compléter des images de contours présentant un manque important d'information.

Une classification des éléments de contours (états) et des différents types de connexions entre deux états est déterminée. Seules les interactions locales au niveau du pixel sont prises en compte, mais chaque site détient une information plus globale : la direction macroscopique.

Pour compléter les données manquantes, on effectue successivement des prolongations et des raccourcissements de lignes de contours, ceci dans le but d'obtenir une configuration stable où apparaissent des « T-jonctions ».

Des modifications importantes ont été apportées par rapport à l'algorithme initialement proposé par J.L. Marroquin afin de pouvoir traiter des données réelles (notamment des images SPOT) et bruitées.

Des améliorations restent à apporter :

- Agrandir le nombre de configurations permettant une meilleure initialisation des états (sans pour autant augmenter le nombre d'états), ceci pour mieux traiter les images bruitées.
- Introduire dans la définition du champ de Markov, un apport supplémentaire d'information provenant de l'image originale (texture, multirésolution...).
- Introduire dans la fonction d'énergie un terme de rappels aux données.

Nous travaillons actuellement sur une autre méthode de restauration de contours, basée également sur une modélisation Markovienne, et prenant en compte un nombre d'états plus restreint et une direction de lignes de contours réelle (et non discrète).

BIBLIOGRAPHIE

- [1] R. AZENCOTT, « Markov fields and image analysis », *Proc. AFCET*, Antibes, 1987.
- [2] J. BESAG, « On the statistical analysis of dirty pictures », *JL of Royal Statistical Society*, series B, Vol. 68, pp. 259-302, 1986.
- [3] A. BLAKE et A. ZISSERMAN, « Visual reconstruction », *MIT Press*, Cambridge-MA, 1987.
- [4] J. CANNY, « A computational approach to edge detection », *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-8 (6), pp. 679-698, 1986.
- [5] B. CHALMOND, « Image restoration using Estimated Markov Model », *Signal Processing*, Vol. 15, pp. 115-129, 1988.
- [6] B. CHALMOND et C. GRAFFIGNE, « Expérience approfondie d'utilisation des techniques markoviennes en imagerie », Rapport « *Signal et Image* » du GDR TdSI, Sept. 1992.

- [7] R. DERICHE, « Using Canny's criteria to derive a recursively implemented optimal edge detector », *International Journal of Computer Vision*, pp. 167-187, 1987.
- [8] D. GEIGER et F. GIROSI, « Parallel and deterministic algorithms for MRFs : surface reconstruction and integration », *Proc. ECCV*, Antibes, Avril 1990.
- [9] G. GEMAN et D. GEMAN, « Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images », *IEEE Trans. Pattern Analysis, Machine Intel.*, Vol. PAMI-6, pp. 721-741, Nov. 1984.
- [10] W.D. HILLIS, *The Connection Machine*, MIT Press, 1985.
- [11] J.L. MARROQUIN, « A Markovian Random field of Piecewise Straight Lines », rapport interne du « *Centro de Investigación en Matemáticas* », Mexico, 1989.
- [12] J.L. MARROQUIN, « A Markovian Random Field of Piecewise Straight Lines », *Biological Cybernetics*, pp. 457-465, 1989.
- [13] S. URAGO, M. BERTHOD et J. ZERUBIA, « Restauration d'image de contours incomplets par modélisation par champs de Markov », Rapport de recherche INRIA n° 1688, juin 1992.
- [14] P. VAN LAARHOVEN et E. AARTS, « Simulated Annealing : Theory and Applications », *D. Reidel Publishing Company*, Dordrecht, Holland, 1987.
- [15] J. ZERUBIA et C. GRAFFIGNE, « Rapport sur la Segmentation : Approches statistiques », Rapport « *Segmentation des images* » du GDR n° 134 TdSI, déc. 1991.
- [16] J. ZERUBIA et F. PLOYETTE, « Détection de contours et lissage d'image par deux algorithmes déterministes de relaxation. Mise en œuvre sur la machine à connexions CM2 », revue *Traitement du Signal*, Vol. 8 - n° 3, p. 165-179, 1991.

LES AUTEURS



Sabine UGARO a obtenu en 1991 un Magistère ESSI et un D.E.A. à l'Université de Nice. Elle a commencé une thèse à l'INRIA Sophia-Antipolis en janvier 1992. Ses travaux de recherche en vision par ordinateur sont essentiellement basés sur la détection et le groupement de contours en utilisant une modélisation Markovienne.



Marc BERTHOD est diplômé de l'École Polytechnique (1969). Il a obtenu une thèse de 3ième cycle sur la reconnaissance de caractères en 1975 et une thèse d'État sur les méthodes de relaxation en 1980. Il est Directeur de Recherche INRIA et chef d'un projet de vision par ordinateur (Projet d'Analyse de Scènes et Traitement d'Images Symboliques). Ses centres d'intérêts comprennent la reconstruction 3-D d'images satellitaires, l'interprétation symbolique, la modélisation Markovienne et les méthodes d'optimisation qui y sont associées.



Josiane ZERUBIA est actuellement chargée de recherche à l'INRIA. Diplômée de l'ENSIEG-INPG (1981), elle a travaillé comme ingénieur de recherche chez Hewlett-Packard jusqu'en 1984. Elle a préparé une thèse de docteur-ingénieur (1986) et une nouvelle thèse (1988) au LASSY - Université de Nice - URA CNRS n° 1376. Elle a ensuite passé une année post-doctorale au « *Signal and Image Processing Institute* » de USC à Los Angeles en 1989. Principaux thèmes de recherche : traitement d'image, modélisation par champs de Markov, estimation de paramètres et connexionisme.

Manuscrit reçu le 2 juillet 1992.