

# Algorithmes stochastiques synchrones pour les architectures cellulaires

## Synchronous stochastic algorithms for cellular architectures

par Jacques-Olivier KLEIN et Antoine DUPRET

Institut d'Électronique Fondamentale - C.N.R.S.UMR. 8622  
Bat. 220 Université de Paris Sud 91405 Orsay France

### *résumé et mots clés*

Dans cet article, nous proposons un algorithme parallèle pour la restauration d'images et la détection de contours. Son originalité réside dans l'emploi de techniques de relaxation stochastique combinées à des réseaux résistifs non-linéaires. Sa pertinence pour l'amélioration d'images et son adéquation à une architecture cellulaire mixte analogique numérique sont simultanément considérées. Les simulations présentées montrent l'efficacité de cet algorithme, y compris dans le contexte d'images particulièrement bruitées.

algorithme stochastique, traitement d'images, architecture cellulaire, réseaux résistifs non-linéaires.

### *abstract and key words*

In this paper, an algorithm dedicated to image restoration and edge detection is addressed. Its principle is based on synchronous stochastic relaxation and resistive fuses. The adequacy to both image enhancement and the implementation on a cellular architecture are considered. Results of simulations are given and demonstrate the efficiency of our algorithm even on very noisy images.

Stochastic algorithm, image processing, cellular architecture, resistive fuses.

## 1. introduction

Le traitement d'image est un domaine de prédilection pour les algorithmes et les architectures parallèles. De nombreux travaux ont porté sur l'adéquation algorithme architecture et le travail présenté dans ce papier s'inscrit dans ce cadre. D'une part, nous développons des architectures intégrées mixtes analogiques et numériques [1-3] pour réaliser un traitement à cadence vidéo, à faible coût, ayant une faible consommation et un encombrement réduit. D'autre part, notre intérêt se porte sur les algorithmes stochastiques. En effet, ces derniers permettent d'effectuer conjointement une régularisation et une détection de contours, optimale au sens d'une fonction de coût. Malheureusement, ces algorithmes restent peu utilisés du fait du volume de calcul trop important qui mène à des temps de simulation excessifs. Des travaux ont conduit à la définition de modèles synchrones [4-5]

permettant une parallélisation massive de ces algorithmes, mais les traitements autorisés par ces modèles sont limités. Il en résulte que les algorithmes conçus à ce jour pour des applications réelles de traitement d'image ne peuvent pas être implantés aujourd'hui sur le type d'architectures que nous visons. Avec l'augmentation rapide de la vitesse des processeurs et de l'intégration des circuits reconfigurables, le traitement des images à la cadence vidéo par des algorithmes numériques déterministes devient de plus en plus accessible [6-7]. A l'inverse, même sur des architectures numériques vectorielles [8] ou massivement parallèles [9], les temps d'exécution des algorithmes itératifs stochastiques restent trop élevés pour accéder à la cadence vidéo. Nous proposons une nouvelle approche, moins formelle, mais qui permet de concilier la faisabilité d'une mise en œuvre matérielle et l'efficacité des algorithmes stochastiques. Dans ce papier, un algorithme stochastique synchrone original est d'abord décrit dans la section 2. Nous l'étudions ensuite par simulation numérique dans

la section 3. Enfin des perspectives de mise en œuvre cellulaires sont présentées dans la section 4.

## 2. algorithme proposé

### 2.1. objectif

Les algorithmes stochastiques permettent d'effectuer simultanément une régularisation d'images et une détection de contours de bonne qualité. Le surcoût du tirage au sort n'est pas forcément rédhibitoire, mais les algorithmes proposés à ce jour ne peuvent pas être implantés dans des architectures compactes. C'est pourquoi nous proposons dans ce papier, un algorithme original bien adapté à la mise en œuvre par une structure parallèle intégrée mixte analogique numérique.

### 2.2. formulation du problème

Le problème de la régularisation et de la détection de contours peut, par exemple, être posé de la manière suivante [10].

Soit  $\mathbf{o} \in \mathbb{R}_+^{N \times M}$ , l'image de dimension  $N \times M$  observée,  $\mathbf{r} \in \mathbb{R}_+^{N \times M}$  l'image améliorée et  $\mathbf{c} \in \{0, 1\}^{2 \times (N-1) \times (M-1)}$  la configuration des contours entre chaque pixel. Nous notons  $c_b$  l'état binaire (l'absence ou la présence) d'un contour  $b$ .

Nous définissons ainsi une fonction de coût  $E(\mathbf{o}, \mathbf{r}, \mathbf{c})$  que nous cherchons à minimiser. Classiquement [11] elle s'exprime :

$$E(\mathbf{o}, \mathbf{r}, \mathbf{c}) = E_1(\mathbf{o}, \mathbf{r}) + E_2(\mathbf{r}, \mathbf{c}) + E_3(\mathbf{c}) \quad (1)$$

où

- $E_1$  est un terme de rappel aux données

$$E_1(\mathbf{r}, \mathbf{o}) = \sum_i \frac{(o_i - r_i)^2}{\sigma^2} \quad (2)$$

- $E_2$  est un terme de régularisation

$$E_2(\mathbf{r}, \mathbf{c}) = \alpha \sum_{(i,j)} (r_i - r_j)^2 (1 - c_{(i,j)}) \quad (3)$$

où  $c_{i,j}$  représente la configuration  $c_b$  du contour  $b$  situé entre les pixels  $i$  et  $j$ . Ce terme prend en compte les contours afin que la régularisation ne les affecte pas.

- $E_3$  est un terme faisant intervenir les potentiels  $V_k(\mathbf{c})$  des cliques  $k$  favorisant certaines configurations locales de contours [10, 11].

$$E_3(\mathbf{c}) = \sum_k V_k(\mathbf{c}) \quad (4)$$

### 2.3. présentation de la méthode de minimisation

Pour minimiser  $E(\mathbf{o}, \mathbf{r}, \mathbf{c})$ , nous allons chercher à minimiser séparément  $E_1 + E_2$  en faisant varier  $\mathbf{r}$  et  $E_2 + E_3$  en faisant varier  $\mathbf{c}$ . L'algorithme se présente donc comme une itération sur les étapes A et B suivantes :

A) chercher  $\mathbf{r}$  qui minimise  $E_1 + E_2$  à  $\mathbf{c}$  constant par un réseau résistif déterministe associé à des interrupteurs (figés durant cette phase),

B) chercher  $\mathbf{c}$  qui minimise  $E_2 + E_3$  en conservant  $\mathbf{r}$  constant par un algorithme stochastique itératif synchrone.

Cet algorithme comprend deux niveaux d'itération. Les itérations de la relaxation stochastique de l'étape « B » constituent la boucle interne de l'algorithme alors que les itérations de succession des étapes « A » et « B » constituent la boucle externe de l'algorithme.

#### 2.3.1. réseau résistif

Pour effectuer la minimisation de l'énergie  $E_1 + E_2$ , nous nous sommes inspirés des réseaux résistifs [12], et plus spécifiquement des réseaux de résistances fusibles qui réalisent un lissage en préservant les discontinuités. L'architecture de notre réseau résistif est représentée sur la figure 1. Les informations de niveau de gris des pixels des images  $\mathbf{o}$  et  $\mathbf{r}$  sont codées par les potentiels électriques des noeuds du réseau. Des interrupteurs sont placés entre chaque pixel de l'image  $\mathbf{r}$  (ex. :  $r_i$  et  $r_j$ ). Ils représentent l'état du contour séparant les pixels  $r_i$  et  $r_j$ . Un interrupteur ouvert correspond à un

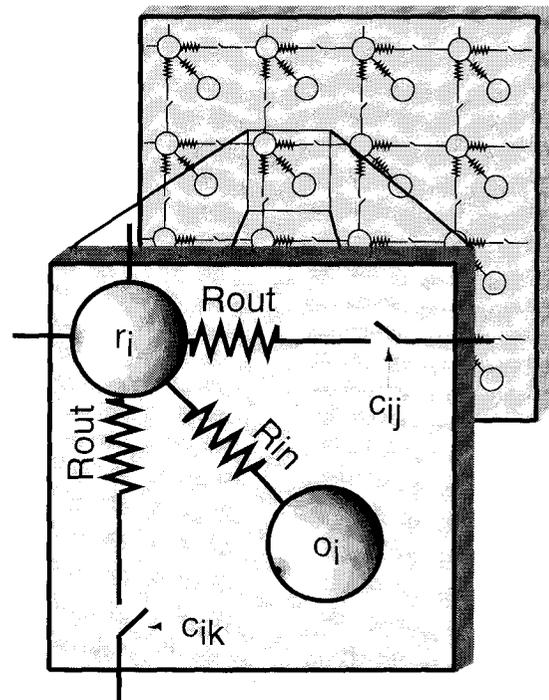


Figure 1. – Réseau de résistances fusibles.

contour présent ( $c_{ij} = 1$ ). Tant que l'état des interrupteurs ne varie pas, le réseau n'est constitué que d'éléments linéaires. Il converge donc sans aucun risque d'instabilité vers le minimum global de la puissance dissipée dans les résistances  $R_{in}$  et  $R_{out}$ . C'est un énorme avantage par rapport aux réseaux non linéaires [13] qui peuvent présenter des oscillations et converger vers des minima locaux de la puissance dissipée. Ainsi, si l'image observée est présentée en entrée, l'image reconstruite en sortie sera celle qui minimise :

$$P_d = P_1(\mathbf{r}, \mathbf{o}) + P_2(\mathbf{r}, \mathbf{c}) \quad (5)$$

où la puissance  $P_1$  dissipée à travers les résistances  $R_{in}$  représente le terme de rappel aux données

$$P_1(\mathbf{r}, \mathbf{o}) = \sum_s \frac{(o_s - r_s)^2}{R_{in}} \quad (6)$$

et où la puissance  $P_2$  dissipée à travers les résistances  $R_{out}$  représente le terme de régularisation :

$$P_2(\mathbf{r}, \mathbf{c}) = \sum_{(s,s')} \frac{(r_s - r_{s'})^2 (1 - c_{(s,s')})}{R_{out}} \quad (7)$$

Bien évidemment,  $E_1 = P_1$  si  $R_{in} = \sigma^2$  et  $E_2 = P_2$  si  $R_{out} = \frac{1}{\alpha}$ . Ainsi notre réseau résistif réalise l'étape « A ». Nous allons donc nous intéresser à l'étape « B ».

### 2.3.2. relaxation stochastique

La détection des contours qui cherche à minimiser  $E_2 + E_3$  est réalisée par une relaxation stochastique. Nous garderons à l'esprit que l'algorithme stochastique doit être suffisamment efficace pour fournir des résultats satisfaisants mais suffisamment simple pour pouvoir être réalisé de façon compacte à l'aide d'opérateurs mixtes intégrés.

L'algorithme proposé consiste à itérer les étapes B.1 et B.2 suivantes.

B.1 D'abord, pour chaque contour éventuel  $b$  est calculé un potentiel

$$V_b = \Phi(N_b, g_b) \quad (8)$$

où  $N_b$  représente l'ensemble des contours voisins de  $b$  et  $g_b$  la norme du gradient dans la direction orthogonale au contour  $b$ . Le potentiel calculé par la fonction  $\Phi$  correspond à l'influence (positive pour excitatrice ou négative pour inhibitrice) de la configuration du voisinage locale des contours et du gradient sur le contour considéré. Son rôle est analogue au potentiel post-synaptique dans les réseaux de neurones ou à l'énergie d'une configuration dans [10]. A l'image des fonctions de cliques utilisées dans le MAP de [10] et dans la machine de Boltzmann synchrone [5] la fonction  $\Phi$  est déterministe et arbitraire. Bien évidemment  $\Phi$  croît avec le gradient  $g_b$ , mais est également modulé par les configurations de contours du voisinage. Par exemple, au

§ 3.1.3., nous favoriserons les configurations formant une frontière continue et défavoriserons toutes les autres.

B.2 Ensuite les états de tous les contours sont tirés au sort simultanément suivant la probabilité :

$$P(c_b^{t+1} = 1 | g_b^t, N_b^t) = f(V_b^t) \quad (9)$$

La fonction d'activation  $f$  est croissante, bornée dans l'intervalle  $]0, 1[$  et telle que  $f(0) = 1/2$ . Classiquement, la fonction d'activation  $f$  choisie est une sigmoïde.

$$f(V_b) = \frac{1}{1 + e^{-V_b/T}} \quad (10)$$

où le paramètre  $T > 0$  représente la température et décroît au cours du recuit simulé. Lors de réalisations matérielles [14, 15] le tirage au sort est réalisé de façon équivalente et plus compacte en comparant  $V_b$  à un bruit pseudo-aléatoire de distribution gaussienne centrée d'écart-type  $4T$  dont la fonction de répartition est assez proche d'une sigmoïde.

## 2.4. discussion

A ce point de la discussion, il est important de rappeler que cet algorithme est destiné à être mis en œuvre par une architecture cellulaire mixte analogique numérique. Le degré de parallélisme est directement lié à la surface des cellules. Celles-ci doivent donc être de surface minimale et utiliser des opérateurs aussi simples que possible.

Cet algorithme est *synchrone* tout comme la *machine de Boltzmann synchrone* introduite par Azencott [4], tous les états des contours sont tirés au sort simultanément. Ce point est primordial car il permet de profiter pleinement du parallélisme des circuits. Il nous différencie notamment de la machine de Boltzmann (séquentielle) de Hinton, du MAP présenté par [10], du recuit quasi-statique [16] et même des travaux de [17] dans lesquels le parallélisme n'est jamais total.

Pour un tel réseau stochastique et synchrone, la loi limite n'est bien connue que dans le cadre des machines de Boltzmann synchrones à états binaires et à interactions symétriques par paires (cliques d'ordre 2) [4]. Même dans ce cas, l'énergie minimisée n'est jamais quadratique. Elle ne peut donc pas être de la forme  $E_2 + E_3$  donnée par l'expression (3). De plus, comme pour la plupart des réseaux neuronaux, les machines de Boltzmann requièrent plusieurs couches de neurones pour réaliser une séparation qui ne peut pas être représentée par un hyper-plan dans l'espace des entrées [18]. Or la prise en compte des configurations locales de contours est une opération non linéaire. Par exemple, à l'intersection de quatre contours, il est souhaitable de favoriser les configurations ayant 0 ou 2 contours actifs par rapport aux configurations ayant 1, 3 ou 4 contours actifs. Pour favoriser de telles configurations il est indispensable de recourir soit à des sous réseaux multicouches soit à des cliques

d'ordre supérieur à 2. Comme la surface dédiée à chaque pixel limite le parallélisme d'une réalisation intégrée, il est clairement impossible d'envisager l'intégration d'un sous réseau neuronal multicouche en chaque point de contours. Nous devons donc renoncer à utiliser le modèle de la machine de Boltzmann à interactions par paires. A ce jour, nous ne connaissons aucun modèle permettant de déterminer l'énergie minimisée et la loi limite du réseau est inconnue. Ainsi, pour un tel algorithme, l'expérimentation sera la seule démonstration (empirique) de son efficacité. De ce fait, plus aucune contrainte (notamment de symétrie) ne pèse sur l'expression de la fonction  $\Phi$ . Nous pouvons la choisir de façon à minimiser la complexité de sa réalisation matérielle. Dans les exemples qui suivent, l'influence de la configuration locale des contours peut être aisément calculée par un opérateur combinatoire local. La faisabilité de sa réalisation parallèle intégrée a déjà été démontrée, notamment dans [6].

### 3. simulations

Afin d'évaluer l'efficacité de notre algorithme, nous avons réalisé des simulations numériques.

#### 3.1. algorithmes simulés

Notre approche est guidée par la perspective d'implantation cellulaire. Les choix algorithmiques sont dictés par un compromis entre la simplicité de mise en œuvre des opérateurs en technologie intégrée et leur efficacité. Ainsi, l'algorithme peut se décliner en différentes versions suivant le choix :

- du calcul de la norme du gradient  $g$  adopté,
- de la fonction  $f$  utilisée,
- de l'expression de la fonction  $\Phi$ .

Nous allons présenter et justifier les différents choix que nous avons effectués pour réaliser nos simulations.

##### 3.1.1. norme du gradient $g$

Pour estimer le gradient suivant la direction orthogonale au contour nous avons choisi l'opérateur le plus simple. Dans toutes nos simulations,  $g$  représente simplement la valeur absolue de la différence de niveau de gris entre les pixels que séparent les contours.

$$g_{(i,j)} = |r_i - r_j| \tag{12}$$

Nous comptons sur la régularisation effectuée par le réseau résistif et la nature stochastique du processus pour rendre le traitement peu sensible au bruit.

##### 3.1.2. fonction d'activation $f$

Le choix de la fonction d'activation repose sur nos précédentes expériences de conception de générateurs pseudo-aléatoires massivement parallèles pour circuits intégrés dédiés [14]. Elle consiste à approcher une fonction sigmoïdale par la fonction erreur. Dans toutes nos simulations, le tirage au sort est effectué en comparant  $V_i$  à une valeur pseudo-aléatoire à distribution  $\mathcal{N}(x/4T)$  gaussienne centrée d'écart type  $4T$ . La fonction d'activation est alors

$$f(v_i) = \int_{-\infty}^{V_i} \mathcal{N}(u/4T) du \tag{13}$$

Tous les algorithmes simulés ont donc la structure suivante illustrée par la figure 2 :

```

T = Tinit
Pour t de 1 à NB.ITERATION
  Recherche de r qui minimise globalement P1 + P2 à c constant
  /* i.e. attendre la stabilisation du réseau résistif et échantillonner
  le résultat r[t] */
  Pour k de 1 à NB_BOUCLE_INTERNE
    Pour tous les contours
      Calculer la norme du gradient ( $g_c$ ) dans la direction orthogonale
      au contour
      Calculer l'influence globale  $V_c$  du réseau sur l'état de ce
      contour
    Fin Pour
    /* synchronisation */
    Pour tous les contours
      Tirer au sort une variable  $\chi_c$  à distribution normale centrée
      d'écart type  $4 \times T$ .
      Mettre à jour  $c[t + 1]$  en comparant  $V_c$  et  $\chi_c$  :
      
$$c[k + 1] = \begin{cases} 0 & \text{si } V_c < \chi_c \\ 1 & \text{si } V_c \geq \chi_c \end{cases} \tag{14}$$

    Fin Pour
  Fin Pour
  Diminuer T
Fin Pour
    
```

Figure 2.

##### 3.1.3. expression de la fonction $\Phi$

La fonction  $\Phi$  représente l'influence du gradient et de la topologie des contours voisins sur la probabilité d'activation d'un contour. Dans sa forme la plus simple, l'expression de  $V_c = \Phi(N_c, g_c)$  ne prend en compte que la norme du gradient, ignorant les contours voisins.

$$V_c = g - V_{seuil}$$

Cependant, afin de favoriser l'apparition de configuration de contours fermés et fins, nous avons repris, dans le contexte d'un processus stochastique, des techniques classiques en détection de contours tels que la suppression des non-maxima locaux et le seuillage à hystérésis.

### 3.1.3.1. seuillage à hystérésis

Pour prendre en compte la configuration locale du voisinage, nous classons les candidats contours en deux catégories. Pour chaque catégorie, le seuil de détection des contours est différent. La première catégorie est constituée des candidats contours poursuivis par un contour et un seul à chaque extrémité. Elle est favorisée par un seuil de détection bas. La seconde catégorie regroupe tous les autres contours. Dans ce cas, la présence du contour est inhibée par un seuil de détection plus élevé. Ce processus est à rapprocher du seuillage à hystérésis classiquement utilisé en traitement d'image pour favoriser la fermeture des contours.

Pour cela, considérons le contour  $c_x$  entouré des contours  $c_a, c_b, c_c, c_d, c_e, c_f$  organisés comme le présente la figure 3.

La proposition  $P_x$  « le contour  $c_x$  est connecté à chaque extrémité à exactement un contour » est équivalente à

$$P_x = (a \oplus b \oplus c) \overline{abc} (d \oplus e \oplus f) \overline{def} \quad (15)$$

où  $\oplus$  représente le ou exclusif.

L'algorithme se présente donc comme suit :

si chaque extrémité de  $c$  est connecté à exactement un contour  
alors

$$V_c = g - V_{seuil\_bas}$$

sinon

$$V_c = g - V_{seuil\_haut}$$

fin si

### 3.1.3.2. suppression des non-maxima locaux

Nous avons également cherché à ne conserver comme candidats contours que les maxima locaux du gradient. Par exemple, dans nos simulations, nous annulons la norme du gradient si elle n'est pas supérieure à celle des deux contours parallèles adjacents. Ceci n'ajoute aucun paramètre supplémentaire dont le réglage aurait été délicat. Il s'agit là d'une opération locale peu coûteuse à réaliser pour une implantation cellulaire. Pour un algorithme déterministe,

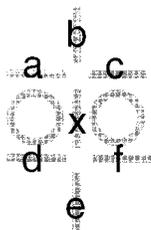


Figure 3. – Configuration locale des contours.

elle évite de fournir des contours épais lorsque le gradient varie lentement. Nous verrons avec les simulations que son rôle est ici différent. L'algorithme est présenté ci-dessous.

si  $g$  est un maximum local du gradient

$$V_c = g - V_{seuil}$$

sinon

$$V_c = -V_{seuil}$$

fin

### 3.1.3.3. seuillage hystérésis et suppression des non-maxima locaux

Enfin, pour obtenir des contours à la fois fermés et fins, nous avons combiné les deux algorithmes précédents (seuillage à hystérésis et suppression des non-maxima locaux) dans l'algorithme suivant.

si  $g$  est un maximum local du gradient

si chaque extrémité de  $c$  est connecté à exactement un contour

$$V_c = g - V_{seuil\_bas}$$

sinon

$$V_c = g - V_{seuil\_haut}$$

fin si

sinon

si chaque extrémité de  $c$  est connecté à exactement un contour

$$V_c = -V_{seuil\_bas}$$

sinon

$$V_c = -V_{seuil\_haut}$$

fin si

fin si

La figure 4 illustre une mise œuvre possible de cet algorithme par une architecture cellulaire mixte.

Sauf indications contraires, nous avons utilisé cette dernière fonction  $\Phi$  dans les simulations présentées ci-dessous.

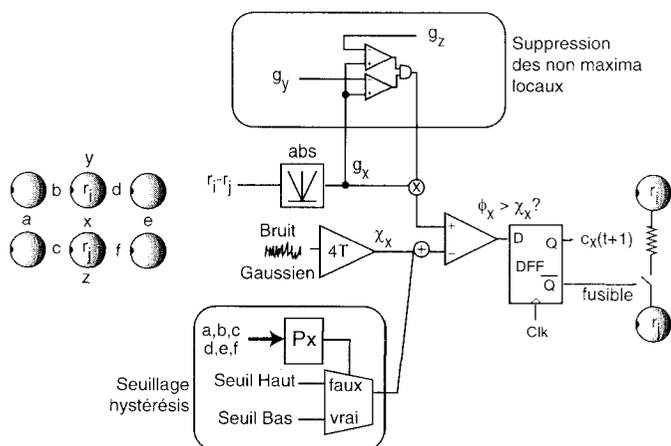


Figure 4. – Structure du calcul d'un contour.

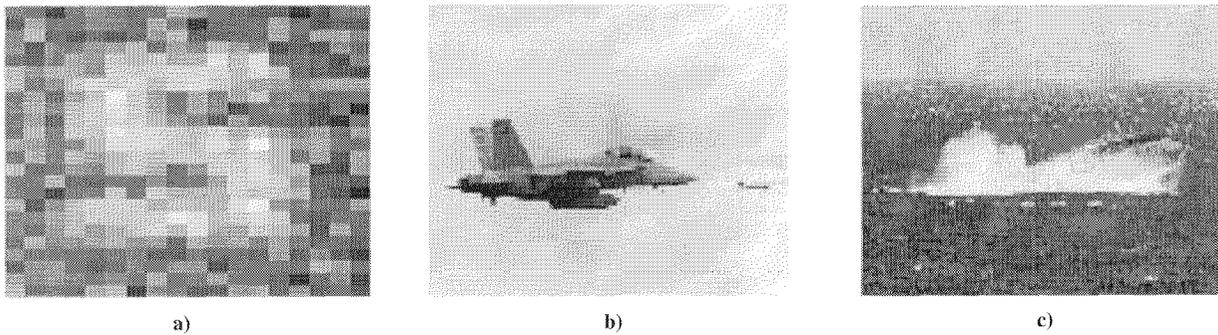


Figure 5. – a) Image originale du « s », b) Image originale de l'avion FA18, c) Image originale du sous-marin.

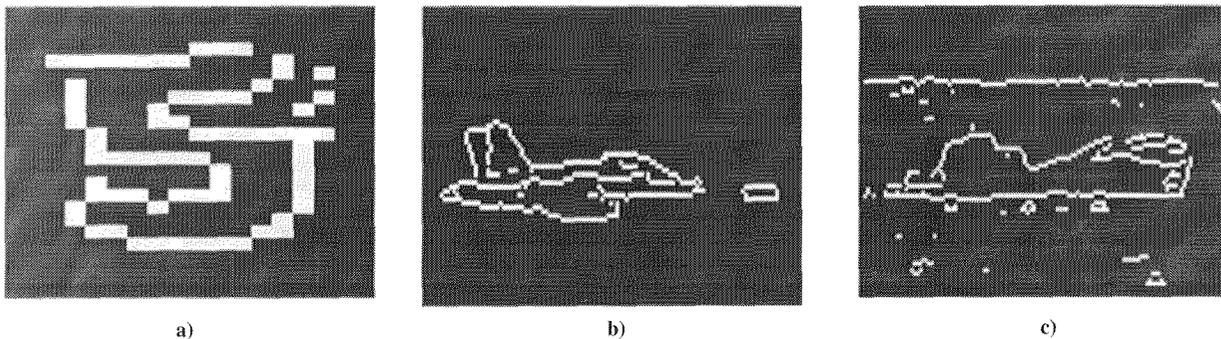


Figure 6.1. – a) Image du S filtre moyennneur  $3 \times 3$  + filtre de Sobel, b) Image du FA 18 Filtre de Canny, c) Image du sous marin filtre moyennneur  $3 \times 3$  + filtre de Sobel.

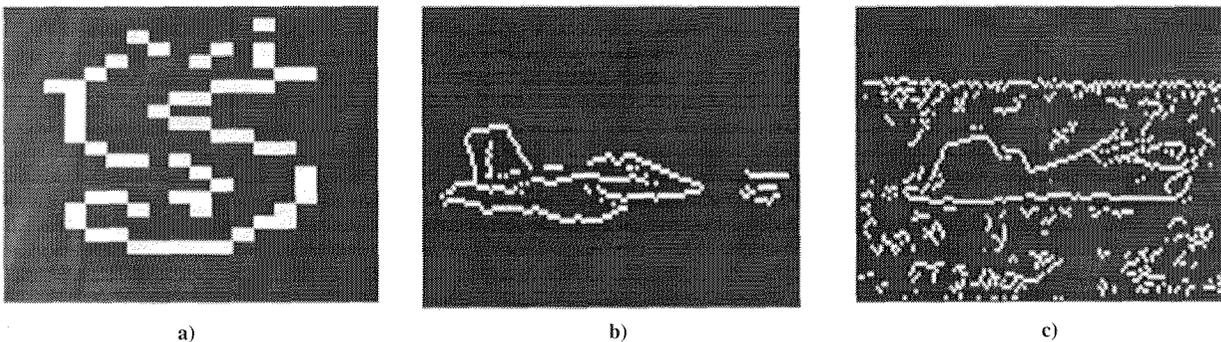


Figure 6.2. – a) Image du S filtre de Canny, b) Image du FA18 filtre de Canny, c) Image du sous-marin filtre de canny.

### 3.2. construction des images de test

Nous avons considéré des images pour lesquelles l'extraction d'une silhouette (un contour continu et fermé) a un sens. Il s'agit donc d'images représentant des objets isolés. De plus, la convergence du réseau résistif et le tirage au sort de l'état des contours sont des opérations assez coûteuses en temps de calcul lorsqu'elles sont simulées, notamment lorsque le rapport  $R_{in}/R_{out}$  est important. Nous avons donc traité des images de petites tailles (cf. figures 5).

La première image est synthétique. Il s'agit d'un signe en forme de « s » de dimension  $24 \times 18$  pixels. Nous l'avons rendue floue en appliquant un filtre moyennneur  $3 \times 3$  et nous avons ajouté du

bruit blanc à distribution gaussienne. Le rapport signal à bruit est de 3,5 dB. Ceci correspond à une image particulièrement difficile à restaurer mais qui reste encore discernable pour l'œil. Les deux autres images proviennent de photographies diffusées sur le site web de l'US Navy. Les originaux étaient en couleurs et de trop grandes dimensions. Nous avons simplement additionné les trois couleurs pour en faire des images en niveaux de gris, puis nous les avons sous échantillonnées d'un facteur cinq pour leur donner des dimensions qui mènent à des temps de simulation raisonnables.

Une à dix minutes sur une station hyper-sparc sont en général suffisantes. La première de ces photographies représente un avion FA18 ( $87 \times 128$  pixels) lançant un missile. Elle est peu bruitée et l'extraction d'un contour presque fermé par une méthode déterministe directe ne pose pas de problèmes (cf. figures 6).

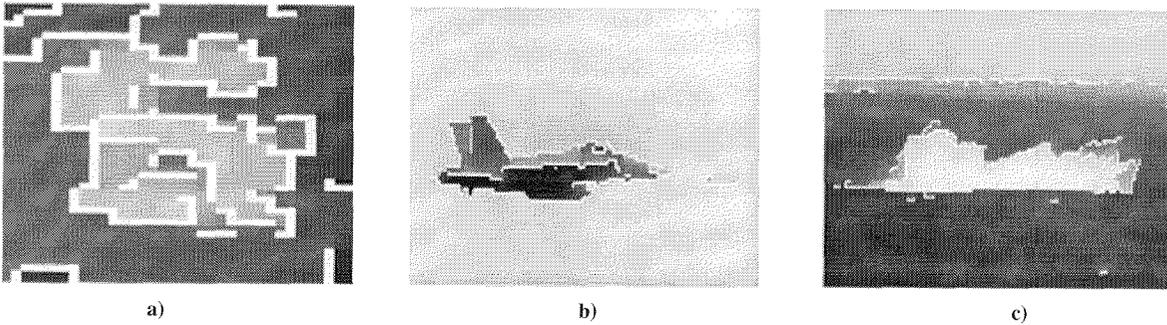


Figure 7. – a) Image du « s » version déterministe, b) Image du FA18 version déterministe, c) Image du sous-marin version déterministe.

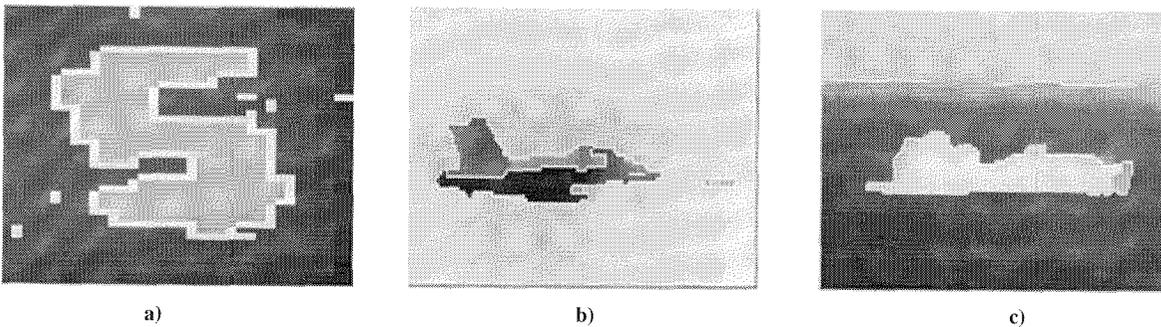


Figure 8. – a) Image du « s » version stochastique, b) Image du FA18 version stochastique, c) Image du sous-marin version stochastique.

Un filtre de moyennage  $3 \times 3$  suivi d'un filtre de Sobel et d'un seuillage avec suppression des non-maxima locaux suffit. La seconde photographie ( $74 \times 124$  pixels) représente un sous-marin faisant surface. L'état de la mer et du ciel donne l'impression d'une image bruitée. En fait, bien que le contraste soit important, l'extraction d'un contour presque fermé sur cette image n'est pas aussi triviale que pour la précédente.

### 3.3. résultats

Les résultats sont représentés sous la forme d'images où les contours sont superposés en blanc sur les images régularisées.

#### 3.3.1. comparaison déterministe versus stochastique

Nous avons comparé nos résultats avec une version déterministe de notre algorithme ainsi qu'à deux filtres optimisés pour la détection de contours : les filtres de Sobel et de Canny [19]. Les résultats présentés sur les figures 7 et 8 montrent clairement l'intérêt du recuit simulé. Ils correspondent à l'algorithme décrit au § 3.1.3.3. : avec suppression des non-maxima locaux et seuillage à hystérésis. Ils ne diffèrent que par la valeur de la température  $T$ . Dans la version déterministe de notre algorithme (à  $T = 0$ ) les contours sont ouverts et irréguliers (cf. figures 7). En réalisant un recuit simulé, les images régularisées apparaissent bien contrastées et entourées par des contours fermés et plus réguliers (cf. figures 8). Les contours superflus sont facilement supprimés par un opérateur de morphologie mathématique (cf.

figures 9). Ce résultat est particulièrement encourageant. En effet, bien qu'aucun modèle connu ne permette de calculer le comportement asymptotique d'une telle relaxation, elle fournit les résultats attendus.

Les résultats présentés sur les figures 6, correspondant aux filtres de Sobel et de Canny (par approximation de la dérivée du filtre gaussien) ont été réalisées pour un grand nombre de paramètres (présence d'un lissage préalable par un filtre moyenneur  $3 \times 3$ , seuils, écart-type du filtre gaussien pour le filtre de Canny) en essayant de fermer les contours à tout prix, quitte à accepter beaucoup de fausses alarmes. Il n'est pas possible de présenter toutes ces simulations, mais aucune d'elles n'a produit des contours totalement fermés comparables à ceux de la figure 8. Nous ne présenterons qu'un résultat par type de filtre et par image. Ces résultats correspondent à la stratégie par défaut de la fonction edge de la boîte à outils traitement d'image de *matlab* qui donne des résultats facilement reproductibles par le lecteur. Le filtre de Sobel est précédé d'un filtre moyenneur  $3 \times 3$  et suivi d'un seuillage avec suppression des non-maxima locaux. Le seuil est fixé à deux fois la moyenne du module du gradient. Le filtre de Canny est suivi d'un seuillage à hystérésis et d'une suppression des non-maxima locaux. Pour le filtre de Canny, l'écart-type du filtre gaussien vaut 1, le seuil haut est déterminé à partir de l'histogramme du module du gradient pour avoir 30 % de candidats contours avant la suppression des non-maxima locaux et le seuil bas vaut 40 % du seuil haut. Les résultats obtenus avec d'autres réglages, y compris un écart-type plus important pour le filtre de Canny, sont en général plus bruités mais ne ferment pas mieux les contours.

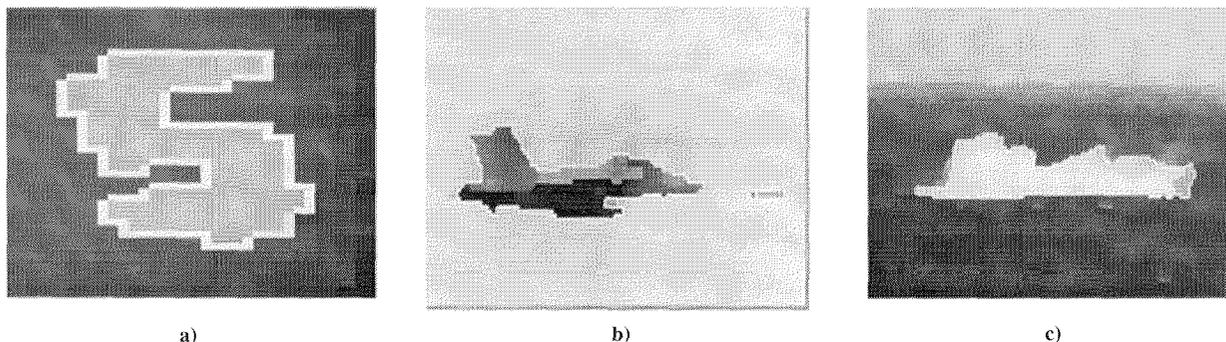


Figure 9. – a) Image du S après suppression des contours ouverts, b) Image du FA18 après suppression des contours ouverts, c) Image du sous-marin après suppression des contours ouverts.

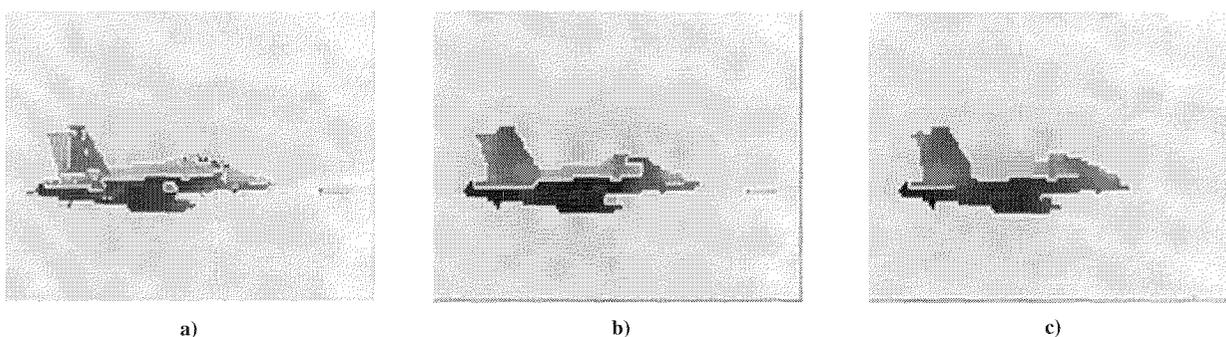


Figure 10. – Image du FA 18 pour différents rapport  $R_{in}/R_{out}$ . a)  $R_{in}/R_{out} = 0$ , b)  $R_{in}/R_{out} = 5$ , c)  $R_{in}/R_{out} = 10$ .

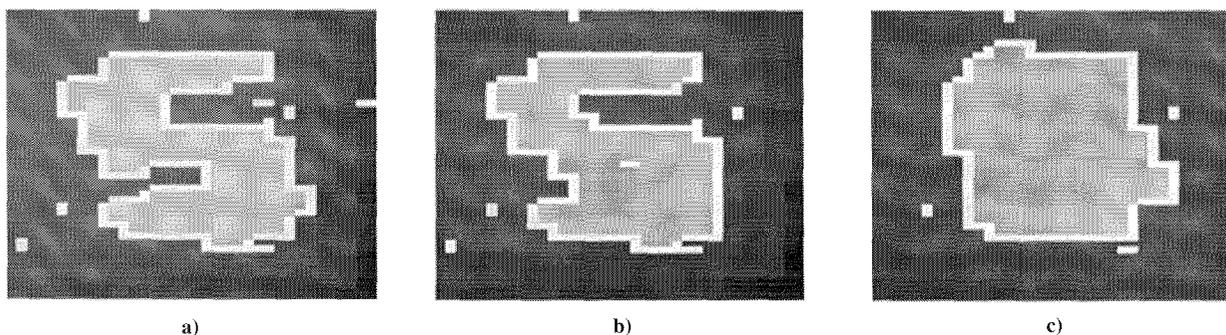


Figure 11. – Image du « s » pour différents rapport  $R_{in}/R_{out}$ , a)  $R_{in}/R_{out} = 5$ , b)  $R_{in}/R_{out} = 7$ , c)  $R_{in}/R_{out} = 15$ .

### 3.3.2. effet du rapport des résistances

Les images 10 et 11 permettent d’apprécier le rôle du rapport entre les résistances  $R_{in}$  et  $R_{out}$ . Dans le cas où aucun contour n’est détecté, le traitement opéré par le réseau résistif est linéaire et isotrope. C’est un filtre passe-bas à réponse impulsionnelle infinie dont la bande passante peut être fixée par le rapport des résistances. Classiquement, la bande passante est directement déduite du niveau de bruit de l’image d’entrée. Malheureusement, les techniques propres aux filtres linéaires et isotropes ne s’appliquent pas ici. En effet, la détection de contours est réalisée par un processus non-linéaire, or la régularisation est contrôlée par les contours et influence leur détection. De ce fait, l’effet du rapport

des résistances sur les résultats (contours et image régularisée) présente un comportement typiquement non-linéaire. Ainsi, pour de faibles valeurs de  $R_{in}/R_{out}$ , le filtrage est quasiment inopérant. Les détails fins sont conservés mais le contour n’est pas fermé. Lorsque le rapport  $R_{in}/R_{out}$  augmente, l’effet du filtrage devient de plus en plus important. Les détails tendent à disparaître mais les contours se ferment. L’intérieur et l’extérieur de l’objet deviennent quasiment uniformes si bien qu’on pourrait presque croire qu’il s’agit d’une segmentation en région. Si nous augmentons exagérément le rapport  $R_{in}/R_{out}$ , le filtrage est tel qu’un effet de craquage des contours apparaît. Dans ce dernier cas, les contours des objets de petites tailles tendent à disparaître. Cela se traduit notamment par la disparition du missile.

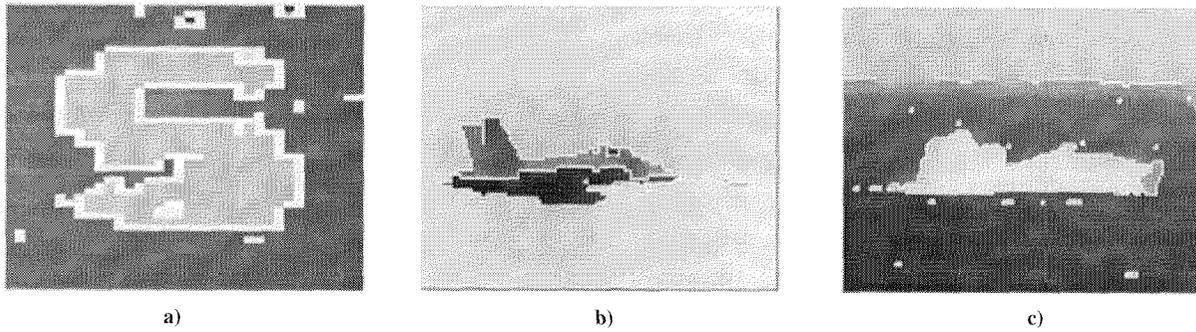


Figure 12. – a) Image du S stochastique. Sans suppression des non-maxima locaux. b) Image du FA 18 stochastique. Sans suppression des non maxima locaux. c) Image du sous-marin versoin stochastique. Sans suppression des non-maxima locaux.

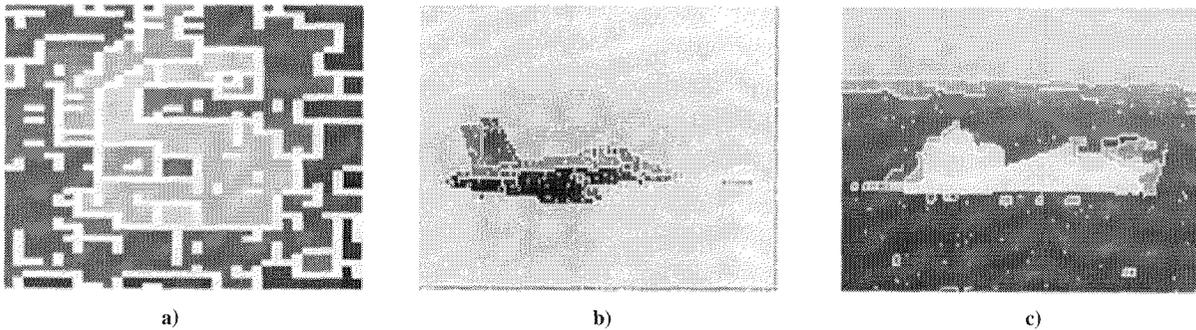


Figure 13. – a) Image du S version stochastique. Sans seuillage hystérésis. b) Image du FA 18 version stochastique. Sans seuillage hystérésis. c) Image du sous marin. Sans seuillage hystérésis.

### 3.3.3. influence de la fonction $\Phi$

Nous avons comparé l'efficacité de notre algorithme pour différentes fonctions  $\Phi$ .

Les figures 12 et 13 montrent les résultats de simulation respectivement en l'absence de suppression des non-maxima locaux et en l'absence de seuillage à hystérésis. Elles permettent ainsi d'apprécier l'efficacité de ces deux méthodes. La suppression des non-maxima locaux était principalement destinée à affiner les contours lorsque le gradient varie lentement. En fait, il apparaît que les contours ne sont pas doublés. La suppression des non-maxima locaux apporte surtout une plus grande immunité au bruit. En revanche, le seuillage à hystérésis était destiné à éliminer les contours ouverts, ce qu'il fait effectivement.

## 4. perspectives d'implantation cellulaire

La partie A de l'algorithme se décrit de façon très simple par une implantation sous forme de réseau résistif (cf. figure 1). De même, la partie B de l'algorithme a été illustrée par une architecture mixte analogique numérique (cf. figure 4). Pour réaliser une

telle architecture totalement parallèle en technologie  $0.6 \mu\text{m}$ , la surface d'un pixel est de  $100 \times 100 \mu\text{m}$  environ dont  $20 \times 20 \mu\text{m}$  pour le photocapteur. Une rétine comprenant une matrice de  $128 \times 128$  pixels et le contrôle associé occuperait une surface un peu inférieure à  $2 \text{cm}^2$ . La vitesse de relaxation serait alors de  $20 \mu\text{s}$  pour 200 itérations. Il est donc possible, avec une technologie très courante aujourd'hui, de disposer d'un circuit réalisant des traitements stochastiques élaborés, qui fournissent des images à une fréquence largement supérieure à la cadence vidéo. Cependant, une telle approche conduit d'une part à un compromis déséquilibré entre les performances en calcul et la surface des pixels et d'autre part, à trop de rigidité dans l'algorithme qui ne pourrait pas s'adapter à la grande variété d'images et d'applications à traiter.

Pour autant, il ne s'agit là que d'une illustration dans la mesure où cet algorithme peut être réalisé par une architecture présentant un moindre degré de parallélisme. Nos recherches actuelles portent notamment sur des architectures mixtes analogiques-numériques programmables, certes moins rapides mais plus compactes et beaucoup plus versatiles [20].

## 5. conclusion

Tout d'abord, nous avons présenté les motivations qui nous ont poussés à définir un algorithme stochastique bien adapté à une

intégration sur un circuit parallèle mixte analogique numérique. Nous avons alors proposé une version stochastique des réseaux de résistances fusibles qui prennent en compte à la fois l'efficacité de l'algorithme et son intégration dans une architecture cellulaire. Il autorise une parallélisation massive des traitements. De plus, il ne requiert que des opérateurs simples qui peuvent être efficacement mis en œuvre par une architecture cellulaire mixte analogique numérique. Enfin, bien qu'il ne se définisse pas comme la minimisation d'une énergie d'expression connue, l'algorithme proposé montre une réelle efficacité pour l'amélioration d'images. Il est particulièrement robuste vis-à-vis du bruit et ferme remarquablement bien les contours.

### BIBLIOGRAPHIE

- [1] H. Pujol, J.-O. Klein, E. Belhaire et P. Garda, "Ra : An analog neurocomputer for the synchronous Boltzmann machine," présenté à MicroNeuro94, Torino, 1994.
- [2] D. Prevost, P. Lalanne, J.-C. Rodier, P. Chavel, A. Dupret, E. Belhaire et P. Garda, "Video-rate simulated annealing for stochastic artificial retinas," *Optics Communications*, vol. 132, pp. 127-431, 1996.
- [3] A. Dupret, E. Belhaire, J.-C. Rodier, P. Lalanne, D. Prevost, P. Garda et P. Chavel, "An Optoelectronic CMOS Circuit Implementing a Simulated Annealing Algorithm," *IEEE Journal of Solid State Circuits*, vol. 31, n° 7, pp. 1046-1050, 7 juillet 1996.
- [4] R. Azencott, "Synchronous Boltzmann Machines and Gibbs Field : Learning Algorithms," présenté à Neurocomputing : algorithms, architectures and applications, Les arcs, 1990.
- [5] R. Azencott, "High-Order Interactions and Synchronous Learning," Proceedings of stochastic models, statistical methods and algorithms in image analysis, Ed. P. Barone, A. Frigessi et M. Piccioni, Lecture Notes in Statistics 74, pp. 14-45, Springer-Verlag, Berlin 1992.
- [6] P. Garda, A. Reichart, H. Rodriguez, F. Devos et B. Zavidovique, "Une rétine électronique automate cellulaire," *Traitement du signal*, vol. 5, pp. 435-449, 1988.
- [7] D. Demigny, "Transformé de Hough, détection de segment temps réel. Test pour une coopération FPGA/DSP?," présenté à Journées Adéquation Algorithme Architecture en traitement du signal et images, CEA/LETI, Saclay, France, 1998.
- [8] J. Lacaille, "Machine de Boltzmann : Théorie et Applications," : Thèse de doctorat, Université Paris XI, 25 juin 1992.
- [9] R. Azencott, A. Doutriaux et L. Younes, "Synchronous Boltzmann Machines and outline based classification," présenté à Proc. of International Neural Network Conf. Paris, Boston, 1990.
- [10] S. Geman et D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian Restoration of Images," *IEEE transaction on pattern analysis and machine intelligence*, vol. PAMI-6, pp. 721-741, 1984.
- [11] C. Graffigne, "Approches frontière : méthodes markoviennes," in *Analyse d'images : filtrage et segmentation*, Masson, Ed., 1995, pp. 177-190.
- [12] C. A. Mead, "Adaptive retina," in *Analog VLSI Implementation of Neural Systems*, vol. 80, C. A. M. a. M. Ismail, Ed. : Kluwer Academic Publishers, 1989, pp. 239-246.
- [13] P. C. Yu, S. J. Decker, H. S. Lee et C. G. Sodini, "CMOS resistive fuse for image smoothing and segmentation," *IEEE Journal of Solid States Circuits*, vol. 27, pp. 545-553, 1992.
- [14] A. Dupret, "A scalable generator of gaussian white noise sources for VLSI implementation," *Electronic Letters*, vol. 31 n°17, pp. 1457-1458, 1994.
- [15] P. Lalanne, J.-C. Rodier, P. Chavel, E. Belhaire et P. Garda, "Optoelectronic device for Boltzmann machines and simulated annealing," optical engineering, vol. 32, pp. 1904-1914, 1993.
- [16] D. Prevost, P. Lalanne, L. Garnerio et P. Chavel, "Quasi-Static Algorithm for image restoration preserving discontinuities," presented at Neural and stochastic methods in image and signal processing III, San Diego, USA, 1994.
- [17] L. Younes, "Synchronous Random Field and Image restoration," ENS Cachan, Publication CMLA 9403, 1994.
- [18] M. L. Minsky et S. A. Papert, *Perceptrons - Expanded Edition*. Cambridge : MA : MIT Press, 1988.
- [19] J.F.Canny, "A computational approach to edge detection," *IEEE transaction on pattern analysis and machine intelligence*, vol. PAMI-8, n°6, pp. 679-698, 1986.
- [20] J.-O. Klein, A. Dupret, S. Moutault et A. Nshare, "Vers une nouvelle génération de rétines programmables", présenté au Colloque GRETSI'99, pp. 596-598 Vannes, 13-17 Septembre 1999.

Manuscrit reçu le 7 décembre 1998.

### LES AUTEURS

Jacques-Olivier KLEIN



Jacques-Olivier Klein (32 ans) est maître de conférences à l'IUT de Cachan. Ses travaux de recherche, qu'il effectue au sein du département AXIS de l'Institut d'Électronique Fondamentale, portent sur les architectures cellulaires pour la vision et la recherche d'algorithmes adaptés aux traitements analogiques. Il codirige avec Antoine Dupret le projet « Parallel Analog Retina-like Image Sensor » (PARIS) qui vise à concevoir des capteurs d'images intelligents incorporant des unités de traitement analogiques versatiles et à définir les algorithmes associés.

Antoine DUPRET



Antoine Dupret (33 ans) est maître de conférences à l'IUT de Villetaneuse. Ses travaux de recherche, qu'il effectue au sein du département AXIS de l'Institut d'Électronique Fondamentale, portent sur les rétines électroniques, l'architecture de leurs cellules de traitement analogiques et de leurs dispositifs de transduction photoélectrique. Il codirige avec Jacques-Olivier Klein le projet « Parallel Analog Retina-like Image Sensor » (PARIS) qui vise à concevoir des capteurs d'images intelligents incorporant des unités de traitement analogiques versatiles et à définir les algorithmes associés.