H. Maître, F. Tupin. M. Sigelle

Polycopié du cours du DEA IARFA

Techniques de traitement d'images

Département TSI - Télécom-Paris

Chapitre 7

Définition et simulation d'un champ de Markov

Chapitre rédigé par Florence TUPIN et Marc SIGELLE

L'information véhiculée par une image va bien au-delà de la seule donnée des niveaux de gris en chaque site (pixel), et la description se fait en termes de zones, contours, structures définis par les contrastes, textures, etc. qui peuvent être présents dans l'image. Le niveau de gris en un site n'est donc souvent pas significatif en lui-même, mais dans ses relations et interactions avec les pixels voisins.

Cette propriété des images, à savoir les interactions locales entre niveaux de gris voisins pour définir les différentes régions de l'image, va nous permettre d'utiliser un formalisme markovien dans de nombreux traitements, qu'il s'agisse de restauration, de segmentation ou plus tard d'analyse complète des images. Le principe est de définir des énergies locales entre groupes de sites reflétant les interactions entre niveaux de gris. L'énergie globale est alors reliée à la probabilité d'apparition de l'image dans le cadre des champs de Gibbs.

Dans ce chapitre, nous introduisons tout d'abord de façon intuitive la notion d'énergie locale avant de définir plus formellement un champ de Markov et d'énoncer le théorème d'équivalence entre champs de Markov et champs de Gibbs. Les algorithmes d'échantillonnage d'un champ de Markov (échantillonneur de Gibbs et algorithme de Métropolis) sont ensuite présentés, ainsi que les modèles markoviens les plus courants. L'utilisation des champs markoviens en traitement d'images dans un cadre bayésien montre la nécessité de pouvoir accéder aux configurations les plus probables d'un champ markovien et nous amène à la présentation du recuit simulé. Dans les parties suivantes, nous abordons les différents estimateurs (MAP, MPM, TPM), le problème de l'estimation des paramètres du champ, les processus de bords, avant de mentionner l'application de la modélisation markovienne à des graphes de primitives de plus haut niveau que les pixels.

7.1 Définition et simulation d'un champ de Markov

7.1.1 Description de l'image

L'image est formée d'un ensemble fini S de sites s_i correspondant aux pixels. S est donc essentiellement un réseau discret fini, partie de $\mathbb{Z}^d$, si on note d la dimension de l'espace (2 le plus classiquement, 3 pour les volumes, etc.). À chaque site est associé un descripteur, représentant l'état du site et qui peut être son niveau de gris, une étiquette, ou une information plus complexe, et prenant ses valeurs dans E.

La notion d'interactions locales nécessite de structurer les relations spatiales entre les différents sites du réseau. Pour ce faire, on munit S d'un système de voisinage $\mathcal{V}$ défini de la façon suivante :

$$\mathcal{V}_s = \{t\} \text{ tels que } \begin{cases} s \notin \mathcal{V}_s \\ t \in \mathcal{V}_s \Rightarrow s \in \mathcal{V}_t \end{cases}$$

À partir d'un système de voisinage, un système de cliques peut être déduit : une clique est soit un singleton de S, soit un ensemble de sites tous voisins les uns des autres. En fonction du système de voisinage utilisé, le système de cliques sera différent et fera intervenir plus ou moins de sites comme illustré sur la figure 7.1. On notera $\mathcal{C}$ l'ensemble des cliques relatif à $\mathcal{V}$, et $\mathcal{C}_k$ l'ensemble des cliques de cardinal k.

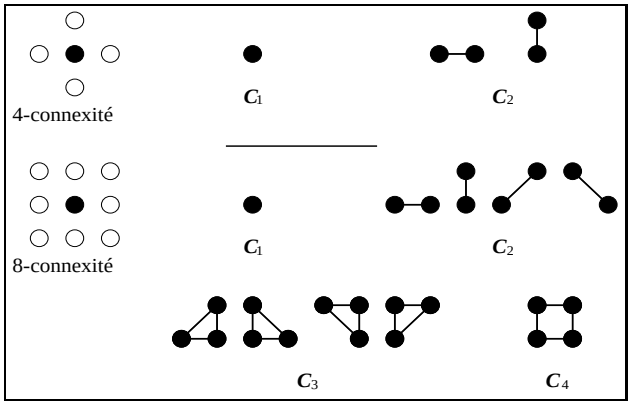


FIG. 7.1 – Les cliques associées à deux systèmes de voisinage en dimension 2.

Les interactions locales entre niveaux de gris (ou descripteurs) de sites voisins peuvent alors s'exprimer comme un potentiel de clique. Soit c une clique, on lui associe le potentiel U_c dont la valeur dépend des niveaux de gris (ou descripteurs) des pixels constituant la clique. En poursuivant ce raisonnement, on peut définir l'énergie globale de l'image comme la somme des potentiels de toutes les cliques :

$$U = \sum_{c \in \mathcal{C}} U_c$$

et l'énergie locale en un site comme la somme des potentiels de toutes les cliques auxquelles il appartient :

$$U_s = \sum_{c \in \mathcal{C} / s \in c} U_c$$

Nous avons jusqu'ici considéré le cas d'une image pour illustrer les notions de voisinage, de clique et de potentiel, mais le formalisme markovien se définit très généralement sur tout graphe. Soit un ensemble de sites S dénombrable (sommets du graphe), et une relation de voisinage, les cliques sont alors définies comme les sous-graphes complets du graphe. C'est l'utilisation de graphes plus généraux que ceux définis sur la grille de l'image qui permet des traitements de plus haut niveau.

7.1.2 Modélisation probabiliste de l'image

La définition des champs de Markov qui sera donnée dans la section suivante nécessite une modélisation probabiliste de l'image. Ainsi, l'image dont nous disposons va être considérée comme une réalisation d'un

champ aléatoire. Soit s un site de l'image, on peut en effet lui associer une variable aléatoire (v.a) X_s prenant ses valeurs dans E. Le niveau de gris x_s en s n'est ainsi qu'une réalisation[1] de la v.a X_s. On définit alors le champ aléatoire $X = (X_s, X_t, ..)$ prenant ses valeurs dans $\Omega = E^{|S|}$. On trouvera aussi le terme de processus aléatoire pour X ; en toute rigueur, " processus " devrait être réservé au cas d'un ensemble d'indexation continu, et champ au cas discret.

Dans ce cadre probabiliste, l'image considérée est simplement une réalisation x du champ. La probabilité globale de x, $P(X = x)$, permet d'accéder en quelque sorte à la vraisemblance de l'image, et les probabilités conditionnelles locales d'une valeur en un site permettent de mesurer le lien statistique entre un niveau de gris et le reste de l'image. L'hypothèse markovienne permet d'évaluer ces quantités.

Notons que nous nous plaçons dans le cas où E, l'espace de valeurs des descripteurs, est quantifié, ce qui nous permet de manipuler des probabilités. Dans le cas où cet espace est continu, il faut remplacer P par une densité de probabilité, mais dans ce cas le théorème que nous allons voir ci-dessous n'est plus valable.

7.1.3 Champs de Markov - Champs de Gibbs

Définition d'un champ de Markov

Considérons x_s la valeur du descripteur prise au site s et $x^s = (x_t)_{t \neq s}$ la configuration de l'image excepté le site s. La définition d'un champ de Markov est alors la suivante :

X est un champ de Markov ssi la probabilité conditionnelle locale en un site n'est fonction que de la configuration du voisinage du site considéré

ce qui s'exprime de façon formelle par :

$$P(X_s = x_s \ / \ x^s) = P(X_s = x_s \ / \ x_t, t \in \mathcal{V}_s)$$

Ainsi, le niveau de gris en un site ne dépend que des niveaux de gris des pixels voisins de ce site. Cette hypothèse markovienne se justifie bien dans le cas de la plupart des images naturelles constituées de zones homogènes ou texturées ainsi que pour une large gamme d'images de synthèse. Plus généralement, une connaissance locale de l'image suffit souvent à réaliser son interprétation partielle et donc cette hypothèse markovienne sera souvent justifiée sur des graphes plus globaux que le graphe des pixels.

Notons qu'en l'absence de contrainte sur le système de voisinage, tous les champs aléatoires peuvent être considérés comme markoviens à condition de prendre un voisinage suffisamment grand. L'intérêt de cette modélisation réside bien sûr dans le cas où la propriété markovienne est vérifiée pour des voisinages restreints permettant des calculs rapides.

Equivalence entre champs de Markov et champs de Gibbs

La modélisation markovienne prend toute sa puissance grâce au théorème que nous allons voir maintenant. En effet, celui-ci permettra d'accéder aux expressions des probabilités conditionnelles locales. Il nous faut au préalable définir un certain nombre de notions relatives aux mesures et champs de Gibbs.

Définition d'une mesure de Gibbs : La mesure de Gibbs de fonction d'énergie (ou d'hamiltonien) $U : \Omega \to \mathbb{R}$ est la probabilité P définie sur Ω par :

$$P(X = x) = \frac{1}{Z} \exp\left(-U(x)\right)$$

[1] On notera généralement en lettres majuscules les variables aléatoires et en minuscules leurs réalisations.

avec

$$U(x) = \sum_{c \in \mathcal{C}} U_c(x)$$

où $\mathcal{C}$ est le système de cliques associé au système de voisinage $\mathcal{V}$ de U[2].

$Z = \sum_{x \in \Omega} \exp(-U(x))$ est une constante de normalisation appelée fonction de partition de Gibbs. En pratique, il est quasi impossible de calculer cette constante à cause du très grand nombre de configurations possibles. Ne serait-ce que dans le cas d'une image binaire (Card $(E) = 2$) et de taille $= 512 \times 512$, on a $card(\Omega) = 2^{262144}$ configurations possibles !

La notation couramment utilisée pour $U(x)$ est abusive car $U_c(x)$ ne dépend pas de l'ensemble de la configuration x mais seulement de x restreinte à la clique c ($U_c(x) = U_c(x_t, t \in c)$).

Nous pouvons maintenant définir le champ de Gibbs de potentiel associé au système de voisinage $\mathcal{V}$: c'est le champ aléatoire X dont la probabilité est une mesure de Gibbs associée au système de voisinage $\mathcal{V}$, ce qui implique :

$$P(X = x) = \frac{1}{Z} \exp\left(-U(x)\right) = \frac{1}{Z} \exp\left(-\sum_{c \in \mathcal{C}} U_c(x)\right)$$

L'énergie globale d'un champ de Gibbs possède donc la propriété de se décomposer sous forme d'une somme d'énergies locales, qui comme on le verra par la suite permettront d'accéder aux probabilités conditionnelles locales. Notons ici que plus une configuration d'un champ de Gibbs a une énergie faible, plus elle est probable.

Le théorème de Hammersley-Clifford [Besag, 1974] établit alors le résultat fondamental suivant sous les hypothèses :

- S fini ou dénombrable,
- le système de voisinage $\mathcal{V}$ est borné,
- l'espace des états E est discret

X est un champ de Markov relativement à $\mathcal{V}$ et $P(X = x) > 0 \quad \forall x \in \Omega$ si et seulement si X est un champ de Gibbs de potentiel associé à $\mathcal{V}$.

Par exemple, si nous considérons un champ de Markov de voisinage 4-connexe, nous pouvons écrire l'énergie de la configuration x sous la forme :

$$U(x) = \sum_{c=(s) \in \mathcal{C}_1} U_c(x_s) + \sum_{c=(s,t) \in \mathcal{C}_2} U_c(x_s, x_t)$$

Notons que rien n'interdit la non-stationnarité du champ, c'est-à-dire la variation des potentiels U_c en fonction de la localisation de la clique c dans l'image [Descombes, 1993]. D'autre part, rien n'impose la symétrie des potentiels et on peut avoir $U_{c=(r,s)}(0,1) \neq U_c(1,0)$.

Le théorème de Hammersley-Clifford, et la forme bien spécifique de probabilité de X qui en résulte, va permettre de lier les probabilités globales et locales comme nous allons le voir maintenant. En effet si nous cherchons à écrire la probabilité conditionnelle locale $P(x_s \ / \ X^s = x^s)$, nous avons grâce au résultat précédent :

$$P(X_s = x_s \ / \ X^s = x^s) = \frac{P(X = x)}{P(X^s = x^s)} = \frac{\exp(-U(x_s, x^s))}{\sum_{\xi \in E} \exp(-U(\xi, x^s))}$$

Définissons l'énergie locale U_s par :

$$U_s(x_s \ / \ x_t, t \in \mathcal{V}_s) = \sum_{c \in \mathcal{C} \ / \ s \in c} U_c(x_s, x_t, t \in \mathcal{V}_s) = \sum_{c \in \mathcal{C} \ / \ s \in c} U_c(x_s, V_s)$$

[2] Il est toujours possible de trouver un système de voisinage $\mathcal{V}$ permettant de décomposer U ; le cas extrême correspondant à des sites tous voisins les uns des autres.

en notant $V_s = (x_t, t \in \mathcal{V}_s)$. Cette énergie locale ne fait donc intervenir que les voisins de s. On peut alors écrire l'énergie globale $U(x)$ sous la forme :

$$U(x) = \sum_{c \in \mathcal{C} \ / \ s \notin c} U_c(x) + \sum_{c \in \mathcal{C} \ / \ s \in c} U_c(x) = \sum_{c \in \mathcal{C} \ / \ s \notin c} U_c(x) + U_s(x_s \ / \ V_s)$$

En simplifiant l'expression de la probabilité conditionnelle locale en supprimant les termes communs qui font intervenir les cliques ne contenant pas le site s au numérateur et au dénominateur, on a :

$$\begin{aligned} P(X_s = x_s \mid X^s = x^s) &= \frac{\exp(-\sum_{c \in \mathcal{C} \ / \ s \notin c} U_c(x) - U_s(x_s \ / \ V_s))}{\sum_{x_s \in E} \exp(-\sum_{c \in \mathcal{C} \ / \ s \notin c} U_c(x) - U_s(x_s \ / \ V_s))} \\ &= \frac{\exp(-U_s(x_s \ / \ V_s)}{\sum_{x_s \in E} \exp(-U_s(x_s \ / \ V_s))} \end{aligned} \tag{7.1}$$

L'expression obtenue, qui ne fait intervenir que les potentiels des cliques contenant le site s (ce qui nous permet de retrouver au passage l'hypothèse markovienne), est très importante. En effet, autant il n'est pas possible partant d'une configuration x d'accéder à sa probabilité à cause de la constante de normalisation, autant il est possible de calculer en chaque site la probabilité conditionnelle locale. Cette expression sera à la base de tous les algorithmes de simulation de champs markoviens que nous verrons dans la section suivante.

Remarque : Le théorème de Hammersley-Clifford n'est valable que lorsqu'aucune configuration n'est interdite (condition $P(X = x) > 0 \quad \forall x$). Des solutions ont été proposées, lorsqu'il est souhaitable de supprimer certaines configurations irréalistes (par exemple avoir une route à l'intérieur d'une zone de mer, ou de l'os dans la matière blanche du cerveau).

7.1.4 Echantillonnage de MRF

Si nous résumons les résultats précédents, la définition d'un champ de Markov passe par la définition de sa fonction d'énergie U. Celle-ci nécessite la définition d'un système de voisinage, qui définit alors le système de cliques, et de fonctions de potentiel associées aux cliques. Ces fonctions de potentiel permettent d'accéder à la probabilité globale d'une configuration, et aux probabilités conditionnelles locales.

Le problème qui se pose alors est, étant défini un champ de Markov, comment pouvons-nous réaliser le tirage d'une configuration (une image ici) en suivant la loi de probabilité de Gibbs caractéristique de ce champ ? Deux algorithmes ont été proposés pour synthétiser des réalisations d'un champ de Markov qui sont :

- l'échantillonneur de Gibbs,
- l'algorithme de Métropolis

que nous allons décrire maintenant.

L'échantillonneur de Gibbs

Cet algorithme, proposé par Geman et Geman [Geman et Geman, 1984a], repose sur la construction itérative d'une suite d'images. A la convergence, i.e après un nombre d'itérations suffisant, les images construites sont des réalisations tirées selon la loi de Gibbs globale.

La méthode de construction de l'image à l'itération n, partant de l'image à l'itération $n-1$ se fait par mises à jour successives des sites de l'image. À l'étape n :

- choix d'un site s ;
- au site s, selon la configuration des voisins V_s pour l'image $x^{(n-1)}$, calcul de la probabilité conditionnelle locale :

$$P(X_s = x_s \mid V_s) = \frac{\exp(-U_s(x_s \mid V_s))}{\sum_{\xi \in E} (\exp(-U_s(\xi \mid V_s))}$$

- mise à jour du site s par tirage aléatoire selon la loi $P(X_s = x_s \mid V_s)$.

On considère que l'algorithme a convergé après un grand nombre d'itérations ou lorsque le nombre de changements est faible. Le choix du site s considéré à l'étape n peut se faire de n'importe quelle façon à condition de balayer tous les sites un très grand nombre de fois (théoriquement un nombre infini de fois). Les méthodes usuelles consistent à tirer un site selon une loi uniforme, ou effectuer un balayage classique, ligne par ligne, de l'image.

Cet algorithme construit en réalité une suite d'images $x^{(n)}$ qui sont les observations d'une suite $X^{(n)}$ de champs aléatoires constituant une chaîne de Markov pour un certain noyau de transition. On peut montrer le théorème suivant, lorsque la séquence balaye chaque site une infinité de fois :

$$\forall x^{(0)} \quad \forall x \in \Omega \quad \lim_{n \to \infty} P(X^{(n)} = x \mid X^{(0)} = x^{(0)}) \; = \; P(x)$$

où P est la mesure de Gibbs associée au champ de Markov considéré. Ainsi, après un grand nombre d'itérations, les images $x^{(n)}$ générées sont des réalisations de la loi globale $P(x)$, et cela indépendemment de la configuration initiale $x^{(0)}$.

La preuve de ce théorème, et donc de la convergence de l'algorithme, est donnée dans [Winkler, 1995].

On parle de l'échantillonneur de Gibbs comme d'un algorithme de relaxation, car il procède par mises à jour successives des sites, et probabiliste car celle-ci est fondée sur un tirage aléatoire.

L'algorithme de Metropolis

L'échantillonneur de Gibbs est un algorithme très utilisé en traitement d'images pour la synthèse de champs de Markov. Néanmoins, un algorithme antérieur et issu de la physique statistique avait été mis au point dans les années 50 par Metropolis [Metropolis et al., 1953].

Cet algorithme repose sur un principe similaire à l'échantillonneur de Gibbs, et il s'agit également d'un algorithme de relaxation probabiliste. Le principe est là encore de construire une suite d'images qui seront des tirages selon la loi du champ de Markov après un nombre suffisamment grand d'itérations. Mais la mise à jour en un site s'effectue de façon différente. Ainsi à l'étape n :

- choix d'un site s
- tirage aléatoire d'un descripteur λ dans E selon une loi uniforme ;
- calcul de la variation d'énergie pour le passage du label du site s de $x_s^{(n-1)}$ à λ :

$$\Delta U = U_s(\lambda \mid V_s^{(n-1)}) - U_s(x_s^{(n-1)} \mid V_s^{(n-1)})$$

- deux cas sont alors possibles :
  1. $\Delta U < 0$, le changement est accepté : $x_s^{(n)} = \lambda$;
  2. $\Delta U \geq 0$, le changement est accepté ou refusé par tirage selon les probabilités $p = \exp(-\Delta U)$ et $1 - p$.

Le système de balayage des sites et le critère d'arrêt sont similaires à ceux de l'échantillonneur de Gibbs. La différence avec l'échantillonneur de Gibbs réside dans le tirage au sort du nouveau niveau de gris (ou descripteur), au lieu de considérer la loi définie par tous les descripteurs. Comme on ne considère que la variation énergétique entre les 2 configurations, l'algorithme de Metropolis est plus rapide à chaque étape que l'échantillonneur de Gibbs, qui lui nécessite le calcul de la fonction de partition locale. Mais la convergence peut être plus lente car le taux d'acceptation est strictement inférieur à 1 (les transitions ne sont pas toujours acceptées, contrairement au cas de l'échantillonneur de Gibbs).

Là encore, le principe est de construire une chaîne de Markov selon un certain noyau de transition (différent de celui intervenant dans l'échantillonneur de Gibbs). Le théorème précédent est alors encore vérifié pour l'algorithme de Metropolis.

7.1.5 Le recuit simulé

Nous avons vu dans les paragraphes précédents comment échantillonner selon la loi de probabilité de Gibbs associée au champ de Markov. À chaque application des précédents algorithmes, une nouvelle réalisation est obtenue. Il peut être utile également de pouvoir calculer la ou les configurations les plus probables qui correspondent aux états d'énergie minimale. C'est l'algorithme du recuit simulé qui permet de trouver ces réalisations.

Avant de présenter cet algorithme, nous avons besoin de quelques résultats sur les distributions de Gibbs avec paramètre de température que nous présentons maintenant.

Distribution de Gibbs avec température

Une distribution de Gibbs avec paramètre de température est une probabilité qui s'écrit :

$$P_T(X = x) = \frac{1}{Z(T)} \exp(-\frac{U(x)}{T})$$

avec $Z(T) = \sum_x \exp(-\frac{U(x)}{T})$ et $T > 0$. Le terme de température provient de l'analogie avec la physique statistique.

Il est intéressant d'étudier le comportement de cette distribution pour des valeurs extrêmes du paramètre de température.

• $T \to \infty$:

On a $\exp(-\frac{U(x)}{T}) \to 1$ et comme $\sum_x P_T(X = x) = 1$, on obtient

$$P_T(X = x) \to \frac{1}{\text{Card } \Omega}$$

Donc P_T converge vers la probabilité uniforme sur Ω, i.e pour une température infinie tous les états sont équiprobables.

• $T \to 0$:

Notons U^* l'énergie minimale et Ω^* l'ensemble des configurations atteignant l'énergie minimale $\Omega^* =$

$\{x_1, ..., x_k\}$ ($x_1, ...x_k$ sont les minima globaux de l'énergie). On peut écrire :

$$P_T(X = x) = \frac{\exp(-\frac{U(x)}{T})}{\sum_y \exp(-\frac{U(y)}{T})} = \frac{\exp(-\frac{U(x)-U^*}{T})}{\sum_y \exp(-\frac{U(y)-U^*}{T})}$$

$$= \frac{\exp(-\frac{U(x)-U^*}{T})}{\sum_{y \notin \Omega^*} \exp(-\frac{U(y)-U^*}{T}) + \sum_{y \in \Omega^*} 1}$$

$\diamond$ Si $x \notin \Omega^*$, on a $U(x) - U^* > 0$ et $\exp(\frac{(U(x)-U^*)}{T}) \to 0$ pour $T \to 0$. Donc $P_T(x) \to 0$ si x n'est pas un minimum global de l'énergie.

$\diamond$ Si $x \in \Omega^*$, on a : $P_T(x_1) = P_T(x_2) = ... = P_T(x_k) = \frac{1}{k}$ (il y a une somme finie de termes qui tendent vers 0 au dénominateur).

Ce qui signifie que lorsque la température est nulle P_T est uniformément distribuée sur les minima globaux de l'énergie, i.e sur les configurations les plus probables. C'est ce résultat qui est à la base de l'algorithme de recuit simulé.

Algorithme du recuit simulé

Cet algorithme est dédié à la recherche d'une configuration d'énergie minimale d'un champ de Gibbs (on ne cherche plus ici à échantillonnner contrairement à précédemment). L'idée d'intégrer un paramètre de température et de simuler un recuit a été initialement proposée par Kirkpatrick [Kirkpatrick et al., 1982] et reprise par Geman et Geman [Geman et Geman, 1984a] qui ont proposé l'algorithme suivant.

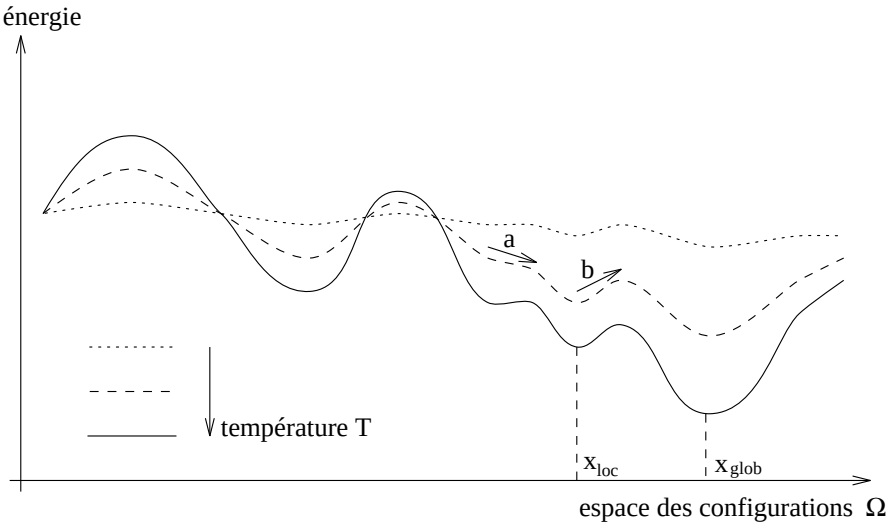
Comme les algorithmes de simulation, c'est un algorithme itératif qui construit la solution au fur et à mesure. Le déroulement de l'algorithme est le suivant (en notant n le numéro de l'itération) :

- choix d'une température initiale $T^{(0)}$ suffisamment élevée
- choix d'une configuration initiale quelconque $x^{(0)}$
- à l'étape n
  - simulation d'une configuration $x^{(n)}$ pour la loi de Gibbs d'énergie $\frac{U(x)}{T^{(n)}}$ à partir de la configuration $x^{(n-1)}$; la simulation peut se faire par l'échantillonneur de Gibbs ou l'algorithme de Métropolis ; on réalise en général un balayage complet de l'image à la température $T^{(n)}$;
  - diminution lente de la température : $T^{(n)} > \frac{c}{\log(1+n)}$
- arrêt lorsque le taux de changement est faible.

La décroissance logarithmique de la température est un rythme très lent ; en pratique des décroissances géométriques sont utilisées, souvent sans dégradation notable des résultats obtenus. La constante c intervenant dans la décroissance dépend de la variation énergétique globale maximale sur l'espace des configurations.

La figure 7.1.5 montre l'évolution du paysage énergétique représenté en 1 dimension au fur et à mesure de la décroissance en température. Au départ, toutes les configurations sont équiprobables puis les minima énergétiques apparaissent et s'accentuent.

Notons que contrairement aux algorithmes de l'échantillonneur de Gibbs et de Métropolis qui échantillonnent selon la loi de Gibbs et qui sont en mesure de donner toutes les configurations possibles, les images obtenues par recuit simulé sont uniques et doivent en théorie correspondre aux minima globaux de l'énergie.



a : $\Delta U < 0$, transition acceptée
b : $\Delta U > 0$, transition selon une probabilité $p = \exp(-\Delta U/T)$

Il existe une preuve de convergence de cet algorithme, qui repose à nouveau sur la construction d'une chaîne de Markov, mais qui est hétéroggène cette fois-ci à cause de la variation du paramètre de température [Geman et Geman, 1984a]. Intuitivement, le recuit simulé permet d'atteindre un optimum global, car il accepte des remontées en énergie. Avec la décroissance de la température, ces sauts énergétiques sont progressivement supprimés au fur et à mesure qu'on se rapproche de l'optimum global. La descente en température doit donc se faire suffisamment lentement pour que l'algorithme ne reste pas piégé dans un minimum local de l'énergie.

Algorithme des modes conditionnels itérés (ICM)

Malheureusement, l'algorithme du recuit simulé est très lourd en temps de calcul puisqu'il demande la génération d'un grand nombre de configurations au fur et à mesure que la température décroît. Des algorithmes sous-optimaux sont donc souvent utilisés en pratique. Besag [Besag, 1986] a ainsi proposé un autre algorithme, beaucoup plus rapide, mais pour lequel nous n'avons pas de preuve de convergence vers un minimum global. Il s'agit de l'ICM, *Iterated Conditional Mode*, que nous allons présenter ici.

Cet algorithme est un algorithme itératif modifiant à chaque étape les valeurs x_s de l'ensemble des sites de l'image. Mais à la différence de ces algorithmes qui étaient stochastiques par essence, la modification d'une valeur se fait ici de façon déterministe.

On construit donc, partant d'une configuration initiale $x(0)$, une suite d'images $x(n)$, convergeant vers une approximation du MAP $\hat{x}$ recherché. Soit un tour la visite de tous les sites de l'image, on parlera dans la suite d'itérations à chaque mise à jour d'un site et d'étape à chaque mise à jour de toute l'image (i.e accomplissement d'un tour).

Le déroulement de l'étape n s'effectue de la façon suivante : on parcourt tous les sites et en chaque site, on effectue les deux opérations suivantes :

1. calcul des probabilités conditionnelles locales, pour toutes les valeurs possibles de λ dans E du site :

$$P(X_s = \lambda \ / \ \hat{x_r}(k), r \in \mathcal{V}_s)$$

(en pratique, calcul plus simplement des énergies conditionnelles locales)

2. mise à jour de la valeur par le λ maximisant la probabilité conditionnelle locale :

$$\hat{x_s}(k+1) = \text{Argmax}_\lambda P(X_s = \lambda \ / \ \hat{x_r}(k), r \in \mathcal{V}_s)$$

(ou de façon équivalente, minimisant l'énergie conditionnelle locale).

Le processus s'arrête lorsque le nombre de changements d'une étape à l'autre devient suffisamment faible.

On peut montrer que l'énergie globale de la configuration $\hat{x}$ diminue à chaque itération. Cet algorithme, contrairement au recuit simulé, est très rapide (une dizaine de balayages permettent d'arriver à convergence) et peu coûteux en temps de calcul puisqu'il ne nécessite que le calcul des énergies conditionnelles locales. En contrepartie, ses performances dépendent très fortement de l'initialisation (par rapport à la forme du paysage énergétique) puisqu'il converge vers un minimum local. L'ICM s'apparente à une descente en gradient (on fait baisser l'énergie à chaque itération) ou à un recuit simulé gelé à température nulle, et peut donc rester bloqué dans le minimum énergétique local le plus proche de l'initialisation. Le recuit simulé, au contraire, grâce au paramètre de température et aux remontées en énergie qu'il autorise permet d'accéder au minimum global.

Notons qu'il a également été proposé d'utiliser la programmation dynamique pour estimer le MAP [Derin et Elliott, 1987a]. Mais il est alors nécessaire d'être dans une configuration simple de segmentation (peu d'étiquettes, dimensions raisonnables) et seule une approximation peut être obtenue.

7.1.6 Quelques MRF fondamentaux

Nous présentons ici quelques uns des champs de Markov les plus utilisés. Comme indiqué précédemment, ces champs sont définis par leur voisinage et leurs fonctions de potentiel. Ils sont illustrés par le tirage de réalisations selon l'échantillonneur de Gibbs.

◇ Modèle d'Ising :

Ce modèle est le plus ancien (1925 [Ising, 1925]) et a été développé lors de l'étude du ferro-magnétisme en physique statistique. L'espace des descripteurs est celui des états des spins, i.e $E = \{-1, 1\}$ (espace binaire), et le voisinage est constitué par les 4 ou 8 plus proches voisins dans un espace bidimensionnel. Les potentiels sont des potentiels en tout ou rien :

$$\begin{aligned} U_{c=(s,t)}(x_s, x_t) &= -\beta \text{ si } x_s = x_t \\ &= +\beta \text{ si } x_s \neq x_t \end{aligned}$$

Ce qui s'écrit également $U_{c=(s,t)}(x_s, x_t) = -\beta x_s x_t$.

Les potentiels des cliques d'ordre 1 (clique constituée par un seul spin) sont de la forme $-Bx_s$. L'énergie totale s'écrit :

$$U(x) = -\sum_{c=(s,t)\in\mathcal{C}} \beta x_s x_t - \sum_{s\in S} B x_s$$

β est la constante de couplage entre sites voisins et B représente un champ magnétique externe. Lorsque β est positif, les configurations les plus probables (i.e d'énergies plus faibles) sont celles pour lesquelles les spins sont de même signe (ferro-magnétisme), alors que dans le cas de β négatif, au contraire, on favorisera l'alternance de spins de signes opposés (anti-ferromagnétisme). La valeur (signe et valeur absolue) de β conditionne donc la régularité du modèle d'Ising. Quant au champ magnétique externe relatif au potentiel d'ordre 1, il favorise a priori par son signe un spin ou un autre.

La figure 7.2 montre des réalisations de modèles d'Ising pour différents paramètres (la régularisation appelée critique correspond à l'apparition des zones homogènes).

◇ Modèle de Potts :

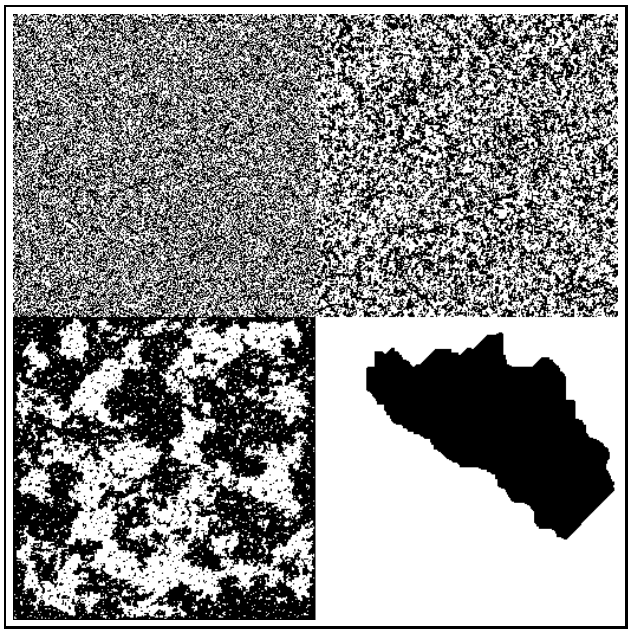


FIG. 7.2 – Le modèle d'Ising plan et 4-connexe pour différentes valeurs de paramètres - Simulations en raccordement torique; a. Image aléatoire : $\beta = 0$, b.régularisation faible, $\beta = 0,2$; c. régularisation critique : $\beta \approx 0,44$; d. régularisation forte : $\beta = 4$.

Il s'agit d'une extension du modèle d'Ising [Wu, 1982] pour un espace m-aire, i.e. $E = \{0, m-1\}$. Il peut s'agir de plusieurs niveaux de gris, mais plus souvent pour ce modèle, d'étiquettes (labels) pouvant représenter une classification de l'image (par exemple les classes *eau, forêt, champ, ville*). Le voisinage considéré est 4- ou 8-connexe et les potentiels sont comme précédemment en tout ou rien mais définis seulement pour les cliques d'ordre 2 :

$$\begin{aligned} U_{c=(s,t)}(x_s, x_t) &= -\beta \text{ si } x_s = x_t \\ &= +\beta \text{ si } x_s \neq x_t \end{aligned}$$

Lorsque β est positif, les configurations les plus probables correspondent à des sites voisins de même niveau de gris ou descripteur, ce qui donne des réalisations constituées par des larges zones homogènes. La taille de ces regions est gouvernée par la valeur de β. Des exemples de réalisations pour différentes valeurs de β sont montrés figure 7.3.

Il est possible de définir des modèles utilisant des pondérations β différentes en fonction des directions des cliques, et de privilégier ainsi certaines directions.

Ce modèle permet également de prendre en compte différentes relations entre les régions (i.e. entre différentes valeurs des descripteurs). On peut par exemple définir des pondérations $\beta(e_s, e_t)$ pour $e_s, e_t \in E$. Dans notre exemple de classification en 4 étiquettes *eau, forêt, champ, ville*, une configuration de sites avec les étiquettes *champ / forêt* peut être supposée plus probable qu'une configuration *ville / forêt*, d'où des valeurs $\beta(champ, forêt)$ et $\beta(ville, forêt)$ différentes [Sigelle, 1993].

◇ Modèle markovien gaussien :

Ce modèle est réservé aux images en niveaux de gris $E = \{0, ..., 255\}$ et ne convient pas bien aux

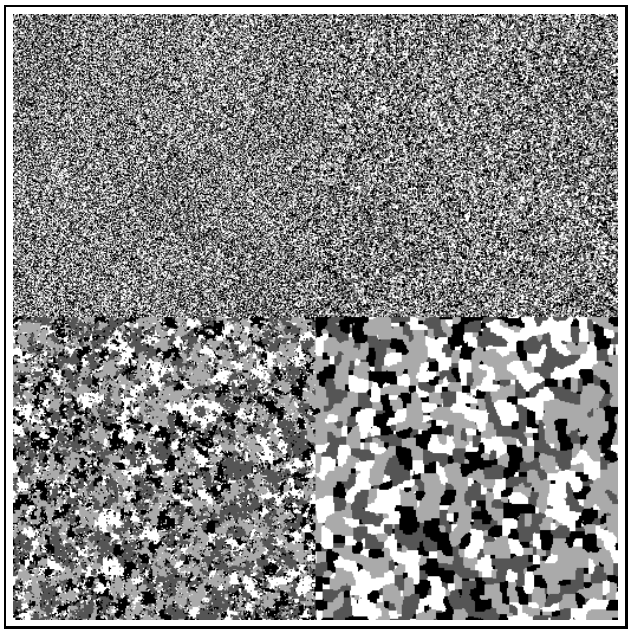


FIG. 7.3 – Le Modèle de Potts 2D et 4-connexe pour différentes valeurs de paramètres (m = 4) - Simulations en raccordement torique ; a. Image aléatoire : $\beta = 0$, b.régularisation faible, $\beta = 0.2$; c. régularisation critique : $\beta \approx 1,099$; d. régularisation forte : $\beta = 4$.

images d'étiquettes. Le voisinage est 4 ou 8-connexe et l'énergie est de la forme :

$$U(x) = \beta \sum_{c=(s,t)} (x_s - x_t)^2 + \alpha \sum_{s \in S} (x_s - \mu_s)^2$$

Le premier terme correspondant aux cliques d'ordre 2 est un terme de régularisation, qui favorise les faibles différences de niveaux de gris entre sites voisins pour $\beta > 0$. Le second terme peut correspondre à un terme d'attache aux données dans le cas où on possède une image de données extérieures. Le rapport $\frac{\alpha}{\beta}$ pondère les influences respectives de l'attache aux données et de la régularisation, et les valeurs absolues des paramètres caractérisent le caractère plus ou moins piqué ou équiréparti au contraire de la distribution.

7.1.7 Applications : restauration et segmentation

Nous avons étudié dans la section précédente des algorithmes de simulation permettant de générer des réalisations d'un champ de Markov donné. Nous abordons ici leur utilisation en tant qu'outils de traitements d'images pour deux grandes applications de bas niveau : la restauration d'images et la segmentation.

Cadre bayésien

Pour ces deux applications, on peut modéliser le problème dans un cadre bayésien de la façon suivante. Nous disposons d'une certaine donnée (image) que nous noterons y et que nous pouvons considérer comme une réalisation d'un champ aléatoire Y. Nous cherchons une réalisation x de l'image restaurée ou segmentée, que nous pouvons modéliser comme un champ de Markov X. X est le champ des étiquettes

(labels) dans le cas de la segmentation, le champ des intensités dans le cas de la restauration. Les espaces de configurations ne sont donc pas nécessairement les mêmes pour X et Y. Ces deux champs sont liés par le processus d'acquisition de l'image, qui conduit du champ idéal X, le processus image originel que nous cherchons, au champ bruité Y que nous observons. La restauration ou la segmentation ont pour objectif d'inverser le processus et donc de remonter à une réalisation de X à partir de l'observation des données bruitées y. On parle dans ce contexte de champ de Markov caché pour X, ou de données incomplètes puisque y n'est pas une réalisation de X.

On peut par exemple utiliser le critère du maximum a posteriori et rechercher la configuration $\hat{x}$ maximisant la probabilité de X conditionnellement à la donnée y i.e $P(X = x \ / \ Y = y)$. Or la règle de Bayes permet d'écrire :

$$P(X = x \ / \ Y = y) = \frac{P(Y = y \ / \ X = x)P(X = x)}{P(Y = y)}$$

Expression dans laquelle il s'agit alors d'analyser chacun des termes $P(Y = y \ / \ X = x)$ et $P(X = x)$, sachant que $P(Y)$ est une constante (indépendante de la réalisation x). Le premier terme $P(Y = y|X = x)$ décrit justement le processus d'observation et d'acquisition des données. L'hypothèse la plus courante (dont la validité reste à justifier) consiste à supposer l'indépendance conditionnelle des pixels (bruit non corrélé par exemple) :

$$P(Y = y \ / \ X = x) = \prod_s P(Y_s = y_s \ / \ X_s = x_s)$$

Cette écriture n'est plus valable lorsqu'il y a une convolution par la fonction de transfert du système d'acquisition, mais on peut montrer que le champ a posteriori reste markovien.

Par ailleurs, on fait sur le champ X recherché une hypothèse markovienne selon un voisinage $\mathcal{V}$ et un modèle donné dépendant de l'application. On peut alors écrire :

$$P(X = x) = \frac{\exp(-U(x))}{Z}$$

Si on revient maintenant à la distribution a posteriori, celle-ci s'exprime par :

$$\begin{aligned} P(X = x \ / \ Y = y) &\propto P(Y \ / \ X)P(X) \propto e^{\ln P(Y \ / \ X) - U(x)} \\ &\propto e^{-\mathcal{U}(x \ / \ y)} \end{aligned}$$

avec :

$$\mathcal{U}(x \ / \ y) = \sum_{s \in S} - \ln p(y_s \ / \ x_s) + \sum_{c \in \mathcal{C}} U_c(x) \tag{7.2}$$

Par conséquent, sous les hypothèses précédentes, on constate que la distribution a posteriori est une distribution de Gibbs et que donc le champ X conditionnellement à y est également un champ de Markov (théorème de Hammersley-Clifford). Ainsi, il est possible de simuler des réalisations de ce champ à l'aide de l'échantillonneur de Gibbs ou de l'algorithme de Metropolis. Mais la configuration x qui nous intéresse est celle maximisant la probabilité a posteriori, donc la réalisation la plus probable du champ de Gibbs, ou encore celle qui minimise l'énergie $\mathcal{U}(x \ / \ y)$. L'algorithme du recuit simulé décrit plus haut permet d'atteindre ce (ou ces) état(s) d'énergie minimale.

Cas de la restauration

Reprenons la démarche précédente et exprimons plus en détails l'énergie $\mathcal{U}(x \ / \ y)$ dans un cas particulier de restauration.

Dans le cas où le processus d'acquisition entraîne une dégradation de l'image sous forme d'un bruit blanc gaussien de variance σ^2, on a la probabilité conditionnelle suivante :

$$p(y_s \ / \ x_s) = \frac{1}{\sqrt{2\pi}\sigma} \exp -\frac{(x_s - y_s)^2}{2\sigma^2}$$

La probabilité a priori $P(X = x)$ permet d'introduire les contraintes que nous souhaitons imposer à la solution (i.e. que nous supposons pour le processus originel). En faisant l'hypothèse que X est markovien nous nous restreignons à des contraintes locales, le plus souvent de régularité entre sites voisins. On choisit fréquemment un modèle avec des potentiels d'ordre 2 :

$$P(X = x) = \frac{1}{Z} \exp(-\beta \sum_{(s,t)\in\mathcal{C}_2} \phi(x_s - x_t))$$

On a alors l'énergie suivante correspondant à la distribution de Gibbs du champ a posteriori :

$$\mathcal{U}(x \ / \ y) = \sum_{s\in S} \frac{(x_s - y_s)^2}{2\sigma^2} + \beta \sum_{(s,t)\in\mathcal{C}_2} \phi(x_s - x_t) \tag{7.3}$$

Le champ X conditionnellement à y est donc un champ de Gibbs pour le même système de voisinage que X. La constante β pondère l'influence entre le terme d'attache aux données (cliques d'ordre 1) qui impose des niveaux de gris x_s de l'image restaurée proches de ceux y_s de la donnée bruitée, et le terme de régularisation (cliques d'ordre 2) qui impose une solution constituée de zones homogènes. Le modèle pour X peut être soit markovien gaussien, soit plus adapté à la restauration des contours avec une fonction ϕ appropriée. En effet, le modèle gaussien qui correspond à un fonction ϕ quadratique favorise des niveaux de gris proches pour des pixels voisins dans tous les cas. Or si on considère une image naturelle cet aspect est néfaste à proximité des contours car il favorisera la présence d'un dégradé. Aussi, de nombreuses fonctions ϕ ont été proposées pour modéliser les potentiels des cliques d'ordre 2 : $U_{c=(s,t)} = \phi(x_s - x_t)$. L'idée est de supprimer la pénalisation lorsque la variation de niveaux de gris est supérieure à une certaine valeur considérée comme représentant un contour. La partie 7.4 détaille ces aspects.

Cas de la segmentation

Dans ce contexte, le champ markovien X est défini sur un autre espace de configurations que Y car seulement quelques étiquettes sont considérées : $E = \{1, ..., m-1\}$ (correspondant aux différentes classes cherchées). Dans ce cas le processus de passage de X (champ des labels) à Y ne décrit pas tant le processus d'acquisition que l'apparence des classes dans l'image. Le terme $P(Y = y \ / \ X = x)$ traduit donc la probabilité de réalisation d'une configuration donnée connaissant son étiquetage (i.e. connaissant la classe de chaque pixel). En supposant l'indépendance des sites les uns par rapport aux autres, et en supposant que le niveau de gris y_s en un site s ne dépend que de l'étiquette x_s en ce site, on a :

$$P(Y = y \ / \ X = x) = \prod_s P(y_s \ / \ x_s)$$

Les valeurs des probabilités conditionnelles sont données par l'histogramme conditionnel des niveaux de gris pour une classe donnée. Par exemple, si on suppose que chaque classe i a une distribution gaussienne de moyenne μ_i et d'écart-type σ_i, on a :

$$P(y_s \ / \ x_s = i) = \frac{1}{\sqrt{2\pi}\sigma_i} \exp(-\frac{(y_s - \mu_i)^2}{2\sigma_i^2})$$

Si comme précédemment on fait une hypothèse markovienne sur X et qu'on se limite aux cliques d'ordre 2, on a :

$$P(X = x) = \frac{1}{Z} \exp(-\beta \sum_{(s,t)\in\mathcal{C}_2} \phi(x_s - x_t))$$

D'où l'énergie a posteriori :

$$\mathcal{U}(x \ / \ y) = \sum_s \frac{(y_s - \mu_{x_s})^2}{2\sigma_{x_s}^2} + \log\sqrt{2\pi}\sigma_{x_s} + \beta \sum_{(s,t)\in\mathcal{C}_2} \phi(x_s, x_t) \tag{7.4}$$

Le champ des étiquettes conditionnellement à y est markovien et d'énergie de Gibbs $\mathcal{U}(x \ / \ y)$. Là encore, comme pour la restauration, le terme d'ordre 1 exprime le respect des données (le niveau de gris doit correspondre à la classe), et le terme d'ordre 2 la contrainte de régularisation introduite. On choisit souvent un modèle de Potts pour X, ce qui donne une image segmentée avec de larges zones homogènes.

La figure 7.4 montre un exemple de segmentation d'une image satellitaire obtenue par le radar à ouverture synthétique ERS-1. L'utilisation du modèle de Potts pour le terme d'attache aux données donne des régions compactes.

Dans les deux applications précédentes il est nécessaire de pouvoir déterminer le ou les états d'énergie minimale qui correspondent au maximum de la probabilité d'un champ markovien. L'algorithme du recuit simulé présenté permet de trouver ces configurations. Nous reviendrons sur ce point en présentant d'autres estimateurs de la solution dans la section suivante.

7.2 Estimateurs dans un cadre markovien

7.2.1 Introduction

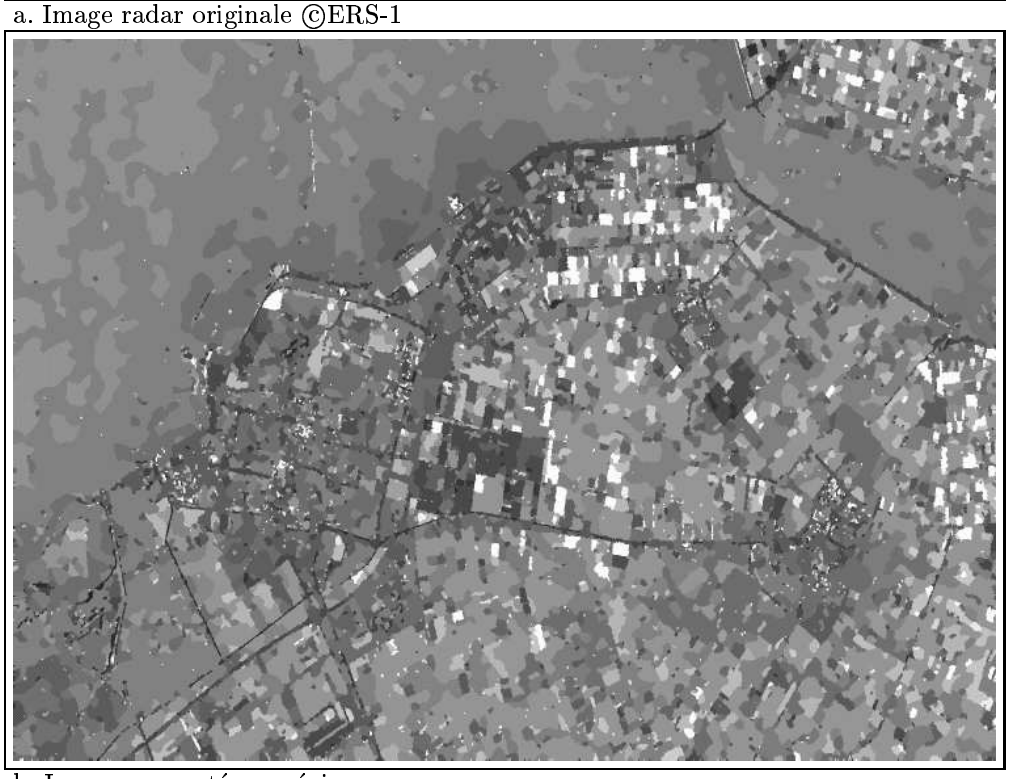
Nous avons vu précédemment comment il était possible d'utiliser le formalisme markovien à des fins de restauration et de segmentation. On se situe alors dans le cadre de données incomplètes (on parle aussi de champs de Markov cachés) car la réalisation dont on dispose est une réalisation bruitée (ou plus généralement vue à travers le système d'acquisition) du champ de Markov originel. En notant Y le champ dont on observe une réalisation, et X le champ initial, l'objectif est alors d'obtenir la meilleure réalisation $\hat{x}$ de X connaissant l'observation y, autrement dit, reconstruire x de manière optimale vis-à-vis d'un certain critère. Dans le précédent paragraphe, nous nous étions intéressés à la réalisation maximisant la probabilité a posteriori $P(X = x \ / \ Y = y)$, et nous avions vu un algorithme permettant d'obtenir cette réalisation : le recuit simulé. En réalité d'autres choix sont possibles, auxquels correspondent d'autres méthodes de résolution, et que nous allons aborder dans ce chapitre.

7.2.2 Modélisation bayésienne et fonction de coût

Si nous reprenons rapidement le raisonnement effectué précédemment, on peut écrire, en appliquant la règle de Bayes :

$$P(X = x \ / \ Y = y) = \frac{P(Y = y \ / \ X = x)P(X = x)}{P(Y = y)}$$

a. Image radar originale ©ERS-1



b. Image segmentée en régions

FIG. 7.4 – Exemple de segmentation markovienne sur une image ERS-1 du Flevoland.

On montre alors que sous certaines hypothèses (indépendance des sites dans la probabilité conditionnelle $P(Y \ / \ X)$ et hypothèse markovienne pour le champ X), la distribution a posteriori est une distribution de Gibbs et donc que le champ X conditionnellement à la donnée y est markovien. Cette propriété n'est pas nécessaire pour les notions suivantes, mais elle sera primordiale pour les algorithmes de résolution.

Le problème est alors de déterminer une estimation $\hat{x}$ optimisant un certain critère, où $\hat{x}$ est une fonction déterministe ϕ de la donnée y :

$$\hat{x} = \phi(y) \text{ avec } \phi : \Omega \to \Omega$$

L'estimation bayésienne procède alors comme suit. On se donne une fonction de coût, L définie de $\Omega \times \Omega$ dans $\mathbb{R}^+$, qui représente le coût de remplacer x par $\phi(y)$, et qui possède les propriétés suivantes :

$$\forall x, x' \in \Omega \times \Omega :$$
$$\bullet L(x, x') \geq 0$$
$$\bullet L(x, x') = 0 \Leftrightarrow x = x'$$

L'estimateur optimal, i.e la fonction ϕ optimale est alors la fonction minimisant l'espérance (notée $E[]$) du coût, c'est à dire :

$$E[L(X, \phi(y)) \ / \ Y = y] = \sum_{x \in \Omega} L(x, \phi(y)) P(x \ / \ y)$$

La fonction ϕ^{opt} minimise donc l'erreur moyenne conditionnellement à y, et l'estimateur optimal est $\hat{x} = \phi^{\text{opt}}(y)$.

Suivant les fonctions de coût envisagées, on obtient différents estimateurs et différentes méthodes de résolution associées.

7.2.3 Estimateur MAP

Considérons la fonction de coût suivante :

$$\begin{aligned} L(x, x') &= 1 \text{ si } x \neq x' \\ L(x, x') &= 0 \text{ sinon} \end{aligned}$$

Cette fonction consiste donc à pénaliser toute différence entre deux configurations, et ce, quel que soit le nombre de sites en lesquels elles diffèrent. Nous pouvons alors écrire :

$$\begin{aligned} E[L(X, \phi(y)) \ / \ y] &= \sum_{x \in \Omega} L(x, \phi(y)) P(x \ / \ y) \\ &= 1 - P(X = \phi(y) \ / \ y) \end{aligned}$$

Par conséquent, la fonction ϕ^{opt} minimisant l'espérance pour cette fonction de coût est celle qui maximimise la probabilité a posteriori :

$$\hat{x} = \phi^{\text{opt}}(y) = \text{Argmax}_{\phi}[P(X = \phi(y) \ / \ y)]$$

Il nous faut donc trouver la réalisation $\hat{x}$, fonction de y, maximisant la probabilité a posteriori $P(X \ / \ y)$. On parle de l'estimateur MAP (maximum a posteriori) ou de maximum de vraisemblance a posteriori.

On retrouve donc, avec cette fonction de coût en tout ou rien, la démarche intuitive que nous avions présentée dans la partie précédente 7.1.7 à savoir chercher la configuration maximisant la probabilité conditionnellement à la donnée disponible.

Les solutions algorithmiques associées à cet estimateur sont le recuit simulé et l'ICM que nous avons présentés dans le paragraphe 7.1.5, utilisés avec la distribution a posteriori avec paramètre de température.

7.2.4 Estimateur MPM

Considérons maintenant la fonction de coût définie par :

$$L(x, x') = \sum_{s \in S} L(x_s, x'_s) = \sum_{s \in S} \mathbf{1}_{x_s \neq x'_s}$$

La fonction de coût pénalise cette fois-ci une configuration proportionnellement au nombre de différences entre deux configurations. Elle paraît donc plus naturelle que la fonction de coût en tout ou rien précédente.

Dans le cas d'une fonction de coût définie comme somme de coûts en chaque site, on peut montrer le résultat suivant :

$$\begin{aligned} E[L(X, \phi(y)) \ / \ Y = y] &= \sum_{x \in \Omega} L(x, \phi(y)) P(x \ / \ y) \\ &= \sum_{s \in S} \sum_{x_s} L(x_s, \phi(y)_s) \sum_{x^s} P(x^s, x_s \ / \ y) \end{aligned}$$

Or $\sum_{x^s} P(x^s, x_s \ / \ y) = P(X_s = x_s \ / \ y)$, donc on peut faire apparaître les probabilités conditionnelles et espérances en chaque site s :

$$\begin{aligned} E[L(X, \phi(y)) \ / \ Y = y] &= \sum_{s \in S} \sum_{x_s} L(x_s, \phi(y)_s) P(X_s = x_s \ / \ y) \\ &= \sum_{s \in S} E[L(X_s, \phi(y)_s) \ / \ y] \end{aligned}$$

On passe donc de la probabilité conditionnelle globale d'une configuration à la probabilité conditionnelle en un site. Il s'agit d'une somme de termes positifs, et par conséquent la fonction ϕ optimale minimise en chaque site l'espérance conditionnelle du coût local $E[L(X_s, \phi(y)_s) \ / \ y]$. Ce résultat est valable pour toutes les fonctions de coût définies par une somme de coûts en chaque site.

Dans le cas de la fonction définie ci-dessus, on a alors comme précédemment :

$$E[L(X_s, \phi(y)_s) \ / \ y] = 1 - P(X_s = \phi(y)_s \ / \ y)$$

Ainsi, la valeur optimale de $\phi(y)$ ou de $\hat{x}$ en chaque site est telle que :

$$\hat{x}_s = \phi^{\text{opt}}(y)_s = \text{Argmax}_{\phi}[P(X_s = \phi(y)_s \ / \ y)]$$

i.e. on maximise en chaque site la marginale a posteriori $P(X_s \ / \ y)$.

On obtient donc des estimateurs du maximum a posteriori locaux, à calculer en chaque pixel contrairement à la recherche précédente qui était globale. L'estimateur est appelé maximum de vraisemblance a posteriori local ou maximum posterior marginal (pour maximum a posteriori de la marginale) abrégé en MPM.

D'un point de vue algorithmique, la taille de l'espace des configurations Ω ne permet pas un calcul direct des quantités $P(x_s \ / \ y)$. Aussi réalise-t-on en pratique des approximations de type Monte-Carlo. En effet, supposons que l'on soit capable de tirer des réalisations de X selon sa loi conditionnelle à y, et notons les $x(1), ..., x(N)$. Il est alors possible de calculer une approximation de l'estimateur MPM. Le tirage des réalisations ne pose quant à lui pas de problème particulier car nous avons vu dans le paragrpahe précédent 7.1.4 comment tirer des réalisations d'un champ de Gibbs avec l'échantillonneur de Gibbs et l'algorithme de Métropolis. Or sous les hypothèses rappelées en début de la section 7.2.1, la probabilité a posteriori $P(X \ / \ y)$ est une distribution de Gibbs.

Supposons donc que nous disposions de N échantillons de X tirés selon la loi a posteriori et que nous cherchions à estimer la distribution conditionnelle en chaque site $P(X_s = \lambda \ / \ y) \ \ \forall \lambda \in E$. Nous allons estimer cette quantité par la fréquence empirique de λ au site s dans les échantillons $x(k)$ de X, i.e :

$$\widehat{P}(X_s = \lambda \ / \ y) = \frac{1}{N} \sum_{k=1}^{N} \mathbf{I}_{x(k)_s = \lambda}$$

L'estimation au sens du MPM est alors donnée en chaque site en choisissant la valeur de x_s dans Ω maximisant $P(X_s \ / \ y)$.

7.2.5 Estimateur TPM

Considérons maintenant la fonction de coût définie par :

$$L(x, x') = \|x - x'\|^2 = \sum_{s \in S} (x_s - x'_s)^2$$

Il s'agit de l'erreur quadratique et elle pénalise cette fois-ci directement la somme des différences entre les deux configurations. Elle peut donc être plus adaptée dans certains cas que les précédentes, puisqu'elle tient compte non seulement du nombre de différences comme le MPM, mais aussi de leurs valeurs.

Dans ce cas, on a en utilisant le résultat établi précédemment :

$$E[L(X, \phi(y)) \ / \ Y = y] \ = \ \sum_{s \in S} E[(X_s - \phi(y)_s)^2 \ / \ y]$$

On cherche donc ϕ, telle que $E[(X_s - \phi(y)_s)^2 \ / \ y]$ soit minimum. Nous allons écrire cette espérance sous une nouvelle forme en utilisant la moyenne conditionnelle au site s, $\overline{x_s} = E[X_s \ / \ y] = \sum_{x_s \in E} x_s P(x_s \ / \ y)$:

$$\begin{aligned} E[(X_s - \phi(y)_s)^2 \ / \ y] &= \sum_{x_s \in E} (x_s - \overline{x_s})^2 P(x_s \ / \ y) + \sum_{x_s \in E} (\overline{x_s} - \phi(y)_s)^2 P(x_s \ / \ y) \\ &= K + (\overline{x_s} - \phi(y)_s)^2 \end{aligned}$$

où K est une constante ne dépendant pas de ϕ donc n'intervenant pas dans la minimisation. Par conséquent, le minimum de l'erreur est atteint pour la fonction ϕ telle que :

$$\phi^{\text{opt}}(y)_s = E[X_s \ / \ y]$$

Cet estimateur consiste à prendre en chaque site la moyenne conditionnelle locale donnée par la loi a posteriori, d'où le nom de TPM (*Thresholded Posteriori Mean*).

D'un point de vue algorithmique, la démarche est similaire à celle effectuée dans le paragraphe précédent pour le MPM. On approxime l'espérance conditionnelle en chaque site par la moyenne empirique en ce site des N échantillons tirés selon la loi a posteriori :

$$\widehat{E}(X_s \ / \ y) = \frac{1}{N} \sum_{k=1}^{N} x(k)_s$$

L'estimation au sens du TPM est alors donnée en chaque site par sa moyenne empirique. Remarquons que cet estimateur est mal adapté à une problématique de segmentation car la moyenne des étiquettes n'a alors aucun sens.

7.2.6 Comparaison des estimateurs MAP, MPM, et TPM

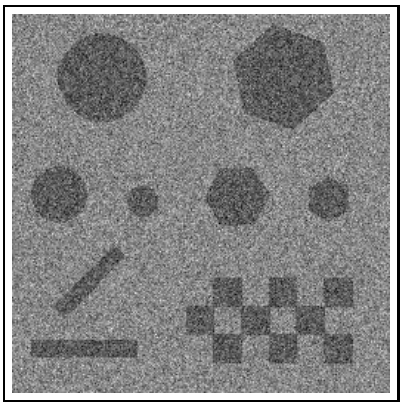
Nous comparons dans cette section les trois estimateurs dans le cadre de la restauration. Dans le cas du MAP, les résultats sont obtenus par ICM et par recuit simulé. L'image à restaurer (figure 7.5.a) est une image bruitée par un bruit blanc gaussien. L'énergie a posteriori utilisée s'écrit :

$$\mathcal{U}(x \ / \ y) = \sum_{s} \frac{(y_s - x_s)^2}{\sigma^2} + \sum_{c \in \mathcal{C}} + \beta \phi(x_r - x_s) \text{ avec } \beta > 0$$

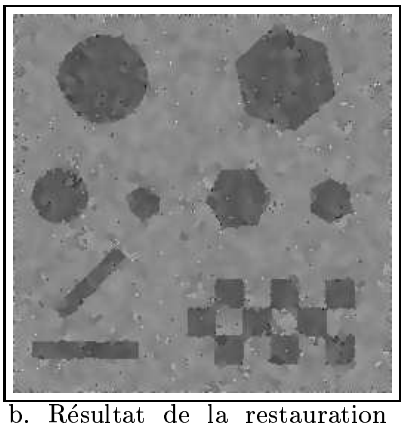
$$\text{et } \phi(u) = \frac{u^2}{1 + u^2}$$

Comme indiqué dans le paragraphe 7.1.7, cette fonction permet de seuiller les pénalités imposées par le terme de régularisation en présence de contours dans l'image.

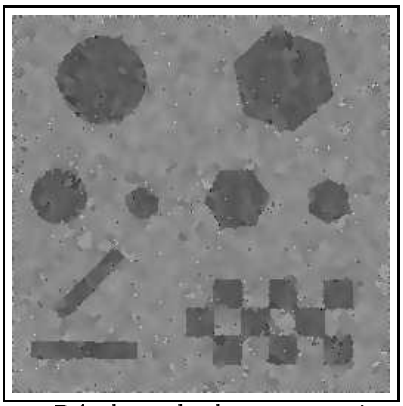
Les paramètres utilisés sont fixés aux valeurs suivantes : $\sigma = 28$ (écart-type du bruit), $\delta = 10$ (saut en amplitude à partir duquel on considère qu'il y a un contour), $\beta = 0,5$ pondération de l'influence relative des deux termes). L'initialisation est donnée par l'image à restaurer. Pour tous les estimateurs, on réalise 600 itérations. Dans le cas du recuit simulé, la température initiale est de 6. Les résultats sont montrés sur la figure 7.5.



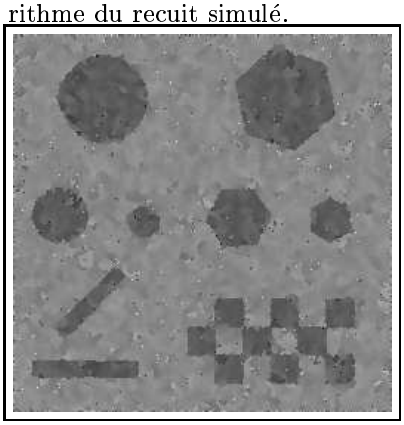
a. Image originale bruitée.



b. Résultat de la restauration avec l'estimateur MAP et l'algorithme du recuit simulé.



c. Résultat de la restauration avec l'estimateur MAP et l'algorithme ICM.



d. Résultat de la restauration avec l'estimateur TPM.

FIG. 7.5 – Comparaison des algorithmes ICM, Recuit Simulé et TPM en restauration.

On constate visuellement que le meilleur résultat est obtenu par le MAP du recuit simulé. Comme l'image originale est une assez bonne initialisation, il n'y a pas de grandes différences entre les algorithmes de recuit simulé et d'ICM pour l'estimateur du MAP. Ce n'est pas vrai dès que l'initialisation s'écarte du résultat à obtenir et les différences avec le recuit simulé peuvent être très importantes. Par ailleurs, on constate que l'estimateur TPM, qui est par définition plus local, donne un résultat plus bruité et moins régularisé. Cette analyse visuelle est confirmée par l'étude statistique qui peut être effectuée sur des zones homogènes de l'image. Le tableau ci-dessous donne les statistiques d'une zone sombre et d'une zone claire de l'image originale et pour les différents résultats de restauration. Les écarts-types les plus faibles sont obtenus pour l'estimateur MAP.

Statistiques sur l'image originale		
zone 1	135.7	27.8
zone 2	90.9	27.1
Estimateur MAP ICM		
zone 1	136.7	11.5
zone 2	91.6	6.9
Estimateur TPM		
zone 1	136.2	11.3
zone 2	91.6	6.9
Estimateur MAP recuit simulé		
zone 1	136.2	10.3
zone 2	91.9	6.1

TAB. 7.1 – Statistiques sur des zones homogènes pour les différents résultats de restauration

On notera également que des points isolés de faible ou fort niveau de gris subsistent dans l'image restaurée. Cela est lié à l'utilisation de la ϕ fonction qui ne régularise plus au-delà d'un certain seuil contrôlé par la valeur de δ.

En ce qui concerne les temps de calcul, les méthodes se répartissent comme suit : l'algorithme le plus rapide pour converger est sans conteste l'ICM, les algorithmes de recuit simulé et de TPM (ou MPM) étant à peu près équivalents. En effet, plus le nombre d'itérations est grand, meilleure est l'estimation de la moyenne a posteriori.

Les conclusions qui sont données ici ne sont pas nécessairement valables pour une application en segmentation. L'estimateur du TPM peut en effet dans certains cas donner de meilleurs résultats que le MAP. Par ailleurs, l'ICM peut s'avérer très utile lorsqu'on connaît une configuration proche de la configuration optimale.

7.3 Estimation des paramètres

7.3.1 Introduction

Le problème de l'estimation de paramètres (encore appelés hyperparamètres dans la littérature), revient très fréquemment en traitement d'image par champs de Markov. Donnons-en plusieurs exemples :

1. On se donne une réalisation d'un champ de Markov associé au modèle d'Ising, mais on ne connaît pas ses paramètres. Quels sont-ils ?
2. On veut généraliser ceci à une image de texture donnée dont on connaît le modèle sous-jacent (par exemple un modèle gaussien en 4-connexité) mais pas les paramètres, qui sont du type :

moyenne locale, variance locale, poids de la régularisation locale. Quels sont-ils ? Leur connaissance pourrait bien en effet servir à la classification d'images composées de zones texturées en se basant sur l'estimation locale de tels paramètres. On classifierait alors selon les valeurs de ces attributs locaux.

3. On veut segmenter une image, et pour cela apprendre les paramètres de chaque classe, ainsi que le coefficient optimal du modèle de régularisation adapté à cette tâche. On sait en effet que le résultat d'une segmentation par estimateur MAP par exemple dépend fondamentalement du poids respectif de la régularisation par rapport à celui de l'attache aux données. Il faut donc là aussi estimer ce poids d'une façon optimale dans un sens à définir.

L'ensemble forme un problème réputé difficile. Nous ne démentirons pas ici la difficuté de ces problèmes, accentuée par le fait que de très nombreuses variantes ont été mises au point dans la littérature pour le résoudre. Certaines tentatives de comparaison de ces approches ont déja été effectuées [Pieczynski, 1994]. Nous exposerons ici les variantes les plus fréquemment utilisées.
Elles se décomposent en deux classes fondamentales :

- le cas des données dites complètes, correspondant aux deux premiers problèmes cités plus haut : un échantillon d'une distribution de Gibbs est connu. Il s'agit de remonter aux paramètres de cette distribution.
- le cas des données dites incomplètes. Là non seulement le résultat de traitement est inconnu, mais les paramètres sont également à estimer.

7.3.2 Données complètes

Notons dans ce qui va suivre x une configuration observée relativement à une distribution de Gibbs donnée P_θ dont l'énergie associée puisse s'écrire sous la forme d'une fonction linéaire d'un paramètre θ, par exemple $U(x) = \theta\ \Phi(x)$, où Φ est un potentiel donné. Un principe naturel en vue de la recherche de θ est d'écrire la vraisemblance de la donnée x :

$$L(\theta) = P_\theta(x) = \frac{\exp - \theta\ \Phi(x)}{Z_\theta}$$

et de chercher par exemple la valeur de l'hyperparamètre $\hat{\theta}$ maximisant cette vraisemblance $L(\theta)$. Le problème essentiel est que l'on ne sait en général pas calculer exactement la fonction de partition Z_θ. Même pour des modèles aussi simples et fondamentaux que ceux d'Ising et de Potts, le résultat (analytique) est obtenu après des calculs excessivement compliqués [Onsager, 1944, Landau et Lifschitz, 1961]. Dans les autres cas on sera donc amené :

- soit à effectuer des approximations de la fonction de partition globale au moyen des fonctions de partition conditionnelles locales (codages paragraphe 7.3.2, pseudo-vraisemblance paragraphe 7.3.2)
- soit à employer des algorithmes itératifs (gradient stochastique paragraphe 7.3.2) à partir de la vraisemblance exacte, mais dont il s'agit alors de prouver la convergence ainsi que le type d'optimum trouvé (local, global).

Méthode des codages

Le principe de la méthode des codages [Besag, 1974] est le suivant. Une fois défini un système de voisinage pour un champ de Markov, nous sommes capables de définir un certain nombre de sous réseaux, chacun formé de sites/pixels indépendants les uns des autres : chacun de ces sous réseaux est appelé un codage. Par exemple avec un voisinage en 4 connexité il existe deux codages comme le montre la figure ci dessous (à gauche), et 4 codages différents dans le cas de la 8-connexité (à droite) :

					3	4	3	4	3
2	1	2	1	2	2	1	2	1	2
1	2	1	2	1	3	4	3	4	3
2	1	2	1	2	2	1	2	1	2
					3	4	3	4	3

Nous allons poser le problème d'estimation dans le cadre de chaque codage. Pour un codage donné les différents sites/pixels le constituant sont indépendants les uns des autres puisqu'ils ne sont pas des voisins pour le champ de Markov de départ. La probabilité globale d'un codage se trouvera donc être le produit des probabilités individuelles de chacun des sites du codage. Or du fait de la structure de Markov du champ de départ cette probabilité individuelle se trouve être la probabilité conditionnelle locale du site/pixel dans le champ de Markov. Pour un codage Cod_n donné nous pouvons donc écrire :

$$P_\theta(\{X_s = x_s\}_{s\in \mathrm{Cod}_n} \ / \ \{X_r = x_r\}_{r\notin \mathrm{Cod}_n}) = \prod_{s\in \mathrm{Cod}_n} P_\theta(X_s = x_s \ / \ V_s) \tag{7.5}$$

Etant donné la structure des probabilités locales telles qu'elles ont été décrites dans les chapitres précédents la fonction de vraisemblance devient calculable, puique les fonctions de partition conditionnelles locales le sont. Dans le cas où la dépendance des énergies locales est linéaire vis-à vis des paramètres, la log-vraisemblance associée,

$$\log P_\theta(\{X_s = x_s\}_{s\in \mathrm{Cod}_n} \ / \ \{X_r = x_r\}_{r\notin \mathrm{Cod}_n}) = -\theta \sum_{c\in\mathcal{C}} U_c(x) - \sum_{s\in \mathrm{Cod}_n} \log(Z_s)$$

est une fonction concave du (des) paramètre(s), car somme de fonctions concaves [Sigelle, 1997]. Elle se prête donc bien à la recherche d'un optimum par une méthode classique de type gradient. On peut également montrer qu'il s'agit dans ce cas d'un simple problème de moindres carrés [Derin et al., 1985].

Pseudo-vraisemblance

Il apparaît en fait expérimentalement que la méthode des codages n'est pas fiable. La méthode du maximum de vraisemblance vrai paraît quant à elle incalculable. Des algorithmes ont cependant été étudiés pour tenter de résoudre ce problème [Younes, 1988], voir paragraphe 7.3.2. En fait nous allons utiliser une méthode intermédiaire qui aura de bonnes propriétés : la méthode du pseudo-maximum de vraisemblance [Graffigne, 1987]. Du maximum de vraisemblance vrai, nous allons prendre l'idée de travailler sur l'ensemble de l'image et non séparément sur des réseaux indépendants. De la méthode de codage, nous conservons l'idée de manipuler une fonction de vraisemblance produit des probabilités locales de chacun des sites/pixels. Cette fonction sera appelée pseudo-maximum de vraisemblance, et elle s'écrira :

$$PL_\theta(X = x) = \prod_{s\in S} P(X_s = x_s \ / \ V_s) \tag{7.6}$$

ou, encore, en considérant le logarithme de cette fonction, et l'expression de la probabilité locale de chaque site/pixel :

$$\log PL_\theta(X = x) = -\theta \sum_{c\in\mathcal{C}} U_c(x) - \sum_{s\in S} \log(Z_s) \tag{7.7}$$

Maintenant l'expression $-\log(Z_s)$ devient calculable, puisque reliée à la fonction de normalisation de la probabilité conditionnelle locale telle qu'elle a été décrite dans le paragraphe 7.1.3. Donc un raisonnement identique à celui du paragraphe précédent conduit au fait que la log-pseudo-vraisemblance est une fonction concave des paramètres lorsque l'énergie en dépend de façon linéaire. Les algorithmes usuels de type gradient ou gradient conjugué s'appliquent donc aussi ici naturellement à la recherche de l'optimum (qui est unique comme précédemment).
Il s'agit alors de qualifier la valeur des paramètres obtenus par cette méthode par rapport à la valeur vraie. Des résultats théoriques importants ont été obtenus à ce sujet [Graffigne, 1987, Guyon, 1992] : la méthode de la pseudo-vraisemblance est consistante et convergente.

Algorithme du gradient stochastique

Partons de la vraisemblance exacte du paramètre. Elle s'écrit bien sûr :

$$L(\theta) = P_\theta(x) = \frac{\exp - \theta\ \Phi(x)}{Z_\theta}$$

La valeur de l'hyperparamètre satisfaisant au principe du maximum de vraisemblance $\hat{\theta} = \arg\max_\theta P_\theta(x)$ doit donc vérifier l'équation :

$$\left(\frac{\partial \log P_\theta(x)}{\partial \theta}\right)_{\hat{\theta}} = -\ \Phi(x) - \left(\frac{\partial \log Z_\theta}{\partial \theta}\right)_{\hat{\theta}} = 0 \tag{7.8}$$

Comme $\mathbf{E}_{\hat{\theta}}[\Phi] = -\left(\frac{\partial \log Z_\theta}{\partial \theta}\right)_{\hat{\theta}}$ [Sigelle, 1997], cette valeur optimale est donc unique et doit satisfaire l'équation suivante :

$$\mathbf{E}_{\hat{\theta}}[\Phi] = \Phi(x) \tag{7.9}$$

Il s'agit là de ce que l'on appelle une équation stochastique. Comme remarqué en introduction de la Section 7.3.2 cette équation ne peut être résolue exactement, pour la raison que $\mathbf{E}_\theta[\Phi]$, qui dérive de la fonction de partition Z_θ, ne peut en général être calculé de façon analytique exacte. On est donc amené à employer des algorithmes basés sur un schéma itératif de type Newton-Raphson, mais adapté à ce cadre stochastique. Un schéma rigoureux conduirait à chaque étape (n) à :

$$\theta_{n+1} = \theta_n - \frac{\mathbf{E}_{\theta_n}[\Phi] - \Phi(x)}{\left(\frac{\partial(\mathbf{E}_\theta[\Phi])}{\partial \theta}\right)_{\theta_n}}$$

c'est-à-dire au schéma itératif :

$$\theta_{n+1} = \theta_n + \frac{\mathbf{E}_{\theta_n}[\Phi] - \Phi(x)}{\mathrm{var}_{\theta_n}(\Phi)}$$

L'idée est alors de remplacer les grandeurs statistiques mises en jeu par leurs valeurs empiriques approchées. Ainsi pour l'espérance du potentiel de régularisation Φ, on prendra sa moyenne empirique au cours d'une seule itération (c'est à dire la valeur effective obtenue !) d'un échantillonneur de Gibbs ou de Metropolis mené avec la valeur courante du paramètre. Quant à la variance de ce potentiel, on l'estime encore plus crûment par une grandeur positive fixée V ! On peut montrer que le prix à payer pour cette approximation est l'introduction d'un terme correctif supplémentaire en $\frac{1}{n+1}$ dans le schéma itératif à l'itération (n). C'est le principe de l'algorithme de gradient stochastique [Younes, 1988] :

$$\left\{ \begin{array}{ll} \theta_0 & \text{arbitraire} \\ x^{(0)} & \text{tiré au hasard} \end{array} \right. , \quad \theta_{n+1} = \theta_n + \frac{\Phi(x^{(n)}) - \Phi(x)}{(n+1)\ V} \quad \text{pour } n \geq 1 \tag{7.10}$$

Il est très important de noter ici que $x^{(n)}$ $(n \geq 1)$, échantillon de la distribution P_{θ_n} obtenu par une dynamique de Gibbs (ou de Metropolis) associée à la valeur courante du paramètre θ_n, est généré à partir de l'échantillon $x^{(n-1)}$ obtenu à l'itération précédente (c'est à dire lors de la valeur précédente du paramètre). On peut alors montrer que cet algorithme stochastique converge presque sûrement, en termes de probabilité, vers la valeur optimale $\hat{\theta}$ lorsque le coefficient V est choisi suffisamment grand.

7.3.3 Données incomplètes

Dans cette section, nous abordons le problème de l'estimation des paramètres dans le cas des données incomplètes dites encore manquantes.

Dans ce cas nous connaissons une observation y, échantillon de la v.a. Y, mais elle est appelée incomplète (ou dégradée), car reliée à une scène originale x, non-dégradée, dont le champ aléatoire correspondant sera noté X. La relation entre y et x s'effectue via une loi de probabilité conditionnelle représentant l'attache aux données (cf. 7.1.7) dont nous explicitons la dépendance p.r. à un paramètre λ positif :

$$P_\lambda(Y = y \ / \ X = x) = \frac{\exp - U_\lambda(y \ / \ x)}{Z_\lambda}$$

Ainsi, dans le cas d'un bruit blanc gaussien additif en restauration ou en déconvolution et l'approximation discrète finie, on a :

$$P_\lambda(Y = y \ / \ X = x) = \frac{\exp - \lambda \ || \ y - Rx \ ||^2}{Z_\lambda}$$

où R est la matrice associée à la réponse impulsionnelle de la fonction de flou (l'identité en restauration) et $Z_\lambda = \left(\sqrt{\frac{\pi}{2\lambda}}\right)^{|S|}$. On notera que la fonction de partition $Z_\lambda = \sum_{y \in \Omega} \exp - U_\lambda(y \ / \ x)$ est ici indépendante de la variable cachée x.
On suppose aussi que l'on dispose d'une connaissance a priori sur la scène à retrouver x, que ce soit en segmentation ou restauration, via la distribution de Gibbs suivante :

$$P_\theta(x) \frac{\exp - \theta \ \Phi(x)}{Z_\theta}$$

Nous aurons besoin dans la suite de ce chapitre d'utiliser la distribution de Gibbs a posteriori :

$$P_{\theta,\lambda}(X = x \ / \ Y = y) = \frac{\exp - U_\lambda(y \ / \ x) - \theta \ \Phi(x)}{Z_{\theta,\lambda}}$$

de fonction énergie $\mathcal{U}(x \ / \ y) = U_\lambda(y \ / \ x) + \theta \ \Phi(x)$, et de fonction de partition associée $Z_{\theta,\lambda}$.
Nous commencerons par généraliser, dans un cadre de Maximum de Vraisemblance (MV), la méthode de gradient stochastique vue au paragraphe précédent lorsque la loi d'observation (attache aux données) est complètement connue, c'est-à-dire que nous nous focaliserons sur l'estimation du meilleur paramètre de régularisation θ. Puis nous comparerons cette méthode à des variantes importantes similaires répertoriées dans la littérature.
Nous aborderons ensuite l'estimation des paramètres d'attache aux données, en particulier dans le cadre de la segmentation d'images. Cela nous permettra de décrire ensuite la seconde grande classe de méthodes adaptée à l'estimation des paramètres : l'EM (Expectation-Maximisation).

Gradient stochastique généralisé [Younes, 1989]

On va prouver dans cette partie que l'estimation du paramètre de régularisation connaissant la forme du potentiel a priori ne peut être dissociée dans la plupart des cas de la forme de l'attache aux données,

supposée connue ici par simplicité, c'est à dire que λ est connu. On suppose accéder par exemple d'une façon ou d'une autre à la variance d'un bruit gaussien en restauration d'image bruitée (et à la réponse impulsionnelle du flou si l'on est en déconvolution).
Dans le cas supposé où aucune information *a priori* sur le paramètre de régularisation n'est disponible, c'est-à-dire lorsque θ suit la distribution uniforme sur $\mathbb{R}$, la vraisemblance de ce paramètre est la grandeur adéquate à étudier connaissant l'observation incomplète y et le paramètre λ. Elle peut se calculer de façon élégante [?] en remarquant que la loi jointe de l'observation et des variables cachées s'écrit :

$$\begin{aligned} P_{\theta,\lambda}(X = x, Y = y) &= P_\lambda(X = x \ / \ Y = y) P_\theta(X = x) \\ &= \frac{\exp - U_\lambda(y \ / \ x)}{Z_\lambda} \cdot \frac{\exp - \theta \ \Phi(x)}{Z_\theta} = \frac{\exp - U_\lambda(y \ / \ x) - \theta \ \Phi(x)}{Z_\lambda \ . \ Z_\theta} \end{aligned}$$

Si au moins une valeur de x n'annule pas $P(Y = y \ / \ X = x)$, on en déduit que :

$$L(\theta) = P_{\theta,\lambda}(Y = y) = \frac{P_{\theta,\lambda}(Y = y, X = x)}{P_{\theta,\lambda}(X = x \ / \ Y = y)} = \frac{Z_{\theta,\lambda}}{Z_\lambda \ . \ Z_\theta} \tag{7.11}$$

Toute valeur optimale du paramètre de régularisation $\hat{\theta}$ satisfait donc l'équation suivante :

$$\left(\frac{\partial \log L(\theta)}{\partial \theta} \right)_{\hat{\theta}} = \mathbf{E}_{\hat{\theta}}[\Phi] - \mathbf{E}_{\hat{\theta},\lambda}[\Phi] = 0 \Rightarrow \mathbf{E}_{\hat{\theta}}[\Phi] = \mathbf{E}_{\hat{\theta},\lambda}[\Phi] \tag{7.12}$$

où l'on rappelle que $\mathbf{E}_\theta[.]$ est l'espérance d'une v.a. sous la distribution de Gibbs a priori d'énergie $\theta \ \Phi(x)$, tandis que $\mathbf{E}_{\theta,\lambda}[.]$ signifie l'espérance statistique sous la distribution de Gibbs a posteriori. Notons également que les deux grandeurs statistiques $\mathbf{E}_{\theta,\lambda}[\Phi]$ et $\mathbf{E}_\theta[\Phi]$ sont des fonctions monotones décroissantes de θ, ce qui implique que plusieurs valeurs optimales de l'hyperparamètre peuvent exister (voir fig.7.3.3).

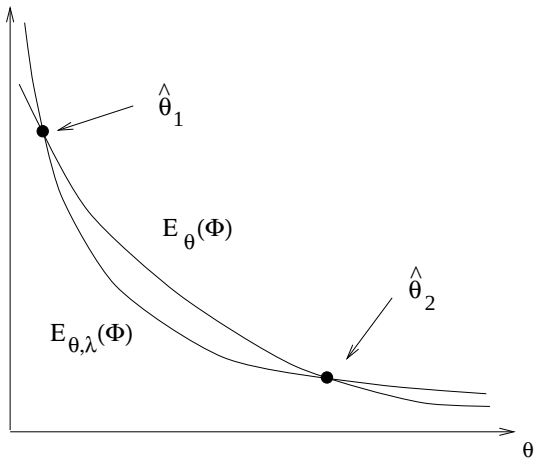


FIG. 7.6 – Valeur(s) optimale(s) $\hat{\theta}$ du paramètre de régularisation en données incomplètes.

Implémentation : Si l'on applique un schéma de Newton-Raphson à l'équation stochastique (7.12) on obtient directement :

$$\theta_{n+1} = \theta_n - \frac{\mathbf{E}_{\theta_n}[\Phi] - \mathbf{E}_{\theta_n,\lambda}[\Phi]}{\left(\frac{\partial (\mathbf{E}_{\theta_n}[\Phi] - \mathbf{E}_{\theta_n,\lambda}[\Phi])}{\partial \theta_n} \right)} = \theta_n + \frac{\mathbf{E}_{\theta_n}[\Phi] - \mathbf{E}_{\theta_n,\lambda}[\Phi]}{\mathrm{var}_{\theta_n}(\Phi) - \mathrm{var}_{\theta_n,\lambda}(\Phi)} \tag{7.13}$$

Il faut d'abord noter que dans ce schéma le dénominateur du dernier terme peut être de signe quelconque contrairement au cas des données complètes (7.10), ce qui est relié à l'existence de plusieurs solutions possibles en données incomplètes. Maintenant, en s'inspirant du raisonnement effectué pour le gradient stochastique pour les données complètes, on s'aperçoit qu'il est nécessaire d'approximer des quantités statistiques (moyenne, variance) reliées aux distributions a posteriori et a priori. On a donc besoin d'échantillonner deux champs de Markov en général : celui lié à la régularisation pure (champ a priori) et le champ de Markov postérieur comprenant l'attachement aux données. Le schéma empirique qui en résulte naturellement est donc le suivant [Younes, 1989] :

$$\begin{cases} \text{un échantillon } x^{(n)} \text{ généré par } P_{\theta_n} \quad \text{(distribution a priori)} \\ \\ \text{un échantillon } \tilde{x}^{(n)} \text{ généré par } P_{\theta_n,\lambda} \quad \text{(distribution a posteriori)} \end{cases}$$

avec la procédure d'évolution de l'hyperparamètre :

$$\theta_{n+1} = \theta_n + \frac{1}{n} \cdot \frac{\Phi(x^{(n)}) - \Phi(\tilde{x}^{(n)})}{(< \mathrm{var}_{\theta_n}(\Phi) > - < \mathrm{var}_{\theta_n,\lambda}(\Phi) >)}$$

Il faut noter en particulier la présence des estimateurs empiriques des variances $< \mathrm{var}_{\theta_n}(\Phi) >$ et $< \mathrm{var}_{\theta_n,\lambda}(\Phi) >$ (qui nécessitent donc en fait au moins deux échantillons pour chacune des distributions et à chaque étape du schéma d'évolution de l'hyperparamètre). Ainsi en segmentation, on doit échantillonner aussi bien la distribution a posteriori, ce qui sera par ailleurs utile en vue de la segmentation à convergence des paramètres par l'un des estimateurs MPM ou TPM, que la distribution a priori qui correspond au modèle de régularisation pur : modèles d'Ising, de Potts, généralisation, et ceci pour la valeur courante du (des) hyperparamètre(s).
La convergence de cette méthode est à montrer dans le cadre général des algorithmes stochastiques [Métivier et Priouret, 1987], qui dépasse largement l'objet de ce chapitre.

Comparaison avec d'autres variantes d'estimation

Dans ce paragraphe on présente d'autres méthodes usuelles d'estimation des hyperparamètres en données incomplètes, qui sont en fait reliées à d'autres estimateurs que le maximum de vraisemblance. Toute la subtilité réside ici dans le choix de la fonction de type vraisemblance à maximiser. On va voir en effet que de très légères différences conduisent à des formes fort différentes d'équations stochastiques, et donc des résultats d'estimation a priori fort différents.
On suppose dans les deux premiers cas qu'un résultat " optimal " de restauration (ou segmentation) x^* est connu à l'étape d'estimation où l'on se situe.

Une première approche consiste à utiliser la loi jointe de l'observation et du résultat connaissant la valeur courante des hyperparamètres [Lakshmanan et Derin, 1989a, Descombes et al., 1999]. On peut alors écrire que le paramètre optimal doit satisfaire :

$$\hat{\theta} = \arg\max_{\theta} P_{\theta,\lambda}(X = x^* \ , \ Y = y)$$

En utilisant comme précédemment le fait que l'attache aux données (resp. la régularisation) est indépendante de l'hyperparamètre θ (resp. de λ), on obtient :

$$P_{\theta,\lambda}(X = x^* \ , \ Y = y) = P_{\lambda}(Y = y \ / \ X = x^*) P_{\theta}(X = x^*)$$

Et comme le premier terme (attache aux données) ne dépend pas de θ,

$$\hat{\theta} = \arg\max_{\theta} P_{\theta}(X = x^*) = \frac{\exp - \theta \ \Phi(x^*)}{[\ Z_{\theta} = \sum_{x \in \Omega} \exp - \theta \ \Phi(x) \ ]}$$

On retombe alors sur l'estimateur au sens du maximum de vraisemblance pour le champ de Markov a priori d'énergie $\theta\ \Phi(x)$ et pour la donnée ici complète x^*, c'est à dire : $\mathbf{E}_{\hat{\theta}}[\Phi] = \Phi(x^*)$. On peut donc utiliser une technique de gradient stochastique classique pour estimer ce paramètre (cf. paragraphe 7.3.2). On peut ensuite itérer en effectuant le traitement désiré (restauration, segmentation) avec les nouvelles valeurs des hyperparamètres obtenues. On obtient donc une nouvelle configuration optimale à partir de laquelle on peut re-estimer les paramètres, et ainsi de suite [Khoumri, 1997, Zerubia et Blanc-Féraud, 1998]. La méthode est théoriquement convergente vers les valeurs optimales des paramètres et de la configuration.

Une seconde approche consiste à utiliser la probabilité du résultat conditionnellement à l'observation et aux hyperparamètres. Dans ce cas, le paramètre optimal doit vérifier :

$$\hat{\theta} = \arg\max_{\theta} P_{\theta,\lambda}(X = x^* \ / \ Y = y)$$

En appliquant la seconde formule de Bayes et en utilisant le même argument d'indépendance des différentes lois de probabilités vis-à-vis des différents hyperparamètres mis en jeu, on obtient :

$$\hat{\theta} \quad = \quad \arg\max_{\theta} \frac{\exp\ -\lambda\ U(y\ /\ x^*) -\ \theta\ \Phi(x^*)}{[\ Z_{\theta,\lambda} = \sum_{x\in\Omega} \exp\ -\lambda\ U(y\ /\ x) -\ \theta\ \Phi(x)\ ]}$$

On tombe cette fois sur l'estimateur au sens du maximum de vraisemblance pour la distribution de Gibbs a posteriori d'énergie $\theta\ \Phi(x) + \lambda\ U(y\ /\ x)$ et pour la donnée complète x^*, c'est-à-dire : $\mathbf{E}_{\hat{\theta},\lambda}[\Phi] = \Phi(x^*)$. Les deux méthodes aboutissent donc à l'estimation d'un paramètre unique, et peuvent donc être implémentées avec une technique de gradient stochastique classique [Younes, 1988, Khoumri, 1997]. Nous citerons également l'alternative des méthodes MCMCML (Monte Carlo Markov Chains Maximum Likelihood) [Descombes et al., 1999].

Il est très intéressant de comparer les deux résultats précédents avec le véritable gradient stochastique généralisé vu précédemment. On rappelle que celui-ci permet de maximiser la probabilité du paramètre de régularisation conditionnellement à l'observation uniquement [Younes, 1989] :

$$\hat{\theta} = \arg\max_{\theta} P_{\theta,\lambda}(Y = y)$$

En toute rigueur, il est l'estimateur exact au sens du MV du paramètre de régularisation. Les deux autres cités précédemment ont essentiellement l'avantage d'être plus rapides, car ne nécessitant qu'un échantillonnage à chaque étape, et permettant de faire alterner le traitement d'image désiré avec l'estimation des paramètres.
En pratique, les trois estimateurs cités aboutissent à des valeurs du paramètre de régularisation semblables, ce qui provient du fait que les modèles adoptés pour l'attache aux données et pour la régularisation sont souvent choisis en cohérence avec l'observation fournie.

Estimation des hyperparamètres d'attache aux données : le cas de la segmentation

Supposons que l'on veuille estimer également les paramètres intervenant dans l'attache aux données, que nous regrouperons sous une variable λ. Ainsi, dans le cas important de la segmentation où le niveau de gris de chacune des m régions d'une image est supposée suivre une loi gaussienne de moyenne μ_i et de variance σ_i^2, λ est l'ensemble $\{\sigma_i,\ \mu_i\}_{i=1,m}$. Nous étudierons ici l'estimation de ces paramètres dans le cas gaussien par raison de commodité. La probabilité des observations conditionellement aux labels s'écrit donc :

$$P_\lambda(Y = y\ /\ X = x) = \prod_{s\in S} \frac{1}{\sqrt{2\pi}\sigma_{x_s}}\ \exp -\frac{1}{2\ \sigma_{x_s}^2}\ (y_s - \mu_{x_s})^2$$

On peut écrire la distribution a posteriori sous la forme :

$$P_{\theta,\lambda}(X = x \ / \ Y = y) = \frac{\exp - \mathcal{W}_{\theta,\lambda}(x \ / \ y)}{Z_{\theta,\lambda}}$$

avec comme définition de la fonction énergie a posteriori effective :

$$\mathcal{W}_{\theta,\lambda}(x \ / \ y) = \sum_{s \in S} \left(\frac{1}{2 \ \sigma_{x_s}^2} \ (y_s - \mu_{x_s})^2 + \ \log \sigma_{x_s} \right) + \theta \ \Phi(x) \tag{7.14}$$

et comme fonction de partition a posteriori associée $Z_{\theta,\lambda}$. La vraisemblance des paramètres d'attache aux données s'écrit donc selon le même principe que précédemment

$$L(\lambda) = P_{\theta,\lambda}(Y = y) = \frac{Z_{\theta,\lambda}}{(\sqrt{2\pi})^{|S|} Z_\theta}$$

Intéressons nous à maximiser cette quantité vis-à-vis d'un des paramètres de λ_i. Il faut remarquer ici que les fonctions énergie mises en jeu dépendent de façon non-linéaire de l'ensemble de paramètres $\lambda = \{\lambda_i\}$. D'où l'on déduit pour chacun des paramètres λ_i :

$$\frac{\partial \log L(\lambda)}{\partial \lambda_i} = \frac{\partial \log Z_{\theta,\lambda}}{\partial \lambda_i} = - \ \mathrm{E}_{\theta,\lambda}[\frac{\partial \mathcal{W}_{\theta,\lambda}}{\partial \lambda_i}] = 0 \tag{7.15}$$

Précisons qu'il s'agit bien de l'espérance a posteriori. Pour analyser la dépendance précise de l'énergie a posteriori vis-a-vis des paramètres $\{\sigma_i, \ \mu_i\}_{i=1,m}$, il est commode de re-écrire la formule (7.14) en employant les fonctions caractéristiques d'appartenance à chaque classe :

$$\mathcal{W}_{\theta,\lambda}(x \ / \ y) \quad = \quad \sum_{i=1}^{m} \left[\sum_{s \in S} \left(\frac{1}{2 \ \sigma_i^2} \ (y_s - \mu_i)^2 + \log \sigma_i \right) \ . \ \mathrm{I}_{x_s = i} \right] + \theta \ \Phi(x)$$

Il en résulte des formules importantes pour la suite :

$$\left\{ \begin{array}{rcl} \dfrac{\partial \mathcal{W}_{\theta,\lambda}(x \ / \ y)}{\partial \mu_i} & = & \dfrac{1}{\sigma_i^2} \ \displaystyle\sum_{s \in S} \ (\mu_i - y_s) \ \mathrm{I}_{x_s = i} \\ \dfrac{\partial \mathcal{W}_{\theta,\lambda}(x \ / \ y)}{\partial \sigma_i} & = & \dfrac{1}{\sigma_i} \ \displaystyle\sum_{s \in S} \left(\frac{- \ (y_s - \mu_i)^2}{\sigma_i^2} \ + 1 \right) \ \mathrm{I}_{x_s = i} \end{array} \right. \tag{7.16}$$

• Examinons d'abord le cas d'un paramètre de moyenne μ_i donné. On peut montrer que (si $\sigma_i < +\infty$) :

$$\forall i \in [1..m] \quad , \quad \mu_i = \frac{\displaystyle\sum_{s \in S} y_s \ P_{\theta,\lambda}(X_s = i)}{\displaystyle\sum_{s \in S} P_{\theta,\lambda}(X_s = i)} \tag{7.17}$$

L'interprétation physique en est claire : on obtient le barycentre des observations en chaque site de l'image pondérées par la probabilité a posteriori d'avoir une classe déterminée.

• De la même façon, on obtient pour les variances de chaque classe :

$$\forall i \in [1..m] \quad , \quad \sigma_i^2 = \frac{\displaystyle\sum_{s \in S} (y_s - \mu_i)^2 \ P_{\theta,\lambda}(X_s = i)}{\displaystyle\sum_{s \in S} P_{\theta,\lambda}(X_s = i)} \tag{7.18}$$

les μ_i pouvant être calculés par la formule (7.17). Le résultat est similaire à une variance empirique, mais avec pondération en chaque site par la probabilité a posteriori que ce site ait le label étudié.

En définitive on aboutit donc à des équations (7.17) et (7.18) auto-cohérentes, de forme bien plus complexe que celles des gradients stochastiques simple et généralisé. On aimerait pouvoir les remplacer par des formes itératives simples : c'est ce qui va justifier l'emploi des méthodes de type EM, à adapter dans le cadre gibbsien défini ici.

Expectation-maximization (EM)

Récapitulons les résultats précédents obtenus concernant l'estimation au Maximum de Vraisemblance des hyperparamètres dans le cas des données incomplètes :

• pour l'hyperparamètre de régularisation nous sommes arrivés à l'équation stochastique :

$$\mathbf{E}_{\hat{\theta}}[\Phi] = \mathbf{E}_{\hat{\theta},\lambda}[\Phi]$$

• pour les paramètres d'attache aux données en segmentation avec m classes nous sommes arrivés au système d'équations :

$$\forall i \in [1..m] \quad , \quad \begin{cases} \mu_i = \dfrac{\sum_{s \in S} y_s \; P_{\theta,\lambda}(X_s = i)}{\sum_{s \in S} P_{\theta,\lambda}(X_s = i)} \\ \sigma_i^2 = \dfrac{\sum_{s \in S} (y_s - \mu_i)^2 \; P_{\theta,\lambda}(X_s = i)}{\sum_{s \in S} P_{\theta,\lambda}(X_s = i)} \end{cases}$$

Nous en avons conclu qu'il serait désirable de résoudre ces équations de manière itérative. Par exemple si les probabilités a posteriori $P_{\theta,\lambda}(.)$ étaient connues (ou apprises) à une étape donnée on pourrait les injecter dans la première équation à condition de pouvoir estimer pour toute valeur de θ l'espérance a priori du potentiel de régularisation ainsi que dans le deuxième système. Ces deux arguments vont nous fournir les principes de l'EM, avec un certain nombre de résultats théoriques très puissants à la clé.

La méthode EM est par essence itérative. Des résultats théoriques très généraux montrent que la vraisemblance des paramètres estimés croît à chaque itération de l'algorithme associé [Baum et al., 1970, Redner et Walker, 1984]. On pourra également consulter en particulier [**?**] pour une présentation claire de cette méthode. De plus cette méthode permet d'une certaine façon l'optimisation séparée des hyperparamètres d'attache aux données et de celui de régularisation. En effet elle considère essentiellement la probabilité jointe de l'observation et des données cachées (un étiquetage de l'image par exemple), c'est-à-dire le produit de la loi d'observation et de la probabilité a priori.
Supposons que les paramètres θ_n et λ_n soient connus à l' étape courante n. On suppose aussi que l'on a accès d'une façon ou d'une autre aux statistiques liées à la loi a posteriori des données cachées courantes (ppartie *Expectation* de l'EM, voir plus loin). On définit alors la notation :

$$Q(\theta, \lambda, \theta_n, \lambda_n) = \mathbf{E}_{\theta_n,\lambda_n}[\log P_{\theta,\lambda}(X = x \; , \; Y = y)]$$

La partie optimisation (*Maximisation*) de l'EM consiste à rechercher :

$$(\theta_{n+1}, \lambda_{n+1}) \quad = \quad \arg\max_{\theta,\lambda} Q(\theta, \lambda, \theta_n, \lambda_n)$$

Du fait de la séparabilité de la loi jointe, la fonction objectif $Q(\theta, \lambda, \theta_n, \lambda_n)$ s'écrit :

$$Q(\theta, \lambda, \theta_n, \lambda_n) = \mathbf{E}_{\theta_n, \lambda_n}[\log P_\lambda(Y = y \ / \ X = x) + \log P_\theta(X = x)]$$

Cela correspond donc à l'optimisation séparée :

$$\left\{ \begin{array}{rcl} \theta_{n+1} & = & \arg\max_\theta \mathbf{E}_{\theta_n, \lambda_n}[\log P_\theta(X = x)] \\ \lambda_{n+1} & = & \arg\max_\lambda \mathbf{E}_{\theta_n, \lambda_n}[\log P_\lambda(Y = y \ / \ X = x)] \end{array} \right.$$

On rappelle que la vraisemblance des paramètres $L(\theta_n, \lambda_n)$ croît en fonction de l'itération n par cette méthode ce qui aboutit à terme à un optimum local pour les valeurs des paramètres.
Examinons plus précisément ce qui se passe pour chacune des catégories de paramètres :

- en ce qui concerne le paramètre de régularisation, cela correspondra à :

$$\mathbf{E}_{\theta_n, \lambda_n}[\frac{\partial \log P_\theta(X = x)}{\partial \theta}] = \mathbf{E}_{\theta_n, \lambda_n}[- \ \Phi - \frac{\partial \log Z_\theta}{\partial \theta}] = 0$$

c'est-à-dire en vertu de résultats supposés maintenant acquis :

$$\mathbf{E}_{\theta_{n+1}}[\Phi] = \mathbf{E}_{\theta_n, \lambda_n}[\Phi] \tag{7.19}$$

Supposons que l'on sache d'une manière ou d'une autre calculer ou estimer l'espérance a posteriori du potentiel de régularisation pour la valeur courante des paramètres. Ainsi, en pratique, on approxime cette quantité par sa moyenne empirique au cours d'un échantillonneur de Gibbs (ou de Metropolis) pris pour la valeur courante des paramètres [3] . On est alors ramené à un problème d'estimation du paramètre de régularisation pour la donnée complète $\mathbf{E}_{\theta_n, \lambda_n}[\Phi]$! On peut donc appliquer la technique du pseudo-maximum de vraisemblance [Chalmond, 1989] ou bien encore celle du gradient stochastique [Younes, 1991] pour re-estimer la nouvelle valeur du paramètre de régularisation.

- pour les paramètres d'attache aux données, on a vu plus haut que

$$\frac{\partial \log P_\lambda(Y = y \ / \ X = x)}{\partial \lambda_i} = -\frac{\partial \mathcal{W}_{\theta,\lambda}(x)}{\partial \lambda_i}$$

Le principe EM s'écrit donc

- pour les paramètres de moyenne :

$$\begin{array}{rcl} \mathbf{E}_{\theta_n, \lambda_n}[\frac{\partial \mathcal{W}_{\theta,\lambda}}{\partial \mu_i}] & = & \frac{1}{\sigma_i^2} \sum_{s \in S} (\mu_i - y_s) \ P_{\theta_n, \lambda_n}(X_s = i) = 0 \\ \Rightarrow \forall i \in [1..m] \ , \ \mu_i(n+1) & = & \frac{\sum_{s \in S} y_s \ P_{\theta_n, \lambda_n}(X_s = i)}{\sum_{s \in S} P_{\theta_n, \lambda_n}(X_s = i)} \end{array} \tag{7.20}$$

- pour les paramètres de variances :

$$\begin{array}{rcl} \mathbf{E}_{\theta_n, \lambda_n}[\frac{\partial \mathcal{W}_{\theta,\lambda}}{\partial \sigma_i}] & = & \sum_{s \in S} \left(\frac{-(y_s - \mu_i)^2}{\sigma_i^2} + 1 \right) P_{\theta_n, \lambda_n}(X_s = i) = 0 \\ \Rightarrow \forall i \in [1..m] \ , \ \sigma_i(n+1)^2 & = & \frac{\sum_{s \in S} (y_s - \mu_i(n+1))^2 \ P_{\theta_n, \lambda_n}(X_s = i)}{\sum_{s \in S} P_{\theta_n, \lambda_n}(X_s = i)} \end{array} \tag{7.21}$$

[3] Ceci correspond à l'étape Estimation de l'algorithme EM.

Deux remarques insistantes :
1. Ce sont bien les distributions a posteriori qui sont mises en jeu dans l'estimation itérative (eqs. 7.20 et 7.21). Elles sont donc à re-estimer à chaque itération n. Une manière bien naturelle est de remplacer ces probabilités à l'étape courante du processus EM par la fréquence empirique d'apparition des labels lors d'une série d'échantillonnages de la distribution de Gibbs a posteriori courante menés à l'aide d'un échantillonneur de type Gibbs ou Metropolis [Chalmond, 1989]. Notons pour cela N le nombre d'itérations effectué avec l'échantillonneur ainsi sélectionné :

$$\underset{(n)}{\bullet} \quad \underset{N}{- - - - - - - - - - -} \quad \underset{(n+1)}{\bullet}$$

Notons également $N_s(i)$ le nombre de fois que le label i a été tiré en un site s au cours de l'ensemble de ces N échantillonnages. On peut donc écrire : $P_{\theta_n,\lambda_n}(X_s = i) \approx \dfrac{N_s(i)}{N}$, et il en résulte les estimations empiriques des paramètres de moyennes et de variance de chaque classe :

$$\forall i \in [1..m] \quad , \quad \left\{ \begin{array}{lcl} \mu_i(n+1) & = & \dfrac{\sum\limits_{s\in S} y_s \; N_s(i)}{\sum\limits_{s\in S} N_s(i)} \\ \\ \sigma_i(n+1)^2 & = & \dfrac{\sum\limits_{s\in S} (y_s - \mu_i(n+1))^2 \; N_s(i)}{\sum\limits_{s\in S} N_s(i)} \end{array} \right.$$

De la même façon, notons $x^{(n)}(k)$ la série d'images échantillons ainsi obtenue pour $k = 1..N$. On obtient pour l'espérance a posteriori du potentiel de régularisation

$$\mathbf{E}_{\theta_n,\lambda_n}[\Phi] \approx \frac{1}{N} \left[\sum_{k=1}^{N} \Phi(x^{(n)}(k)) \right]$$

C'est donc ensuite que l'on procéde à l'étape d'estimation EM comme indiqué plus haut, à partir de ces valeurs empiriques a posteriori.
2. On retrouve le caractère itératif que nous voulions anticiper à propos des équations plus haut. Là aussi, à convergence, on doit satisfaire à la forme exacte des équations (7.17) et (7.18) obtenues par le Maximum de Vraisemblance.

Variantes Une variante importante, appelée ICE ou Iterative Conditional Estimation est la suivante [Pieczynski, 1994]. Considérons les N échantillons $x^{(n)}(k)$ obtenus précédemment à une étape courante n de l'EM. On pourrait songer à estimer moyenne et variance des classes pour chacun d'eux considéré comme segmentation courante, puis prendre la moyenne empirique (arithmétique) des valeurs ainsi obtenues

$$\forall i \in [1..m] \quad , \quad \left\{ \begin{array}{lcl} \mu_i(n+1) & = & \dfrac{1}{N} \sum\limits_{k=1}^{N} \left(\dfrac{\sum\limits_{s\in S} y_s \; \mathbf{I}_{x_s^{(n)}(k)=i}}{\sum\limits_{s\in S} \mathbf{I}_{x_s^{(n)}(k)=i}} \right) \\ \\ \sigma_i(n+1)^2 & = & \dfrac{1}{N} \sum\limits_{k=1}^{N} \left(\dfrac{\sum\limits_{s\in S} (y_s - \mu_i(n+1))^2 \; \mathbf{I}_{x_s^{(n)}(k)=i}}{\sum\limits_{s\in S} \mathbf{I}_{x_s^{(n)}(k)=i}} \right) \end{array} \right.$$

Cette estimation est différente de l'EM. On peut montrer qu'elle correspond en fait à l'estimateur de la moyenne a posteriori des paramètres.

Conclusion pour l'estimation en données incomplètes

On voit donc au terme de cette partie que l'estimation en données incomplètes se prête à un nombre très riche de variantes. On pourrait ainsi parfaitement juger préférable d'estimer certains types de paramètres comme moyenne et variance par l'EM, tandis que l'on adopterait la méthode de Lakhsmanan-Derin pour le paramètre de régularisation, ou réciproquement! Ces variantes méritent encore d'être examinées et comparées entre elles de façon exhaustive, dans la lignée de l'approche suivie dans [Pieczynski, 1994]. Il faut aussi préciser que les méthodes utilisant un échantillonnage de la distribution a posteriori (comme l'EM) permettent en même temps de fournir une série de configurations échantillons se prêtant favorablement à la segmentation ou à la restauration dans le cadre des estimateurs de type TPM ou MPM, lorsque la convergence des paramètres vers leur valeur optimale est supposée atteinte.

7.4 Processus de bords

L'un des grands domaines d'application des modèles markoviens est la restauration comme nous l'avons mentionné au paragraphe 7.1.7. Connaissant une réalisation y, on cherche une solution x minimisant le critère :

$$\mathcal{U}(x \ / \ y) = U_1(x, y) + \beta U_2(x)$$

Le terme de régularisation $U_2(x)$ permet alors d'agir directement sur les dérivées de la solution cherchée. Dans le cas d'un potentiel quadratique, il s'agit d'une régularisation de Tikhonov qui entraîne un fort lissage de la solution au détriment des discontinuités souvent naturellement présentes dans l'image.

De nombreux travaux ont porté sur l'introduction de discontinuités et sur les avantages potentiels que cela impliquait dans la recherche de la solution. Nous nous limiterons ici à des cliques d'ordre deux, c'est-à-dire à l'introduction de contraintes sur la dérivée première de la solution cherchée. Nous mettons en évidence l'équivalence entre processus de bords et certaines fonctions avant de présenter quelques algorithmes déterministes d'optimisation.

7.4.1 Processus de bords explicites et implicites

Dès l'article fondateur de Geman et Geman [Geman et Geman, 1984a], l'introduction de processus de bords permettant de désactiver l'effet de lissage entre deux sites est envisagé. Celui-ci, que nous noterons B dans la suite est défini sur une grille duale de celle de l'image et est souvent décomposé en deux sous champs, l'un représentant les interactions verticales B^v et l'autre les interactions horizontales B^h. Il s'agit souvent (mais pas nécessairement) de processus de bords booléens, prenant la valeur 1 en présence de discontinuités et 0 en son absence. Le terme de régularisation s'écrit alors [**?**] [Geiger et Girosi, 1991] :

$$U_2(x, b) = \sum_{(s,t)\in\mathcal{C}^v} [(x_s - x_t)^2(1 - b^v_{st}) + \gamma b^v_{st}] + \sum_{(s,t)\in\mathcal{C}^h} [(x_s - x_t)^2(1 - b^h_{st}) + \gamma b^h_{st}] \qquad (7.22)$$

D'une façon intuitive, en l'absence de discontinuités, on a $b_{st} = 0$ et on retrouve un potentiel quadratique visant à lisser la solution. En revanche, en présence de discontinuité ($b_{st} = 1$), la pénalité est arbitraire et vaut γ[4] de façon à limiter le nombre de discontinuités introduites par le modèle. Ce modèle (modèle de la membrane mince -*weak membrane*-) est très simple et n'introduit pas d'interactions entre les éléments du processus bords.

En réalité, il n'est pas nécessaire d'introduire explicitement un processus de bords et un choix de fonction judicieux peut procurer le même résultat. En effet, notons $\phi^v(x_s, x_t, b^v_{st})$ le terme $(x_s - x_t)^2(1 -$

[4] Ce paramètre peut varier pour des modèles anisotropes ou non stationnaires (on a alors γ^v_{st} et γ^h_{st}).

$b^v_{st}) + \gamma b^v_{st}$ (et de même pour ϕ^h), alors on cherche la solution (x, b) telle que $U_2(x, b)$ soit minimale. Or :

$$\begin{aligned}
\min_{(x,b)} U_2(x,b) &= \min_{(x,b)} \sum_{(s,t)\in\mathcal{C}^v} \phi^v(x_s, x_t, b^v_{st}) + \sum_{(s,t)\in\mathcal{C}^h} \phi^h(x_s, x_t, b^h_{st}) \\
&= \min_{x} \sum_{(s,t)\in\mathcal{C}^v} \min_{b^v_{st}} \phi^v(x_s, x_t, b^v_{st}) + \sum_{(s,t)\in\mathcal{C}^h} \min_{b^h_{st}} \phi^h(x_s, x_t, b^h_{st}) \\
&= \min_{x} \sum_{(s,t)\in\mathcal{C}^v} \psi(x_s - x_t) + \sum_{(s,t)\in^h} \psi(x_s - x_t)
\end{aligned}$$

en notant :

$$\psi(u) = \min(u^2, \gamma)$$

Par conséquent, la quadratique est remplacée dans ce modèle par une quadratique tronquée (figure 7.7). L'utilisation d'un processus de bords explicite est dans ce cas équivalente à l'utilisation de la fonction ψ (on parle de processus de bords implicite). La valeur de γ détermine à partir de quelle valeur du gradient on introduira une discontinuité.

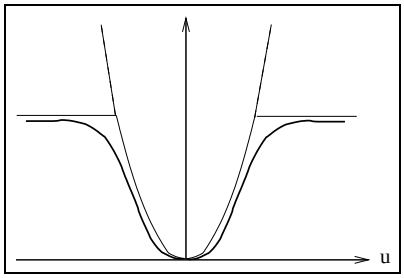


FIG. 7.7 – Fonctions quadratique, quadratique tronquée et la fonction de régularisation proposée par Geman et McClure préservant les discontinuités.

Un très grand nombre de fonctions de régularisation préservant les discontinuités ont été proposées et étudiées (on pourra se référer à [Charbonnier et al., 1996] pour une étude comparative). La fonction suivante, qui présente l'intérêt d'être dérivable partout, est très utilisée en restauration [Geman et McClure, 1985] :

$$\phi(u) = \frac{u^2}{1+u^2}$$

Notons qu'il est intéressant d'utiliser des fonctionnelles convexes pour la recherche de la solution, mais la prise en compte des discontinuités repose sur le comportement de $\frac{\phi'(u)}{2u}$.

Le choix d'un processus de bords implicite ou explicite va donner lieu à différents algorithmes de minimisation. En effet, le recuit simulé permettant d'accéder au minimum global de l'énergie est un algorithme long et coûteux. Il peut dans de nombreux cas être remplacé par un algorithme déterministe pour accélérer la recherche de la solution.

7.4.2 Algorithmes de minimisation

Nous donnons ici le principe des quelques algorithmes déterministes :

– GNC [Blake et Zissermann, 1987] : Blake et Zisserman ont proposé un algorithme déterministe appelé le GNC (Graduated Non Convexity) pour l'optimisation du critère (sous sa forme implicite). Le principe de l'algorithme consiste à approximer le critère par une fonction convexe qui permet

de définir une bonne solution initiale (calculée par une descente de gradient par exemple, puisque la solution est dans ce cas unique). Puis le critère est modifié graduellement perdant sa propriété de convexité pour se rapprocher du critère initial. À chaque étape une solution est trouvée de façon déterministe en utilisant pour initialisation la solution donnée par l'étape précédente. Si des preuves de convergence peuvent être obtenues dans certains cas particuliers, cette démarche n'assure cependant pas de trouver le minimum global pour toutes les fonctions d'énergie.

- MFA [Geiger et Girosi, 1991] : une autre approche est le recuit par champ moyen MFA (Mean Field Annealing) qui utilise cette fois un processus bords explicite. L'algorithme met en œuvre une descente de température au cours de laquelle à chaque étape une solution (image restaurée et processus de bords) est estimée au sens du champ moyen. On a des expressions explicites pour le processus de bords, tandis que x est estimée de façon itérative.
- Artur et Legend [Charbonnier, 1994] : Charbonnier a également proposé deux algorithmes déterministes appelés Artur et Legend pour l'optimisation. Il utilise un processus de bords explicite et exploite le fait qu'à b fixé, le critère est quadratique en x (donc convexe et minimisable rapidement par une descente de type gradient), tandis qu'à x fixée, il existe une expression analytique du processus de bords.

Malgré de nombreuses différences (notamment sur les propriétés de convergence et sur leur généralité), il s'agit toujours d'algorithmes itératifs minimisant des suites d'énergies (GNC ou MFA) ou faisant varier les variables auxiliaires (le processus de bords pour Artur et Legend) au cours des itérations. Une comparaison est effectuée dans [Charbonnier, 1994] et [Zerubia et Chellapa, 1993].

7.5 Quelques applications des champs markoviens

Cette partie présente quelques applications des champs markoviens en traitement d'images pour illustrer les potentialités de ce domaine. On s'intéressera dans un premier temps à des applications de bas niveau (analyse de textures et segmentation) avant de présenter des applications manipulant des graphes construits à partir de primitives plus complexes (régions, segments).

7.5.1 Applications sur le graphe des pixels

Cette première partie présente trois applications de bas-niveau, i.e qui travaillent sur le graphe des pixels :

- La première est une analyse de textures s'appuyant sur un modèle markovien ; elle permet de discriminer différentes textures de l'image à l'aide des paramètres du champ extraits ; un exemple d'applications pour la détection des zones urbaines en imagerie satellitaire est donné ;
- La seconde est un simple exemple de segmentation appliquée en imagerie radar mettant en évidence la souplesse de ce modèle pour prendre en compte des statistiques très variées dans les images et détaillant quelques aspects pratiques de segmentation ; on se placera dans un cadre supervisé pour le paramètre de régularisation, c'est à dire que celui-ci sera fixé de façon empirique ;
- La troisième application présentée est un schéma de fusion markovien, relativement général qui permet de combiner plusieurs sources d'informations ; un exemple est donné dans le cas de l'analyse d'images satellitaires NOAA dans plusieurs bandes spectrales.

Analyse de textures

Nous nous intéressons dans cette partie à l'utilisation des modèles markoviens pour discriminer différents types de textures dans les images. L'idée est de se placer dans le cas de données complètes, i.e de faire l'hypothèse que l'image dont nous disposons est la réalisation d'un champ markovien, et d'extraire

les paramètres de ce champ. La variabilité des paramètres en fonction du type de textures devant alors permettre de réaliser une classification de l'image.

Nous avons déjà vu au paragraphe 7.3.2 un certain nombre de techniques (méthode des codages, maximum de pseudo-vraisemblance, gradient stochastique) pour calculer les paramètres d'un champ markovien. Toutes ces méthodes, qui ne présupposent pas de modèle pour le champ, sont assez lourdes à mettre en œuvre. Nous allons supposer ici que nous nous situons dans le cadre d'un champ markovien gaussien pour lequel nous pouvons obtenir une expression exacte des paramètres. Dans le cas d'un champ markovien gaussien, nous pouvons écrire l'énergie $U(x)$ sous la forme :

$$U = \frac{1}{T}\left(\lambda \sum_{s\in S}(x_s-\mu)^2 + \sum_{c=(s,t)\in\mathcal{C}}(x_s-x_t)^2\right)$$

Avec cette écriture, les paramètres caractérisant le champ et que nous cherchons à estimer sont donc la température T (correspondant à une sorte de "variance généralisée"), la moyenne μ et la pondération du terme d'attache aux données de l'énergie, λ.

La méthode des queues de comètes La méthode que nous allons voir a été proposée dans [Descombes, 1993]. Considérons la probabilité conditionnelle en un site :

$$P(X_s = x_s \ / \ x_t, t\in\mathcal{V}_s) = \frac{1}{Z(V_s)}\exp\{-\frac{1}{T}\left(\lambda(x_s-\mu)^2 + \sum_{t\in\mathcal{V}_s}(x_s-x_t)^2\right)\}$$

En notant n le nombre de sites voisins de s, et m_s la moyenne locale des niveaux de gris des voisins du site s, $m_s = \dfrac{\sum_{t\in\mathcal{V}_s} x_t}{n}$, on peut montrer qu'on a :

$$\begin{aligned} P(X_s = x_s \ / \ x_t, t\in\mathcal{V}_s) &= \frac{1}{Z(m_s)}\exp\{-\frac{n+\lambda}{T}\left(x_s - \frac{1}{n+\lambda}(nm_s+\lambda\mu)\right)^2\} \\ &= P(X_s = x_s \ / \ m_s) \end{aligned}$$

On a donc remplacé le conditionnement local par l'ensemble des voisins de s, par une seule variable conditionnante m_s, ce qui améliorera la robustesse des estimateurs. En effet, le nombre de sites concernés par un conditionnement par m_s sera bien plus grand que celui des sites concernés par une configuration $V_s = (x_t, t\in\mathcal{V}_s)$ du voisinage.

Nous constatons que $P(X_s \ / \ m_s)$ est une loi gaussienne, de moments (moyenne et variance) suivants :

$$\begin{aligned} \mathbf{E}(X_s \ / \ m_s) &= \frac{nm_s+\lambda\mu}{n+\lambda} \\ \mathrm{var}(X_s \ / \ m_s) &= \frac{T}{2(n+\lambda)} \end{aligned}$$

L'espérance et la variance de X_s conditionnellement à m_s étant fonction des paramètres λ, T et μ que nous recherchons, il nous suffit d'estimer ces moments de façon empirique. La variance ne dépendant pas de la valeur conditionnante m_s, on notera le moment théorique par σ^2. Pour faire ces estimations, on construit l'image dite des "queues de comètes" en raison de son aspect, qui est simplement l'image des fréquences normalisées des niveaux de gris x_s conditionnellement à la moyenne m_s du voisinage. Si on met les valeurs de m_s en colonne et celles de x_s en ligne, chaque ligne de l'image des queues de comètes représente $P(X_s \ / \ m_s)$, i.e une gaussienne de moyenne $\dfrac{nm_s+\lambda\mu}{n+\lambda}$ et de variance $\dfrac{T}{2(n+\lambda)}$ (figure **??**).

On considère alors pour la variance, l'estimateur suivant :

$$\widehat{\sigma^2} = \sum_{m_s} P(m_s)\widehat{\mathrm{var}}(X_s \ / \ m_s)$$

en notant $\widehat{\mathrm{var}}(X_s \ / \ m_s)$ l'estimation empirique de la variance faite selon une ligne des queues de comètes. Cet estimateur permet d'accorder à chaque probabilité conditionnelle locale une importance proportionnelle au nombre d'échantillons qui la constituent et entraîne une plus grande robustesse de l'estimation.

En ce qui concerne l'espérance, nous avons la relation suivante :

$$\begin{aligned} \mathbf{E}(X_s \ / \ m_s) &= \frac{n}{n+\lambda} m_s + \frac{\mu}{n+\lambda} \\ &= \alpha m_s + \beta \end{aligned}$$

Par conséquent, les espérances des probabilités conditionnelles se situent sur une droite. L'estimation empririque des moyennes $\widehat{\mathbf{E}}(X_s \ / \ m_s)$ permet de faire une estimation aux moindres carrés des paramètres α et β de la droite. Une fois estimés empiriquement $\widehat{\sigma^2}$, $\widehat{\alpha}$, $\widehat{\beta}$, on peut déduire λ, μ, et T par les relations :

$$\begin{aligned} \widehat{\sigma}^2 &= \frac{\widehat{T}}{2(n_s + \widehat{\lambda})} \\ \widehat{\alpha} &= \frac{n_s}{n_s + \lambda} \\ \widehat{\beta} &= \frac{\widehat{\lambda}\widehat{\mu}}{n_s + \widehat{\lambda}} \end{aligned}$$

En explicitant l'estimation aux moindres carrés des paramètres de la droite, on peut exprimer directement λ, μ, et T en fonction des moments d'ordre 1 et 2, conditionnés ou non. On retrouve dans ce cas pour μ, la moyenne empirique des x_s.

Application à la détection des zones urbaines

Cet exemple d'application est tiré de la thèse de X. Descombes [Descombes, 1993].

On peut utiliser les résultats précédents pour analyser les textures présentes sur une image satellitaire SPOT. L'image étant par essence non stationnaire (sinon l'analyse aurait peu d'intérêt !), le calcul des paramètres se fait localement, sur une fenêtre glissante centrée en chaque pixel. La fiabilité des estimateurs s'accroît avec la taille de la fenêtre, en même temps, et de façon antagoniste, que le risque de considérer des mélanges de textures différentes à l'intérieur de la fenêtre d'étude. Une solution pour remédier à ce problème peut être de ne considérer que les échantillons les plus représentés. En cas de mélange, les échantillons appartiendront à la texture la plus présente dans la fenêtre.

Le paramètre de température, est un bon indicateur du milieu urbain qui se présente sur une image SPOT sous une forme assez texturée, type "poivre et sel" avec alternance de niveaux de gris faibles et forts. En effet, ce paramètre qui mesure en quelque sorte le chahut de la zone, a des valeurs plus élevées dans les régions urbaines. Il permet d'obtenir une bonne discrimination du milieu urbain.

7.5.2 Segmentation

Nous allons aborder dans cette partie une application très classique des champs markoviens qui est la segmentation. Nous commençons par rappeler le principe de la segmentation markovienne en prenant l'exemple des images radar [Tupin, 1997], puis présentons une méthode de fusion dans un cadre markovien [Descombes, 1993].

Segmentation d'une image radar *Cet exemple d'application est tiré de la thèse de F. Tupin [Tupin, 1997].*

La modélisation est similaire à celle qui a été présentée dans le paragraphe 7.1.7. Ecrivons à nouveau les deux termes intervenant dans la probabilité a posteriori. Pour la probabilité du champ des observations conditionnellement au champ des étiquettes, en supposant l'indépendance des pixels, on a :

$$P(Y \ / \ X = x) = \prod_s P(Y_s = y_s \ / \ X_s = x_s)$$

Les images radar sont des images très bruitées par le phénomène de speckle. En revanche, le processus d'acquisition est bien modélisé statistiquement et on a l'expression suivante pour une image radar en amplitude :

$$p(Y_s = y_s \ / \ X_s = i) = \frac{2L^L}{\mu_i^L \Gamma(L)} y_s^{(2L-1)} \exp(-\frac{L y_s^2}{\mu_i})$$

avec L un paramètre du système connu[5] appelé nombre de vues, Γ la fonction Gamma, et μ_i les moyennes en intensité (carré de l'amplitude) des différentes classes i considérées.

Le champ des étiquettes est supposé markovien avec un modèle de Potts qui vise à obtenir des zones homogènes compactes sur l'image segmentée :

$$P(X = x) = \frac{1}{Z} \exp(-\beta \sum_{(s,t) \in \mathcal{C}_2} \phi(x_s - x_t))$$

avec $\beta > 0$, $\phi(0) = 1$ et $\phi(x) = 1 \quad \forall x \neq 0$.

Le champ a posteriori résultant est donc markovien et son énergie s'écrit :

$$\mathcal{U}(x \ / \ y) = L \sum_s (\frac{y_s^2}{\mu_{x_s}} + \ln \mu_{x_s}) + \sum_{c=(s,t)} \phi(x_s - x_t)$$

En choisissant l'estimateur MAP, la solution est obtenue par recuit simulé avec une décroissance géométrique en température, et une température initiale fixée arbitrairement. Le choix des classes se fait de la façon suivante. On se fixe le nombre de classes (15 dans les illustrations ci-dessous) et on applique un algorithme de k-moyennes dont le résultat[6] sert à calculer les valeurs des moyennes en intensité μ_i des différentes classes. Notons que ces classes n'ont pas de contenu sémantique et que la segmentation correspond ici à un "découpage" de l'image.

Le choix de β qui pondère l'influence entre attache aux données et régularisation se fait de façon ad hoc après quelques essais. L'augmentation de β entraîne une augmentation de la taille des zones obtenues par la segmentation. Il serait bien sûr possible d'estimer ce paramètre à l'aide d'une des méthodes décrites au chapitre 3. Notons que ce modèle n'est pas adapté à la préservation des cibles ponctuelles et des lignes qui sont des configurations de forte énergie pour le champ des étiquettes (il faut donc une forte attache aux données pour que ces configurations subsistent dans le résultat final). Il est bien sûr possible de prendre en compte un champ externe pour mieux respecter les lignes par exemple [Tupin et al., 1996] ou d'utiliser un modèle plus approprié [Descombes et al., 1995].

Le tableau ci-dessous 7.2 résume les paramètres utilisés et la figure 7.4 montre le résultat de la segmentation.

[5] Pour les produits PRI du satellite ERS1 $L = 3$.

[6] Le résultat des k-moyennes est très bruité à cause du bruit multiplicatif présent sur les images radars et l'absence de modes dans l'histogramme. Il ne peut donc pas dans ce cas être utilisé directement comme résulat de segmentation.

TAB. 7.2 – Valeurs des paramètres de la segmentation

Température initiale	5
Facteur de décroissance géométrique	0,95
Paramètre de régularisation β	0,4
Nombre de classes	15
Nombre d'itérations pour les k-moyennes	20

Schéma de fusion dans un cadre markovien • Principe du schéma de fusion

Nous nous plaçons maintenant dans le cas où plusieurs sources d'information sont disponibles pour réaliser la classification de l'image. Ces différentes sources peuvent provenir de l'extraction de différents paramètres à partir d'une même image ou directement de plusieurs capteurs.

Notons y le vecteur d'attributs correspondant aux différentes sources d'informations $y = (y^1, ..., y^K)$ avec K le nombre de données (ou canaux) et M le nombre de classes $E = \{\lambda_1, ..., \lambda_M\}$. La probabilité a posteriori s'écrit :

$$p(X \ / \ Y = y) \propto p(Y \ / \ X)p(X)$$

Si nous faisons l'hypothèse que les sources sont **indépendantes** entre elles, et les pixels de chaque source entre eux, on a :

$$\begin{aligned} p(Y \ / \ X = x) &= \prod_{s \in S} P(Y_s \ / \ X_s = x_s) \\ &= \prod_{s \in S} P(\{Y_s^1, Y_s^2, ..., Y_s^K\} \ / \ X_s = x_s) \\ &= \prod_{s \in S} P(Y_s^1 \ / \ X_s = x_s)...P(Y_s^K \ / \ X_s = x_s) \\ &= \prod_{s \in S} \prod_{k=1}^{K} P(Y_s^k \ / \ X_s = x_s) \end{aligned}$$

Le potentiel d'attache aux données résultant s'exprime alors sous forme d'une somme des potentiels individuels de chaque source :

$$U_s(y_s = (y_s^k)_k \ / \ x_s = \lambda) = \sum_k U_s(y_s^k \ / \ \lambda)$$

Les différentes sources ne renseignant pas de la même façon sur toutes les classes, il est possible d'introduire des coefficients de pondération exprimant la confiance (la fiabilité) qu'on veut accorder à chacune des sources par rapport à une classe. On notera $\gamma_{(k,\lambda)}$ la "confiance" accordée à la source k pour la classe λ. Il faut par ailleurs ne favoriser aucune classe, donc les coefficients doivent vérifier $\sum_k \gamma(k, \lambda) = 1 \ \ \forall \lambda$. L'expression de l'attache aux données s'écrit alors :

$$U_s(y_s = (y_s^k)_k \ / \ x_s = \lambda) = \sum_k \gamma_{(k,\lambda)} U_s(y_s^k \ / \ \lambda)$$

En pratique, il n'est pas toujours nécessaire de faire des modélisations statistiques compliquées pour calculer les potentiels d'attache aux données. Souvent des potentiels très simples, linéaires par morceaux,

permettent d'obtenir de bons résultats. On attache un potentiel faible (typiquement 0) à la plage de niveaux de gris qui correspond à la classe considérée et un potentiel élevé (typiquement 1) ailleurs. Cette définition peut se faire de façon supervisée en analysant l'histogramme ou de façon automatique par une recherche des modes de l'histogramme de l'image (analyse multi-échelle [Aurdal, 1997], recuit simulé sur l'histogramme [Bloch et Maitre, 1997], etc.).

La définition des coefficients de fiabilité des sources est souvent plus problématique. On se limite en général à des remarques de bon sens en affectant $\gamma_{k,\lambda} = 0$ lorsque la source k n'est pas significative pour la classe λ, 0.5 si l'information délivrée est approximative, et 1 si la source est pertinente (avant normalisation).

• **Application à l'analyse des images SPOT**

Cet exemple d'application est tiré de la thèse de X. Descombes [Descombes, 1993]. On trouvera d'autres exemples dans [Tupin et al., 1996] [Aurdal, 1997].

Nous décrivons ici l'application de ce schéma à l'analyse de plusieurs canaux délivrés par le satellite NOAA. Il s'agit de 5 canaux de basse résolution (1.1km) correspondant aux domaines visible, proche infra-rouge, moyen infra-rouge et 2 canaux thermiques, montrés sur la figure **??**. Les classes qu'on cherche à discriminer sont les suivantes : mer, nuages, continent sans relief, continent avec relief et icebergs.

Une première étape consiste à définir les "sources" que nous allons utiliser. La méthodes des queues de comète que nous avons présentée plus haut permet en effet de déduire de chaque image, 3 images de paramètres (moyenne locale, température et paramètre d'attache aux données λ). Au total 20 images sont donc disponibles, dont les plus significatives pour notre objectif, sont sélectionnées : l'image 1 (visible), l'image de température et l'image de moyenne associées, l'image 3 (moyen infra-rouge) et l'image de température associée (soit 5 en tout).

Pour chacune de ces images, une analyse supervisée de l'histogramme est effectuée, permettant de définir les potentiels (linéaires par morceaux) et la pertinence du canal pour chaque classe. Le terme d'attache aux données de l'énergie a posteriori est alors défini comme mentionné précédemment. Quant au terme de régularisation, les classes recherchées étant relativement compactes, on utilise un modèle de Potts.

7.6 Applications sur des graphes de primitives

Comme nous l'avons mentionné au chapitre **??**, le formalisme markovien est défini sur tout graphe et son champ d'applications est donc bien plus vaste que la simple grille des pixels. De nombreux problèmes se prêtent à la manipulation de primitives plus complexes, soit pour des raisons de rapidité comme dans le premier exemple que nous décrivons ci-dessous, soit parce qu'il s'agit d'un problème plus proche de l'interprétation d'image, difficile à traiter au niveau du pixel et nécessitant l'introduction d'informations de haut niveau comme dans le second exemple décrit.

7.6.1 Graphes de régions : application à la segmentation d'une image d'IRM cérébrale

Cet exemple d'application est tiré de la thèse de T. Géraud [Géraud et al., 1995].

Pour accélérer la segmentation des images, plusieurs applications partent d'une sur-segmentation de l'image qui est ensuite améliorée en fusionnant les régions. Cette fusion, qui procure la segmentation finale, peut se faire dans un cadre markovien. Le graphe est construit à partir des régions de la sur-segmentation, chaque région correspondant à un sommet du graphe et la relation de voisinage étant définie par la relation d'adjacence entre les régions. Le terme d'attache aux données dépend alors des

attributs de la région (niveau de gris moyen des pixels la constituant, moments d'ordre supérieur, etc.) et le terme de régularisation dépend de l'application, un potentiel de Potts pouvant être utilisé lorsqu'on essaye de trouver des zones relativement compactes.

L'objectif de l'exemple décrit ici est de réaliser une segmentation d'images IRM cérébrales. Les classes considérées sont la matière grise, blanche, le liquide céphalo-rachidien, les ventricules et une classe ASI représentant les autres structures internes (noyaux caudés, thalamus, putamen) qui sont difficiles à segmenter et auxquelles on s'intéresse particulièrement dans cette application. Le nombre de pixels de ces images volumiques ($256 \times 256 \times 128$) limite l'utilisation de méthodes markoviennes en raison du temps de calcul. Par contre, l'utilisation d'un graphe de régions construit à partir d'une sur-segmentation, en réduisant drastiquement le nombre de sites permet de réaliser un recuit simulé à un coût raisonnable.

• **Sur-segmentation :** L'étape de sur-segmentation 3D est réalisée par un algorithme calculant la ligne de partage des eaux sur l'image du gradient après fermeture morphologique pour réduire le nombre de bassins. L'image résultat est constituée de zones de niveaux de gris homogènes, auxquelles on associe les attributs suivants : volume (noté vol), niveau de gris moyen (qui sera l'observation y du champ des données), coordonnées du centre du bassin. Le résultat de cette méthode appliquée à l'image 7.8.a est montré sur la figure 7.8.b.

• **Relaxation markovienne :** Un graphe est construit comme indiqué précédemment à partir de la sursegmentation (fig. 7.8.c). On associe aux arcs du graphe la surface d'adjascence entre les deux régions (notée surf). Les potentiels du champ markovien sont alors définis comme suit :

– Terme d'attache aux données :

$$P(Y_s \ / \ x_s) = \exp\left(-\sum_s \frac{\mathrm{vol}_s}{2\sigma^2_{x_s}}(y_s - \mu_{x_s})^2\right)$$

(déduit d'une étude statistique des classes [Géraud et al., 1995])

– Terme de régularisation :

$$U_{c=(s,t)}(x) = \mathrm{surf}_{s,t} Q_{(x_s,x_t)}$$

Seuls les potentiels des cliques d'ordre 2 sont choisis non nuls. La matrice Q est une matrice d'adjascence permettant de pondérer les voisinages entre classes suivant qu'ils sont favorisés ou non [Sigelle, 1993].

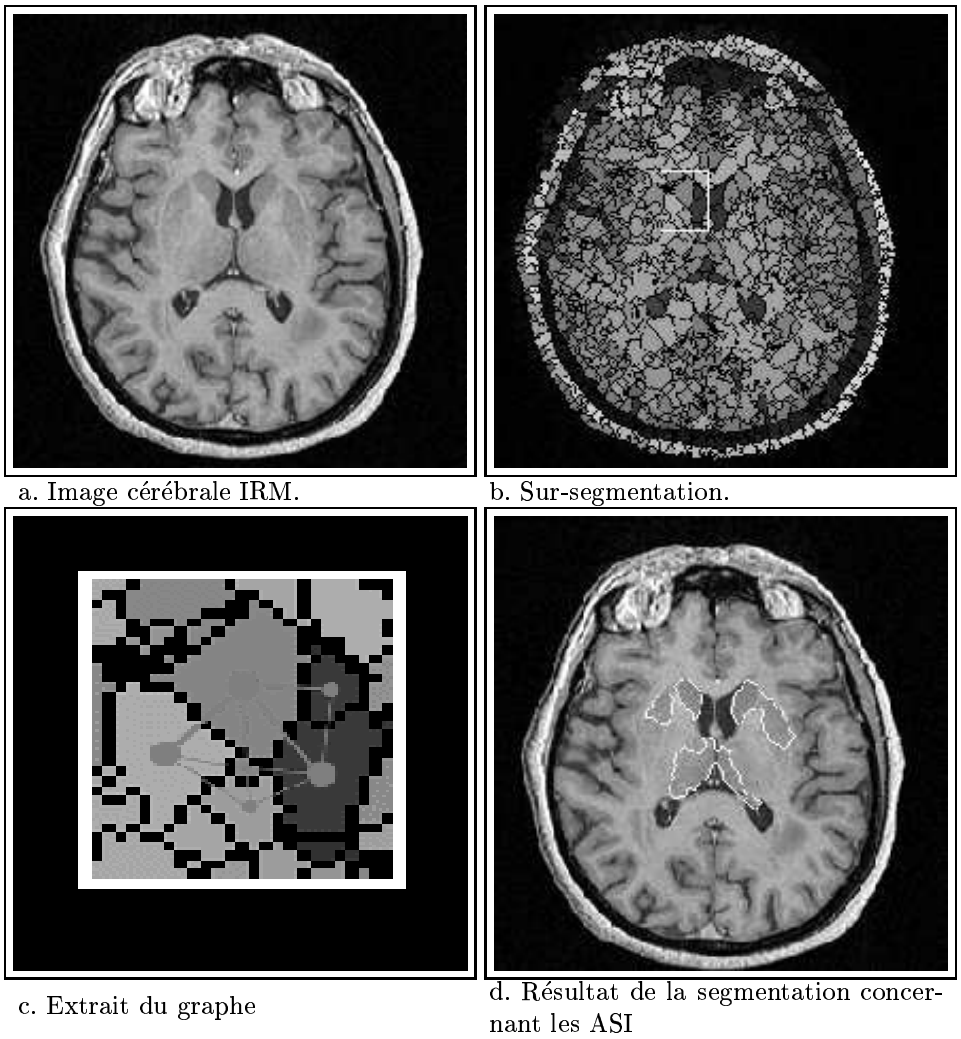
Le critère utilisé est un critère MAP et la solution est obtenue par un recuit simulé. Le temps de calcul est de moins d'une minute pour un graphe de 32 000 sites. Les résultats pour les ASI sont montrés sur la figure 7.8.c.

7.6.2 Graphes de segments : application à la détection du réseau routier

Cet exemple d'application est tiré de la thèse de F. Tupin [Tupin, 1997].

Nous nous intéressons dans cette application à la détection du réseau routier et hydrographique dans le cas des images radar. Le bruit de speckle présent sur ces images ne permet pas d'obtenir de très bons résultats de bas niveau et les détecteurs de lignes même adaptés à ce type d'imagerie ont des taux de fausses alarmes élevés si on veut obtenir des taux de détection suffisants.

L'idée est donc de faire suivre l'étape de bas niveau de détection des lignes par une étape de plus haut niveau, dans laquelle on injectera des informations a priori sur la forme des routes. Le cadre markovien, par l'intermédiaire du terme a priori se prête bien à l'introduction de connaissances sur les objets recherchés, à condition que celles-ci puissent s'exprimer de façon locale à l'échelle du graphe. Dans le cas du réseau cette hypothèse (qui assure que le champ soit markovien) est vérifiée, puisse que la pratique montre qu'il nous suffit d'informations locales au niveau des segments pour prendre une décision (présence ou absence de réseau routier).



a. Image cérébrale IRM.

b. Sur-segmentation.

c. Extrait du graphe

d. Résultat de la segmentation concernant les ASI

FIG. 7.8 – Etapes de la segmentation sur le graphe des régions

La démarche adoptée pour la détection du réseau est donc la suivante. L'étape de bas niveau permet de détecter des segments candidats. Parmi ceux-ci, certains appartiennent aux objets à détecter, quand d'autres sont de fausses détections. On fait alors l'hypothèse que les segments détectés et toutes les connexions possibles entre ces segments contiennent le réseau routier. Les connexions "possibles" ne sont pas toutes les connexions, mais les connexions raisonnables : entre des segments suffisamment proches, et à peu près "alignés" par exemple. L'ensemble de ces segments (ceux détectés et les connexions) consituent les sommets du graphe (voir figure 7.9). La relation de voisinage entre deux segments est définie par le partage d'une extrémité par ces 2 segments. Les figures ci-dessous illustrent les étapes de la construction du graphe sur un extrait d'image radar (fig. 7.12).

Une fois ce graphe construit, on définit un champ d'observation associé Y et un champ d'étiquettes X (les étiquettes étant simplement 0 pour "non-route" et 1 pour "route") dont on cherche la configuration optimale au sens du critère MAP.

Les termes de l'énergie sont définis de la façon suivante :

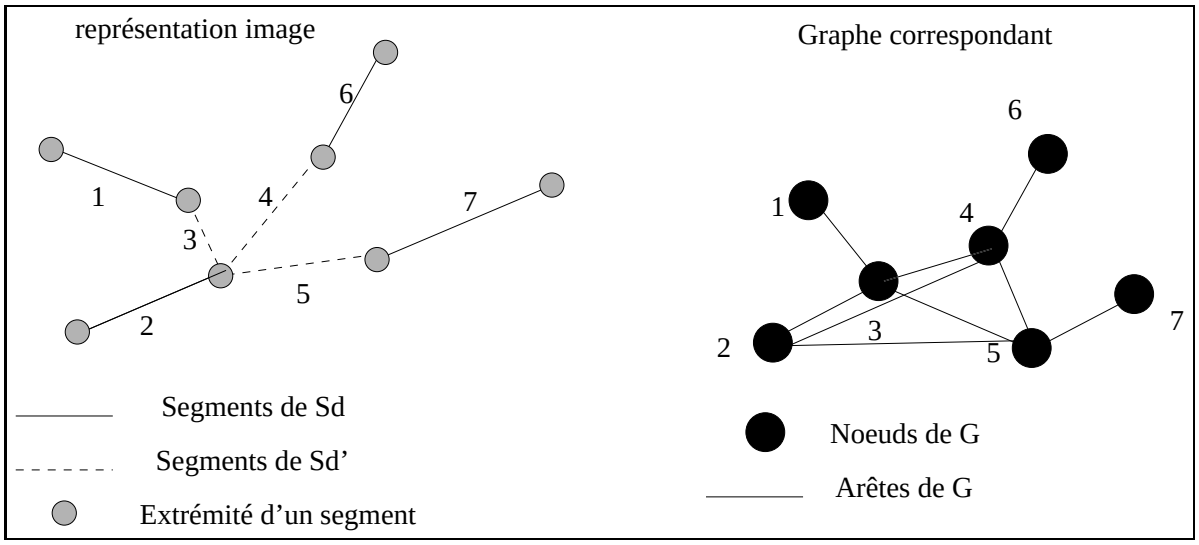
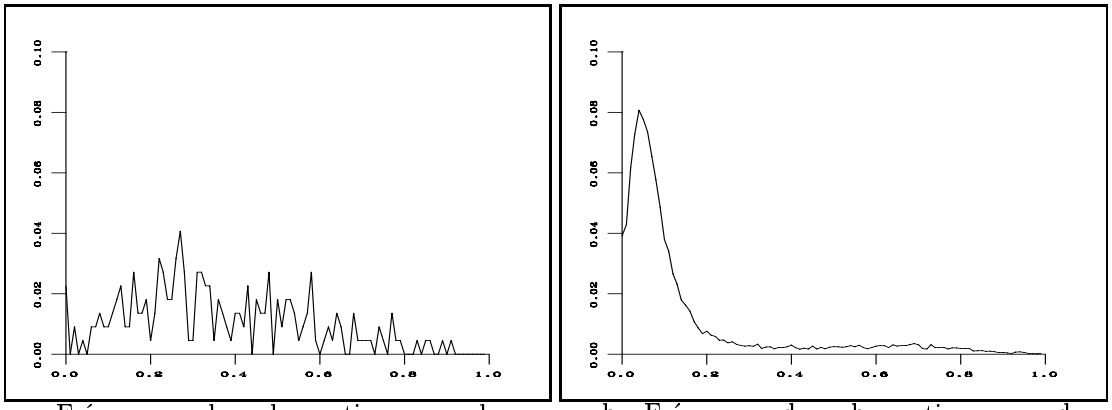


FIG. 7.9 – Construction du graphe de segments

– Terme d'attache aux données : l'observation en un site est définie comme la réponse moyenne du détecteur de lignes le long de ce segment ; les potentiels $U_s(y_s \ / \ x_s)$ pour les labels 0 et 1 sont alors obtenus par une étape d'apprentissage le long de quelques segments appartenant à une vraie route et de quelques segments de "fausse alarme" ; la figure 7.10 montre les fréquences des observations y (approximant les $P(y_s \ / \ x_s)$) et les potentiels linéaires par morceaux qui en sont déduits (fig. 7.11) ;



a. Fréquences des observations sur des segments-routes détectés manuellement (approximant $P(Y_s \ / \ X_s = 1)$).

b. Fréquence des observations sur des segments-non routes (approximant $P(Y_s \ / \ X_s = 0)$).

FIG. 7.10 – Fréquences conditionnelles des observations sur une zone test.

– Énergie a priori : elle permet justement d'intégrer toutes les connaissances a priori qu'on veut prendre en compte pour la détection du réseau, par exemple du type :
  - (i) les routes sont longues et, dans l'absolu, elles ne s'arrêtent pas ;
  - (ii) elles ont une courbure relativement faible ;
  - (iii) un segment de route est plus souvent connecté en une extrémité à un unique segment de route qu'à plusieurs.

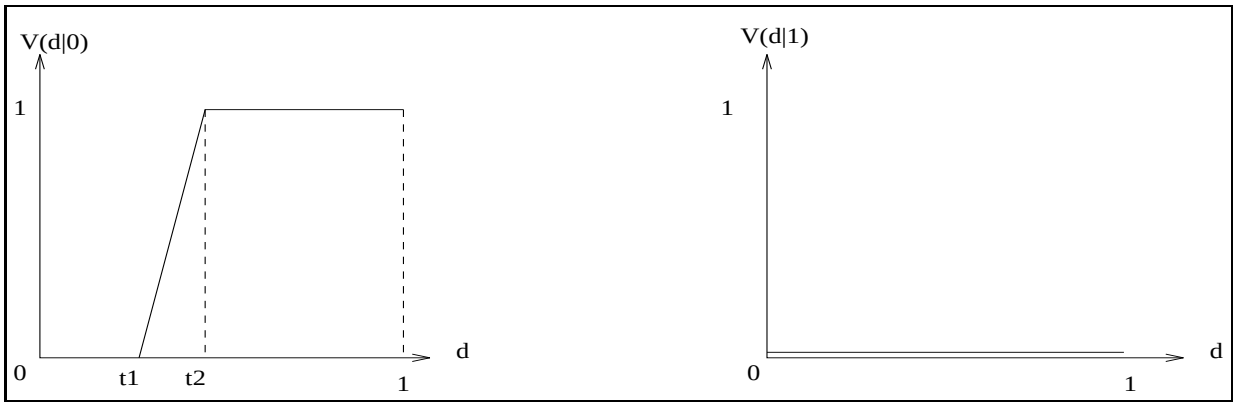


FIG. 7.11 – Potentiels linéaires par morceaux utilisés

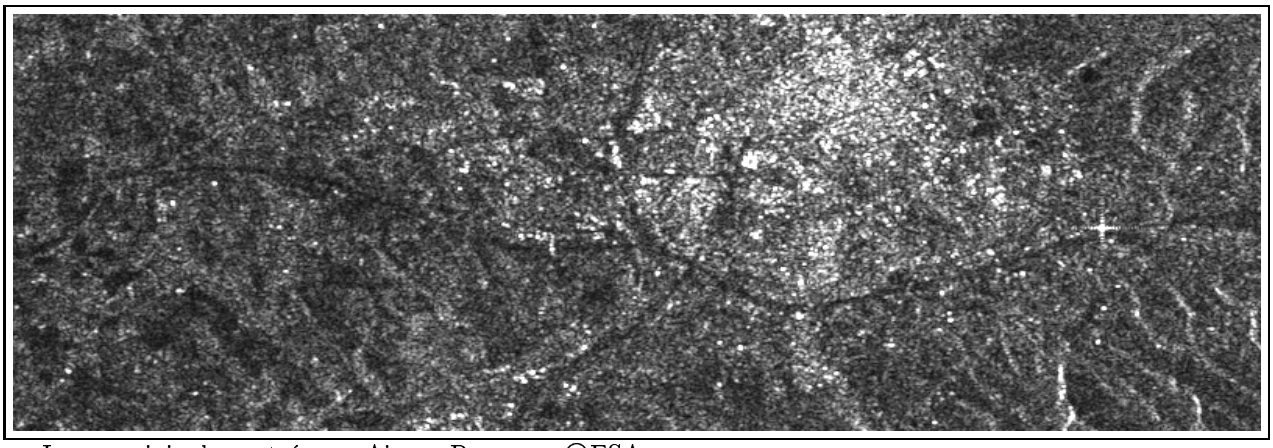
Ces a priori peuvent s'exprimer en définissant les potentiels des cliques d'ordre maximal (rappelons qu'une clique est un ensemble de sites tous voisins les uns des autres, donc dans notre cas, un ensemble de segments qui se rejoignent en un même point) ; quatre paramètres suffisent pour les contraintes mentionnées précédemment : un paramètre contrôlant les extrémités (qui vise à défavoriser la configuration d'une clique où un seul segment a le label "route") ; deux paramètres contrôlant la longueur des routes (qui visent à favoriser des configurations où il y deux segments routes qui se joignent en une extrémité et qui sont "alignés") ; un paramètre contrôlant les carrefours (qui vise à défavoriser les configurations où une multitude de segments sont "route").
L'apprentissage de ces paramètres se fait difficilement par les méthodes du chapitre 3. En revanche l'étude de configurations extrêmes (chaîne de segments tous à "route", etc.), plus connue sous le nom de "Boîtes qualitatives d'Azencott" [Azencott, 1992], permet de fixer des intervalles de valeurs pour ces paramètres.

Un exemple de résultat obtenu par recuit simulé est montré sur la figure 7.13.

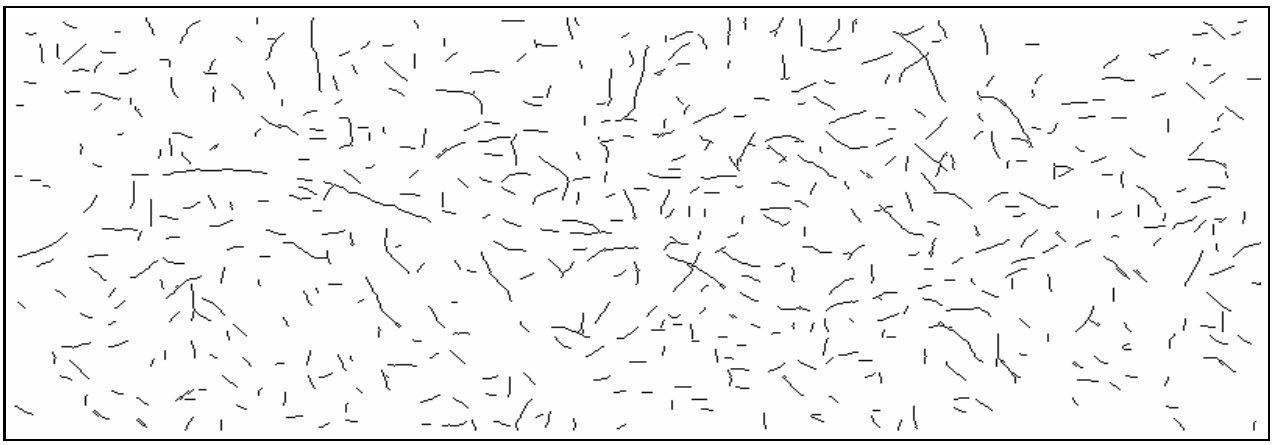
7.6.3 Conclusion

Ce chapitre ne se veut pas une présentation exhaustive des applications possibles des champs markoviens, mais présente quelques unes de leurs potentialités dans des domaines et à des niveaux de traitement d'image variés. Leur grande souplesse permet en effet d'introduire toutes sortes d'informations, que ce soit pour le terme d'attache aux données, que ce soit pour les a priori possibles, ou même encore sur la forme du graphe à utiliser.

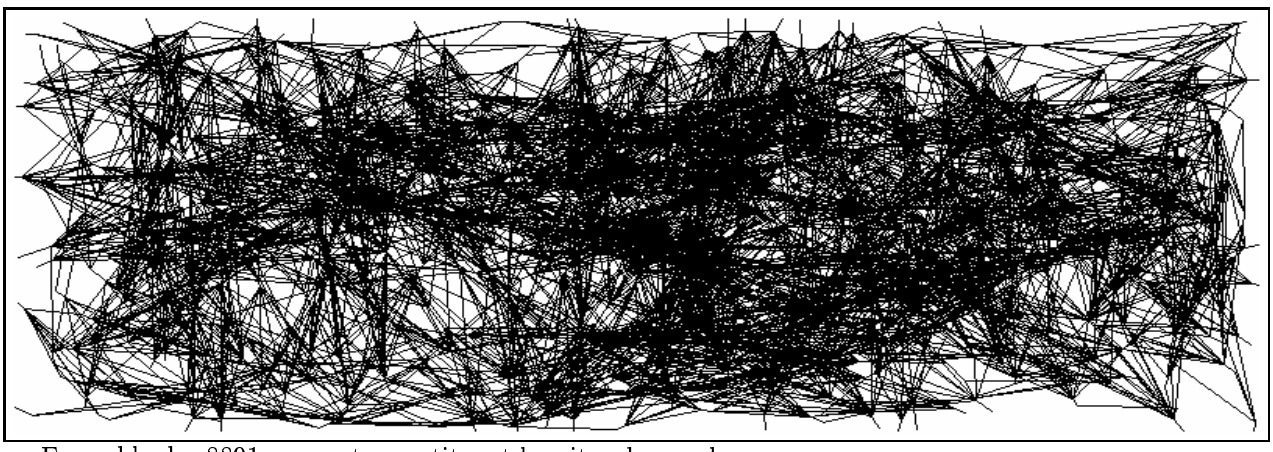
A côté de la multitude d'applications qui peuvent trouver une solution dans un cadre markovien, s'ouvrent des champs de recherche plus théoriques sur l'estimation des paramètres et l'accélération des techniques de recherche de solution.



a. Image originale centrée sur Aix en Provence ©ESA .



b. Ensemble des 839 segments détectés.



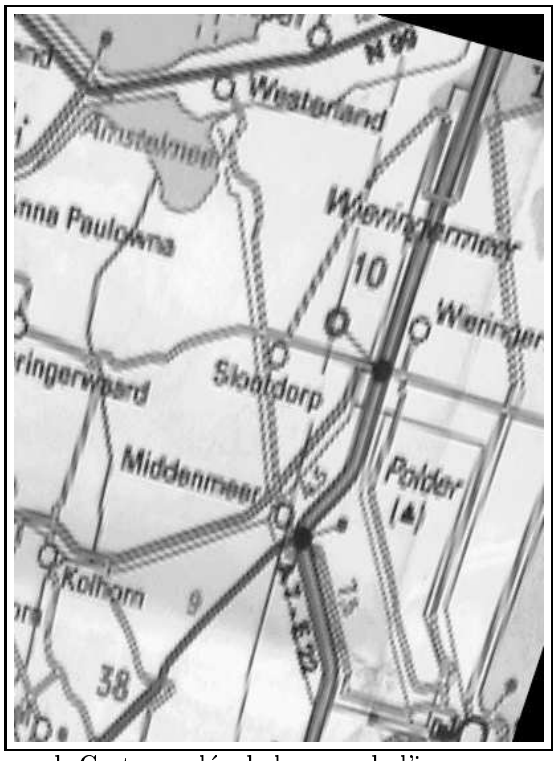
c. Ensemble des 8891 segments constituant les sites du graphe.

FIG. 7.12 – Les étapes de la construction du graphe

a. Image originale (Wieringermeer) ©ESA.

b. Image des segments détectés.

c. Résultat de la détection markovienne.



d. Carte recalée de la zone de l'image.

FIG. 7.13 – Détection des routes sur le Wieringermeer

Chapitre 11

La détection des contours dans les images

Chapitre rédigé par Henri MAÎTRE

La détection de contours dans les images a débuté de façon extrêmement empirique par des opérateurs locaux qui, soit estimaient un gradient, soit convoluaient l'image par des masques caractéristiques des contours [Haralick et Shapiro, 1985]. Dans les années 80, des approches plus systématiques ont été mises en place par Marr [Marr et Hildreth, 1980], puis Canny [Canny, 1986], pour obtenir des contours plus significatifs. Ces travaux ont abouti à une bonne compréhension de ce qu'il faut faire pour détecter les contours, mais la définition même des contours demeure très vague, ce qui rend ces techniques encore peu efficaces sur un problème concret. De plus, on a pu montrer que le problème de détection de contours est généralement mal posé (au sens de la résolution des systèmes) [Torre et Poggio, 1986]. Les seuls modèles de contours utilisables sont ceux de contours idéalisés, comme ceux présentés sur la figure 11.1 ; ils sont bien loin de la réalité. C'est pourquoi, même si de très gros progrès ont été accomplis dans ce domaine, les techniques empiriques d'estimation du gradient proposées dans les années 70-80 restent souvent encore employées en concurrence de techniques plus modernes. Une excellente référence à ce problème est l'ouvrage collectif [Cocquerez et Philipp, 1996]. Plusieurs articles existent sur la comparaison des performances de détecteurs de contours, en particulier [Palmer et al., 1996, Heath et al., 1997].

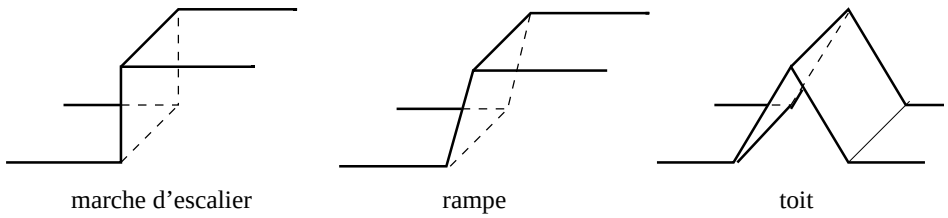


FIG. 11.1 – Quelques modèles de contours. Le plus utilisé est celui en marche d'escalier.

Nous présenterons tout d'abord une approche formelle (mais malheureusement stérile !) de la détection des contours, puis les méthodes empiriques encore utilisées seront vues dans la partie 2. Une approche plus analytique, proposée par Canny, sera ensuite présentée, avec ses dérivées qui sont aujourd'hui les plus employées en raison de leur efficacité. Enfin les techniques fondées sur le principe des **contours actifs** de Kass, Witkins et Terzopoulos seront présentées, ainsi que les approches par **ensembles de niveaux** (*level sets*). Nous terminerons par une brève présentation des méthodes de poursuite et de fermeture des

contours.

11.1 Un modèle continu de contour.

Dans une image supposée continue $f(x,y)$, un contour apparaît comme une ligne où sont localisées les très fortes variations de f. Soit $\overrightarrow{G}$ le gradient de f :

$$\overrightarrow{G} = \overrightarrow{\nabla f} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

On associe à f une image du module du gradient de f :

$$G = |\overrightarrow{\nabla f}| = \left[\left(\frac{\partial f}{\partial x} \right)^2 + \left(\frac{\partial f}{\partial y} \right)^2 \right]^{1/2} \tag{11.1}$$

ainsi qu'une image des directions du gradient à partir de :

$$\overrightarrow{g} = \frac{\overrightarrow{\nabla f}}{|\overrightarrow{\nabla f}|}$$

et par application de $[-\pi, \pi[$ sur $\{0, 255\}$. Un contour est alors défini comme **le lieu des maximums du gradient dans la direction $\overrightarrow{g}$ du gradient**. Un point de contour vérifie donc ;

$$\frac{\partial G}{\partial g} = 0 \qquad \text{et} \qquad \frac{\partial^2 G}{\partial g^2} \leq 0 \tag{11.2}$$

avec :

$$\frac{\partial}{\partial g} = \overrightarrow{g}.\overrightarrow{\nabla} \tag{11.3}$$

L'équation obtenue est complexe et non linéaire :

$$\frac{\partial f}{\partial x}.\frac{\partial}{\partial x}\sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2} + \frac{\partial f}{\partial y}.\frac{\partial}{\partial y}\sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2} = 0$$

Elle n'a donc en général pas de solution explicite et ne peut être résolue que par tâtonnement. On procède alors en deux étapes : on calcule tout d'abord le gradient, puis on recherche les extrémums dans la direction du gradient.

Afin de garantir la double dérivation même en présence de discontinuités de type *marche d'escalier*, on prétraite l'image f par convolution avec une fonction au moins deux fois dérivable. On a beaucoup utilisé pour cela la gaussienne, mais on peut également prendre des polynômes de faible degré.

On peut également se placer dans les axes locaux définis par la tangente $\overrightarrow{t}$ et la normale $\overrightarrow{g}$ à la surface $f(x,y)$ (cf. figure 11.2). Ce repère est tourné d'un angle Φ par rapport au repère $\{x, y\}$:

$$\Phi = Arctg\left[\frac{\partial f}{\partial x} / \frac{\partial f}{\partial y}\right] \tag{11.4}$$

On a alors :

$$\frac{\partial f}{\partial g} = \frac{\partial f}{\partial x}\cos\Phi + \frac{\partial f}{\partial y}\sin\Phi$$

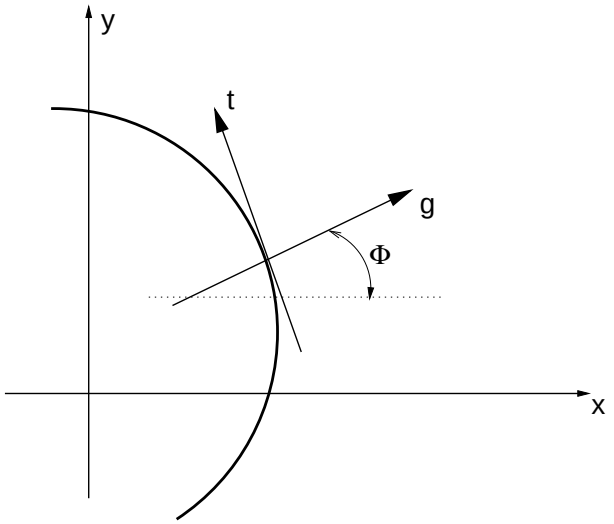


FIG. 11.2 – Repère local défini par le vecteur normal et le vecteur tangent à l'image.

et les équations 11.2 donnent :

$$\frac{\partial^2 f}{\partial g^2} = \frac{\partial^2 f}{\partial x^2}\cos^2\Phi + 2.\frac{\partial^2 f}{\partial x \partial y}\cos\Phi\sin\Phi + \frac{\partial^2 f}{\partial y^2}\sin^2\Phi$$

Cette équation n'est pas plus simple à résoudre que la précédente. Cependant elle se relie assez aisément à l'équation du laplacien de f :

$$\Delta f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

qui s'exprime également (par invariance du laplacien par changement de repère euclidien) dans le repère local $\{t, g\}$:

$$\Delta f = \frac{\partial^2 f}{\partial g^2} + \frac{\partial^2 f}{\partial t^2}$$

On voit que si le terme $\frac{\partial^2 f}{\partial t^2}$ est négligeable (c'est-à-dire si le contour a une courbure très faible), alors :

$$\Delta f = 0 \qquad \approx \qquad \frac{\partial^2 f}{\partial g^2} = 0 \tag{11.5}$$

Cette propriété sera utilisée dans les approches dites par passage par zéro du laplacien (cf détecteur de Marr).

11.2 Les approches classiques

Cette section présente un ensemble de méthodes qui ont eu historiquement une grande importance en traitement des images. Bien que certaines soient encore régulièrement employées, un lecteur intéressé par les techniques plus actuelles pourra se rendre directement à la section 11.3.

11.2.1 Les détecteurs de gradient par filtrage

Ces détecteurs reposent tous sur une recherche d'un extremum de la dérivée première (ou d'un passage par zéro d'une dérivée seconde), celle-ci étant calculée de diverses manières, mais généralement par un filtrage passe-haut précédé d'un léger filtrage passe-bas pour s'affranchir des bruits. Ces approches par **filtrage linéaire** sur des critères très simples de sélection en fréquence ont reçu une base théorique dans [Modestino et Fries, 1977, Shanmugam et Green, 1979]. La somme des connaissances sur ce sujet se trouve dans [Torre et Poggio, 1986].

Ainsi, Modestino et Fries ont proposé de détecter les contours par un filtrage de Wiener, le but de ce filtrage étant d'effectuer sur l'image non bruitée une opération de laplacien qui sera suivie par la suite d'une détection de zéros alliée à un seuillage (nous verrons ces étapes ultérieurement).

Le filtre de base de Modestino et Fries est un filtre récursif de la forme :

$$H_0(z_1, z_2) = \frac{\sum_i^n \sum_j^n b_{i,j} z_1^{-i} z_2^{-j}}{1 + \sum \sum a_{i,j} z_1^{-i} z_2^{-j}}$$

où les coefficients $a_{i,j}$ et $b_{i,j}$ sont déterminés à partir de 3 paramètres spécifiques de l'image à traiter : λ, ρ et ϵ. λ est le paramètre poissonnien de densité de contours (il exprime la distance moyenne entre 2 contours lors d'un balayage de l'image), ρ la constante de la loi exponentielle de l'auto-corrélation de l'image (cf. chapitre 2) et ϵ la puissance du bruit. Comme un tel filtre ne filtre que du coin supérieur gauche au coin inférieur droit, on lui associe les 3 autres filtres obtenus par rotation de $\pi/2$, pour obtenir un filtre final. Dans la pratique pour assurer des temps de calcul raisonnables, on choisit des filtres d'ordre 1 ($n = 1$).

11.2.2 Les détecteurs de gradient par masques

A côté de ces approches très inspirées du traitement du signal, des filtres de dérivation plus empiriques ont été proposés à partir d'estimateurs locaux de l'image f ou de ses dérivées $\frac{\partial f}{\partial x}$. Ces estimées sont obtenues à l'aide de masques (cf. figure 11.3) appliqués sur des fenêtres de 2×2 pixels ou 3×3 pixels (exceptionnellement, en cas d'images très bruitées, sur des fenêtres plus grandes). On note sans surprise que la somme des coefficients de ces filtres est nulle (fonction de transfert nulle à la fréquence 0), et que les coefficients sont anti-symétriques. Les filtres les plus utilisés sont, dans l'ordre décroissant :

Sobel > Roberts > Gradient > Prewitt

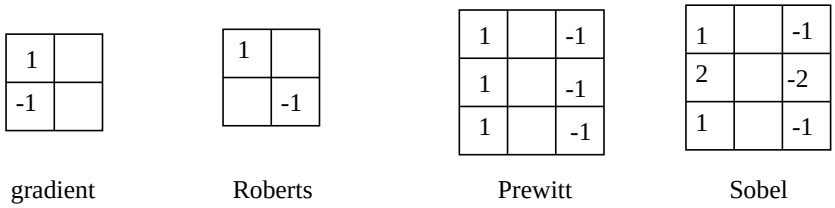


FIG. 11.3 – Quatre filtres de détection de contours par estimation du gradient. Les filtres représentés estiment une seule dérivée. Par rotation de $\pi/2$, on calcule la seconde dérivée.

Les filtres 3×3 sont un peu moins précis (c'est-à-dire que les contours qu'ils détectent sont moins bien localisés et souvent épais), mais les images ainsi obtenues sont généralement plus fiables et permettent des post-traitements plus poussés, ils sont également centrés sur un pixel et non entre des pixels (filtres à phase nulle).

La réponse de l'un quelconque de ces filtres s'obtient de la façon suivante : le filtre est centré en chaque pixel successivement (au coin supérieur gauche pour les filtres 2×2, au point central pour les filtres 3×3). Le produit du masque par les valeurs des pixels correspondant étant fait, la valeur absolue de la somme est retenue. Puis le masque est tourné de 90^o autour de son centre et la même mesure est répétée. Les deux mesures sont alors ajoutées et leur somme constitue la mesure du gradient en ce point selon la formule de l'équation 11.1, ou selon la formule approchée (en norme $\mathcal{L}_1$), un peu plus rapide à calculer :

$$\hat{G} = |\frac{\partial f}{\partial x}| \ + \ |\frac{\partial f}{\partial y}|.$$

On peut également mesurer l'orientation Φ du contour au point donné en faisant le rapport des réponses des deux filtres comme dans l'équation 11.4.

De tels filtres sont particulièrement simples de mise en œuvre, et rapides de calcul, permettent en particulier d'assurer des calculs en temps réel pour des applications industrielles.

11.2.3 Pré- et post traitements

La qualité de la détection est très liée à la qualité des contours dans l'image (cela sera vérifié pour toutes les méthodes, mais particulièrement pour ces méthodes peu élaborées). Ainsi, elles sont souvent bonnes pour les applications de robotique, en environnement artificiel, pour des images aériennes à forte résolution (en milieu urbain par exemple), elles se dégradent très vite pour les scènes d'extérieur, les images médicales, et deviennent totalement inapplicables en imagerie bruitée (thermographie, imagerie ultrasonore, radar SAR, etc.). On a alors intérêt à revenir à des démarches moins intuitives comme celles exposées dans [Modestino et Fries, 1977, Shanmugam et Green, 1979].

Ces détecteurs doivent toujours être suivis d'une étape de post-filtrage, et souvent précédés d'une étape de pré-filtrage.

Parmi les filtres de **pré-traitement**, on recherche ceux qui diminuent le bruit tout en préservant les discontinuités. Ils ont fait l'objet du chapitre **??** : filtre médian par exemple, mais aussi filtre toboggan [Fairfield, 1990] (cf. section 10.3.1), ou filtre de diffusion anisotrope de Perona et Malik [Perona et Malik, 1987] (cf section 10.2.4). Ces filtres ont cependant le défaut de créer des contours artificiels qui peuvent être par la suite difficiles à éliminer.

Les **post-traitements** commencent le plus souvent par l'élimination des points de contour trop faibles par un seuillage (seuillage adaptatif si l'image est très hétérogène). On élimine ensuite les points qui ne sont pas des extrémums locaux dans la direction du gradient. Cela s'obtient à partir de la direction Φ mesurée précédemment et une comparaison simple des points rencontrés (ou interpolés si l'on souhaite une précision meilleure que le pixel). On peut opérer également des seuillages *par hystérésis*, afin de ne conserver que les composantes les plus importantes des contours. Pour cela, on procède à deux seuillages, le second étant très tolérant (c'est-à-dire laissant un très grand nombre de contours candidats) ; on ne garde des contours sélectionnés par ce second seuil que les points connexes d'un point de contour préservé par le premier (cela se fait aisément par un double étiquetage des images seuillées) cf. figure 11.4.

Enfin on procède à des étapes de **poursuite** et de **fermeture** des contours. La première opération a pour objectif de supprimer les petites disparitions de contours qui peuvent se produire par suite du bruit ou d'une perte de contraste, la seconde se propose de détecter des objets topologiquement clos, c'est-à-dire des zones fermées. Ces étapes seront vues plus en détail plus loin.

11.2.4 Les détecteurs de passage par zéro du laplacien

Ces méthodes ont été proposées en 1976 [Marr et Hildreth, 1980]. Elles ont eu une grande importance historique, étant considérées comme le prototype du détecteur de contour inspiré des systèmes biologiques

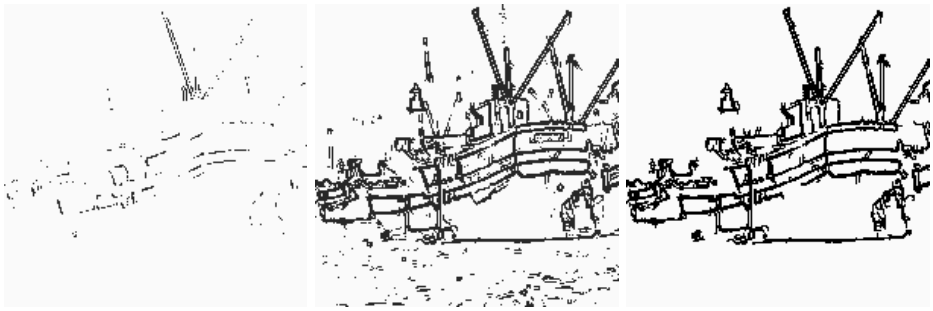


FIG. 11.4 – Filtrage de contours par hystéresis : à gauche, application d'un seuil sévère (70) au gradient de l'image bateau, seuls les contours les plus fiables sont détectés. Au centre, application d'un seuil tolérant (25), beaucoup de contours sont détectés, mais également beaucoup de bruit. A droite, résultat du filtre par hystérésis : seuls les contours du second filtrage connexes à ceux du premier sont conservés. Le détecteur de contours est un détecteur de Sobel. Aucun filtrage n'est appliqué pour réduire l'épaisseur des contours.

(*primal sketch* de Marr). Elles utilisent le fait que le passage par zéro du laplacien permet de bien mettre en évidence les extrémums de la dérivée. Nous avons eu l'occasion de voir (cf. équation 11.5) qu'en l'absence de forte courbure, le passage par zéro du laplacien correspond en effet bien au maximum du gradient dans la direction du gradient. Ces méthodes tirent en outre profit du fait que les zéros de la dérivée seconde constituent un réseau de lignes fermées (évitant donc, en principe, les étapes de poursuite et de fermeture). Il en est de même du réseau des lignes de crête du gradient, mais le premier est plus aisément détecté à partir d'un simple étiquetage des zones positives et négatives. Mais l'estimation de la dérivée seconde étant très sensible aux bruits, il convient de filtrer très fortement l'image avant d'en mesurer le laplacien. Cela conduit au filtrage suivant :

$$\psi * \left(\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \right) = \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) [\psi * f] .$$

où ψ est un filtre passe-bas ; ce qui se récrit symboliquement :

$$image\ des\ contours = passage\ par\ zéro \left(f * \left[\frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} \right] \right).$$

cela exprime qu'une image de contours est obtenue par filtrage de l'image par la dérivée seconde d'un filtre passe-bas, puis détection des zéros de la fonction ainsi obtenue.

Les filtres les plus utilisés pour ces filtrages passe-bas sont les filtres gaussiens [Marr et Hildreth, 1980]. Marr a montré en effet qu'avec de tels filtres on pouvait approcher de très près les effets donnés par le système visuel humain. Le filtre obtenu par convolution avec le laplacien d'une gaussienne est connu sous le nom de LOG (cf figure 11.5). Un autre filtre très utilisé et très proche du LOG est le DOG (*Difference of Gaussians*) (figure 11.6, qui procède par différence de deux gaussiennes dont les écarts-type sont dans un rapport 1,6 (pour simuler au mieux le système visuel). Les gaussiennes sont bien sûr affectées des facteurs adéquats pour que la somme des parties positives du filtre soit égale à la somme des parties négatives (valeur nulle à la fréquence 0 de la fonction de transfert). Sa mise en œuvre peut tirer profit des diverses qualités de la gaussienne : séparabilité, noyau auto-reproducteur, limite de suites de polynômes, approximation par des splines, etc., ce qui permet d'en accélerer la réalisation.

Dans des approches très sommaires, le DOG est parfois remplacé par le DOB (*Difference of Box*), filtre très rapide même pour des grandes tailles (on tire profit alors du fait que les filtres de moyenne

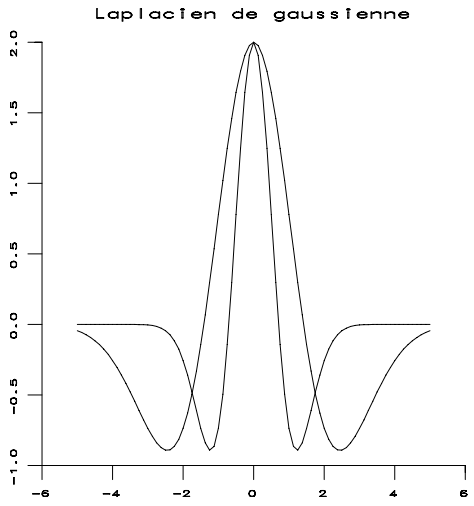


FIG. 11.5 – Filtre LOG pour deux valeurs différentes de σ.

(*Box*) ont une complexité de calcul indépendante de la taille de la fenêtre d'intégration au prix d'un coup mémoire important). Les résultats de ces filtres est cependant généralement de piètre qualité.

Les filtres LOG et DOG, n'ont plus aujourd'hui qu'un intérêt historique. On leur préfère les filtres analytiques que nous verrons plus loin.

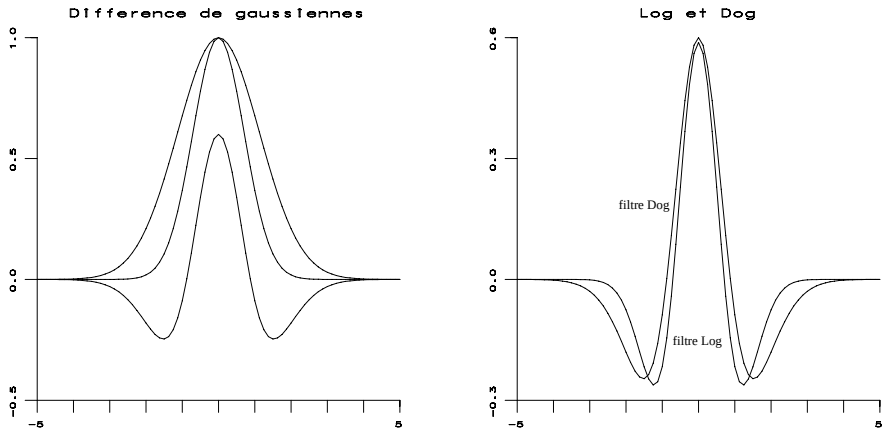


FIG. 11.6 – A gauche : filtre DOG et les deux gaussiennes dont il est la différence (leurs écarts-types sont dans un rapport 1,6). A droite filtre LOG et filtre DOG comparés.

11.2.5 Les détecteurs par masquage adapté

Dans ces techniques, on recherche en chaque point de l'image la présence d'une configuration conforme à un gabarit appartenant à un *dictionnaire de contours*. Pour cela, on définit une distance entre un contour type et une fenêtre de l'image. C'est par exemple l'inverse du produit scalaire des deux fenêtres : si l'image

est dénotée $f(x,y)$, et le $k^{ème}$ gabarit $g_k(x,y)$, on calcule par exemple :

$$r_k(x,y) = \frac{\left(\sum_{x,y \in fenêtre} f^2(x,y) \cdot \sum_{x,y \in fenêtre} g_k^2(x,y)\right)^{1/2}}{\sum_{x,y \in fenêtre} f(x,y) g_k(x,y)}$$

pour toutes les valeurs de k. On ne conserve que la meilleure des valeurs et seulement si elle est suffisamment faible. L'approche par masques adaptés (*template matching*), permet de connaître aisément la direction du contour, elle permet également de soumettre, en parallèle, ces calculs à des architectures adaptées. Mais la qualité des contours, ainsi que les post- traitements qu'ils requièrent sont très semblables à ceux que l'on a vus pour les filtres de maximum du gradient.

Les filtres adaptés les plus utilisés sont le filtre de Kirsch (cf. tableau 11.1), le filtre de Nevatia et Babu (cf. tableau 11.2) et les boussoles directionnelles de Prewitt (cf. tableau 11.3).

5	5	5
-3	0	-3
-3	-3	-3

-3	5	5
-3	0	5
-3	-3	-3

TAB. 11.1 – Masques de Kirsch : il y a 8 filtres issus de ces 2 fenêtres par rotation de $\pi/4$.

-100	-100	0	100	100
-100	-100	0	100	100
-100	-100	0	100	100
-100	-100	0	100	100
-100	-100	0	100	100

-100	32	100	100	100
-100	-78	92	100	100
-100	-100	0	100	100
-100	-100	-92	78	100
-100	-100	-100	32	100

TAB. 11.2 – Masques de Babu et Nevatia : il y a 12 filtres issus de ces 2 fenêtres par rotation, la première de $\pi/2$, la seconde de $\pi/4$.

1	1	1
1	-2	1
-1	-1	-1

1	1	1
-1	-2	1
-1	-1	1

TAB. 11.3 – Masques de boussole directionnelle : il y a 8 filtres issus de ces 2 fenêtres par rotation.

Au-delà de ces approches souvent heuristiques, Hueckel a proposé une démarche beaucoup plus rationnelle, s'appuyant sur la décomposition du signal d'image sur une base de fonctions orthogonales en coordonnées polaires (polynômes d'Hermite utilisés également pour l'oscillateur harmonique), tronquée à ses 8 premiers termes [Hueckel, 1971]. Malheureusement les résultats n'ont pas été au niveau des investissements et cette voie est maintenant abandonnée.

11.3 Les approches analytiques

Nous allons voir maintenant une approche qui a permis une bien meilleure compréhension des conditions d'une bonne détection de contours et qui a ainsi conduit à des détecteurs de très bonne qualité. On les voit émerger dans les années 85, à partir des travaux : [Torre et Poggio, 1986, Shen et Castan, 1986, Canny, 1986].

11.3.1 Les critères de Canny

Canny, dans une approche originale, [Canny, 1986] a proposé un filtre déterminé analytiquement à partir de 3 critères :

1. garantir une bonne détection, c'est-à-dire une réponse forte même à de faibles contours,
2. garantir une bonne localisation,
3. assurer que pour un contour il n'y aura qu'une seule détection (éviter les effets de *rebonds* dus, par exemple, à la troncature des filtres).

Ces 3 critères s'expriment par l'optimisation conjointe de 3 fonctionnelles qui permettent de définir le **filtre linéaire optimal** pour la détection d'une **marche d'escalier** sous l'hypothèse d'un bruit additif indépendant du signal [1].

Si l'on considère que le filtre a pour réponse impulsionnelle $\psi(x)$, ces fonctionnelles s'écrivent :

$$\textit{bonne détection} : \quad \Sigma = \frac{\int_0^\infty \psi(x)dx}{\sqrt{\int_{-\infty}^\infty \psi^2(x)dx}}$$

$$\textit{bonne localisation} : \quad \Lambda = \frac{|\psi'(0)|}{\sqrt{\int_{-\infty}^\infty \psi'^2(x)dx}}$$

$$\textit{réponse unique} : \quad \frac{|\psi'(0)|}{\sqrt{\int_{-\infty}^\infty \psi''^2(x)dx}} = k\frac{\int_{-\infty}^0 \psi(x)dx}{\sqrt{\int_{-\infty}^\infty \psi^2(x)dx}}$$

La résolution du système est assez complexe (on maximise le produit $\Sigma\Lambda$ sous la contrainte du $3^{ème}$ terme constant). Par ailleurs, Canny souhaite obtenir un filtre monodimensionnel (dans la direction orthogonale au contour), à réponse impulsionnelle finie (RIF) ; celui-ci a alors une expression complexe, composée de 4 termes, chacun combinant des lignes exponentielles et trigonométriques :

$$\psi(x) = a_1 e^{x/\sigma}\sin\omega x + a_2 e^{x/\sigma}\cos\omega x + a_3 e^{-x/\sigma}\sin\omega x + a_4 e^{-x/\sigma}\cos\omega x$$

où les coefficients a_i et ω sont déterminés à partir de la taille du filtre. Le paramètre σ est un paramètre de grande importance que nous retrouverons dans tous les autres filtres dérivés de l'approche de Canny. C'est un paramètre d'échelle qui indique en-deça de quelle distance deux contours parallèles seront confondus en un seul. Canny montre que la dérivée d'une gaussienne est une bonne approximation de son filtre.

$$\psi(x) \approx -x\exp\left(-\frac{x^2}{2\sigma^2}\right)$$

C'est donc la réduction à 1 dimension du filtre proposé par Marr à 2 dimensions (le maximum de la dérivée de la gaussienne est obtenu pour le passage par zéro du laplacien de gaussienne). Ce filtre donne des résultats de bonne qualité. Le critère de Canny (critère $\Sigma\Lambda$) vaut 1,12 pour le filtre RIF optimal, 0,97 pour le filtre gaussien. Remarquons cependant que le filtre gaussien n'est pas RIF.

[1] Par rapport aux approches proposées précédemment, on note que le filtre reste linéaire, mais que le critère d'optimisation est plus complet.

11.3.2 Les filtres de Deriche et Shen et Castan

Au filtre de Canny, on préfère souvent le détecteur de Deriche, qui répond exactement aux mêmes critères de qualité que celui de Canny, mais qui possède une réponse impulsionnelle infinie (filtre RII). Il a pu donc être synthétisé de façon récursive particulièrement efficace [Deriche, 1987]. Le filtre de Deriche a une expression générale de la forme :

$$\psi(x) = -cx\exp(-\alpha|x|),$$

avec :

$$c = \frac{[1-\exp(-\alpha)]^2}{\exp(-\alpha)}.$$

avec les critères même de Canny il est supérieur au filtre de Canny ($\Sigma\Lambda = 2$). Le paramètre α de Deriche représente alors l'inverse de l'écart type σ de la gaussienne du filtrage de Canny ($\alpha = \frac{\sqrt{\pi}}{\sigma}$). Le filtre de Deriche s'écrit de façon récursive en fonction des valeurs $f(x)$ de l'image, par l'intermédiaire de 2 filtres l'un décrivant l'image de gauche à droite et l'autre de droite à gauche :

$$z^+(x) = c\epsilon + 2\epsilon z^+(x-1) - \epsilon^2 z^+(x-2)$$

$$z^-(x) = -c\epsilon + 2\epsilon z^-(x+1) - \epsilon^2 z^-(x+2)$$

$$z(x) = z^+(x) + z^-(x)$$

avec :

$$\epsilon = \exp(-\alpha)$$

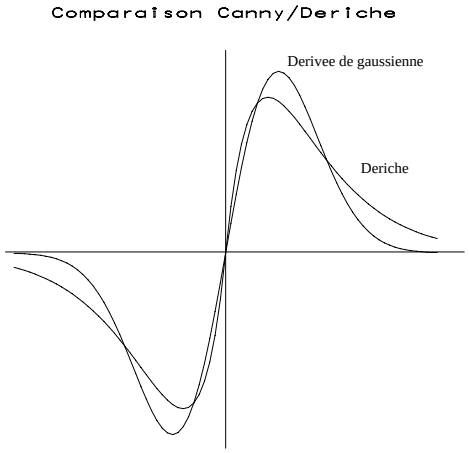


FIG. 11.7 – Le filtre de Deriche : $x.exp(-|x|)$ et la dérivée de la gaussienne : $\frac{x}{\sqrt{\pi}}.exp(-x^2/\pi)$.

Le filtre de Shen et Castan [Shen et Castan, 1986] possède également de très bonnes qualités de détection, il s'apparente à la même famille des filtres exponentiels (mais comme il n'est pas dérivable à l'origine, il ne se prête pas à une comparaison avec le critère de Canny) :

$$f(x) = c.\mathrm{sgn}(x)\exp(-\alpha|x|)$$

Le filtre de Shen et Castan, en raison de sa forme à l'origine, donne une localisation très précise des contours, mais il est sensible aux bruits.

Dans [Demigny et Kamlé, 1997], Demigny a étendu les critères d'optimalité des filtres de Canny à des images discrètes. Il les a ensuite étendus à des profils en rampe et non plus en marche d'escalier.

11.3.3 L'extension à 2D

Tous ces filtres sont mono-dimensionnels, le filtrage 2D est donc obtenu par l'action de deux filtres croisés, l'un en x, l'autre en y.

Pour définir l'un de ces deux filtres, lorsque l'on a choisi le gabarit du filtre dans la direction perpendiculaire au contour, afin de mieux intégrer le signal, on utilise, dans la direction du contour, une **fonction de prolongement** qui permet de filtrer les bruits. C'est souvent une gaussienne de même écart-type que celle qui est dérivée dans le détecteur. Par exemple la dérivée en x pourra être estimée par convolution de l'image par :

$$\psi(x,y) \;=\; -x\exp(-(\alpha x^2))\exp(-(\alpha y^2))$$

Ces filtres ne sont donc généralement pas isotropes. C'est-à-dire qu'ils ne répondent pas de façon identique à des contours identiques mais d'orientations différentes. Les seules fonctions séparables et isotropes sont les fonctions gaussiennes. Mais même si l'on utilise des filtrages gaussiens, la mise en œuvre discrète du filtre sur une maille carrée rend les filtres anisotropes. Des travaux importants ont été consacrés à la réduction de cet effet [Kunduri et al., 1999] utilisant une combinaison de deux détecteurs à $\pi/4$ l'un de l'autre.

En conclusion, dans une approche moderne de la détection des contours, un filtre de détection se compose de deux estimateurs de dérivées, l'un selon Ox, l'autre selon Oy. L'un de ces détecteurs (choisissons celui selon Ox) se compose du produit de 2 fonctions :

- **selon Oy c'est une fonction passe-bas, symétrique, (la fonction de prolongement) dont l'étendue est fonction de l'importance du bruit et de la distance que l'on souhaite respecter entre deux contours ;**
- **selon Ox c'est une fonction passe-haut (anti-symétrique), souvent égale à la dérivée de la précédente.**

Les couples « fonctions de prolongement - dérivées » sont typiquement issus de la gaussienne (Canny), de l'exponentielle décroissante (Shen et Castan) ou du produit de l'exponentielle par x (Deriche) (cf. figure 11.8), mais on peut également les concevoir très librement pour une application spécifique.

11.3.4 Variantes autour des filtres analytiques

D'autres filtres aux performances comparables, mais résultats d'optimisations différentes, ont été proposés dans :

- [Spacek, 1986] : On modifie le critère de Canny, et on cherche pour filtre une spline cubique de forme $x^3 - 2x^2 + x$.
- [Petrou et Kittler, 1991] : On considère alors que le contour est également le résultat d'un flou, ce n'est donc plus une marche, mais une rampe. Les critères de sélection sont ceux de Spacek et le filtre est une combinaison non-linéaire de lignes trigonométriques et d'exponentielles décroissantes.
- [Sarkar et Boyer, 1991] : On modifie dans ce cas les critères de Canny, on adopte une approche variationnelle et on choisit la représentation récursive d'un filtre RII comme Deriche.

Une excellente comparaison et analyse des filtres de détection des contours est donnée dans [Cocquerez et Philipp, 1996].

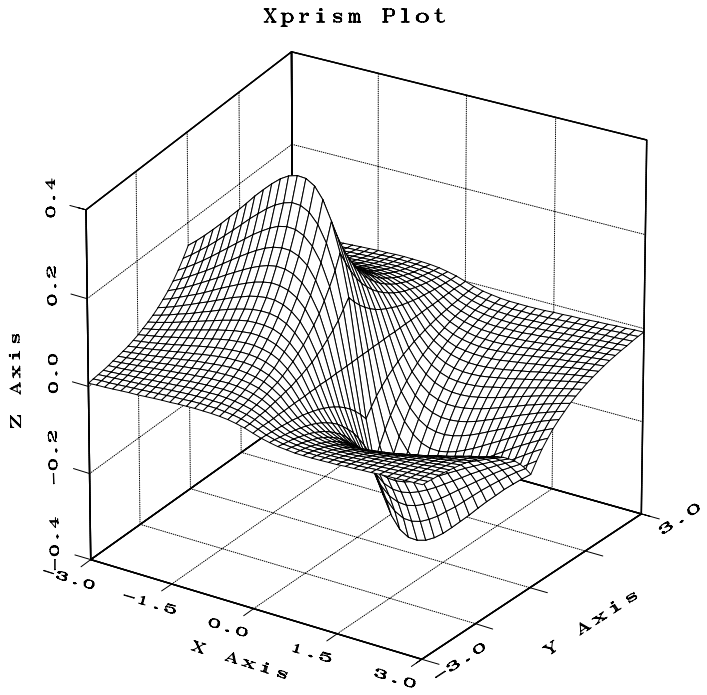


FIG. 11.8 – Profil général d'un filtre d'estimation de la dérivée dans la direction Ox. Selon Oy le filtre est un intégrateur (ici $exp(-|y|)$, selon Ox, c'est un dérivateur (ici $x.exp(-|x|)$).

Utilisant les résultats précédents ainsi que des méthodes éprouvées d'optimisation semi-globale développées dans les contours actifs (cf. section 11.4), des solutions variationnelles ont également été proposées qui sont fort efficaces et permettent d'ajouter des informations connues a priori sur l'allure des contours lors de leur détection. Voir par exemple [Fua et Leclerc, 1990].

11.4 Les contours actifs = les *snakes*

11.4.1 Une approche physique des contours

Une approche très différente des méthodes antérieures de détection de contours a été proposée en 1987 par Kass, Witkins et Terzopoulos [Kass et al., 1988] (voir aussi [Blake et Isard, 1998]), appelée contours actifs ou *snakes*. Il s'agit d'une méthode semi-interactive dans laquelle l'opérateur place dans l'image, au voisinage de la forme à détecter, une ligne initiale de contour. Cette ligne sera amenée à se déformer sous l'action de plusieurs forces :

- une énergie propre, assimilée à l'énergie mécanique de tension et de torsion d'une ligne matérielle,
- une énergie potentielle imposée par l'image qui vise à plaquer la courbe sur les contours,
- une énergie externe, introduite par l'utilisateur pour traduire les contraintes spécifiques du problème qu'il se pose.

Sous ces énergies, le contour *actif* va évoluer pour rechercher la position d'énergie minimale, qui sera ainsi un compromis entre les diverses contraintes du problème.

L'écriture formelle du problème passe par la définition paramétrique du contour, en fonction d'une variable s généralement l'abscisse curviligne :

$$v(s) \;=\; [x(s)\;,\; y(s)]^t \qquad s \;\in\; [0,1]$$

$$E_{totale} \;=\; \int_0^1 \left[E_{interne}(v(s)) \;+\; E_{image}(v(s)) \;+\; E_{externe}(v(s))\right] ds \tag{11.6}$$

avec :

$$E_{interne} \;=\; \alpha(s)\left(\frac{dv}{ds}\right)^2 \;+\; \beta(s)\left(\frac{d^2v}{ds^2}\right)^2$$

où :

- la première dérivée prend en compte les variations de longueur de la courbe, (c'est donc un terme de tension (résistance à la rupture), qui est contrôlé par l'élasticité que l'on attribue au contour),
- tandis que la seconde exprime les variations de la courbure (c'est un terme de flexion contrôlé par la raideur du contour).

Ces deux termes agiront donc pour produire une courbe régulière (cf. figure 11.9).

Le second terme d'énergie : E_{image} caractérise les lignes que l'on souhaite suivre. Dans le cas de détection de contours, ce sont des lignes de fort gradient, il vaut donc généralement :

$$E_{image} \;=\; -\nabla f$$

mais il peut être adapté pour suivre les maximums des niveaux de gris (dans ce cas $E_{image} = f$), ou toute autre fonction définie à partir de ceux-ci.

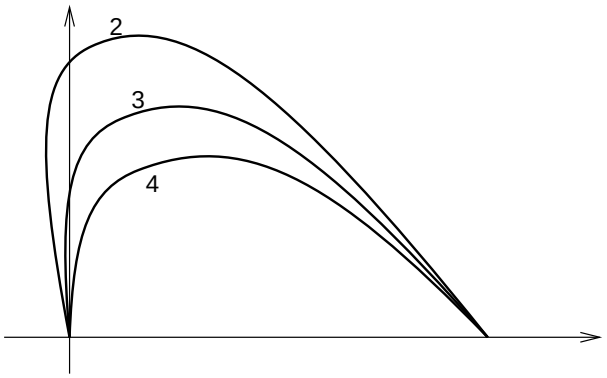


FIG. 11.9 – Déformation d'un contour actif à extrémités fixes pour des valeurs du rapport α/β (élasticité/raideur) égales à 2, 3 et 4, en l'absence d'une énergie liée à l'image.

Enfin le dernier terme d'énergie : $E_{externe}$ est choisi par l'utilisateur. Il peut avoir des formes très variées afin, par exemple, de contraindre le contour à resembler à un gabarit donné, à s'approcher d'un contour déjà détecté sur une autre image (suivi de séquences, ou images à 3D)[2].

En l'absence de ce dernier terme, et dans les nombreux cas où l'on recherche les lignes de fort gradient, l'équation 11.6 s'écrit alors :

$$E_{totale} \;=\; \int_0^1 \left[-\nabla f + \alpha(s)\left(\frac{dv}{ds}\right)^2 \;+\; \beta(s)\left(\frac{d^2v}{ds^2}\right)^2\right] ds \tag{11.7}$$

[2] On notera cependant que, dans le cas de volumes numériques tridimensionnels $f(x,y,z)$, il est possible d'étendre les contours actifs à des surfaces actives pour lesquelles l'énergie s'exprime comme pour des plaques minces.

11.4.2 Mise en équation des contours actifs

L'équation intégrale 11.7 est résolue généralement de façon variationnelle. On suppose que le contour évolue vers un minimum d'énergie, soit vers un zéro de $\frac{\partial E_{totale}}{\partial t}$. En désignant par v' et v'' les dérivées de v le long de la courbe, on obtient une équation différentielle vectorielle :

$$\gamma \frac{\partial v}{\partial t} \; + \; (\alpha v')' \; - \; (\beta v'')'' \; = \; \frac{\partial |\nabla f|^2}{\partial v}$$

où α, β et γ sont potentiellement variables le long de s. Il y a plusieurs façons de concevoir la discrétisation de la courbe :

1. selon les différences finies : les éléments de la courbe sont réduits en des points auxquels sont attachés les éléments mécaniques (masse, raideur, etc.) de la courbe considérée concentrée en ces points ;
2. selon les éléments finis on remplace chaque portion de courbe par le segment élémentaire et les éléments mécaniques sont calculés sur ces segments.

Nous ne décrivons ici que la première approche la plus fréquemment employée. Après discrétisation de la courbe en un nombre n de points et en posant :

$$V^t \; = \; [v_0^t, \; v_1^t, \; v_2^t, \; ....., \; v_{n-1}^t]^t$$

et :

$$h_v \; = \; -|\nabla f(v)|^2$$

on obtient une équation matricielle de la forme :

$$(A \; + \; \gamma I) V^t \; = \; \gamma V^{t-1} \; - \; h_v(V^{t-1})$$

où A est une matrice pentadiagonale[3] (figure 11.11) de taille $n \times n$ fonction de α et β, I est la matrice unité de taille $n \times n$ et γ exprime une *inertie* de la courbe aux déplacements. Sa résolution donne :

$$V^t \; = \; (A \; + \; \gamma I)^{-1} (\gamma V^{t-1} - h_v(V^{t-1}))$$

Si les paramètres α, β et γ sont invariants le long de la courbe, il suffit alors de calculer une seule fois $(A + \gamma I)^{-1}$ pour résoudre le système pour tous les temps t. Sinon, on inverse la matrice à chaque instant. Les choix des paramètres α, β et γ qui ne sont pas dictés par le problème demeurent souvent délicats pour garantir une convergence convenable, ainsi que le choix de l'estimateur de ∇f. L'utilisation d'un estimateur par filtre de Deriche est parfois préconisée.

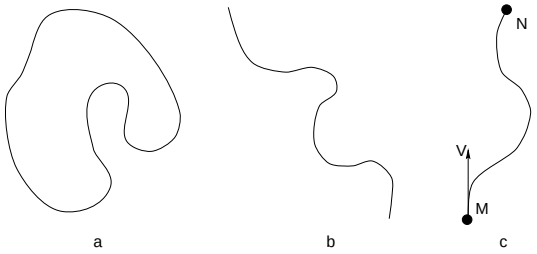


FIG. 11.10 – Contour actif fermé en a, ouvert à extrémités libres en b et ouvert à extrémités fixes en c (on impose en M une extrémité fixe et une orientation fixe donnée par le vecteur V).

Malgré les difficultés de réglage de la convergence, les contours actifs apportent une solution heureuse à la détection de contour, intermédiaire entre les solutions purement locales (opérateurs de type Sobel) et les segmentations globales. Trois types de contours actifs différents sont utilisés (cf. figure 11.10) :

[3]Dans le cas d'une discrétisation par éléments finis, la matrice est heptadiagonale

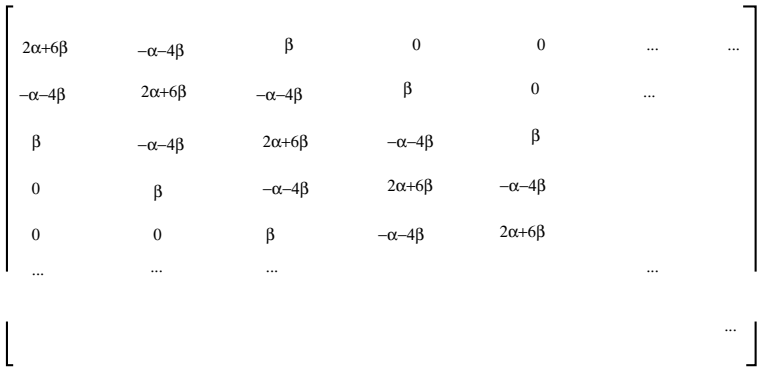


FIG. 11.11 – Un contour actif fermé est représenté par une matrice circulante pentadiagonale.

1. les contours actifs fermés (où $v_0^t = v_{n-1}^t$),
2. les contours actifs à extrémités libres,
3. les contours actifs à extrémités fixes (où les positions v_0 et v_{n-1} sont fixes dans le temps, ainsi que, potentiellement les premières dérivées en ces points).

Selon le type de contours actifs, les matrices A présentent la propriété d'être cieculante (contours actifs fermés), Töplitz (contours actifs à extrémités libres) ou quelconques (figure 11.11).

Laissé libre d'évoluer seul, en l'absence de forces d'attraction dues au gradient dans l'image, le contour actif a tendance à se réduire à un point s'il est fermé, à une droite s'il est à extrémités libres. Pour compenser cette tendance, on est parfois amené à introduire des forces internes de *gonflage*. On a alors affaire à des *ballons* [Cohen, 1991]. Comme les équations des forces de gonflage ne dérivent plus d'un potentiel, les écritures s'en trouvent un peu complexifiées.

Des solutions élégantes ont également été proposées pour rechercher la meilleure position du snake en en déplaçant ses nœuds sur toute la grille discrète de l'image par une technique de programmation dynamique [Amini et al., 1990] (cf. figure 11.12).

Par ailleurs, le couplage de techniques de contours actifs avec des méthodes s'appuyant sur des propriétés statistiques distinguant la forme de son fond, et optimisées par un maximum de vraisemblance au sens bayésien, a montré sa très grande efficacité, permettant de faire le lien entre techniques de contours et de régions [Chesnaud et al., 1999].

11.4.3 Les ensembles de niveaux (*level sets*)

Ce sont également des représentation variationnelles des contours qui conduisent donc à des solutions qui évoluent au cours du temps dans l'image, régies par un critère global. Ces modèles ont cependant le mérite remarquable de pouvoir changer de topologie si les contours l'imposent : par exemple un contour simple peut évoluer en deux contours séparés, ou, à l'inverse, deux contours séparés se réunir en un seul contour. Cela est rendu possible par l'utilisation de fonctions d'une dimension supérieure à celle des contours recherchés (si l'on cherche des lignes de contours, on introduira pour inconnue une surface de $\mathbf{R}^3$, si on cherche des surfaces de $\mathbf{R}^3$, on introduit des volumes de $\mathbf{R}^4$). Le contour est alors défini comme l'ensemble de niveau zéro (*level sets*) de cette fonction : $v(s)$ est une ligne de niveau $z = 0$ de la surface $z = f(x, y)$, souvent prise comme la distance au contour [Osher et Sethian, 1988, Sethian, 1996].

Sur le contour :

$$\frac{dz}{ds} = 0 = \frac{\partial z}{\partial x}\frac{dx(s)}{ds} + \frac{\partial z}{\partial y}\frac{dy(s)}{ds}$$

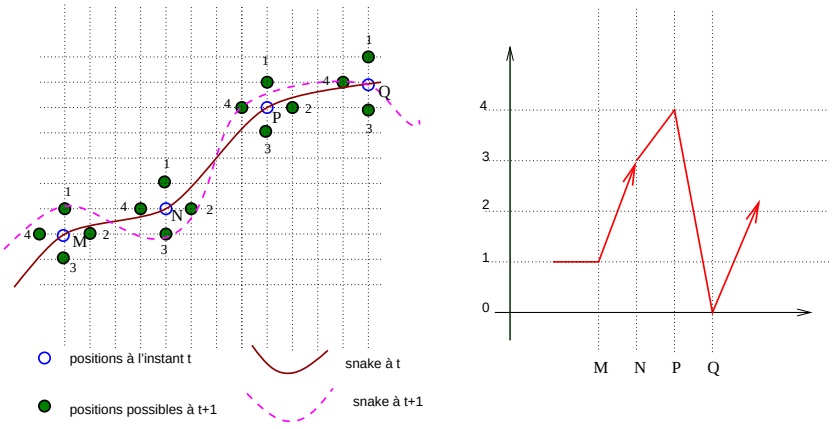


FIG. 11.12 – Dans la technique d'optimisation des contours actifs par programmation dynamique, chaque nœud M, N, P, Q peut se déplacer dans l'un des nœuds $1, 2, 3$, ou 4, ou rester en place (position 0). L'optimisation se fait dans le graphe de droite par propagation de gauche à droite.

et la courbe de niveau évolue en fonction des itérations t selon l'équation fondamentale :

$$\frac{\partial z}{\partial t} = ||\vec{grad}(z)||\vec{N} \wedge \frac{\vec{dv}}{dt}$$

où $\vec{N}$ représente la normale au contour v.

Des équations semblables existent à 3D [Zhao et al., 2000].

11.4.4 Les modèles géodésiques actifs

Les approches par contours actifs peuvent être reliées à la famille des problèmes d'optimisation géodésique dans des espaces dont la géométrie est fixée par la dynamique de l'image (*cf.* chapitre 12.5.1). Ces techniques, appelées modèles géodésiques actifs, ont été proposées dans [Caselles et al., 1997] et le lien avec les contours actifs y est discuté.

Le problème du snake, s'il n'y a pas d'élasticité, peut s'écrire comme la minimisation de l'énergie :

$$J_1(v) = \int_a^b |v'(s)|ds + \lambda \int_a^b \psi(|\nabla f(v(s))|)^2 ds$$

où ψ représente le détecteur de contour. Caselles a introduit une fonctionnelle du type :

$$J_2(v) = 2\sqrt{\lambda\alpha} \int_a^b |v'(s)|\psi(|\nabla f(v(s))|)ds$$

Minimiser J_2 revient à trouver le chemin de longueur minimale dans l'espace riemannien dont la métrique est induite par l'image. On a pu montrer un certain nombre d'équivalences entre les propriétés des minimums de J_1 et de J_2 dans [Aubert et Blanc-Féraud, 1999].

Les contours géodésiques utilisent des solutions proches de celles des ensembles de niveaux pour le codage et la représentation de la ligne de contours.

11.5 La poursuite et la fermeture des contours

Si l'on excepte les contours actifs et les passages par zéro des laplaciens, la plupart des détecteurs de contours fournissent des contours ouverts, c'est-à-dire qu'ils ne séparent pas les composantes de l'image dans des objets topologiquement distincts. Cela rend souvent plus difficile l'étape de reconnaissance des formes qui doit suivre. Pour cela, on a proposé plusieurs méthodes permettant d'obtenir des contours clos. C'est ce que l'on appelle la **fermeture de contours**.

De nombreuses solutions ont été proposées pour cette étape fondamentale, reposant sur des principes très différents (les contours actifs à extrémités fixes que nous venons de voir en sont un exemple). La qualité des résultats obtenus est généralement directement liée au coût informatique consenti, de très bonnes solutions existant, mais à des coûts prohibitifs pour la plupart des applications.

11.5.1 Méthodes de recherche dans des graphes

Ce sont les méthodes qui reposent sur les bases les plus solides et susceptibles de donner les meilleurs résultats, mais leurs coûts sont souvent très élevés, car elles se proposent d'explorer l'espace de toutes les solutions et de retenir la meilleure en fonction d'un critère que l'opérateur s'est donné. L'écueil de la recherche combinatoire exhaustive peut être évité par des recherches polynomiales assez complexes, surtout fondées sur la programmation dynamique.

La technique de base, présentée par Martelli [Martelli, 1972] et Montanari [Montanari, 1971], est celle d'une optimisation combinatoire minimisant la fonction de coût (ou maximisant une fonction de mérite). Les algorithmes de programmation dynamique (Viterbi, A^*), sont particulièrement bien adaptés à ce problème, mais, malgré leur complexité polynomiale sont généralement limités par des profondeurs de recherche de quelques pixels si le graphe est la totalité de l'image.

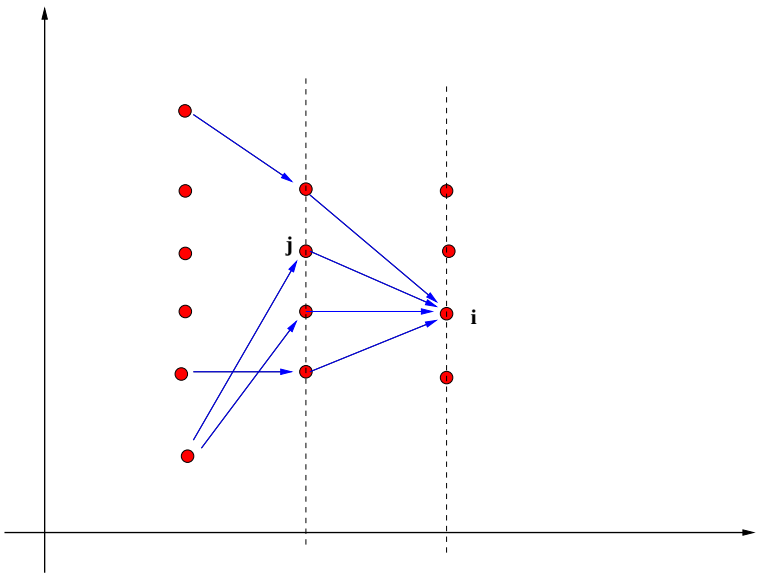


FIG. 11.13 – Résolution d'un problème de recherche de chemin optimal dans un graphe par programmation dynamique : la propagation se fait de la gauche vers la droite. On trouve le meilleur chemin arrivant à l'état i en optimisant la fonction de coût sur tous les états j qui ont été calculés à l'étape antérieure.

Dans une approche de type Viterbi, on considère l'évolution pas à pas d'un contour. À chaque

pas le coût du contour est évalué pour toutes les provenances possibles du contour et n'est retenu pour tout point du parcours que le seul chemin qui minimise le coût pour arriver à ce point (cf. figure 11.13). On garde alors en chaque point ce coût minimum ainsi que l'adresse du *père* qui a permis d'y aboutir. Un nouveau pas est alors fait et toutes les destinations sont explorées que l'on peut atteindre en un pas à partir des points déjà atteints. La fonction de coût se calcule de la façon suivante :

$$\Gamma(i) = \gamma(i) + min_{j \in antécédents} [\Gamma(j) + \delta(i,j)] = \gamma(i) + \epsilon(i)$$

où $\gamma(i)$ est inversement proportionnel à la qualité du point i comme point de contour et $\delta(i,j)$ exprime l'incompatibilité de i et j le long d'un contour. Le terme γ est un terme d'innovation le long du chemin tandis que ϵ est un terme de mémoire. La pondération de ces deux termes est un problème délicat en programmation dynamique.

Un contour est fermé lorsque le point i est confondu avec un point d'un autre contour. Le coût de la fermeture est alors l'intégrale le long de la chaîne des coûts de liaison δ et des coûts individuels γ. Pour éviter de trop pénaliser les chaînes longues, il est possible de calculer le *coût moyen* le long du parcours en pondérant le terme de mémoire ϵ par le chemin parcouru.

En limitant les domaines de recherche à des champs convexes, ou plus généralement en introduisant des limitations sur les types de chemin que l'on s'autorise à trouver, on peut étendre notablement la portée de ces fermetures (cf. figure 11.14). Des approches multi-résolutions permettent de travailler de façon hiérarchique, d'une fermeture grossière à une fermeture précise, en limitant progressivement le domaine de recherche à une bande étroite autour du contour détecté à l'étape précédente. Dans d'autres cas, on se limite à des horizons faibles pour entreprendre une recherche par programmation dynamique qui est validée, puis relancée à partir du résultat précédent.

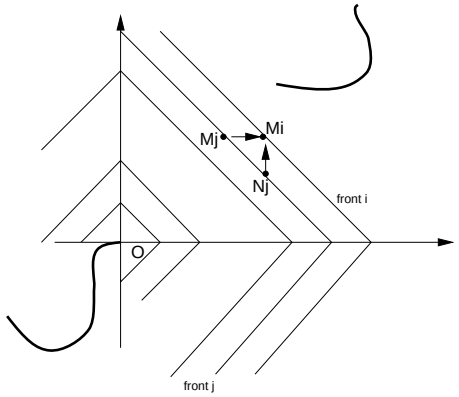


FIG. 11.14 – Exemple d'une fermeture de contours par recherche d'un chemin optimal dans un graphe. Le contour à fermer s'arrête en O. Les horizons de recherche à chaque étape de la programmation dynamique sont des carrés (fronts d'ordre i), centrés en O. Chaque point M_i peut avoir 2 antécédents (M_j et N_j). Une fermeture s'obtient lorsqu'un des points appartient à un autre contour. Pour éviter que le contour ne se ferme trop rapidement sur lui-même (près de O), il faut invalider les points du contour lui-même. Une telle propagation contraint les contours à une courbure toujours de même signe.

Dans une recherche de type A^*, on dispose d'une estimée de la fonction de coût ($\hat{\Gamma} > \Gamma_{opt}$) et la fonction de coût doit être croissante le long d'une fermeture. On fait une recherche en largeur d'abord en comparant à chaque nœud la valeur locale du coût à cette estimée et on invalide le nœud si le coût dépasse l'estimée. On a tout intérêt à trouver donc une estimée très proche du coût optimal réel.

Les fonctions de coût sont les points déterminants de la qualité de la méthode. Cette fonction de coût doit être locale pour préserver le caractère markovien nécessaire à la programmation dynamique (le

coût en un point ne dépend que de γ et du coût au point précédent), mais elle doit également incorporer toutes les informations que l'on connaît sur un *bon contour* : contraste entre les deux plages séparées par le contour bien sûr, mais aussi, longueur et courbure, voire écart à un contour modèle dans certains cas. C'est dans cette fonction de coût que l'on rajoute l'information qui a manqué au détecteur local placé en amont pour détecter le contour.

11.5.2 Les automates

Ces solutions, au contraire des précédentes, ne proposent que des solutions très sous-optimales, fondées sur de seuls critères locaux (il n'y a pas d'optimum global comme par programmation dynamique). Elles sont particulièrement rapides de mise en œuvre [Giraudon, 1987], mais peuvent parfois faire diverger la combinatoire si l'on gère mal la prolifération des chemins par ramification. Les résultats d'une détection par masquage adapté sont les candidats idéaux à ces techniques de fermeture puisqu'ils fournissent des directions de recherche.

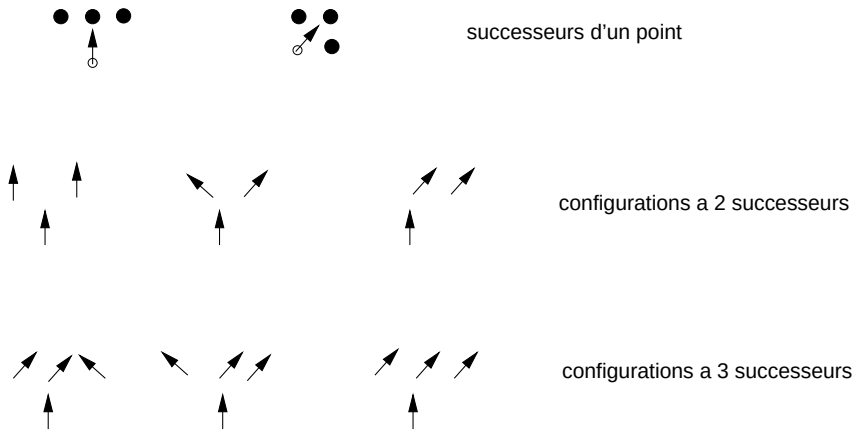


FIG. 11.15 – Configurations retenues par l'automate de Nevatia et Babu pour orienter un contour vers 3 successeurs au plus.

Les diverses méthodes se distinguent par le choix des successeurs d'un point (parmi des configurations pré-définies dans [Robinson, 1977, Cocquerez et Philipp, 1996, Nevatia et Babu, 1980], par prédiction de type Kalman) et le mode de balayage de l'image (par exemple 1 balayage dans [Nevatia et Babu, 1980], 2 balayages, l'un horizontal l'autre vertical dans [Cocquerez et Philipp, 1996], traitement de listes chaînées dans [Giraudon, 1987]). Il est également important de définir des fonctions de mérite pour valider les chaînes ainsi trouvées, car les automates, comme la programmation dynamique, trouvent toujours un contour, mais ne garantissent pas toujours qu'il correspond véritablement au contenu de l'image.

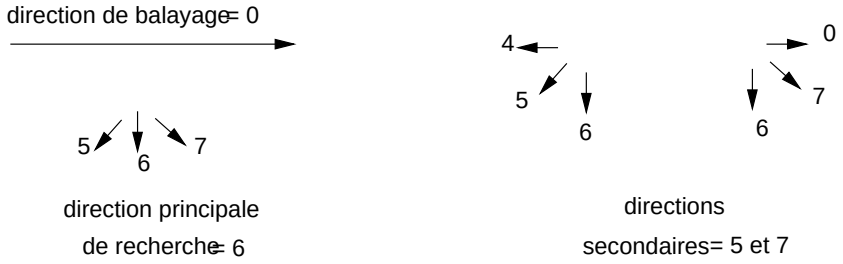


FIG. 11.16 – L'automate de Cocquerez lors d'un balayage horizontal utilise les directions 5, 6 et 7 de Freeman (cf. chapitre 14.4) comme directions principales. Lorsque l'une d'elle a été retenue, les directions adjacentes (par exemple le 4 et le 6 pour la direction principale 5) sont utilisées pour élargir la recherche. Cocquerez propose de faire 2 balayages successifs, l'un horizontal, l'autre vertical.

Chapitre 12

La segmentation par régions

Chapitre rédigé par Henri MAÎTRE

L'approche duale de la détection des contours pour la décomposition d'une image en ses formes élémentaires est l'approche par régions. Elle repose sur la recherche de zones possédant des attributs communs, soit de luminosité, soit, plus rarement, de textures (nous n'aborderons pratiquement pas ce second point dans ces lignes, mais au chapitre 13).

Nous allons voir ici, tout d'abord, les méthodes utilisant l'histogramme, qui sont souvent très proches des méthodes de classification conventionnelles, mais dans lesquelles serait renforcé l'aspect de cohérence de zone. Nous verrons ensuite les méthodes pyramidales, prototypes des méthodes de subdivision hiérarchique descendante, puis les méthodes de croissance de régions, qui s'inspirent des méthodes de conquête de l'optimisation combinatoire, et les méthodes de partage et réunion qui tirent profit simultanément des avantages des deux méthodes précédentes.

Enfin, nous donnerons très brièvement un aperçu de méthodes à base de règles qui se proposent de combiner approches par contours et approches par régions dans un seul vaste programme de segmentation.

12.1 Les méthodes sur histogramme

Ces méthodes sont de mise en œuvre assez simple et de performances souvent réduites car elles ne tirent pas profit de l'aspect spatial de l'information d'image. Elles sont recommandées dans les cas suivants :

1. lorsque les images présentent des classes évidentes : documents écrits ou schémas en noir et blanc ou en couleur, objets très contrastés (par exemple cellules d'une biopsie ou avion sur un ciel), etc.

2. lorsque les images sont définies sur de nombreux canaux (images multi- ou hyper-spectrales), ce qui enrichit l'information portée par l'histogramme.

L'idée générale de ces méthodes consiste à isoler des pics de l'histogramme. À une dimension on procède donc à des seuillages ou des multi-seuillages. À n-dimensions on procède à des classifications [Dubuisson, 1990].

12.1.1 Avec apprentissage

Seuillage bayésien

Si l'on dispose d'une connaissance sur les classes que l'on recherche (en particulier la probabilité d'occurrence d'un niveau de gris pour chaque classe, et la probabilité à priori des classes), alors on peut aisément se replacer dans les conditions de la théorie bayésienne de la décision [Dubuisson, 1990, Duda et Hart, 1973] et choisir les seuils qui minimisent le coût des fausses erreurs.

Pour des histogrammes à 1 dimension et pour 2 populations X et Y, en dénotant par :
- $P(X)$ et $P(Y)$ les probabilités à priori des classes X et Y (sous la contrainte $P(X) + P(Y) = 1$),
- $P(n/X)$ et $P(n/Y)$ les probabilités conditionnelles qu'un pixel de la classe X ait un niveau de gris n (et idem pour Y),
- C_X et C_Y les coûts des mauvaises classifications de X et Y,

en supposant ces grandeurs toutes connues, et sous l'hypothèse de stationarité de l'image (c'est-à-dire que les propriétés statistiques sont invariantes en tout point de l'image), le seuil optimal est défini comme celui minimisant le coût global :

$$\Gamma(s) = C_x \int_s^N P(X).P(n/X)dn + C_y \int_0^s P(Y).P(n/Y)dn \tag{12.1}$$

Ce seuil est obtenu en testant pour tout n le rapport de vraisemblance $\Lambda(n)$ face au seuil s :

$$\Lambda(n) = \frac{P(n/Y)}{P(n/X)} \quad \text{et} \quad s = \frac{C_y P(Y)}{C_x P(X)}$$

Seuillage de Neyman-Pearson

Une autre décision peut se faire selon le critère de Neyman-Pearson. Dans ce cas on s'intéresse en particulier à une classe (ici on choisit X). On définit la probabilité de fausse alarme pour cette classe :

$$P_f = \int_0^s P(Y)P(n/Y)dn$$

On maximise alors la probabilité de détection pour une probabilité de fausse alarme donnée. Dans ce cas on considère que la fausse alarme est l'erreur la plus grave. On fixe sa probabilité à une valeur acceptable : $P_f = \alpha$. La probabilité de détection de X est donnée par :

$$P_d = \int_0^s P(X)P(n/X)dn$$

Le seuil de Neyman-Pearson est donné par la méthode du lagrangien :

$$L = P_d - \lambda(P_f - \alpha) \tag{12.2}$$

$$= \lambda\alpha - \int_0^s [P(n/Y)P(Y) - P(n/X)P(X)]dn \tag{12.3}$$

et la décision se fait en comparant le rapport de vraisemblance $\Lambda(n)$ à α.

Cette décision est moins sensible aux probabilités à priori et conduit en particulier à des décisions plus proches de nos choix intuitifs que la décision bayésienne dans le cas où un événement est très rare.

D'autres critères existent bien sûr qui s'appuient sur d'autres choix statistiques (par exemple minimum d'entropie).

12.1.2 Seuillage sans apprentissage

Lorsque l'on dispose du seul histogramme pour extraire des classes, on recherche généralement des modes de cet histogramme qu'on isole par des seuillages au creux des vallées. Souvent les histogrammes sont trop irréguliers et il convient alors de les filtrer, soit par des fenêtres glissantes, soit par des équations de diffusion (éventuellement isotropes). De nombreuses règles empiriques ont été proposées pour choisir les seuils automatiquement qui ne sont guère généralisables [Kittler et al., 1985, Weszka, 1978].

12.1.3 Méthodes de classification

Disposant d'un histogramme, éventuellement multidimensionnel, la plupart des techniques de classification s'appliquent à sa segmentation. Les plus utilisées sont :

- les techniques de nuées dynamiques (*k-means*) qui procèdent alternativement en classifiant au plus proche voisin le nuage des points, selon une distance à des noyaux donnés, puis en estimant la position des meilleurs noyaux de ces classes ainsi obtenues. Il est important pour cette méthode de disposer du nombre de classes recherchées. Si l'on ne le connaît pas, on choisit souvent des critères entropiques (comme le critère d'Akaike ou le critère de Hannan et Quinn [Olivier et al., 1997]) qui permettent d'évaluer des classifications obtenues avec des nombres de classes différents[1].
- les réseaux neuromimétiques et en particulier les cartes de Hopfield.
- et, si l'on dispose d'un certain nombre d'échantillons déjà classés, les plans, ou les courbes, séparateurs, ainsi que les k-plus-proches voisins.

Dans les espaces de grande dimension (imagerie hyperspectrale par exemple), on peut avoir intérêt à réduire tout d'abord la dimension des espaces pour éviter d'avoir à estimer des probabilités très faibles. On peut le faire par ACP (analyse en composantes principales) ou par analyse de Karhunen-Loeve. Ces deux transformations sont identiques à une normalisation près des axes. Elles procèdent en projetant le nuage de points sur le sous-espace construit à partir d'un nombre limité des vecteurs propres de la matrice de covariance des données.

Dans les espaces de dimension p, les distances utilisées entre deux nuages de points caractérisés par leur moyenne (vectorielle) m_i et leur matrice de covariance Γ_i sont, par ordre de complexité croissante [Fukunaga, 1990, Zhang et Modestino, 1990] :

- la distance euclidienne qui pondère également toutes les variables :

$$(m_1 - m_2)^t (m_1 - m_2)$$

- la distance de Mahalanobis (par exemple du nuage 2 par rapport au nuage 1), qui rend compte de l'orientation des inerties des nuages dans l'espace :

$$(m_1 - m_2)^t \Gamma_2^{-1} (m_1 - m_2)$$

- la distance de Bhattacharyya, qui permet de distinguer des lois de mêmes moyennes, mais de variances différentes. :

$$1/4(m_1 - m_2)^t (\Gamma_1 + \Gamma_2)^{-1} (m_1 - m_2) + \frac{1}{2} \ln \frac{\frac{1}{2}|\Gamma_1 + \Gamma_2|}{\sqrt{|\Gamma_1||\Gamma_2|}}$$

12.1.4 Sélection sur l'histogramme et dans l'image

Le défaut des approches par classification est de complètement négliger les relations spatiales entre pixels, pour ne s'attacher qu'à leurs propriétés de radiométrie. Pour pallier ce défaut, dans une approche

[1] On note cependant avec [Olivier et al., 1997] que ces modèles ont tendance à sur-évaluer le nombre de classes trouvées.

proposée dans [Ohlander et al., 1978] et souvent reprise, on procède de façon itérative dans l'espace image et sur l'histogramme :

1. sur l'histogramme on sélectionne un mode isolé,
2. parmi les zones de l'images contribuant à ce mode on sélectionne la zone connexe la plus grande.

On itère ce processus, soit en subdivisant à nouveau l'histogramme de la zone retenue, soit en s'occupant d'un autre mode de l'histogramme original.

12.1.5 Sélection sur histogramme et régularisation

Mais si l'on souhaite améliorer les propriétés spatiales des zones obtenues par sélection de modes sur l'histogramme ou par classification, l'une des méthodes les plus appropriées est de modéliser le problème par un champ markovien (voir chapitre 7). On considère que les régions forment une partition sur l'image (cf. section 12.2). Chaque région est représentée par une fonction caractéristique et identifiée par une étiquette $\lambda_i \in E$. Le champ d'étiquettes est un champ caché à l'utilisateur qui n'observe que la réalisation bruitée de l'image $f(x,y)$ en chaque pixel, et qui va essayer d'estimer la meilleure distribution des étiquettes λ_i connaissant les n (généralement selon le critère du Maximum A Posteriori de Λ (MAP)).

Dans la formalisation markovienne on passe d'une représentation probabiliste à une représentation en énergie (généralement au moyen d'une formalisation bayésienne de la probabilité à postériori). Très souvent l'énergie (représentant la probabilité à postériori de la classe sachant l'observée) est constituée de deux termes : l'un traduit la similarité de la classification aux données que l'on observe, c'est le terme d'attache aux données (probabilité des observées conditionnellement aux classes), le second exprime les a priori que nous avons sur les classes (par exemple classes compactes, aux contours réguliers ou de forme prédéfinie, dans des relations de voisinage particulières, probabilité à priori de classe, etc.). Des hypothèses peuvent également être faites sur les distributions des niveaux de gris (par exemple lois gaussiennes sur les régions), des étiquettes (dépendance ou indépendance spatiale), ainsi que des distributions conditionnelles des niveaux de gris sachant les étiquettes.

Dans la démarche la plus simple on utilise alors souvent un terme d'attache aux données de la forme :

$$U_0 = (f(x,y) - \mu_i)^2$$

où $f(x,y)$ est le niveau de gris du pixel, μ_i le niveau de gris de la classe λ_i dans laquelle on a placé le pixel (x,y). Un tel terme est équivalent à une probabilité gaussienne d'appartenance à la classe λ_i :

$$p(f(x,y)|\lambda_i) \propto \exp(-U_0) = \exp\left[-\frac{(f(x,y)-\mu_i)^2}{2\sigma_i^2}\right]$$

Le terme de régularisation est calculé sur les cliques $\{c\}$ constituant le voisinage que l'on souhaite donner à (x,y) (cf. figure 7.1). Il prend en compte la compatibilité $\Phi(\lambda_i, \lambda_k(x',y'))$ entre l'étiquette λ_i attribuée à (x,y) et celle $\lambda_k(x',y')$ attribuée à (x',y') :

$$E_1 = \sum_{(x',y')\in\{c\}} \Phi(\lambda_i, \lambda_k(x',y'))$$

Par exemple, dans le cas le plus simple, on peut attribuer une énergie négative si $\lambda_i = \lambda_k$ et positive dans le cas contraire (modèles de Potts et d'Ising, voir chapitre 7). Des modèles plus complexes peuvent prendre en compte des compatibilités plus subtiles (classes plus ou moins proches). On recherche alors un minimum de $U = U_0 + \theta U_1$ par toutes les techniques usuelles d'optimisation itératives (ICM, recuit simulé, etc.). Le coefficient θ permet de pondérer les effets de la régularisation par rapport à ceux de

la classification. Pour un θ fort, on obtiendra des classes très compactes et grandes au détriment d'une bonne séparation des classes sur l'histogramme [Derin et Elliott, 1987b, Lakshmanan et Derin, 1989b, Heitz et al., 1994, Won et Derin, 1992].

Dans d'autres approches, on peut se dispenser d'introduire des étiquettes de classes en introduisant explicitement un processus de bord, (cf. section 7.4 et [Geman et Geman, 1984b]), processus b discret binaire (0 ou 1) ou continu (entre 0 et 1) qui prend, généralement, ses valeurs entre les pixels. Dans le cas binaire, lorsqu'il vaut 1 le voisin concerné n'a plus d'effet sur le site, selon la formule :

$$U_1 = \sum_{(x',y')\in\{c\}} (1-b).\Phi(g_i, g_k(x',y'))$$

C'est donc l'équivalent d'un contour introduit dans l'image. Mais on pénalise usuellement l'introduction d'un contour par un terme de la forme :

$$U_1' = \sum_{(x',y')\in\{c\}} b$$

ou par un coût de type MDL (Minimum Description Length) (cf. section 12.4).

Enfin, on peut également supprimer le processus de bord en le « cachant » à l'intérieur du potentiel U_1. Pour cela on choisit une fonction Φ qui sature pour des valeurs fortes des différences entre niveaux de gris de sites voisins. On perd alors, avec la convexité de la fonctionnelle d'énergie, des bonnes propriétés de convergence du champ markovien et l'optimisation devient plus complexe (cf. figure 7.7 et section 7.4) [Blake et Zisserman, 1987, Black et Rangarajan, 1996].

12.2 Les méthodes par transformation de régions

Les méthodes que l'on va examiner maintenant s'appuient toutes sur la notion de **prédicat** et sur celle de **partition.**

Un prédicat $\mathcal{P}$ est une proposition logique dont la valeur dépend de son argument. Nous nous intéresserons aux prédicats sur les régions $\mathcal{R}_i$ de l'image.

Le prédicat de base de la segmentation est : *la région $\mathcal{R}_i$ est homogène.* Pour vérifier le prédicat, parmi les arguments les plus utilisés en segmentation d'images, nous retiendrons les suivants :

- le contraste sur la région : $\mathcal{P}(\mathcal{R}_i) = vrai \iff max_{\mathcal{R}_i}[f(x,y)] - min_{\mathcal{R}_i}[f(x,y)] < \sigma$
- l'écart-type de la région : $\mathcal{P}(\mathcal{R}_i) = vrai \iff \sqrt{\frac{1}{N}\sum_{\mathcal{R}_i}(f(x,y)-m)^2} < \sigma$ (avec $N = Card(\mathcal{R}_i)$ et $m = \frac{1}{N}\sum_{\mathcal{R}_i} n(x,y)$),
- la distance interquartile sur la région, c'est-à-dire la distance séparant les 25 % inférieurs des 25 % supérieurs de l'histogramme,
- les différences limitées : $\mathcal{P}(\mathcal{R}_i) = vrai \iff \forall (k,j)\ voisins, |f(k) - f(j)| < \sigma$,
- l'entropie : $\mathcal{P}(\mathcal{R}_i) = vrai \iff -\sum_{\mathcal{R}_i} p(f)\log(p(f)) < \sigma$.
- etc.

Un autre prédicat également utilisé est : *la région $\mathcal{R}_i$ est distincte de ses voisines.* On utilise des arguments qui mettent en jeu les différences de valeurs moyennes, les distances inter-médianes, les contrastes moyens ou minimums aux frontières, etc.

Une partition Π est un ensemble de sous-ensembles $\mathcal{R}_i$ (appelées régions) de l'image, vérifiant :

$$\Pi \iff \begin{cases} \forall i,j \quad \mathcal{R}_i \cap \mathcal{R}_j = \emptyset \\ \bigcup_i \mathcal{R}_i = \text{support(image)} \\ \forall i \quad \mathcal{R}_i \neq \emptyset \end{cases}$$

Lorsque toutes les régions d'une partition vérifient le prédicat, on dit que la partition le vérifie. La partition triviale où tous les pixels sont dans des régions différentes de cardinal 1 vérifie tout prédicat qui peut se vérifier sur une image. Tout prédicat qui ne peut être vérifié par la partition triviale est impossible. On connaît donc au moins une partition qui vérifie tout prédicat non impossible.

Il existe bien sûr un très grand nombre de partitions d'une image et il existe également généralement un très grand nombre de partitions qui vérifient le prédicat (il suffit, partant d'une telle partition, de subdiviser n'importe quelle région pour obtenir une nouvelle partition vérifiant le prédicat). On ne sait pas trouver toutes les partitions vérifiant le prédicat (la combinatoire est généralement inaccessible) mais on ne sait non plus généralement pas choisir entre des partitions vérifiant un prédicat. Des critères empiriques peuvent être utiles :

- le cardinal de la partition (à minimiser),
- la taille de la plus petite région (à maximiser),
- une « distance entre régions » (par exemple somme des distances entre zones adjacentes) (à maximiser).

Dans de nombreux cas par exemple, on ne recherche que des partitions maximales, c'est-à-dire telles que deux régions adjacentes ne vérifient pas le prédicat :

$$\forall(i,j : i \text{ et } j \text{ adjacents}) \Rightarrow \mathcal{P}(\mathcal{R}_i \cup \mathcal{R}_j) = \text{faux}$$

En l'absence de stratégie pour trouver la meilleure partition vérifiant un prédicat donné, nous allons décrire des stratégies de base qui sont très souvent utilisées.

12.2.1 La croissance de région

Les méthodes de croissance de région *(region growing)* n'aboutissent pas à des partitions car la propriété $\bigcup_i \mathcal{R}_i = image$ n'est généralement pas satisfaite. Partant de **germes**[2] on applique successivement à l'image des prédicats plus sévères que le prédicat $\mathcal{P}$. Ainsi on commence à associer aux germes les seuls pixels qui sont en très bon accord avec le prédicat. On réduit cette sévérité progressivement, et on se rapproche ainsi du prédicat objectif. La décision d'associer un pixel à une région se fait alors le plus souvent sans ambiguïté à moins que ses « distances » à deux régions soient égales (et en ce cas un choix quelconque est peu décisif). Plus importante est la décision de regrouper 2 régions qui sont adjacentes et vérifient le prédicat. Il a été montré qu'il convient alors d'être assez sévère pour fusionner des régions et qu'il est souvent préférable de traiter ce point lors d'une étape ultérieure en acceptant donc une **sur-segmentation** de l'image, plutôt qu'une fusion abusive qui ne serait plus récupérable.

Les méthodes de croissance de région cessent souvent de créer de nouveaux germes bien avant que le prédicat ne vaille $\mathcal{P}$ et rejettent les régions de cardinal 1. Ainsi tous les pixels ne se retrouvent pas dans des régions (cf. figure 12.1) [Adams et Bishof, 1994, Beaulieu et Goldberg, 1989, Chang et Li, 1994].

12.2.2 Le partage de région

Dans une démarche par partage de région *(region splitting),* on part de l'image entière (à laquelle généralement le prédicat $\mathcal{P}$ ne s'applique pas). On appelle $\mathcal{R}$ cette région. On lui applique plusieurs divisions δ produisant de nouvelles régions $\mathcal{R}_i^\delta$. On teste $\mathcal{P}$ sur chaque $\mathcal{R}_i^\delta$, et on retient la meilleure subdivision δ, c'est-à-dire :

- celle qui conduit à des sous-régions vérifiant toutes $\mathcal{P}$,
- ou celle qui donne le plus de sous-régions vérifiant toutes $\mathcal{P}$,
- ou, si aucune $\mathcal{R}_i^\delta$ ne vérifie $\mathcal{P}$, celle qui fournit la meilleure valeur à un critère $\mathcal{E}$ dit critère d'échec.

[2] les germes sont souvent des régions où le prédicat est trivialement vérifié, par exemple des zones où l'image est constante

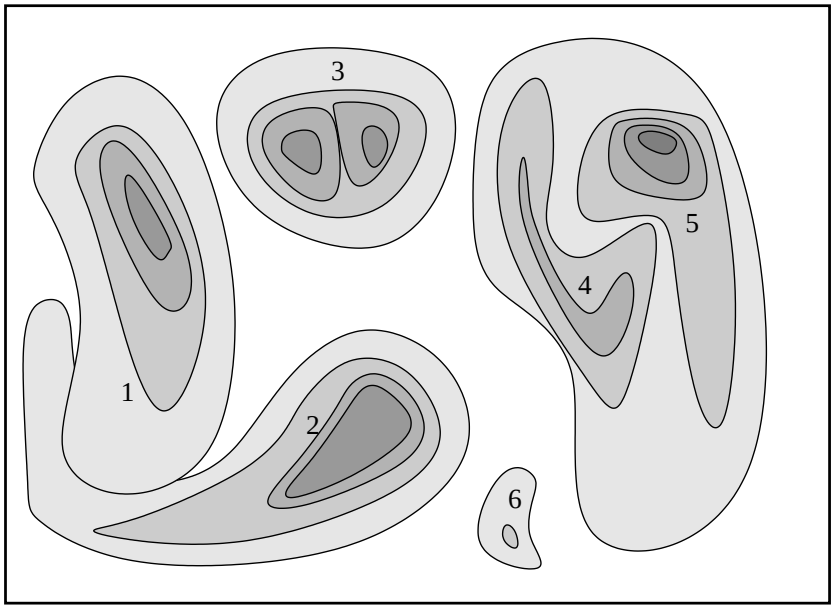


FIG. 12.1 – Croissance de région : 6 régions ont été obtenues par 4 prédicats successifs. La zone 6 n'a été créée qu'au second prédicat. La zone 3 regroupe 2 sous-régions qui ont été fusionnées car leur union vérifie le postulat et des critères annexes sur la forme résultante sont vérifiés. Ce n'est pas le cas de 1 et 2.

Pour δ choisi, chaque sous-région $\mathcal{R}_i^\delta$ ne vérifiant pas $\mathcal{P}$ devient alors une région $\mathcal{R}$ passible du traitement ci-dessus.

Les stratégies de subdivision sont peu nombreuses, on utilise généralement les 2 bipartitions régulières horizontale ou verticale (cf. figure 12.2), parfois des tripartitions régulières.

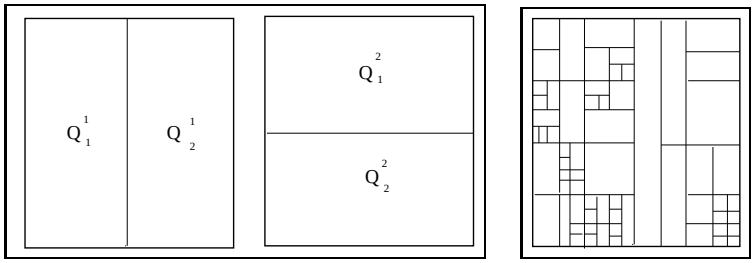


FIG. 12.2 – A gauche : partition d'une zone : on choisit entre la partition verticale ou la partition horizontale. A droite, à la fin du partage, l'image partitionnée.

Les critères d'échec $\mathcal{E}$ sont souvent les mesures de variance ou de dynamique qui sont testées dans les prédicats $\mathcal{P}$. Lorsque toutes les régions ont été récursivement testées, il peut se trouver que des zones adjacentes le long d'une frontière d'ordre supérieur vérifient le prédicat. On peut alors les réunir avec les mêmes précautions que celles signalées au paragraphe 12.2.1.

Les algorithmes de partage sont mal adaptés à une mise en œuvre informatique sur machine séquentielle car les calculs effectués sur tous les pixels de la zone $\mathcal{Q}$ ne sont généralement pas réutilisables au niveau inférieur $\mathcal{Q}_i$.

12.2.3 La réunion de région

Les techniques de réunion (*region merging*) prennent l'exact contre-pied des précédentes. Ce sont des méthodes ascendantes (*bottom-up*) lorsque les autres étaient descendantes (*top-down*). Les pixels sont visités systématiquement. Pour chaque carré de 2×2 pixels le postulat $\mathcal{P}$ est testé, et s'il est accepté les pixels sont regroupés en une région. Après le parcours de toute l'image, les groupes de 2×2 régions se voient appliquer le même test et, éventuellement, les mêmes conséquences (réunion en une région de niveau 2) [Zhu et Yuille, 1996].

Les tests de réunion de région sont fréquemment faits sur des tests statistiques [Saporta, 1978]. On se place souvent dans l'hypothèse de bruits gaussiens sur des fonctions à valeur moyenne constante. Les tests les plus utilisés sont :

- le Chi2 (χ^2),
- le test de Wilcoxon, il travaille sur les pixels triés par ordre croissant de niveaux de gris des deux régions. Pour chaque pixel de la liste issue de la région 1 on compte combien de pixels de 2 sont plus sombres. Ces nombres sont additionnés pour donner une variable U de distribution asymptotiquement gaussienne de moyenne $n_1 n_2/2$ et de variance $n_1 n_2(n_1 + n_2 + 1)/12$. On teste donc cette valeur U face à sa distribution asymptotique.
- le test de Student d'égalité des espérances, on teste la variable de Student :

$$T(n_1 + n_2 - 2) = \frac{(m_1 - m_2)\sqrt{n_1 + n_2 - 2}}{\sqrt{(n_1\sigma_1^2 + n_2\sigma_2^2)(1/n_1 + 1/n_2)}}$$

- le test de Fisher-Snedecor d'égalité des moyennes et des variances, on teste la variable de Fisher :

$$F(n_1 - 1; n_2 - 1) = \frac{n_1\sigma_1^2(n_2 - 1)}{n_2\sigma_2^2(n_1 - 1)}$$

12.2.4 Les pyramides

Afin de bénéficier des avantages des deux méthodes précédentes, Horowitz a proposé une approche par pyramide [Horowitz et Pavlidis, 1976]. L'image est représentée sur une pyramide appelée *quad-tree*, constituée de N niveaux, l'image originale étant au niveau 0. Chaque pixel au niveau ν a 4 fils au niveau $\nu - 1$. Le pixel au niveau ν est la moyenne de ses 4 fils. Un schéma plus complet a été proposé par Burt [Burt, 1981] où les passages entre niveaux se font par filtrage gaussien. Une abondante littérature a été proposée depuis pour améliorer ces filtrages par une approche en ondelettes.

La segmentation procède directement à un niveau intermédiaire (par exemple $\nu = 2$), et tous les pixels fils sont testés avec le prédicat $\mathcal{P}$. Si le prédicat n'est pas vérifié, le pixel considéré est étiqueté comme candidat à un partage. Sinon il est candidat à une réunion avec ses voisins. C'est donc une technique de partage et réunion, où l'on profite du passage au niveau supérieur (ici $\nu = 3$) pour accélérer la procédure (cf. figure 12.3).

Mais la stratégie du quad-tree est trop rigide et conduit à des partitions trop régulières que les techniques de croissance ne permettent plus de rattraper. Des techniques inspirées de la pyramide ont été proposées qui donnent de bien meilleurs résultats. Ainsi, la méthode proposée dans [Suk et Chung, 1993]), autorise beaucoup plus de fusion que les simples réunions de pixels 2×2. Douze fenêtres différentes sont autorisées, avec une priorité aux plus grandes (cf figure 12.4) (voir aussi [Chen et al., 1991]).

12.3 Les graphes d'adjacence

Les techniques par graphes d'adjacence sont beaucoup utilisées à partir de sur-segmentations (c'est-à-dire de segmentations où les zones sont subdivisées trop finement). Ces sur-segmentations sont par

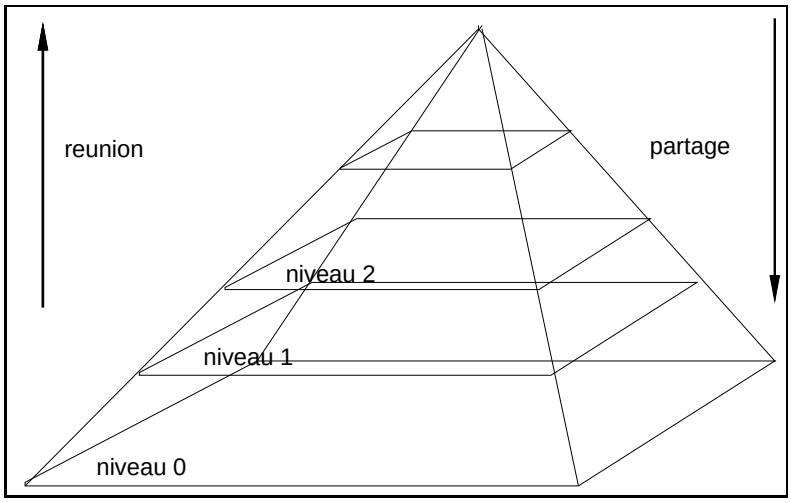


FIG. 12.3 – Méthode de pyramidage par quad-tree.

exemple le résultat d'un prédicat trop sévère dans la phase de segmentation où d'un algorithme très sensible aux variations locales comme la technique des lignes de partage des eaux (cf. chapitre 6 et [Schmitt et Mattioli, 1994b]).

L'idée de base consiste à plonger les régions obtenues dans une structure de graphe où une région est un nœud et un arc une relation d'adjacence. Puis on définit une fonction de similarité entre 2 nœuds. On trie tous les couples de nœuds adjacents dans une liste ordonnée. On regroupe les 2 meilleurs candidats. On remet à jour la liste et on itère.

La méthode proposée par Beaulieu et Goldberg [Beaulieu et Goldberg, 1989] est un bon exemple d'une telle technique. Le critère d'homogénéité $H(\mathcal{R}_i)$ d'une zone $\mathcal{R}_i$ est le résidu de son approximation par un polynôme de degré d fixé :

$$H(\mathcal{R}_i) = \sum_{(x,y)\in\mathcal{R}_i} \left[f(x,y) - \sum_{p+q\leq d} a^i_{p,q} x^p y^q \right]^2$$

et plus particulièrement d'une approximation par une constante. En partant d'une partition très fine en régions très homogènes, et en acceptant progressivement des réunions de régions de moins en moins semblables, on s'approche de l'optimisation du critère global de minimum de variance sur les partitions.

D'autres schémas plus complexes sont mis en œuvre, par exemple dans [Wang, 1998] pour obtenir des segmentations meilleures.

12.4 La méthode MDL = Minimum Description Length

C'est une technique issue de la théorie stochastique de l'information [Rissanen, 1984, Rissanen, 1987, Rissanen, 1989] pour optimiser la représentation de données. Elle a été reprise de façon simplifiée dans divers domaines du traitement des images (en particulier en codage). L'idée du MDL consiste à exploiter l'analogie, au sens de la théorie de l'information, entre longueur minimale de description et quantité d'information. On cherche donc à optimiser le choix d'un modèle pour décrire les données conduisant à la plus courte description :

– d'une part par le choix d'un modèle qui s'adapte bien aux données,

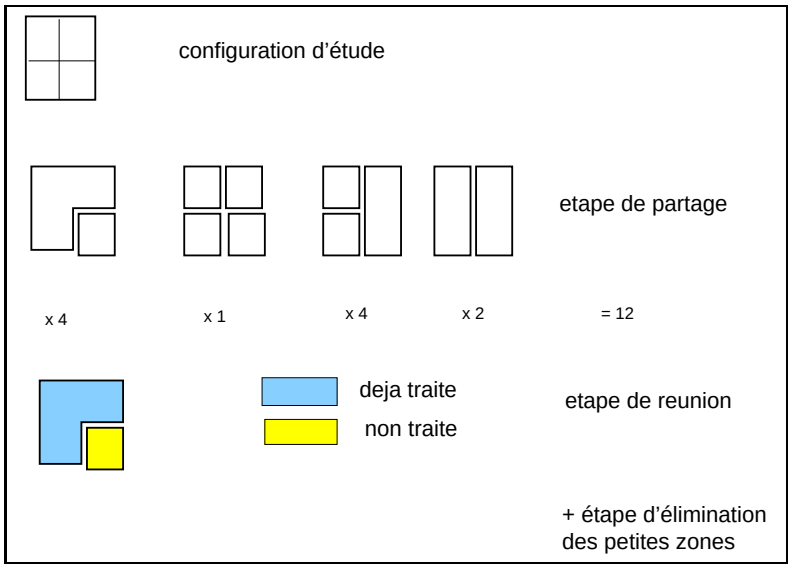


FIG. 12.4 – Les 3 étapes de la méthode de partage et réunion de Suk. Pour chaque étape il y a un prédicat différent. Celui du partage et de la réunion est généralement un prédicat de contraste, celui sur l'élimination des petites zones est un prédicat de taille. Les petites zones sont associées à la zone la plus proche en niveaux de gris.

– d'autre part par le choix d'un modèle simple qui nécessite peu de paramètres.
L'optimisation au sens du MDL procure un compromis entre ces deux termes. Dans le cas de la description d'une image, le modèle doit décrire les contours des régions et approcher au mieux l'image entre les contours.

On se place donc dans le domaine de la représentation paramétrique des données. On dispose d'un modèle dépendant de paramètres et on recherche le meilleur jeu de paramètres pour approcher les données. Soit $\phi = (\phi_1, ..., \phi_k)$ le vecteur à k composantes des paramètres. Soit x l'observée (l'image). Dans le cadre de l'estimation au maximum de vraisemblance, on cherche le meilleur $\phi \in \mathbb{R}^k$ qui maximise $p(x/\phi)$. Cela revient à maximiser la log-vraisemblance :

$$\mathcal{L}(x/\phi) \;=\; -\log\; p(x/\phi) \tag{12.4}$$

Si l'on dispose de plusieurs modèles, chacun ayant des jeux différents de ϕ_i, la transmission de la représentation de x nécessite la transmission d'une part des paramètres décrivant la forme dans ce modèle, d'autre part les éléments permettant de reconstruire le modèle puisqu'il existe de nombreux modèles potentiellement disponibles. On sait que selon la théorie de l'information, il existe des codes optimaux pour transmettre le premier terme en utilisant un nombre de bits égal au logarithme de la probabilité $p(x/\phi)$, c'est à dire le terme de l'équation 12.4. On devra ensuite transmettre le second terme. Pour rechercher la meilleure description au sens de la théorie de l'information, on minimisera donc une expression plus complète que celle utilisée pour la représentation au maximum de vraisemblance (équation 12.4) :

$$\mathcal{L}'(x/\phi) \;=\; -\log\; p(x/\phi) + \mathcal{L}(\phi) \;=\; \mathcal{L}(x/\phi) + \mathcal{L}(\phi) \tag{12.5}$$

où $\mathcal{L}(\phi)$ représente la quantité d'information nécessaire pour transmettre le modèle. Dans le cadre général de la classification ou du traitement du signal, il est parfois possible de mener une optimisation complète et explicite du MDL.

Utilisé dans le cadre de la segmentation d'image, le MDL introduit dans la segmentation un terme $\mathcal{L}'(x/\phi)$ de pénalité lors d'une représentation en régions. Il vient limiter d'une part le nombre de contours (le premier terme $\mathcal{L}(x/\phi)$), d'autre part la complexité des contours par le choix si possible d'un petit nombre de paramètres très simples à coder (le second terme : $\mathcal{L}(\phi)$).

Dans la pratique, il n'est généralement pas possible en traitement des images d'obtenir une solution explicite de l'équation 12.5 et l'on recherche très souvent des solutions itératives (par modification des lignes de contour) sous le contrôle du terme 12.5. Le MDL est utilisé en segmentation d'images de différentes façons. On exprime généralement :

1. le coût de la représentation d'une région par une fonction (par exemple constante). Ce coût s'exprime par le nombre de bits nécessaires à coder la constante plus le nombre de bits nécessaires à coder l'erreur résiduelle,
2. le coût de la représentation des contours (coût de codage de la chaîne des contours). Une chaîne de contours peut être codée de façons très diverses : Freeman (cf. chapitre 14.4), approximations polygonales, splines, processus de bords, etc. et conduira alors à des codages très différents.

Le MDL tend à procurer un compromis entre ces deux termes, c'est-à-dire à donner peu de contours réguliers mais au bon endroit. Les techniques mises en œuvre en MDL sont à base de champs de Markov (optimisation par recuit simulé) [Lee, 1998, Leclerc, 1989, Zhu et Yuille, 1996], ou d'optimisation de courbes [Darell et al., 1990, Keren et al., 1990].

12.5 L'approche de Mumford et Shah

12.5.1 Approche formelle

C'est une façon formelle de poser le problème de segmentation [Mumford et Shah, 1989] qui n'a malheureusement pas de solution exacte. Soit $f(x,y)$ l'image[3], de support $\mathcal{I}$, on la remplace sur des domaines $\mathcal{R}_i$ par des fonctions régulières $g_i(x,y)$ qui se rejoignent le long de contours Γ_j (cf. figure 12.5). La fonctionnelle à minimiser se compose de :

$$U(\Gamma,g,f) = \mu^2 \iint_{\mathcal{I}} (f(x,y)-g_i(x,y))^2 dxdy + \iint_{\mathcal{R}_i} ||\nabla g_i(x,y)||^2 dxdy + \nu \int_{\Gamma_i} dl \qquad (12.6)$$

On montre aisément que les 3 termes sont simultanément nécessaires sous peine de convertir l'expression en une formule triviale. Mumford et Shah ont déduit des propriétés particulières des solutions Γ_j et g, que nous examinons ci-dessous.

On sait résoudre ce problème dans quelques cas simples (par exemple si g_i est une constante qui ne prend que deux valeurs (généralement choisies comme +1 et -1), on retrouve le problème d'Ising qui a une solution - en théorie exacte - par recuit simulé (cf. chapitre 7)[4].

Mumford et Shah ont montré par ailleurs que l'équation générale 12.6 peut évoluer vers deux formes limites.

– Tout d'abord la minimisation de l'énergie U_o :

$$U_o(\Gamma,f) = \frac{1}{\mu^2} U(\Gamma,f,g) = \sum_i \iint_{\mathcal{R}_i} (f-g_i)^2 dxdy + \nu . \int_{\Gamma_j} dl$$

[3] Attention dans le texte original de Mumford et Shah, [Mumford et Shah, 1989] les rôles de f et g sont interchangés, afin de garder la cohérence de nos notations nous avons préféré désigner par f l'image et par g son approximation.

[4] On se souvient cependant que les conditions de convergence vers le maximum absolu par recuit simulé sont liées à une descente en température infiniment lente.

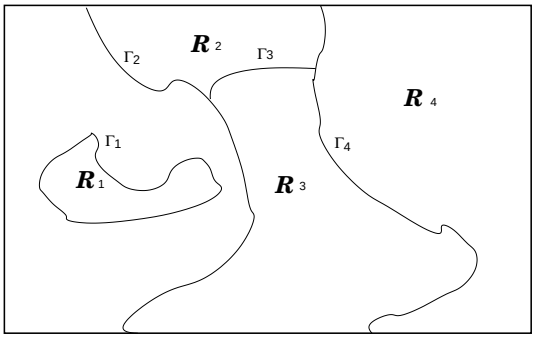


FIG. 12.5 – Approche de Mumford et Shah : on recouvre chaque domaine $\mathcal{D}_i$ d'une fonction continue, on cherche à minimiser la distance entre la fonction de représentation et l'image, ainsi que la longueur des côtés.

que l'on obtient en imposant à g d'être constante par morceaux sur les $\mathcal{R}_i$: $g(x,y) = g_i \ \ ssi \ \ (x,y) \in \mathcal{R}_i$. Il est alors aisé de montrer que $g_i = \frac{1}{s_i} \iint_{\mathcal{R}_i} f(x,y)dxdy$ où s_i représente la surface de $\mathcal{R}_i$ (c'est donc la valeur moyenne sur la surface).
U est la limite de U_o quand μ tend vers 0. La minimisation de U_o est un problème bien posé. Le problème d'Ising évoqué plus haut en est un sous-problème réduit au cas à 2 classes, le problème de Potts l'étend à un plus grand nombre de classes.

– Puis la minimisation de l'énergie :

$$U_\infty(\Gamma) = \int_\Gamma \left[\nu_\infty - \left(\frac{\partial f}{\partial n} \right)^2 \right] dl$$

où ν_∞ est une constante et $\frac{\partial f}{\partial n}$ mesure la composante normale à Γ du gradient de f.
Ce problème peut se ramener à un problème de géodésique : on cherche simultanément à déterminer les trajectoires Γ_i de longueur minimale (terme en ν_∞) et à maximiser le gradient transverse de f le long de Γ_i[5]. Ce problème est généralement mal posé si les Γ sont quelconques ; sur des contours de forme restreinte, le problème peut être bien posé.
On peut montrer que U_∞ est la limite de U quand μ tend vers l'infini.

On montre que si les Γ sont fixés (c'est-à-dire si l'on a fait par ailleurs une détection de contours), alors la minimisation de U est la recherche du minimum d'une fonction quadratique définie positive possédant un minimum unique. Elle est solution de :

$$\Delta g = \mu^2 (g - f) \tag{12.7}$$

avec pour conditions aux limites :

$$\frac{\partial g}{\partial n}|_{\Gamma_j} = 0$$

Si les contours Γ_j sont des courbes régulières (généralement on recherche des solutions parmi les courbes de classe $\mathcal{C}^2$)[6], on déduit alors les propriétés suivantes :

[5] On a également abordé ce problème lorsqu'on a évoqué les modèles géodésiques actifs (*cf.* paragraphe 11.4.4)
[6] Une courbe de classe $\mathcal{C}^n$ possède n dérivées continues.

- les contours Γ_j ne peuvent posséder que des points singuliers de deux types :
  - soit des points triples où 3 courbes se joignent à 120^o,
  - soit des points extrémités d'où démarre une courbe unique (démarrage d'un pli) ;
- les contours se rattachent aux bords de l'image selon des angles droits ;
- par ailleurs, les fonctions g ont une dérivée normale aux Γ_j horizontale le long des Γ_j.

12.5.2 Les variantes autour de la formulation de Mumford-Shah

Des variantes du problème de Mumford et Shah ont été proposées qui abordent le problème de nombreuses façons. Par exemple des méthodes déterministes issues de la mécanique et de la résistance des matériaux ont été proposées pour apporter des solutions en termes de *membranes* ou de *plaques minces* [Blake et Zisserman, 1987]. On cherche alors à résoudre l'équation 12.7. Ces solutions conduisent à des équations données par :

- dans le cas d'une membrane :

$$U(g,f) = \alpha_1 . \sum_i \iint_{\mathcal{D}_i} (f(x,y) - g_i(x,y))^2 dxdy + \alpha_2 . \sum_i \iint_{\mathcal{D}_i} (||\nabla g_i(x,y)||)^2 dxdy \quad (12.8)$$

- dans le cas d'une plaque mince :

$$U(g,f) = \alpha_1 . \sum_i \iint_{\mathcal{D}_i} (f(x,y) - g_i(x,y))^2 dxdy + \alpha_2 . \sum_i \iint_{\mathcal{D}_i} (\Delta g_i(x,y))^2 dxdy \quad (12.9)$$

  Il est également possible de prendre en compte dans ce second cas un terme de torsion (dérivées croisées) dans le terme de régularisation.

Dans les 2 cas les discontinuités Γ sont fixées (par exemple par un détecteur de contour préalable).

La résolution de ces problèmes (connaissant la position des contours on recherche la valeur en tout point de g_i) se fait traditionnellement par des techniques convolutionnelles. À chaque fonctionnelle est attachée une réponse impulsionnelle - ou fonction de Green - G. C'est-à-dire que, sur chaque domaine, $\mathcal{D}_i$, $g(x,y)$ est de la forme :

$$g(x,y) = \int \int_{\mathcal{D}_i} G(x,y,u,v).f(u,v)dudv$$

Par exemple en l'absence de contours, la membrane (équation 12.8) a pour fonction de Green K_o, fonction de Bessel modifiée de seconde espèce :

$$G(\rho = \sqrt{(x-x')^2 + (y-y')^2}) = \frac{1}{2\pi\lambda^2} K_o\left(\frac{\rho}{\lambda}\right)$$

équivalente à $\log(1/\rho)$ près de Γ et équivalente à $\frac{1}{\rho} exp(-\rho)$ pour ρ grand. C'est la solution de l'équation sans second membre issue de l'équation 12.7 :

$$g - \frac{\alpha_2}{\alpha_1} . \nabla^2 g = \delta(x,y)$$

La plaque mince (équation 12.9) a pour fonction de Green :

$$G(\rho) = \frac{i}{4\pi\mu^2} \left[K_o\left(\frac{\rho\sqrt{i}}{\mu}\right) - K_o\left(\frac{\rho}{\mu\sqrt{i}}\right) \right]$$

et elle est solution de l'équation :

$$g + \mu^4 \nabla^4 g = \delta(x, y)$$

avec $\mu^4 = \frac{\alpha_2}{\alpha_1}$.

Dans le cas où il n'y a pas de discontinuités de g le long des Γ, le problème est assez simple mais peu satisfaisant en termes de segmentation d'image [Grimson, 1981, Terzopoulos, 1983]. La solution est obtenue par des techniques de relaxation [Grimson, 1981] ou de relaxation multi-grilles [Terzopoulos, 1983] [7].

Dans le cas discret on étend les solutions continues par des approches par éléments finis permettant de calculer de façon discrète les dérivées (on est alors dans un schéma très semblable à celui adopté pour les contours actifs vus au paragraphe 11.4).

On définit un processus de bord b booléen (cf figure **??**) et on calcule les divers termes de l'énergie en fonction des valeurs $b = 0$ (pas de bord) ou $b = 1$ (présence d'un bord) de ce processus.

Si l'on ne connaît pas la position exacte des contours, on met en œuvre des techniques itératives et (souvent) sous-optimales qui partent de contours estimés et les améliorent. La résolution se fait par des techniques de Markov avec processus de bord explicites [Geman et Geman, 1984b] - cf paragraphe 12.1.5 - ou implicites et des schémas d'optimisation adaptés (par exemple la méthode GNC) ou des techniques à base de réseaux neuromimétiques de Hopfield. La méthode GNC (*Graduated Non Convexity* de Zisserman [Blake et Zisserman, 1987]), est particulièrement adaptée à ce problème. Elle remplace les fonctions d'énergie non convexes à optimiser par une famille de fonctions convexes qui, successivement optimisées par une technique de gradient, permettent d'atteindre l'optimum global.

Dans ces schémas, on peut aussi introduire un terme de pénalité sur la longueur des contours comme dans l'équation 12.6. On trouve dans la littérature des formes variées de cette énergie :

1. la plus simple est celle de Mumford et Shah et s'exprime par :

$$U = \int_\Gamma dl$$

2. une minimisation de la courbure des contours :

$$U = \int_\Gamma \left(\frac{d\theta}{dl} \right) dl$$

3. une pénalisation pour les seuls forts changements de direction des contours.

Des techniques où l'on mélange des termes de dérivées premières et secondes (comme dans les contours actifs - cf. paragraphe 11.4) sont également possibles, mais réputées moins bonnes.

[7] Les techniques multi-grilles sont des techniques d'optimisation qui utilisent la régularité spatiale de la solution pour trouver tout d'abord une solution exacte sur une grille d'échantillonnage très grossière, puis par un raffinement itératif de la solution, de la calculer sur des grilles de plus en plus fines pour aboutir à la résolution ultime de l'image.

Chapitre 14

Description de contours et de formes

Chapitre rédigé par Henri MAÎTRE

Dans de nombreuses applications de traitement des images, une fois l'image segmentée (on peut le faire, de la façon la plus simple, en seuillant l'image [Sahoo et al., 1988, Cham et al., 1998] ou à partir des méthodes décrites aux chapitres 11 et 12), on essaie de reconnaître les divers objets qui la composent à partir de leur seule silhouette. C'est particulièrement vrai dans les applications de tri dans lesquelles on est intéressé à saisir ou trier des objets arrivant sur un convoyeur[1], dans les applications de reconnaissance de caractères, ainsi que pour des applications de surveillance ou de guidage, par exemple en imagerie militaire.

Il est utile alors que l'utilisateur dispose d'une représentation de la forme vérifiant plusieurs propriétés :

1. une bonne fidélité à la forme initiale,
2. une bonne discrimination de formes différentes,
3. une bonne adaptation aux opérations de reconnaissance des formes, et en particulier une insensibilité aux déformations qui sont susceptibles d'entacher l'objet,
4. une certaine compacité pour permettre l'archivage de nombreuses formes et potentiellement de ces mêmes formes sous divers aspects.

De nombreuses représentations des formes ont été développées concurremment, chacune pour répondre à un problème parfois assez spécifique et donc mettant l'accent sur l'une ou l'autre des propriétés ci-dessus.

14.1 Fonction caractéristique

La représentation naturelle d'une forme dans une image est une représentation par fonction caractéristique, c'est à dire sous forme d'une image binaire, les seuls pixels non-nulles étant ceux couverts par la forme :

$$\begin{aligned} f(i,j) &= 0 \qquad \text{si } (i,j) \notin \text{ objet} \\ &= 1 \qquad \text{si } (i,j) \in \text{ objet} \end{aligned}$$

Dans une image où de nombreux objets existent, on utilise une image d'étiquettes (ou de *labels*) qui généralise la notion de fonction caractéristique :

$$f(i,j) = k \qquad \text{si } (i,j) \in \text{ objet } k$$

[1] Le tri se partage en tri planaire (où les objets sont préalablement mis à plat, généralement sur un tapis roulant), et en tri en vrac, où ils sont présentés dans un désordre total.

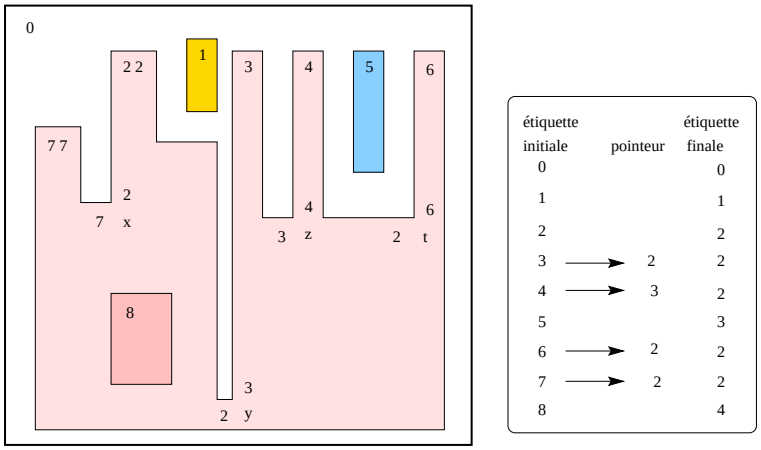


FIG. 14.1 – Etiquetage d'une image en 4 connexité. L'image de gauche est le résultat du premier passage jusqu'au point noté x. Les étiquettes ont été attribuées à chaque point soit en donnant une nouvelle étiquette si les 2 antécédents du point sont à 0, soit en donnant l'étiquette des antécédents si ces antécédents ont une étiquette semblable différente de 0. En x les étiquettes sont de valeur différente. Le pixel x se voit attribué l'étiquette 2 car $2 = \min(2,7)$, le pixel y l'étiquette 2, le pixel z l'étiquette 3 et le pixel t l'étiquette 2. Le point x crée le pointeur $7 \sim 2$, y le pointeur $3 \sim 2$, z $4 \sim 3$ et t $6 \sim 2$. Après un second passage sur l'image, on détermine la liste finale des étiquettes de 0 à 4. Pour recréer une image d'étiquettes en conformité avec cette liste, il faut donc faire un deuxième passage sur l'image.

Une image d'étiquettes résulte d'une procédure d'étiquetage. Elle se fait par le choix d'une connexité qui permet de définir les propriétés de topologie des divers objets (4- ou 8- connexité sur les trames carrées, 6-connexité en trame hexagonale). L'étiquetage vient attribuer une étiquette semblable à tous les points connexes au sens de la segmentation. Par exemple en 4-connexité un algorithme d'étiquetage fonctionne de la façon suivante (cf. figure 14.1).

1. une liste d'étiquettes est initialisée à 0 ;
2. le balayage (par exemple vidéo) de l'image est initialisé ; au premier point est attribuée l'étiquette 0 ;
3. un point nouveau est pris dans l'ordre du balayage ; on examine sa connexité à son voisin supérieur et à son voisin de gauche (ses antécédents) :
   - si le point courant est connexe à ses deux antécédents (au sens de la segmentation), et si ses deux antécédents ont la même étiquette, on lui attribue l'étiquette de ces antécédents ;
   - si le point courant est est connexe à ses deux antécédents, mais que ses deux antécédents n'ont pas la même étiquette, on donne au point l'étiquette la plus petite parmi les deux, puis dans la liste des étiquettes, on crée un pointeur qui renvoie de l'étiquette la plus grande vers l'étiquette la plus petite ;
   - si le point courant est connexe à l'un seulement de ses antécédents, on lui attribue l'étiquette de cet antécédent ;
   - si le point courant n'est pas connexe à ses deux antécédents, on incrémente la liste des étiquettes, puis on attribue au point courant cette nouvelle étiquette ;
4. on retourne en 3 jusqu'à atteindre le dernier point de l'image.

Lorsque l'ensemble de l'image a été ainsi traité (après donc un passage complet sur l'image), on réorganise l'image des étiquettes. Pour cela on parcourt cette image et chaque valeur d'étiquette est

remplacée par l'étiquette terminale rencontrée en remontant la liste des pointeurs. La liste des étiquettes est alors filtrée de toutes les étiquettes qui possèdent au moins un pointeur. On constitue ainsi une liste d'étiquettes qui occupe tous les entiers entre 0 et k_{max} s'il y a k_{max} objets se découpant sur un fond.

14.2 Description de formes

Les descripteurs (ou paramètres) de forme sont des nombres qui représentent chaque forme et permettent de les classer (par exemple par nuées dynamiques ou plans séparateurs). Ces paramètres ont été abondamment proposés dans la littérature [Coster et Chermant, 1985], beaucoup étant adaptés à des formes particulières. Citons les plus utilisés, par exemple :

- le rapport iso-périmétrique, proportionnel au rapport du carré du périmètre de l'objet à sa surface (dans des images continues, il est maximum pour le cercle) ;
- l'indice d'allongement [Schmitt et Mattioli, 1994b], proportionnel au rapport du carré du diamètre géodésique à la surface de l'objet, (minimal pour les disques au sens du voisinage choisi).

Les paramètres de forme sont bien adaptés pour des objets de taille assez grande car ils ont souvent été définis pour des formes continues. Pour des objets de petite taille (quelques pixels), la discrétisation du maillage induit des comportements souvent peu satisfaisants.

14.2.1 Représentation par les moments

Connaissant l'objet par sa fonction caractéristique $f(x, y)$, une représentation classique de sa forme consiste à en mesurer les divers moments :

$$M_{mn} = \iint x^m y^n f(x, y) dx dy$$

En particulier les moments centrés (rapportés au centre de gravité $(X_g\ Y_g)$ de la forme) sont invariants par translation :

$$\bar{M}_{mn} = \iint (x - X_g)^m (y - Y_g)^n f(x, y) dx dy$$

Sur une image discrète, ces moments s'écrivent :

$$\bar{M}_{mn} = \frac{1}{J^m K^n} \sum_{j=1}^{J} \sum_{k=1}^{K} (j - X_g)^m (k - Y_g)^n f(j, k)$$

Les moments d'inertie (valeurs propres de la matrice d'inertie, matrice 2×2 de terme courant $\bar{M}_{mn},\ m + n = 2$) sont invariants par rotation. Les moments d'inertie normés par la plus grande valeur propre sont invariants par similitude (rotation et facteur d'échelle). Les moments d'inertie décrivent bien l'allongement de formes régulières comme des ellipses ou des distributions gaussiennes. Ils sont plus ambigus sur des formes complexes (cf. figure 14.2).

14.2.2 Représentation par les moments invariants

Des moments invariants ont été proposés (nommés moments de Hilbert), invariants par translation, rotation et changement d'échelle (dans la limite de l'approximation par une maille discrète) [Hu, 1962] . Ils se construisent à partir du moment normé :

$$N_{nm} = \frac{J^m K^n}{\bar{M}_{00}^{\alpha}} \bar{M}_{mn}$$

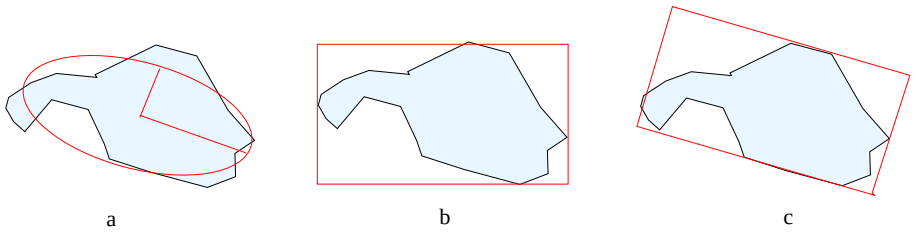


FIG. 14.2 – Description de formes par : a - ellipse et axes d'inertie, b - boîte englobante, c - boîte minimale.

où $\alpha = \frac{1}{2}(m+n)+1$, par les formules :

$$
\begin{aligned}
h_1 &= N_{20} + N_{02} \\
h_2 &= (N_{20} - N_{02})^2 + 4N_{11}^2 \\
h_3 &= (N_{30} - 3N_{12})^2 + (N_{03} - 3N_{21})^2 \\
h_4 &= (N_{30} + N_{12})^2 + (N_{03} - N_{21})^2 \\
h_5 &= (N_{30} - 3N_{12})(N_{30} + N_{12})[(N_{30} + N_{12})^2 - 3(N_{03} + N_{21})^2] \\
&\quad +(3N_{21} - N_{03})(N_{03} + N_{21})[3(N_{30} + N_{12})^2 - (N_{03} + N_{21})^2] \\
h_6 &= (N_{20} - N_{02})[(N_{30} + N_{12})^2 - (N_{03} + N_{21})^2] \\
&\quad +4N_{11}(N_{30} + N_{12})(N_{03} + N_{21}) \\
h_7 &= (3N_{12} - N_{30})(N_{03} + N_{21})[3(N_{30} + N_{12})^2 - (N_{03} + N_{21})^2] \\
&\quad +(3N_{21} - N_{03})(N_{30} + N_{12})[(N_{30} + N_{12})^2 - 3(N_{03} + N_{21})^2]
\end{aligned}
$$

14.2.3 Boîtes englobantes et boîtes minimales

Lorsque les formes sont plus irréguliéres, on préfère souvent des descriptions de l'allongement par la forme de la boîte englobante :

- soit la boîte alignée sur les axes et donc simplement définie par ses dimensions $(x_{max} - x_{min})$ et $(y_{max} - y_{min})$,
- soit la boîte orientée selon le diamètre minimal de l'objet (de calcul un peu plus complexe) (cf. figure 14.2).

14.3 Polygones de Guzman

C'est l'une des méthodes les plus anciennes [Guzman, 1968]. L'approche de Guzman consiste à envelopper l'objet à reconstruire dans des « boîtes » de formes de plus en plus précisément adaptées. Elle peut donc s'appliquer soit à partir d'une image d'étiquettes de la forme à analyser, soit à partir d'une liste des pixels de contour de la forme. Les formes prototypes auxquelles seront comparées les objets sont construites sur un maillage carré de façon systématique et se classent en niveaux en fonction de la longueur de leur périmètre. Cette longueur (exprimée en pixels) étant obligatoirement paire, on décrit chaque niveau par le demi-cardinal du nombre de pixels constituant la frontière. Dans chaque niveau, les divers prototypes sont repérés par un indice qui les identifie uniquement. Ainsi, la forme la plus simple (le carré unitaire) a une longueur 4 (et donc appartient au niveau 2). C'est la forme $\Phi(2,1)$. Afin de tenir compte des symétries et des rotations de $\pi/2$, toutes les formes identiques par rotation modulo $\pi/2$

et toutes les formes identiques par symétrie droite (par rapport à un axe horizontal ou vertical) sont rapportées à une même prototype du dictionnaire des formes (cf. figure 14.3). Un objet quelconque est donc décrit par une succession d'indices décrivant l'objet à divers niveaux de résolution. Deux objets sont généralement identiques jusqu'à un niveau n, puis différents à partir du niveau $n+1$ (cf. figure 14.4).

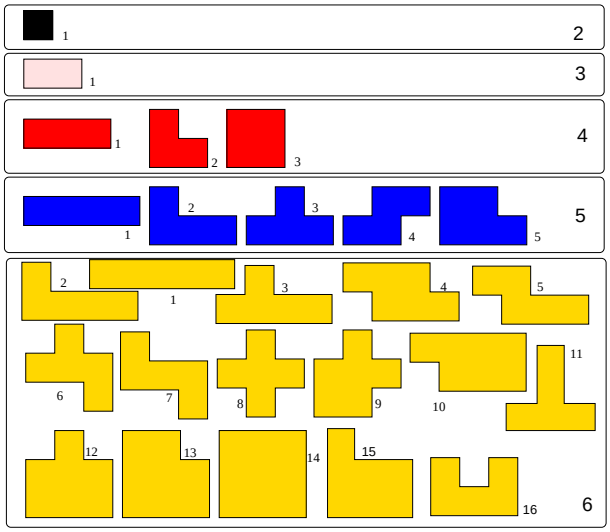


FIG. 14.3 – Les premières formes du dictionnaire de Guzman.

Cette approche rencontre plusieurs limites :
- pour les ordres grands, les formes deviennent très nombreuses et le dictionnaire trop grand. La recherche du représentant se faisant par comparaison au prototype devient très longue ;
- les distances d'une forme à un prototype ne s'imposent pas de façon unique (cf. figure 14.5), laissant place à des classements différents selon le critère adopté ;
- les relations « verticales » entre niveaux sont complexes et ne permettent pas d'accélérer le parcours de l'arbre de façon commode ;
- certaines formes se retrouvent identiques dans des niveaux différents, à un facteur d'échelle près (par exemple le carré élémentaire se retrouve dans tous les niveaux impairs).

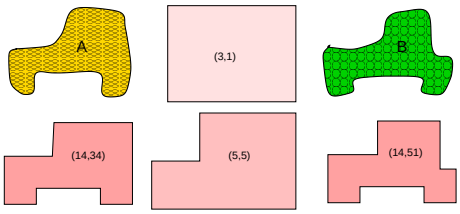


FIG. 14.4 – Représentation de Guzman. Les 2 formes A et B sont semblables aux niveaux de représentation 3 et 5, mais diffèrent au niveau 14, où l'une est représentée par le prototype 34 et l'autre par le prototype 51.

L'approche de Guzman a été abandonnée aujourd'hui, mais elle apporte des concepts intéressants à la description des formes :
- la notion de raffinement de la description,
- la notion de similarité entre tous les objets « vus de très loin »,

– la notion d'invariance par rotation, symétrie et changement d'échelle que l'on pourrait étendre éventuellement à d'autres transformations.

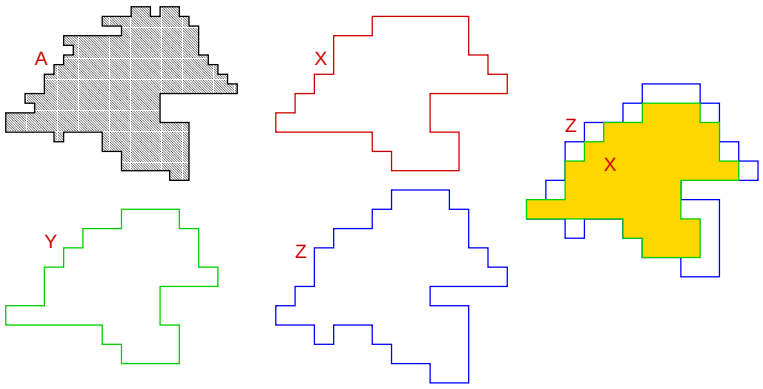


FIG. 14.5 – L'arbitraire de l'approximation des formes : la forme A est l'objet. Les 3 formes X, Y et Z en sont des approximations. La distance d'une représentation à la forme initiale peut se mesurer par la surface entre les 2 contours. Il y a de nombreuses formes à distance minimale. X est une forme la plus proche en terme de surface et « la plus simple » au sens du nombre d'angles droits lorsqu'il y a plusieurs solutions de même distance. Y est la forme la plus proche par défaut (entièrement contenue dans A), Z est celle par excès. Z et Y sont superposées à droite.

14.4 Chaînes de Freeman

C'est la méthode la plus ancienne de description des contours dans les images et aussi la plus utilisée encore aujourd'hui [Freeman, 1961, Freeman, 1977]. Si elle est moins utilisée en reconnaissance des formes qu'elle le fut, elle est très fréquemment utilisée dans des applications récentes comme la transmission des images par zones (par exemple dans MPEG-).

14.4.1 Définition

C'est une technique de représentation des directions du contour (on code la direction le long du contour dans un repère absolu lors du parcours du contour à partir d'une origine donnée). Les directions peuvent se représenter en 4-connexité (codage sur 2 bits) ou en 8-connexité (codage sur 3 bits)[2].

Le codage d'un contour se fait donc de la façon suivante :

1. transmission des coordonnées absolues du point de départ,
2. transmission de la liste des codes de déplacement d'un point du contour au suivant sur le maillage.

Les codes des contours sont donnés par la figure 14.6.

Dans d'autres techniques, on code de façon différentielle le changement de direction d'un point au suivant. Cela peut se justifier en codage sur 3 bits si une direction est très dominante par rapport aux autres.

Deux exemples de codages par chaînes de Freeman sont présentés sur la figure 14.7 (le codage du positionnement absolu a été omis). Dans le cas de la 4-connexité, le codage de la suite de contours occupe

[2]des essais ont été faits pour coder également les chaînes de Freeman sur 4 bits en ajoutant les directions reliant le point central à la seconde couronne des voisins qui ne sont pas dans la direction des voisins de la première couronne. Mais cette approche n'a pas eu un très grand succès.

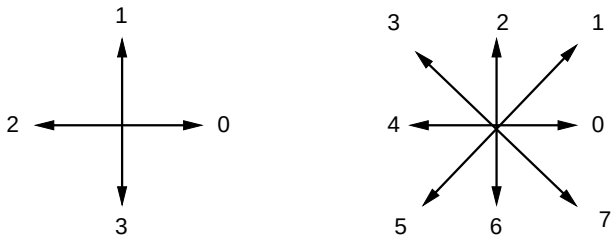


FIG. 14.6 – Les codes de Freeman en 4-connexité (à gauche) et en 8-connexité (à droite).

124 bits (62 × 2), tandis qu'en 8 connexité elle occupe 126 bits (42 × 3). Il n'y a pas de règle générale sur l'efficacité des représentations en termes de compression. Si la courbe est très complexe (beaucoup de changements de direction), le codage en 8-connexité est généralement plus compact. La qualité du codage est cependant toujours au moins aussi bonne en 8-connexité.

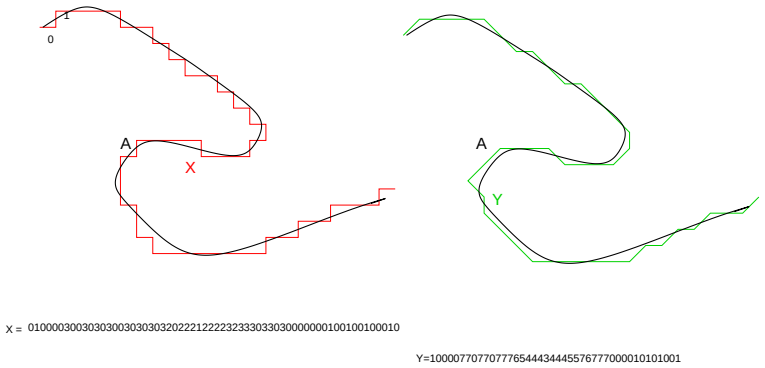


FIG. 14.7 – Codage de Freeman. Approximation de la courbe A par la chaîne X (en 4-connexité) et par la chaîne Y (en 8-connexité).

	0	1	2	3	4	5	6	7
0			1	2		6	7	
1					2		0	
2	1				3	4		0
3	2						4	
4		2	3				5	6
5	6		4					
6	7	0		4	5			
7			0		6			

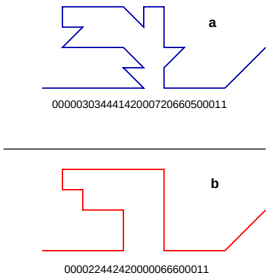


FIG. 14.8 – En 8-connexité, il est possible de simplifier les courbes en remplaçant certains couples de descripteurs par le descripteur de la table ci-dessus. Dans la figure de droite, la figure b est une simplifiée de la figure a par remplacement des seuls angles aigus. Il existe plusieurs simplifiées selon l'ordre de substitution que l'on adopte.

14.4.2 Les propriétés des chaînes de Freeman

Les chaînes de Freeman se prêtent à un certain nombre de manipulations commodes.

1. On obtient une dilatation de la courbe d'un facteur k en répétant $k$fois chaque descripteur.
2. On ne peut généralement pas réduire une courbe sans distorsion.
3. On fait tourner une courbe de $k \times \frac{2\pi}{n}$ (dans le cas d'une chaîne de Freeman en $n-$connexité) en ajoutant (ou retranchant) k modulo n à la chaîne initiale.
4. On mesure la longueur d'une courbe par les formules suivantes :
   - en 4-connexité : $L =$ nombre de descripteurs,
   - en 8-connexité : $L =$ nombre de descripteurs pairs $+ \sqrt{2}$ nombre de descripteurs impairs.
5. Inversion d'un chemin : on inverse tous les descripteurs et on inverse la séquence. L'inverse d'un descripteur j est $\bar{j} = n/2 + j \mod(n)$.
   exemple en 4-connexité : $X = 001321 \rightarrow \bar{X} = 301322$.
6. Simplification d'un chemin : c'est un chemin dont on a supprimé des détails sans changer globalement la forme. Cela s'obtient en remplaçant des séquences de p descripteurs consécutifs par des descripteurs équivalents reliant les mêmes points (cf. figure 14.8) :
   exemple en 4-connexité : $\{012\} \rightarrow \{1\}$,
   en 8-connexité : $\{03\} \rightarrow \{2\}$.
7. Réduction d'un chemin : c'est l'un des chemins de longueur minimale reliant les 2 extrémités de la courbe initiale. On associe 2 par 2 des descripteurs inverses de la chaîne et on les supprime.
   exemple en 4-connexité : $X = \{00132122\} \rightarrow \tilde{X} = \{21\}$. On obtient tous les chemins réduits en changeant l'ordre des associations.
   en 8-connexité la réduction est un peu plus complexe car il faut aussi regrouper des ensembles de 3 ou 4 descripteurs qui s'annulent : exemple $\{025\}$ ou $\{7225\}$.
8. Fermeture d'un contour : on teste la fermeture d'un contour en vérifiant que la chaîne réduite est nulle.
   On ferme un contour en lui ajoutant le chemin inverse d'un de ses chemins réduits. Il y a beaucoup d'autres fermetures possibles que par addition de l'inverse d'un réduit, mais les fermetures obtenues ainsi sont de longueur minimale. Il y a d'autres fermetures minimales que celles obtenues par addition de l'inverse d'un réduit.
9. Courbe qui s'intersecte : pour savoir si une courbe décrite par sa chaîne se recoupe, on procède à une réduction de chemin systématique en partant de son origine et en testant si chaque nouveau descripteur possède un inverse dans la chaîne déjà parcourue. Si à un instant la chaîne déjà parcourue se réduit à une chaîne nulle, on a trouvé un point double.
10. Changement d'origine : le changement de l'origine d'une chaîne de longueur L revient à une permutations circulaires des descripteurs modulo L.
11. Sens de parcours d'un contour : le contour fermé d'une forme simplement connexe peut être décrit dans le sens direct ou dans le sens inverse. Pour connaître le sens d'un contour fermé, on réduit ce contour jusqu'à n'avoir que 4 descripteurs. En 4-connexité, il n'existe que 2 chaînes possibles (et celles qui s'en déduisent par changement d'origine) :
    - le carré direct : $X_D = 0123$,
    - le carré inverse : $X_I = 0321$.

    En 8-connexité les configurations sont un peu plus nombreuses.
12. Surface d'une région simplement connexe : la technique est assez complexe mais ne nécessite pas la reconstruction de la région [Freeman, 1977].

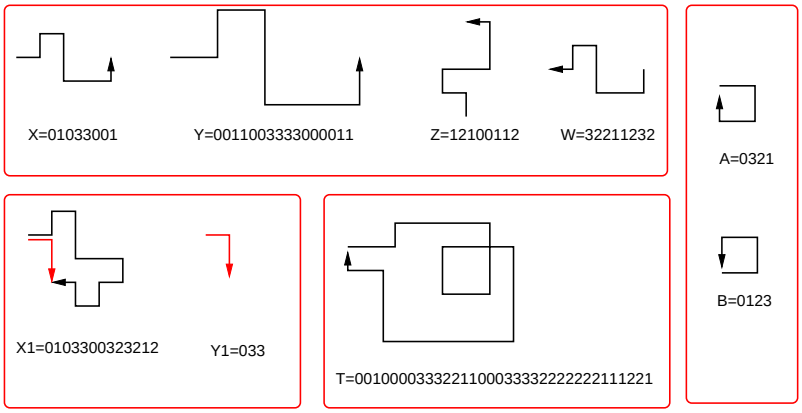


FIG. 14.9 – Quelques propriétés des chaînes de Freeman : X = chaîne originale, Y = contour double, Z = rotation de $\pi/2$, W = contour inverse. X1 = chaîne originale et Y1 = chaîne réduite. T = exemple de boucle dans une chaîne. A et B sont les deux contours élémentaires (inverse et direct) en 4-connexité.

14.4.3 Reconnaissance des formes par des chaînes de Freeman

Pour que 2 formes X et Y soient identiques, il faut tout d'abord qu'elles aient le même nombre de descripteurs. Puis, comme elles peuvent différer par le sens de description, il faut comparer X à Y d'une part et à $\bar{Y}$ d'autre part. Enfin, comme elles peuvent avoir des origines différentes, il convient de tester les deux chaînes descripteur à descripteur sous toutes les hypothèses de permutation circulaires. Des techniques rapides ont été développées pour faire ces recherches (d'une complexité sous-linéaire en L), [Boyer et Moore, 1977, Miclet, 1984] [3] .

Dans le cas où l'on ne recherche pas une similarité exacte mais seulement approchée (par exemple pour prendre en compte la présence du bruit lors de la détection des contours), on est amené à utiliser des techniques de **distance d'édition** pour lesquels on chiffre les coûts des erreurs possibles : omission de contours, adjonction de contours, permutation de descripteurs, etc. On utilise alors généralement des techniques de programmation dynamique (algorithme de Wagner et Fisher [Wagner et Fisher, 1974,

[3] Ce sont des automates finis semblables à ceux qui sont utilisés en particulier pour les compilateurs et les éditeurs de texte.

Miclet, 1984], élargi pour des substitutions de plusieurs descripteurs consécutifs dans [Lowrance et Wagner, 1975]) ou de relaxation (algorithme de Davis) pour retrouver la distance minimale entre X et Y.

14.5 Descripteurs de Fourier

14.5.1 Descripteur par tangente

Dans cette approche on considère le contour comme une courbe continue qui peut être décrite par son abscisse curviligne s à partir d'une origine A choisie (cf. Fig 14.10). On paramètre la courbe par l'angle fait par le vecteur tangent en chaque point et celui au point origine : $\phi(s)$ et on crée la variable réduite t qui prend ses valeurs entre 0 et 2π : $t = 2\pi s/L$, où L est la longueur complète du contour. On construit alors la fonction $\Phi(t)$:

$$\Phi(t) = \phi\left[\frac{2\pi.s}{L}\right] - \frac{2\pi s}{L}$$

le terme correctif prenant en compte l'enroulement de 2π de la tangente pour un tour de contour. La fonction $\Phi(t)$ est une fonction périodique sur $[0, 2\pi[$ qui admet donc une série de Fourier : $\Phi(t) = \sum_{k=0}^{\infty} a_k \exp(-ikt)$. On appelle descripteurs de Fourier l'ensemble des modules des a_k : $\{|a_k|\}$.

Ils bénéficient des propriétés suivantes :

1. ils sont invariants par translation de la forme,
2. ils sont invariants par changement d'échelle (puisque t est normalisé),
3. ils sont invariants par rotation, puisque l'on a choisi la différence d'angle entre 2 tangentes,
4. ils sont invariants par changement d'origine, car passer d'une origine A à une origine A' revient à :
   - retrancher $\Phi(t_{A'})$ à toutes les valeurs $\Phi(t)$,
   - changer t en $t - t_{A'}$.

   Donc : $\Phi_{A'}(t) = \Phi_A(t) * \delta(t - t_{A'}) - \Phi(t_{A'})$ et : $a_k \to a_k \exp(-kt_{A'})$ si $k \neq 0$

Pour comparer des formes on compare leurs descripteurs par ordre croissant. Si de plus on veut simplifier le contour, il suffit de supprimer les ordres k élevés dans le développement. malheureusement, dans cette représentation, si un contour est fermé, le contour obtenu en filtrant les hautes fréquences ne l'est généralement plus. C'est pourquoi on préfère souvent les descripteurs de Fourier par représentation complexe qui n'ont pas ce défaut.

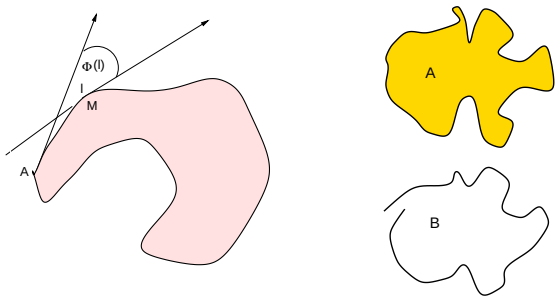


FIG. 14.10 – A gauche : descripteurs de Fourier par tangente. Le point de départ A est choisi arbitrairement. Son vecteur tangent sert de référence à la paramétrisation de la courbe. A droite : après troncature du développement de Fourier, la forme A, initialement fermée devient B, plus régulière, mais non fermée.

14.5.2 Représentation complexe

Dans cette représentation, on décrit la forme par un ensemble $\{M_j\}$ de points de contours, et on représente la forme dans le plan complexe. On attache donc à chaque M_j un nombre complexe $z_j = x_j + \mathrm{i}y_j$. On appelle alors descripteurs de Fourier, les coefficients de la TF Z de z :

$$Z_k = \sum_{j=1}^{N} z_j \exp(-2\pi \mathrm{i} jk)$$

Les coefficients Z_k, pour $k \in [-N/2+1, N/2]$, jouissent d'intéressantes propriétés [Bertrand et al., 1982].

1. Pour $k = 0$, Z_0 est le centre de gravité de la forme. Si l'on l'omet, la description est invariante par translation.
2. Si tous les Z_k sont nuls sauf pour $k = 1$, la forme est un cercle de rayon Z_1 (ou un polygone régulier à N côtés), donc Z_1 joue le rôle de facteur d'échelle. La normalisation par Z_1 rend la forme invariante par homothétie.
3. Les coefficients $Z_{|k|}$ et $Z_{|1-k|}$ (pour $k \neq 1$ et $k \neq 0$) jouent des rôles symétriques (mais opposés) de la façon suivante :
   - l'ordre k indique le nombre d'actions sur le cercle unité (entre 0 et 2π) : 1 action pour $k = 2$ et $k = -1$, 2 actions pour $k = 3$ et $k = -2$, 3 actions pour $k = 4$ et $k = -3$, etc. Ces actions sont réparties régulièrement autour du cercle unité,
   - les valeurs de $k > 0$ indiquent des actions de traction sur la courbe, pour la déformer vers l'extérieur du cercle unité, les valeurs de $k < 0$ indiquent des actions de pression sur la courbe, pour creuser la courbe vers son centre,
   - la phase du nombre complexe Z_k : ϕ_k exprime le lieu, sur le cercle unité, où s'exerce l'action.
4. Plus les coefficients sont nombreux, plus la forme est complexe. Et plus les coefficients sont élevés plus les détails sont fins sur la courbe.

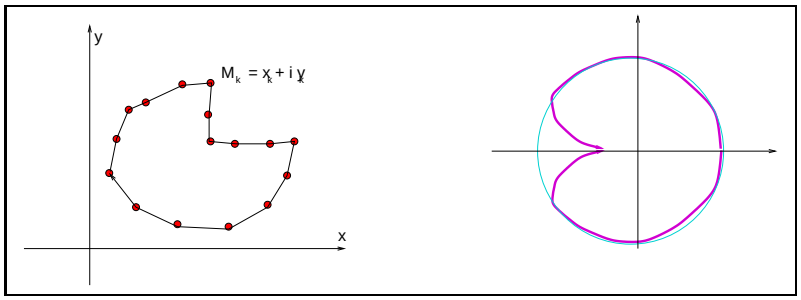


FIG. 14.11 – A gauche : descripteurs de Fourier par représentation complexe. Le point courant M_k est décrit par ses coordonnées complexes dans le plan image. A droite : le cercle résulte de la prise en compte du seul coefficient Z_1, la courbe présentée résulte de l'adjonction d'un coefficient de pression (donc à $k < 0$), en 1 seul point (donc $k = 2$ ou $k = -1$, ici $k = -1$), de déphasage nul (puisque situé en π).

On voit que les descripteurs de Fourier par représentation complexe ont le même type de comportement que les descripteurs par tangente. Ils sont mieux adaptés aux formes discrètes puisqu'ils garantissent toujours que la forme demeure fermée après troncature du développement de Fourier. Ils peuvent être invariants par rotation si l'on s'intéresse aux seuls modules des coefficients Z_k.

Parmi leurs inconvénients, il faut noter que l'on ne peut pas aisément garantir qu'un contour de forme simplement connexe ne donnera pas, après troncature, un contour qui s'auto-intersectera.

14.6 Approximations polynomiales

Il y a essentiellement 3 types de telles approximations :

1. les approximations analytiques par ajustements de nuages de points par des polynômes du premier ordre (conduisant donc à des polygones) ;
2. les approximations polygonales par des critères géométriques obtenues en parcourant la courbe des points ordonnés selon une abscisse curviligne ;
3. les approximations par des polynômes de degré > 1 et en particulier les approximations par des fonctions splines.

Les approximations polygonales des formes décrites par des points de contour sont les représentations les plus employées. Dans les cas les plus simples, on se contente de relier 2 à 2 des points de contour dans un ordre déterminé au préalable (donc selon une abscisse curviligne croissante). Deux problèmes se posent alors :

- peut-on éliminer d'éventuels points parasites qui n'appartiennent pas au contour ?
- peut-on faire l'économie de certains sommets de la ligne polygonale pour obtenir des formes plus simples ?

Ces problèmes seront abordés dans le paragraphe 14.6.2.

Mais avant d'en arriver à cette situation simple, se posent les problèmes difficiles de déterminer quels points appartiennent à un même contour et dans quel ordre les relier lorsque ces points sont issus d'un détecteur de contour par nature local. C'est ce que nous abordons tout d'abord.

14.6.1 Approximation d'un nuage de points par une droite unique

C'est un problème bien classique mais qui mérite pourtant un peu d'attention. Soit $M_i = (x_i, y_i)$ les points, en nombre N, que l'on cherche à approcher par une droite. Il y a 2 façons de prendre le problème.

Approximation par régression linéaire

C'est une approche aux moindres carrés : on recherche la droite $\Delta : y = a_0 + a_1 x$ qui minimise la distance (cf. Fig 14.12) :

$$d_1^2 = \sum_i^N [y_i - (a_0 + a_1 x_i)]^2 \tag{14.1}$$

La solution est donnée par : $A = X^{\#} Y = (X^t X)^{-1} X^t Y$, où $X^{\#}$ dénote la matrice pseudo-inverse de la matrice X, où X, Y et A sont donnés par :

$$X = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ .. & .. \\ 1 & x_N \end{bmatrix}$$

$$Y = [y_1,\ y_2,\ ..,\ y_N]^t$$

$$A = [a_0, a_1]^t$$

Ces formules s'étendent très aisément aux espaces de dimensions supérieures, ainsi qu'aux approximations par des polynômes d'ordre plus élevé. la distance minimisée est celle mesurée selon le seul axe y. C'est donc une mesure généralement mal adaptée en traitement d'image, puisque x et y y jouent habituellement un rôle équivalent. On lui préfère donc les méthodes par axe d'inertie.

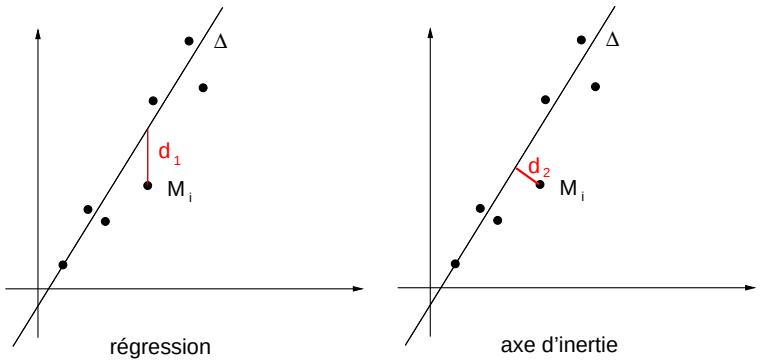


FIG. 14.12 – Deux approximations linéaires aux moindres carrés minimisant des distances différentes : à gauche par régression linéaire, à droite par axe d'inertie.

Approximation par axe principal d'inertie

C'est aussi une approche aux moindres carrés, mais on minimise dans ce cas la somme des distances de tous les points à la droite Δ :

$$d_2^2 = \sum_{i=1}^{N} \frac{[(a_0 + a_1 x_i - y_i)]^2}{a_1^2 + 1} \tag{14.2}$$

C'est l'équation de l'axe d'inertie des points, qui passe par leur centre de gravité X_g, Y_g et qui est donné comme vecteur propre de plus grande valeur propre de la matrice de forme quadratique :

$$S = \sum_{N} V_i V_i^t = \begin{bmatrix} \Sigma x_i x_i & \Sigma x_i y_i \\ \Sigma x_i y_i & \Sigma y_i y_i \end{bmatrix}$$

Ces équations s'écrivent sans problème en dimensions supérieures (pour estimer des variétés d'ordre variable). Elles se transcrivent beaucoup plus difficilement à des polynômes d'ordre plus grand car on ne sait pas, en règle générale, exprimer la distance d'un point courant à une telle fonction.

Estimations robustes

Si l'on repose le problème précédent dans le cadre plus général de l'estimation robuste, on est amené à considérer ce problème de façon un peu plus complète [Meer et al., 1991, Rousseeuw et Leroy, 1987]. On cherche ici la meilleure droite représentant au mieux l'ensemble des points M_i sous l'hypothèse d'un bruit entachant la position des points.

Lorsque le bruit est gaussien, les estimations aux moindres carrés nous assurent d'une bonne qualité de l'estimée. Mais il faut souvent tenir compte de la présence, parmi les points, de détections erronées dont les écarts à la droite ne sont pas gaussiens (ce sont des *outliers*[4]).

Une première façon de prendre en compte ces points consiste à faire une première estimation aux moindres carrés, puis à éliminer les points trop éloignés de la droite (par exemple à l'aide d'un test statistique tel celui de Cramer-von Mises[5] . Ces méthodes ont reçu une base théorique dans l'approche

[4] on appelle *outlier* un point qui n'est pas régi par la même distribution statistique que les points dont on recherche une approximation.

[5] Le test de Cramer-von Mises calcule une distance entre la loi gaussienne théorique et la répartition expérimentale des erreurs. Un point est rejeté si sa distance à la distribution des autres points est trop improbable sous l'hypothèse gaussienne [Saporta, 1990].

dite par **moindres carrés tronqués** (LTS : *Least Trimmed Squares*) ainsi que toutes ses formes dérivées : moindres quantiles carrés (LQS), moindre k^e carré (LKS) [Lee et al., 1998].

Mais cette approche est réputée dangereuse (on peut éliminer successivement les points appartenant à la droite si un *outlier* est trop éloigné de celle-ci). On lui préfère des techniques d'estimation par médiane ou par médianes itérées, plus coûteuses en temps de calcul, mais capables d'éliminer beaucoup plus de points erronés.

Dans une estimation par médiane, on choisit 2 points quelconques M_i et M_j de l'ensemble. On leur associe une droite qui fournit des coefficients a_{ij} et b_{ij}. Par combinatoire sur l'ensemble des points, on obtient deux ensembles de coefficients : $a = \{a_{ij}\}$ et $b = \{b_{ij}\}$. Les ensembles a et b sont triés séparément et l'on choisit les valeurs médianes $\bar{a}$ et $\bar{b}$ comme estimateurs des paramètres de la droite. Le filtrage médian peut tolérer jusqu'à $50\% - \epsilon$ d'*outliers*. Dans le cas de l'estimation simultanée de deux variables, l'estimation par médiane permet de tolèrer jusqu'à 25% d'*outliers* $(0, 5^2)$.

De nombreux estimateurs robustes ont été calculés à partir de la notion de médiane. Ainsi, le minimum des médianes des distances quadratiques (LMS *Least Median of Squares*) détermine le centre d'un nuage de points en calculant pour chaque point la distance à tous les autres, puis en caractérisant la distance de ce point au nuage par la valeur moyenne des distances et enfin en choisissant comme centre du nuage celui dont la médiane est minimale [Rousseeuw et Leroy, 1987].

D'autres approches robustes reviennent à une estimation par filtrage (comme le font la moyenne ou la régression), mais en pondérant chaque point i par un coefficient $\rho(x_i)$ qui tend vers 0 pour des points loin de la solution. Une autre approche consiste à intégrer cette pondération des points lointains directement dans la distance (c'est le cas des M-estimateurs [Huber, 1981]). Dans ce cas, les « distances » [6] croîssent moins vite que la loi quadratique ou décroîssent même pour tendre vers 0 pour les grandes distances. On les appelle des M-estimateurs.

Une autre famille d'estimateurs robustes s'appuie sur des recherches systématiques. c'est par exemple le cas de la méthode RanSac (*Random Sample Consensus*) [Fischler et Bolles, 1981]. Cette méthode consiste à choisir 2 points et à adopter la droite qui les joint comme approximation. On mesure alors le nombre de points qui sont en accord avec cette hypothèse (par exemple à l'aide d'un critère de distance du point à la droite). On choisit finalement le couple de points qui conduit au plus fort consensus. On montre qu'il n'est pas nécessaire de tester tous les couples mais qu'un petit nombre choisis aléatoirement (de l'ordre de quelques pourcents) est usuellement suffisant.

Estimation d'un mélange de droites

Si l'on sait que n droites existent dans l'ensemble des points, il est possible de faire une classification au sens des nuées dynamiques par exemple. Pour cela on initialise le processus en choisissant n droites (issues par exemple de l'ensemble des points par tirage aléatoire) représentées par leurs paramètres $\{a_0^i, a_1^i, i = 1, \ldots, n\}$. On attribue chaque point à la droite qui minimise la distance 14.1 ou mieux 14.2. Après classification de tous les points on estime pour chaque droite ses paramètres $\{a_0^i, a_1^i\}$ par l'une des méthodes vues ci-dessus. On itère le processus enchaînant classification et estimation. On converge vers une solution qui dépend assez fortement de l'initialisation.

Une autre famille de méthodes utilise la version floue des nuées dynamiques (les C-moyennes-floues ou *Fuzzy-C-means*), dans laquelle chaque point k « appartient » à l'une des droites i avec une appartenance $\mu_i(k)$ fonction de sa distance à la droite. Pour tout point k, on vérifie : $\sum_{i=1}^{n} \mu_i(k) = 1$ [Bezdek, 1981].

[6] Les distances ainsi définies ne vérifient en fait plus les propriétés d'une distance.

Extension de l'estimation à des coniques

Dans le cas où l'on recherche des formes représentées par des cercles ou des ellipses, on peut choisir une paramétrisation matricielle de la forme quadratique les représentant. Chaque conique indicée par i est alors décrite par son centre q_i, son rayon r_i et sa matrice d'ellipticité A_i :

$$[(\mathbf{x} - q_i)^t A_i (\mathbf{x} - q_i)] = r_i^2$$

où $\mathbf{x}$ est le point courant du plan. La distance d'un point $\mathbf{x}_k$ à cette conique s'exprime par :

$$d_i^2(k) = \left[[(\mathbf{x}_k - q_i)^t A_i (\mathbf{x}_k - q_i)]^{1/2} - r_i \right]^2$$

L'estimation des paramètres se fait de la même façon que précédemment à partir d'une conique initiale souvent circulaire.

14.6.2 Approximations polygonales, simplification de contours polygonaux

Lorsque l'on dispose d'une courbe continue ou finement échantillonnée et que l'on souhaite la réduire à une ligne polygonale, on dispose de très nombreux algorithmes qui proposent soit des critères d'approximation soit des mises en œuvre différents. Le plus connu est l'algorithme de la corde.

Algorithme de la corde (ou de Ramer)

C'est un processus de subdivision qui peut être entrepris soit de façon récursive (chaque segment créé fait l'objet d'une nouvelle subdivision), soit de façon itérative (la courbe Γ est considérée globalement à chaque étape). C'est cette dernière version que nous examinons sur l'exemple de la figure 14.13. Les sommets du polygone sont choisis successivement comme les points de Γ les plus éloignés des cordes précédemment tirées. Le processus s'arrête lorsque la nouvelle distance candidate est inférieure à un seuil ϵ fixé. Très employé dans de nombreuses applications par la simplicité de sa mise en œuvre, l'algorithme de la corde n'est pas très rapide. Il ne garantit pas non plus une convergence uniforme vers la courbe finale car, dans son implémentation itérative, il se peut que la distance à l'étape n soit supérieure à celle à l'étape $n-1$.

Algorithme de Dunham

Dans cette méthode on cherche à supprimer les cordes successives qui sont presqu'alignées. On se fixe donc une tolérance angulaire ϵ. Partant de A, on couvrira par une même corde tous les points de la courbe qui se trouvent dans la tolérance et l'on choisira le premier point du polygone comme le dernier point de cet ensemble. Le processus est répété en ce point.

Algorithme de Wahl et Danielsson

Son objectif est de minimiser l'aire laissée entre la courbe et la ligne polygonale.

Algorithme progressif

Enfin, un algorithme progressif utilise le critère de distance de tout point de la courbe au segment d'approximation, mais dans un schéma progressif. Dans ce schéma, partant d'une extrémité, la courbe est parcourue, et le point courant est retenu comme extrémité d'un segment d'approximation s'il est le dernier point visité permettant une approximation de tous les points déjà visités, à une tolérance ϵ donnée.

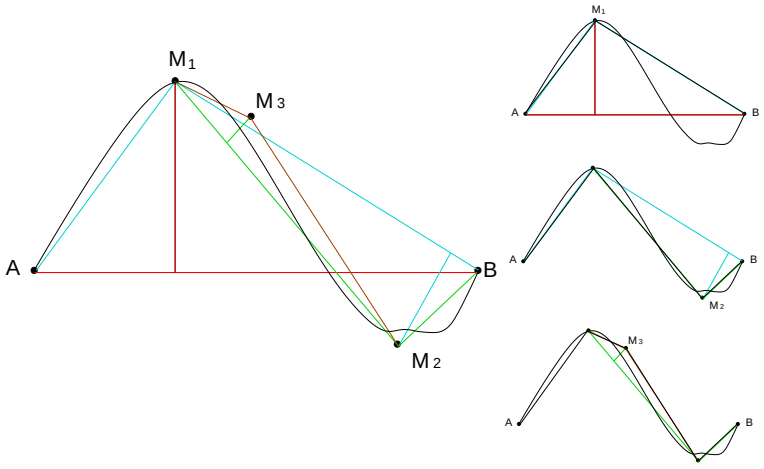


FIG. 14.13 – Algorithme de la corde. Les points M_1, M_2 et M_3 sont successivement sélectionnés pour créer la ligne polygonale représentant la courbe AB. Le critère de sélection est la distance maximale à la corde polygonale précédemment obtenue.

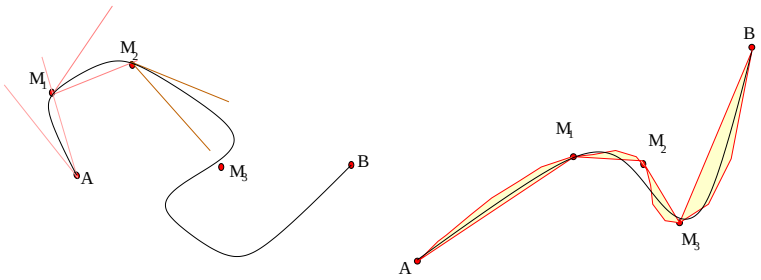


FIG. 14.14 – Algorithme de Dunham, à gauche, les points M_1, M_2 et M_3 sont choisis lorsque la tolérance angulaire ϵ est insuffisante à approximer la courbe. A droite, algorithme de Wahl et Danielsson : il cherche un minimum des surfaces laissées entre la ligne polygonale et la courbe.

14.6.3 Approximation par des splines

L'approximation d'un ensemble de points de contours par des polynômes d'ordre plus élevé que les droites est une bonne façon de représenter ce contour [Chalmond, 2000]. Le choix du degré du polynôme se fait généralement de façon à assurer des propriétés de continuité au contour obtenu :

- les segments de droite (degré 1) assurent la continuité du contour,
- les approximations par polynômes d'ordre 2 permettent d'avoir des dérivées continues,
- les polynômes d'ordre 3 permettent d'avoir des courbures continues.

Les fonctions splines sont de bons candidats pour ce type d'approximation. Nous en avons déjà vu l'usage à diverses occasions : au chapitre 10.4.2 lorsque nous avons discuté du rééchantillonnage des images, au chapitre 12.5.2 lorsque nous avons estimé des fonctions régulières lors de segmentation par régions. Deux types de fonctions peuvent être utilisées [Unser et al., 1993a, Unser, 1999] :

1. les **fonctions interpolantes** qui passent exactement par tous les points de contours qui sont utilisés pour calculer la spline ;
2. les **fonctions approximantes** qui ne passent pas nécessairement par les points mais qui s'en approchent de façon contrôlée.

Approximation

Dans le cas des fonctions approximantes, on définit la spline d'ordre k par morceaux en fonction d'une variable continue u, et des m points de contrôle P_i qui la déterminent. Q_i et P_i sont des fonctions de même dimension : des vecteurs de $\mathbb{R}^2$ si les points P_i sont définis par leurs coordonnées du plan ou des vecteurs de $\mathbb{R}^3$ dans l'espace.

Le morceau i est défini par :

$$Q_i(u) = \sum_{r=0}^{k-1} P_{i+r} b_r^k(u) \qquad i = 0, ...m - k + 1 \tag{14.3}$$

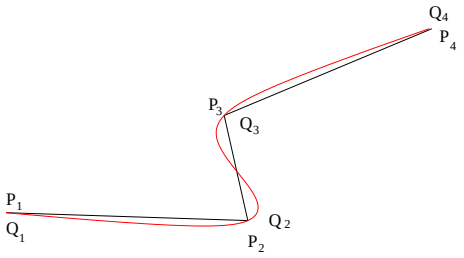


FIG. 14.15 – Dans le cas d'une interpolation spline, les jonctions entre les morceaux de splines sont les points de contrôle eux-mêmes.

Les fonctions $b_r^k(u)$ sont définies à partir des fonctions $\beta^k(u)$ qui ont été introduites au chapitre 10.4.2 équations 10.32 et suivantes :

$$b_r^k(u) = \beta^k \left[u - \frac{r-1}{2} \right]$$

Les positions des points de raccord entre splines sont données par $u = 0$ dans l'équation 14.3 :

$$Q_i(0) = \sum_{r=0}^{k-1} P_{i+r} b_r^k(0) \tag{14.4}$$

Il y a de nombreuses façons de définir les variables u. La plus naturelle consiste à considérer que les échantillons sont régulièrement distribués en fonction de u (attention cela conduit à une paramétrisation où u n'est pas l'abscisse curviligne le long du contour), alors $b_0^k(0) = 0$ et $b_{k-1}^k(1) = 0$, on peut réécrire cette équation [Goshtasby et al., 1990] sous la forme :

$$Q_i = \sum_{r=0}^{k-2} P_{i+r} b_r^k(0)$$

qui prend la forme d'une convolution : $Q = P * H_k$. Ceci montre que les Q sont des points filtrés issus des P.

Interpolation

Dans ce cas, les coefficients des splines de l'équation 14.3 sont inconnus et l'équation devient :

$$Q_i(u) = \sum_{r=0}^{k-1} c_{i+r} b_r^k(u) \qquad i = 0, ...m - k + 1$$

et les coefficients inconnus c_i sont déterminés en contraignant les splines à passer par les points de contrôle P_i :

$$P_i(0) = \sum_{r=0}^{k-1} c_{i+r} b_r^k(0)$$

dont on a vu au chapitre 10.4.2 les divers modes de résolution soit par inversion de matrice soit par filtrage (cf. figure 14.15).

14.7 Transformation de Hough

La transformation de Hough est un outil de traitement des images dont l'usage déborde largement l'application pour laquelle nous la présentons maintenant [Ballard, 1978, Sklansky, 1978, Maître, 1985b]. Ces applications seront vues plus tard. Nous nous intéressons actuellement au sous-problème suivant :

Connaissant un ensemble de pixels P (potentiellement bruité) appartenant à une frontière que l'on sait polygonale, comment déterminer le nombre de segments impliqués dans cette ligne polygonale et leur position ?

Exprimé dans les termes que nous avons vus précédemment, on se retrouve devant un difficile problème de **mélange**. Les solutions combinatoires (où l'on recherche systématiquement les classes possibles) sont généralement inacceptables. La transformation de Hough nous propose au contraire une solution élégante.

Elle consiste à transformer un problème inconnu (retrouver des droites) en un problème mieux connu : retrouver des nuages de points. Ceci s'obtient en associant à l'espace de l'image (dénoté $\mathcal{I}$ et défini par ses variables d'espaces $\{x, y\}$), un espace de paramètres, dénoté par $\mathcal{H}$ (dans le cas où l'on recherche des droites représentées par l'équation $y = ax + b$, on a $\mathcal{H} = \{a, b\}$). Pour faire cette transformation il y a 2 façons : la transformation de Hough de *1 à m* et la transformation de Hough de *m à 1*.

14.7.1 Définitions

Transformation de *1 à m*

Dans cette définition, on associe à tout point $M_i = (x_i, y_i)$ de P toutes les droites du plan. Elles sont définies dans $\mathcal{H}$ par : $b = -x_i a + y_i$, ce qui est l'équation d'une droite. A un point de $\mathcal{I}$ est associée une droite de $\mathcal{H}$ (d'où le nom de *1 à m* : à 1 point sont associés m points, *many* en anglais) et à un point de $\mathcal{H}$ une droite de $\mathcal{I}$.

Lorsque l'on transforme tous les points de P par la transformation, on associe à $\mathcal{I}$ un ensemble de droites qui, idéalement, se coupent en des points H_k. Le nombre de transformations à faire est $\nu = N = Card(P)$. Les transformés des H_k sont les droites cherchées dans $\mathcal{I}$. Ces points H_k s'obtiennent aisément en ne conservant que les intersections des droites de $\mathcal{H}$ et en recherchant les nuages d'intersections (figure 14.16).

Transformation de *m à 1*

Dans cette définition, on choisit d'associer à tout bi-point $\{M_i, M_j\}$, une droite Δ_{ij} de $\mathcal{I}$ qui se transforme en un unique point[7] Q_{ij} de $\mathcal{H}$. Les coordonnées a_{ij} et b_{ij} du point Q_{ij} se déduisent immédiatement de l'équation de Δ_{ij}. En combinant tous les couples de points de P, on obtient tous les points Q de $\mathcal{H}$. Le nombre de transformations à accomplir est $\nu' = \frac{1}{2}N(N-1)$. L'espace de Hough est alors constitué de

[7] la dénomination « *m à 1* » se traduirait alors précisément par « 2 à 1 », attention, m n'a pas la même signification dans les 2 définitions.

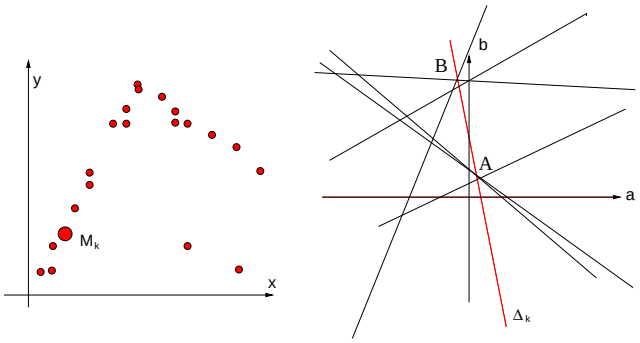


FIG. 14.16 – Transformation de Hough de *1 à m*. Un point M_k est transformé en une droite Δ_k de l'espace des paramètres $\{a, b\}$. On a représenté quelques-unes seulement de toutes les droites existantes. Elles définiront finalement 2 points d'intersection, l'un en A, l'autre en B.

ν' points que l'on identifie immédiatement aux intersections des ν droites obtenues par la transformation de *1 à m*.

On se ramène donc, comme précédemment, à la recherche des nuages de points les plus denses du plan $\mathcal{H}$.

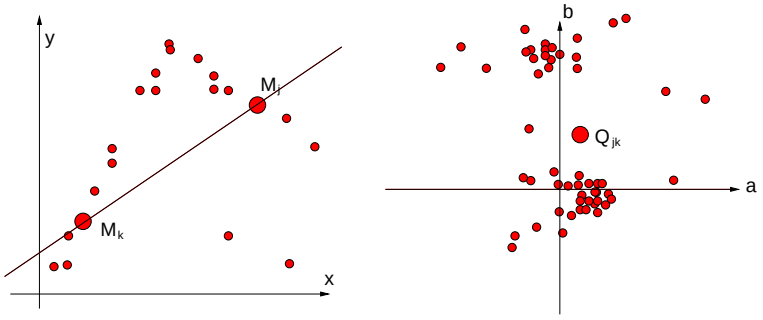


FIG. 14.17 – Transformation de Hough de *m à 1*. Deux points M_j, M_k définissent une droite qui se transforme en 1 point Q_{jk} de l'espace des paramètres (a, b).

Passage d'une TH à l'autre

Il est assez simple de montrer que l'accumulateur de la TH de m à 1 se déduit de celui de la TH de 1 à m par un seuillage au niveau 1. Ces deux transformations sont donc tout à fait équivalentes dans leurs résultats.

Généralisation

On appellera transformation de Hough associée à une forme paramétrée par les variables $\{a_i\}$ la transformation qui fait passer de l'espace image $\mathcal{I}$ à l'espace des paramètres $\mathcal{H}$.

La transformation est de *1 à m* si on associe à un point de $\mathcal{I}$ la variété de $\mathcal{H}$ décrivant toutes les formes passant par ce point. La transformation est de *m à 1* si l'on associe à toutes les combinaisons m points de $\mathcal{I}$ une seule forme (donc un point unique de $\mathcal{H}$).

Il existe généralement de nombreuses transformations de *n à m* ($n < m$), où l'on se place dans une situation intermédiaire : en associant n points de $\mathcal{I}$ on limite le sous-espace d'arrivée à une partie de la variété de la transformation de *1 à m*. Par exemple, on peut utiliser simultanément la position d'un point de contour et la direction de la tangente en ce point. Ces problèmes n'étant pas directement liés à la recherche de contours, des exemples seront vus plus loin.

TH et filtrage adapté

Sklansky [Sklansky, 1978] a démontré qu'il y a équivalence entre la TH et la détection par filtrage adapté (par corrélation) dans une situation particulière mais fréquente. Si les seules inconnues d'un problème de détection sont les deux translations du plan (δx et δy), c'est à dire si la forme ne subit ni changement d'échelle ni rotation, ni transformation perspective. Soit $f(x,y)$ la forme caractéristique de l'objet disponible ($f(x,y) = 1$ si (x,y) appartient au contour de l'objet, et 0 sinon). Soit $g(x,y)$ la forme recherchée. Le filtrage adapté conduit à une détection optimale au sens de l'erreur quadratique moyenne, en recherchant les maximums de la fonction de corrélation :

$$C_{fg}(\delta x, \delta y) = \iint f(x,y)g(x-\delta x, y-\delta y)dxdy = f(x,y) * g(-x,-y)$$

Cette démonstration est assez simple si l'on voit que chaque point M de coordonnées (x,y) de l'image f est candidat à être associé à un point quelconque de g de coordonnées (u,v) si l'on décale g du vecteur $x-u, y-v$ (cf. figure 14.18). Cette contribution à l'accumulateur de position $\delta x = x-u, \delta y = y-v$ vaut exactement : $f(x,y)g(u,v)$. Sommé sur toute la courbe f on obtient :

$$\begin{aligned} \gamma(u,v) = \gamma(-\delta x, -\delta y) &= \iint f(x,y)g(x-\delta x, y-\delta v)dxdy \\ &= f(x,y) * g(-x,-y) \end{aligned}$$

Ce qui est bien le résultat cherché. En fait on montre que la TH agit en pratique en inversant l'ordre des intégrales par rapport au filtrage adapté. Celui-ci procède usuellement en calculant l'intégrale ci-dessus pour un décalage u, v donné, en sommant sur tous les points du plan tandis que la TH choisit un point du plan et incrémente toutes les intégrales (les compteurs) qui lui sont asssociées par un déplacement u, v quelconque. Cette inversion des opérations peut permettre de tirer profit d'a prioris importants pour accélerer les calculs.

Grâce à cette propriété la TH peut hériter des nombreux résultats obtenus en filtrage adapté : pour prendre en compte le rôle du bruit si celui-ci à une densité de probabilité connue, pour déterminer la taille des accumulateurs lorsque la PSF est connue, etc. [Maître, 1985b, Princen et al., 1992].

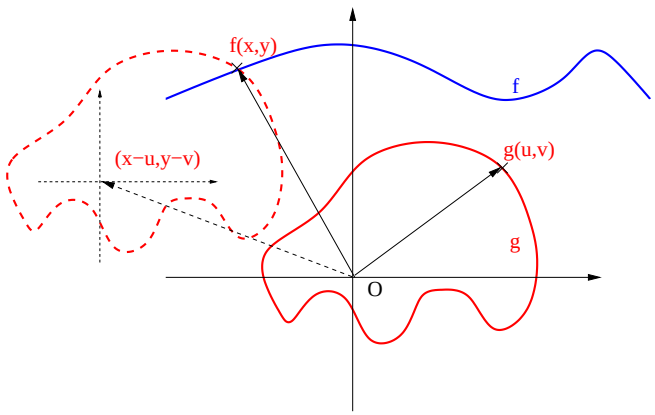


FIG. 14.18 – Transformation de Hough : le point de coordonnées x, y de f sera associé au point $g(u, v)$ de g si l'on fait subir à cette courbe g une translation de $(x - u, y - v)$.

14.7.2 Mise en œuvre de la TH

Discrétisation de $\mathcal{H}$

Dans la pratique le plan $\mathcal{H}$ est discrétisé. Chaque cellule de position a_i, b_i de largeur $\delta a, \delta b$ est appelée accumulateur. Elle contiendra un nombre d'autant plus important que la droite $y = a_i x + b_i$ sera plus probable dans $\mathcal{I}$.

Au début de la transformation, tous les accumulateurs sont à zéro. Les points sont alors visités successivement (individuellement dans la TH de *1 à m*, combinatoirement pour les autres). Chaque hypothèse donne naissance à des votes pour des cellules particulières. Pour chaque vote, l'accumulateur de la cellule est incrémenté de 1.

Lorsque tous les candidats ont voté, on recherche les accumulateurs de $\mathcal{H}$ de comptes localement maximaux.

Le choix de la taille des accumulateurs est donc un problème délicat.

1. Des accumulateurs trop petits recevront très peu de votes et donneront des statistiques peu significatives. On ne saura décider de la position d'un maximum entre plusieurs cellules voisines de comptes faibles et proches.
2. Des accumulateurs trop grands donneront une mauvaise précision dans la détection de la droite.

Par ailleurs, la mise en œuvre de la TH nécessite un espace mémoire de taille $\frac{\Delta a \Delta b}{\delta a \delta b}$ où Δa et Δb représentent les domaines de variation de a et b.

Des solutions ont donc été proposées pour adapter localement la taille des accumulateurs (très précis

près des lieux où les votes sont nombreux). Cela peut se faire dynamiquement par des approches de subdivision récursive des cellules (selon le principe des **quad-tree** cf. 12.2.4), ou par des passages successifs sur l'ensemble des points de P, le premier passage permettant de dimensionner les cellules.

Ce problème est particulièrement important dans le cas où l'on recherche des formes possédant de nombreux paramètres, puisque l'espace $\mathcal{H}$ est l'espace produit de tous ces paramètres. Il est alors de grande dimension.

Paramétrisation des formes

Le choix d'une bonne paramétrisation des formes recherchées apparaît immédiatement à la lecture du paragraphe ci-dessus si l'on souhaite appliquer la démarche proposée à la recherche de droites. Dans le cas d'une paramétrisation cartésienne ($y = ax + b$), les domaines de variation de a et b sont potentiellement de moins l'infini à plus l'infini, conduisant à un insoluble problème de résolution de cellule. Par ailleurs, si l'on examine la variable a par exemple, on voit que, statistiquement, elle décrit 25% des droites du plan pour $a \in [0, 1[$ et 25% pour $a \in]1, \infty[$, conduisant donc à des cellules qui seront très inégalement remplies si leur taille est uniforme.

On peut cependant choisir une paramétrisation plus heureuse pour la droite, par exemple la paramétrisation normale (cf. figure 14.19). Dans ce cas la droite $y = ax + b$ est repérée par la distance ρ et l'angle θ.

La transformation de *1 à m* s'écrit : $\rho = x_i \cos\theta + y_i \sin\theta$, et la forme associée à un point est donc une sinusoïde dans l'espace $\rho = f(\theta)$, d'équation :

$$\rho = \sqrt{x_i^2 + y_i^2}\cos(\theta + \phi)$$

avec : $\cos(\phi) = x/\sqrt{x_i^2 + y_i^2}$.

Cette paramétrisation est tout d'abord homogène puisque les angles θ sont a priori équiprobables, ainsi que les distances ρ. Elle fournit également des domaines de variation finis : $\Delta\theta = [-\pi, +\pi[$ et $\Delta\rho = [0, \sqrt{2}L]$ si L est le côté de l'image, puisqu'une droite dont ρ serait supérieur à la longueur de la diagonale de l'image ne serait pas visible dans l'image.

La transformation de *m à 1* associe au doublet M_i, M_j le point de $\mathcal{H}$ de coordonnées :

$$\begin{aligned} \rho_{ij} &= \frac{|x_i y_j - x_j y_i|}{\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}} \\ \theta_{ij} &= -\text{Arctg}\left[\frac{x_j - x_i}{y_j - y_i}\right] \end{aligned}$$

14.7.3 Détections par TH

La TH est très souvent utilisée, comme nous l'avons vu précédemment, pour détecter des droites. Les autres courbes paramétrées détectées par la TH sont :

1. les paraboles (3 paramètres si l'on connaît la direction de leur axe, 5 dans le cas général). Par exemple pour détecter des côtes dans les radiographies du corps humain) ;
2. les cercles (3 paramètres) par exemple pour détecter des cuves ou des citernes en imagerie aérienne. On paramètre alors le cercle par son centre et son rayon et la forme associée dans cet espace est un cône, centré au centre du cercle.
3. les ellipses (5 paramètres : les 3 du cercle + la direction de l'axe et l'applatissement ou l'ellipticité), également utilisée pour détecter des formes circulaires vues en perspectives.

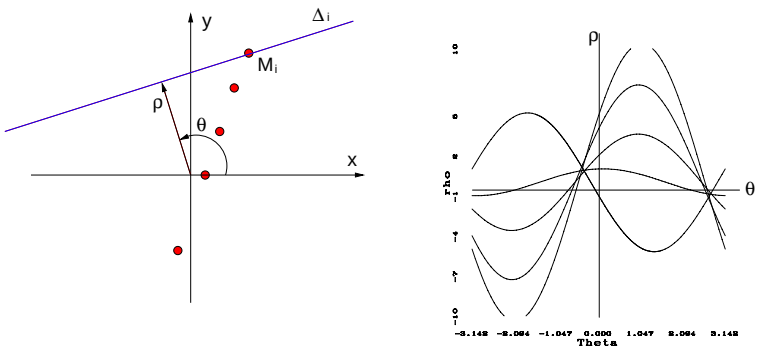


FIG. 14.19 – Transformation de Hough de *1 à m* en coordonnées normales. Une droite Δ_i est repérée par l'angle θ et la distance au pied de sa normale ρ. Dans l'espace $\mathcal{H}$, chaque point donne une sinusoïde.

4. des traces sinusoïdales (5 paramètres), forme fréquemment recherchée en expérimentation en physique ou en géologie.

La TH est également utilisée pour suivre des formes quelconques sous des hypothèses de transformations particulières. Les paramètres sont alors ceux qui caractérisent la déformation : changement d'échelle, rotation à 2 ou 3 D, projection perspective, etc.

14.8 Conclusion

Il n'est pas possible d'épuiser toutes les variétés des représentations et des modélisations qui ont été adoptées pour décrire les formes des objets en traitement des images. en effet, beaucoup de ces représentations tiren profit de spécificités attachées à la famille particulière des objets à traiter pour une application donnée dans un contexte précis. Nous avons cependant vu que toutes les représentations cherchent à marier la simplicité de la description et la fidélité à la forme. En cela, elles relèvent déjà toutes de la reconnaissance des formes, puisque ces deux objectifs sont aussi ceux que la reconnaissance des formes se donne.