

SEGMENTATION HIERARCHIQUE DE TEXTURES A L'AIDE D'UNE DECOMPOSITION EN ONDELETTES

Georges LOUM^(1,2), Slim M'HIRI⁽²⁾,
Jacques LEMOINE⁽²⁾, Eric PETIT⁽²⁾, Pierre PROVENT⁽³⁾

(1) Departement Genie Electrique, INSET BP 1093 Yamoussoukro COTE D'IVOIRE.

(2) L.E.R.I.S.S., Université Paris XII Val-de-Marne,
61, Avenue du Général De Gaulle, 94010 Creteil Cedex, FRANCE.

(3) Laboratoire des signaux et systèmes, CNAM, 292, Rue St Martin 75003 Paris,
FRANCE.

Mots clés: Segmentation, Transformée en Ondelettes, Facteur de forme, Classification floue

RESUME

Dans cet article, nous présentons une méthode de segmentation hiérarchique de textures fondée sur une décomposition en ondelettes de l'image. Une segmentation grossière est effectuée au niveau de résolution le plus élevé à partir de prototypes de chaque classe de texture déterminés avec un classificateur flou. Une analyse de voisinage permet ensuite de créer une classe de pixels ambigus. La segmentation initiale est progressivement affinée en reportant la prise de décision pour les pixels ambigus au niveau de résolution immédiatement inférieur. Ce processus est itéré jusqu'au niveau de résolution de l'image originale.

ABSTRACT

In this paper, we present a method for texture image segmentation based on wavelet decomposition. A rough segmentation is done at the highest level from prototypes of each class of texture determined with a fuzzy classifier. A neighbourhood analysis allows to create a class of ambiguous pixels. The first segmentation is gradually enhanced by switching to the lower level once ambiguous pixel is reached. The process is repeated until reaching the given level of the image.

1 INTRODUCTION

La segmentation des images texturées a été largement étudiée durant ces dernières années. Ce type de segmentation suppose l'extraction d'attributs pouvant permettre de caractériser au

mieux les différentes textures de l'image afin de les différencier. Les travaux de recherche récents sur le système visuel humain qui n'éprouve aucune difficulté apparente à reconnaître des textures différentes, ont inspiré le modèle de filtrage multicanaux pour l'analyse de textures. Ce filtrage peut être modélisé mathématiquement par les fonctions de Gabor et les ondelettes. Nous proposons dans ce papier, une méthode de segmentation qui se fonde sur une décomposition en ondelettes.

Les ondelettes ont déjà été appliquées à la segmentation d'images [1, 2]. Notre méthode diffère des précédentes par le mode de caractérisation des textures et par le processus de segmentation employé. A chaque niveau de résolution, des attributs de texture appelés *facteurs de forme* sont déterminés en combinant les informations des différents niveaux de résolution. Une classification de type flou est ensuite effectuée au niveau de résolution le plus élevé (le nombre c de classes est supposé a priori connu). Au terme de cette étape les "meilleurs" prototypes de chaque classe sont obtenus. On classe par la suite les pixels de l'image du niveau de résolution courant en réalisant une mesure de distance dans l'espace des attributs de texture. Une image partitionnée en c classes est ainsi obtenue. Une analyse de voisinage permet de déterminer un ensemble de pixels ambigus (pixels isolés ou frontaliers). La classification de ces pixels ambigus est reportée au niveau de résolution inférieur. Ce processus est itéré jusqu'au niveau de résolution le plus bas de la décomposition.

Nous présentons tout d'abord le *facteur de forme* avant de décrire l'algorithme de segmentation mis en oeuvre.

2 TRANSFORMEE EN ONDELETTE ET FACTEUR DE FORME

La Transformée en Ondelettes (T.O.) permet de représenter une image à différents niveaux de résolutions. Elle fournit au niveau de résolution j :

- une approximation I_j de l'image de départ, et
- trois images de détail D^1_{-j} , D^2_{-j} , D^3_{-j} , privilégiant chacune une orientation particulière: horizontale (indice 1), verticale (indice 2), et diagonale (indice 3). Ces images renferment les informations perdues lors du passage du niveau de résolution précédent $j-1$ au niveau j .

La mise en oeuvre pratique de la T.O. est généralement réalisée au moyen de deux filtres H (passe-bas) et G (passe-haut) conjugués en quadrature, dont les sorties sont sous-échantillonnées (un échantillon sur deux est conservé) [3]. Le filtrage réalisé par ces filtres est d'abord effectué suivant les lignes puis les colonnes de l'image. Ce qui permet d'obtenir dans un premier temps deux sous-images intermédiaires, puis les quatre sous-images résultats de la transformation. La décomposition est orthogonale, complète et s'effectue sans redondance d'informations.

Le processus de base peut être appliqué de façon récursive aux images d'approximation successives. Il conduit alors à une représentation hiérarchique pyramidale de l'image originale.

La décomposition en ondelettes sur J niveaux d'une image originale discrète I_0 supposée de niveau de résolution 0, s'écrit :

$$I_0 = I_{-J} + D_{-J}^1 + D_{-J}^2 + D_{-J}^3 + D_{-J+1}^1 + D_{-J+1}^2 + D_{-J+1}^3 + \dots + D_{-1}^1 + D_{-1}^2 + D_{-1}^3$$

Si les coefficients de l'image d'approximation I_{-j} et de l'image de détail D_{-j}^l d'orientation l ($l=1,2$ et 3) sont désignés respectivement par $i_{-j,k}$ et $d_{-j,k}^l$, alors, le *facteur de forme* associé au pixel k du niveau de résolution J est défini par [4]:

$$FF_j^l(k) = \frac{\sqrt{(i_{-j,k})^2 + \sum_{j=1}^J \sum_{b \in V_j(k)} (d_{-j,b}^l)^2}}{|i_{-j,k}|}$$

où j est l'indice de résolution et $V_j(k)$ le voisinage du pixel k correspondant au niveau de résolution j . Pour j fixé, $V_j(k)$ correspond à un voisinage de taille $2^{j-1} \times 2^{j-1}$ (Figure 1).

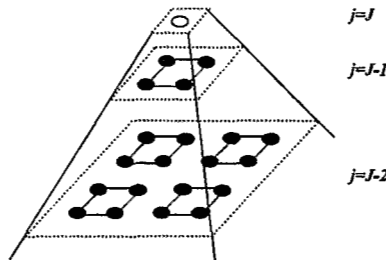


Figure 1 : Exemple de voisinages $V_j(k)$

Il est intéressant de remarquer que :

- Le facteur de forme s'obtient par combinaison des informations des différents niveaux de résolutions de la décomposition. Ce qui constitue la particularité de cet attribut par rapport à ceux utilisés dans les méthodes précédentes, qui ne prennent en compte pour leurs calculs que les informations d'un seul niveau de résolution.
- Les différentes directions préférentielles de la T.O. permettent de calculer à chaque niveau de résolution trois facteurs de forme. Ces facteurs de forme directionnels sont utiles pour la discrimination des textures. Le pouvoir discriminant de ces facteurs a été éprouvé dans un travail antérieur [4].

3 ALGORITHME DE SEGMENTATION

La méthode de segmentation que nous proposons s'articule autour de trois points essentiels : la détermination des prototypes, la segmentation primaire, et la localisation précise des contours.

3-1 Détermination des prototypes

Cette phase du traitement consiste à déterminer les "meilleurs" prototypes de chaque classe. Elle se déroule au niveau le plus élevé de la pyramide de décomposition et requiert la connaissance du nombre c de classes (nombre de textures présentes dans l'image). C'est une phase importante puisque le choix de ces prototypes conditionnent les performances de la segmentation.

Dans les techniques habituelles de classification, l'appartenance d'un pixel à une classe est binaire. A la sortie de ces classificateurs, on ne dispose d'aucune mesure de degré d'appartenance d'un pixel à une classe permettant de dire si ce pixel est ou n'est pas un bon prototype de la classe. Les techniques de classification floue présentent l'avantage de considérer les classes comme des ensembles flous de l'image. Cela permet d'attribuer à un pixel donné, un degré d'appartenance aux différentes classes en présence. Ainsi dans une classe donnée, les pixels ayant un degré d'appartenance élevé, peuvent être considérés comme les meilleurs prototypes de la classe.

L'algorithme de classification floue que nous avons utilisé est celui des c -moyennes floues FCM [5]. Cet algorithme procède de façon itérative :

A chaque pixel x_i est associé un vecteur d'attribut formé des trois facteurs de forme et un vecteur d'appartenance $U_i = (u_i^{<1>}, u_i^{<2>}, \dots, u_i^{<c>})$ où $u_i^{<k>}$, désigne le degré d'appartenance du pixel x_i à la classe C_k tel que :

$$u_i^{<k>} \in [0,1] \quad \text{et} \quad \sum_{k=1}^c u_i^{<k>} = 1$$

A chaque itération, le centre V_k de la classe C_k , est calculée à partir des pixels x_i (avec $i=1, \dots, n$) de l'image par :

$$V_k = \frac{\sum_{i=1}^n (u_i^{<k>})^m x_i}{\sum_{i=1}^n (u_i^{<k>})^m}$$

où m est un facteur de flou généralement compris entre 1,5 et 2.

La détermination des centres des c classes permet d'actualiser le vecteur de degré d'appartenance associé au pixel x_i . Les nouvelles composantes de ce vecteur s'obtiennent par :

$$u_i^{<k>} = \frac{1}{\sum_{q=1}^c \left(\frac{d_i^{<k>}}{d_i^{<q>}}\right)^{\frac{2}{m-1}}} \quad \text{avec} \quad (d_i^{<k>})^2 = \|x_i - V_k\|^2$$

L'itération est poursuivie jusqu'à ce que l'on converge vers un minimum. A ce stade, une règle de décision (par coupe de niveaux ou par maximum d'appartenance) est appliquée pour décider de l'appartenance d'un pixel à une classe.

Pour notre application, la détermination des meilleurs prototypes de chaque classe se fera par la règle de décision par coupe de niveaux α suivante :

$$x_i \text{ est prototype de } C_k \text{ si } u_i^{<k>} \geq \alpha \text{ avec } 0.5 < \alpha \leq 1$$

3-2 Segmentation primaire

A l'aide des prototypes déterminés précédemment, on calcule le nouveau centre de chaque classe. Ces centres permettent d'affecter aux différentes classes C_k , tous les pixels du niveau de résolution courant par une mesure de distance dans l'espace des attributs.

Une analyse du voisinage de chaque pixel est ensuite effectuée afin de déterminer les pixels ambigus. Un pixel est considéré comme ambigu, si il n'appartient pas à la même classe que la majorité des pixels de son voisinage immédiat. Les pixels ambigus sont généralement représentatifs de zones singulières ou des frontières. Leur classification est reportée à une résolution supérieure.

Ce report permet d'analyser beaucoup plus finement les régions représentées par les pixels ambigus. En effet, le passage d'un niveau de résolution donnée à celui qui lui est immédiatement inférieur, revient à représenter selon la construction pyramidale un noeud par ses quatre parents, et donc à scinder une région ambiguë en quatre sous-régions.

3-3 Localisation précise des contours

Au début de cette phase, on dispose : des prototypes de chaque classe, des pixels ambigus, et d'une image résultat formée par des pixels affectés à une classe.

Seuls les pixels ambigus font l'objet d'une classification lors de cette phase. L'image résultat obtenue est interpolée pour être ramenée au niveau de résolution inférieur. A ce niveau, les facteurs de forme associés aux parents des pixels prototypes sont utilisés pour calculer les

nouveaux centres de classes. Une mesure de distance à ces centres permettent de classer les pixels ambigus. Cette opération permet d'affiner les contours.

Le processus est itéré jusqu'au niveau de résolution 1 sans que l'on ait à nouveau recours à l'algorithme de classification floue pour la détermination des prototypes. A ce dernier niveau, tous les pixels sont attribués à une classe. L'image résultat du niveau 1 est ensuite interpolée au niveau 0. Afin de lisser les frontières obtenus après cette dernière phase d'interpolation l'image du niveau 0 est soumise à un filtrage médian.

Le schéma général de l'algorithme est le suivant :

Etape 1. Réalisation la décomposition en ondelette de l'image jusqu'au niveau de résolution J.

Etape 2. Calcul des trois facteurs de forme pour chaque niveau de résolution j ($1 \leq j \leq J$).

Etape 3. Au niveau de résolution J:

Classification floue (vecteur attribut : les trois facteurs de forme).

*Si le degré d'appartenance d'un pixel à une classe $q > \alpha$,
alors marquer ce pixel comme prototype de la classe q ,
donner à ce pixel le label de la classe q .*

Initialisation de j (variable du niveau de résolution courant) : $j = J$

Etape 4. Calcul du centre des prototypes de chaque classe au niveau j .

Etape 5. Affectation des pixels du niveau $j = J$ ou des pixels ambigus de niveau $j \neq J$,

*Mesure de distance d'un pixel au centre d'une classe
dans l'espace (tridimensionnel) des attributs de texture.*

*Si $j = J$, alors pixels à considérer = pixels du niveau de résolution j
sinon pixels à considérer = pixels ambigus*

Pour chaque pixel à considérer,

*Si la distance d'un pixel au centre d'une classe k est inférieure
à la distance de ce pixel à tous les autres centres de classe,
alors donner à ce pixel le label de la classe k .*

Etape 6. Détermination des pixels ambigus de l'image de résolution courante.

Analyse du voisinage de chaque pixel de l'image de niveau de résolution j .

*Si j est différent de 1,
et que la classe du pixel considéré n'est pas majoritaire dans le voisinage de ce pixel,
alors marquer ce pixel comme ambigu.*

Etape 7. Extension de l'image de niveau de résolution j , au niveau de résolution $j-1$.

*Pour chaque pixel de l'image de résolution j ,
donner à ses quatre parents la même étiquette (classe, prototype, ambigus) que lui.*

Décrémenter j

Etape 8. Amélioration de la segmentation

Si $j \neq 0$, retourner à l'Etape 4.

sinon filtrer l'image du niveau 0 par un médian pour obtenir l'image segmentée finale.

4 RESULTATS

La taille des images traitées est de 256×256 avec 256 niveaux de gris. La décomposition en ondelettes a été effectuée jusqu'au niveau J=4 qui correspond à un voisinage de taille 16×16 sur l'image originale. L'ondelette utilisée est celle de Haar qui présente pour notre application le double avantage d'être d'un coût faible en temps de calcul, et de conduire à un calcul des attributs sur des régions qui ne se recouvrent pas. Ce qui implique une réduction de l'ambiguïté sur la localisation exacte des contours. La valeur de α de l'algorithme de classification floue, qui permet de déterminer les meilleurs prototypes, a été fixée à 0.96 pour l'ensemble des expérimentations. L'analyse du voisinage est réalisée sur une fenêtre de taille 3×3 pour J>2 et 5×5 pour J≤2. Ce choix tient compte du nombre de pixels des images des différentes résolutions. Le filtre médian final est de taille 7×7. L'algorithme de segmentation a été implanté en langage C sur un micro-ordinateur de type PC, 486DX4, 50 MHz.

Les résultats obtenus par l'algorithme sont satisfaisants. La figure 2 illustre les différentes étapes de l'algorithme pour une image (figure 2.a) composée de deux textures. La classe de la texture de gauche est dénommée classe 1.1, et celle de droite, classe 1.2. Toutes les images ont été ramenées au niveau 0 pour faciliter leur lecture. On peut noter sur la figure 2.b, qui montre le résultat de la classification floue, que très peu de pixels ne sont pas prototypes (pixels en noir). Cela illustre le fort pouvoir discriminant du facteur de forme. Le résultat de la segmentation grossière est présenté à la figure 2.c. L'analyse de voisinage permet de déterminer les pixels ambigus (en blanc sur les figures 2.d, 2.f et 2.h). Du niveau 3 au niveau 1, la localisation de la frontière des deux textures devient de plus en plus précise et il reste très peu de pixels ambigus au niveau 1 (figure 2.h). Le résultat final de la segmentation est présenté à la figure 2.j. On peut constater l'effet de lissage du filtre médian. Notons toutefois que bien que utile, ce filtre introduit souvent une délocalisation légère des contours.

Les résultats de la segmentation de deux mosaïques de textures (figures 3.a et 4.a) sont présentés aux figures 3.b et 4.b. La première est constituée de pullover (classe 2.1), siège (classe 2.2), tissu écossais (classe 2.3), et bois (classe 2.4). La seconde, des textures canevas (classe 3.1) et (lierre 3.2) tirées de l'album de Brodatz [6]. Tous les résultats chiffrés sont indiqués dans le tableau 1.

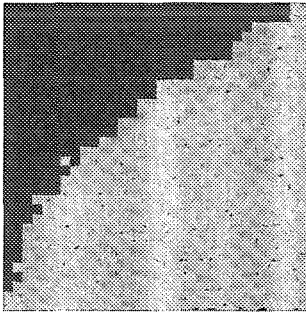
	<i>Image 1</i>		<i>Image 2</i>		<i>Image 3</i>	
<i>Niveau de départ de la segmentation</i>	4		4		4	
<i>Nombre de pixels flous</i>	7		14		7	
<i>Taux correct de classification de la classe (x.x) en (%)</i>	(1.1)	97.3	(2.1)	98.6	(3.1)	96.5
	(1.2)	97.8	(2.2)	98.9	(3.2)	95.7
			(2.3)	99.9		
			(2.4)	98.8		
<i>Durée de l'opération (s)</i>	33.5		37		35	

Tableau 1 : Résultats de la segmentation des images test. La classe est indiquée entre parenthèses

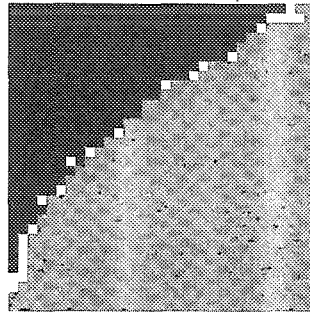
5 CONCLUSION

Dans cet article, nous avons décrit un algorithme de segmentation hiérarchique d'images texturées. Le processus proposé consiste à effectuer une segmentation grossière au niveau le plus élevé de la décomposition, puis à affiner progressivement le résultat de la segmentation en reconsidérant aux niveaux inférieurs, la classification d'un nombre réduit de pixels. Ce procédé

NIVEAU 3

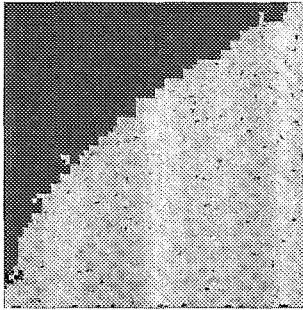


[e]

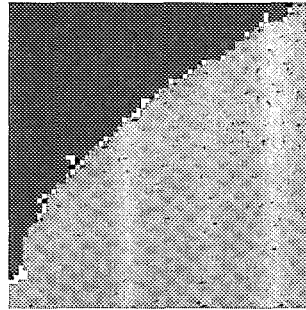


[f]

NIVEAU 2



[g]



[h]

IMAGE ORIGINALE

NIVEAU 1

RESULTAT FINAL



