Stable Feature Extraction Using Wavelet Filters for Texture Segmentation

Sylvain Meignen, Valérie Perrier

Abstract

This paper describes a new unsupervised texture segmentation algorithm when the number of textures is a priori unknown. We first extract a new feature from a wavelet filter decomposition on a stability criterion. The decomposition uses wavelet frames associated with symmetric orthogonal filters. The rest of the article is devoted to the computation of the number of textures and of their features and to the description of the segmentation algorithm. Finally, a comparison with other existing methods is provided.

Keywords

Multiscale texture image analysis, image segmentation, wavelet filters.

I. Introduction

Texture discrimination is basic in many applications in computer vision. In its simpler form, the problem is to design a method to decide whether two given texture samples are similar or significantly different. The first step is thus to find a qualitative description of textures. One important category of texture discrimination method is the category of signal processing method, where typically a filtering step followed by a nonlinear step are used as feature extractor. Methods based on multichannel or multiresolution analysis have received a lot of interest for the designing of filters [6][15][23][4][17]. The biological foundation for those methods dates back to the discovery that the impulse response of simple cells in the striate cortex could be approximated by Gabor filters [9]. These filters have been widely used for texture segmentation [4][22][11][8][10], but they depend on parameters whose determination requires an a priori knowledge on the textures.

Orthogonal wavelet or orthogonal wavelet packet decompositions allow multiresolution analyses while preserving the volume of data and have been used for texture classification [6][14][17]. The main drawbacks of the orthogonal wavelet (packet) decompositions for texture segmentation consist in the loss of translation invariance and in bad spatial resolution [16]. One therefore prefer to use for segmentation tasks redundant decomposition such as wavelet frames [1][2][23][15].

Once a special kind of decomposition has been chosen (filtering step), a nonlinear transformation is applied to the coefficients of the decomposition to extract texture features. This step basically consists in rectifying the filter response (i.e. transforming negative amplitude to the corresponding positive amplitude) and smoothing. Smoothing filters are determinant to extract local energies that make up texture features. Commonly applied smoothing filters to compute local energies are rectangular windows [23][7] or Gaussian windows [1][11][4]. The latter provides more robust features for texture segmentation task. In any cases, one has to determine the appropriate window size which depends on the frequency content of the rectified filter response [20][1]. The output from the local energy is a set of images which are the basis for the segmentation step.

In most texture segmentation applications, the number of textures in the image is assumed to be known. Unlike classical approaches we wish to build an algorithm that enables to determine the number of textures before the segmentation step. In our case, this will be done through a split and merge procedure.

We first determine a new feature, which is the direction of the vector of local energies in the space of overcomplete wavelet coefficients. We will establish the stability of this feature, showing its low sensitivity to texture sample size. Then we define a split and merge procedure based on this new feature which allows to compute the number of textures in the image and its associated feature. The computed features serve as an initial guess for the segmentation algorithm and thus reduce the possibility of the algorithm to get trapped in a local minimum.

S. Meignen is with LMC-IMAG Laboratory, Grenoble, France. E-mail: sylvain.meignen@imag.fr
V. Perrier is with LMC-IMAG Laboratory, Grenoble, France. E-mail: valerie.perrier@imag.fr

Our paper is organized as follows: section II is devoted to filter design. Section III brings about a stable texture feature while section IV details the segmentation algorithm (hierarchical splitting step, agglomerative merging step and final segmentation). Section V compares our segmentation algorithm with some other existing methods.

II. Channel Decomposition and Tight Frame Representation

Let an image of textures be a function f of $l_2(Z^2)$ with support $I = [1, N] \times [1, N]$. We assume that f is filtered by $F = \{F_i \in l_2(Z^2); \quad i = 1, \cdots, P\}$ a frequency selective filter bank which can be of different kind. In a multiresolution framework, it may be generated by Gabor filters [4][11][3][1], orthogonal or biorthogonal wavelet filters [6][23][14][17] or wavelet packet filters [6].

Referring to the study made by Randen in [20] on the choice of wavelet filters for texture segmentation, one had rather chose an overcomplete wavelet filter representation. The main reasons for better performances are, first, that critically sampled filter banks imply inaccurate edge localization and, second, that overcomplete representations such as wavelet frames are shift invariant [23].

A. Tight Frames

Our approach is based on the design of a discrete tight frame F using dilation and translation of filters h and g that define perfect reconstruction filter banks of $l_2(Z^2)$ [24]. These are entirely specified by a prototype filter h that satisfies the standard quadrature mirror filter condition

$$|\hat{h}(\omega)|^2 + |\hat{h}(\omega + \pi)|^2 = 1$$

where $\hat{h}(\omega) = \sum_{-\infty}^{\infty} h[n]e^{-in\omega}$ is the Fourier series of h. The filter h also satisfies the condition $\hat{h}(0) = 1$. A complementary high-pass filter g is obtained by symmetry with respect to $\frac{\pi}{2}$:

$$\hat{g}(\omega) = \hat{h}(\omega + \pi).$$

In this case, filters h and g are called conjugate mirror filters. Such filters can both be symmetric or antisymmetric about $n = 0$. In applications, filters with symmetry or antisymmetry are clearly favored since they are advantageous in alleviating boundary artifacts through the simple method of mirror extension [15].

Let $<, >$ be the scalar product on $l_2(Z^2)$ and $\|.\|$ be the corresponding norm. Then, a frame c_n of $l_2(Z^2)$ is defined by the existence of constants A and B such that:

$$A\|f\|^2 \leq \sum_n < f, c_n >^2 \leq B\|f\|^2 \quad \forall f \in l_2(Z^2)$$

When $A = B = 1$ the frame is said to be tight. In other words, a tight frame is a frame that preserves the energy of f. It is possible to define tight frames of $l_2(Z^2)$, through the equivalence [21]

Theorem 1: (tight frames of $l_2(Z^2)$) The family

$$F = \{h_J[n-l]h_J[m-k], \{g_j[n-l]h_j[m-k], \\ h_j[n-l]g_j[m-k], g_j[n-l]g_j[m-k]\}_{1 \leq j \leq J}\}_{(l,k) \in Z^2}$$

is a tight frame of $l_2(Z^2)$, for all $J \geq 1$ if and only if:

$$|\hat{h}(\omega)|^2 + |\hat{g}(\omega)|^2 = 1 \tag{1}$$

where

$$h_{j+1}[k] = \sum_n h[n]h_j[k - 2^j n]$$

$$g_{j+1}[k] = \sum_n g[n]h_j[k - 2^j n]$$

and $h_0 = h$.

This theorem enables us to define tight frames of $l_2(Z^2)$ using conjugate mirror filters. The filter bank F we use is of the kind defined by Theorem 1 and is made up with $3*J+1$ filters F_i. More precisely, we consider $F_1 = \{g_1[n-l]g_1[m-k]\}_{(l,k)\in Z^2}$, $F_2 = \{g_1[n-l]h_1[m-k]\}_{(l,k)\in Z^2}$, $F_3 = \{h_1[n-l]g_1[m-k]\}_{(l,k)\in Z^2}$,..., $F_{3J+1} = \{h_J[n-l]h_J[m-k]\}_{(l,k)\in Z^2}$. The result of the filtering of f by F_i is called f_i. The overcomplete wavelet decomposition of f is then given by $(f_1, f_2, \cdots, f_{3J+1})$, f_{3J+1} being the coarse approximation of f. The computational cost of the whole decomposition, based on recursive discrete convolutions is $O(N^2 log(N))$ operations.

B. Choice of Filter h

A major drawback associated to the definition of tight frames in the context of Theorem 1 is that it is impossible to define symmetric filters h and g about $n=0$ of finite length that satisfy condition (1). To use symmetric filters h and g about $n=0$ is however crucial to prevent shifts through filtering.

We make the choice to consider symmetric filters of infinite length satisfying condition (1), which we truncate. In such a framework, Battle-Lemarié filters are good candidates. They are defined by:

$$\hat{h}(\omega) = \exp(\frac{-i\epsilon\omega}{2})\sqrt{\frac{S_{2m+2}(\omega)}{2^{2m+1}S_{2m+2}(2\omega)}}$$

with

$$S_m(\omega) = \sum_{-\infty}^{\infty} \frac{1}{(\omega + 2k\pi)^m}$$

with $\epsilon = 0$ for m odd and $\epsilon = 1$ otherwise. For m odd $h[n]$ is symmetric about $n = 0$, so we restrict ourselves to such cases. To get a filter with high frequency selectivity, one is inclined to choose m large but such filters are much more sensitive to truncation. For instance, if we decide to use only the coefficients in h with amplitude larger that 0.01, we should keep 9 coefficients if $m = 1$, 15 coefficients if $m = 3$. For larger m, the number of coefficients larger than 0.01 becomes prohibitive in the sense that the decomposition of f over filter F with the same precision on the filter coefficients requires 10 times more CPU than that with $m = 1$. Furthermore, it has been evidenced by Unser in [23], that increasing the regularity of the filter bank (i.e. m) does not necessarily imply better performances in terms of texture classification and segmentation. Referring to the classification results in [23], the Battle-Lemarié filter with $m = 1$ which corresponds to linear spline is a good choice. Furthermore, keeping 9 coefficients limits the effect of truncation on energy conservation defined by (1).

C. Determination of the Depth of Decomposition J

Once the frame representation and the filter h have been chosen to define filter bank F, one needs to determine the appropriate depth of decomposition J associated with the filter F. We make the hypothesis that f has a zero mean and we determine J as follows. For any f, denote:

$$E_i = \sum_{[n,m]\in I} f_i[n,m]^2 = \sum_{[n,m]\in I} (f * F_i[n,m])^2$$

As the image is with zero mean, E_{3J+1} is decreasing to zero with J, since asymptotically it tends to the average value of f. We also recall that as we use a tight frame representation, if we neglect the truncation errors, we have the energy conservation that is

$$\|f\|^2 = \sum_{i=1}^{3J+1} E_i.$$

Given a threshold R in $]0,1[$ we define the depth of decomposition $\hat{J}$ as:

$$\hat{J} = \underset{J\geq 1}{argmin} \left\{ \frac{\sum_{i=1}^{3J} E_i}{\sum_{i=1}^{3J+1} E_i} > R \right\}. \qquad (2)$$

If R is chosen close to 1, the decomposition at depth $\hat{J}$ mostly preserves the energy in the detail subspaces (indices $i = 1, \cdots, 3\hat{J}$).

D. Mirror Extension

To avoid boundary problems, we use a mirror extension of the initial image f and we use a symmetric filter h (Battle-Lemarié filter with $m = 1$ and 9 coefficients). Although the filtering process creates filtered images f_i whose respective sizes are bigger than the initial image f, due to the symmetry, we can only retain the coefficients that are inside I, the support of f. Those coefficients define at each point $[n, m]$ in I a vector

$$\left(\tilde{f}_i[n, m]\right)_{i=1,\cdots,3\hat{J}+1}. \tag{3}$$

III. Stable Features Determination

In this section we consider an image f of $l_2(Z^2)$ with support $I = [1, N] \times [1, N]$ and with zero mean, made of a single texture. We assume that the depth of decomposition is fixed to $\hat{J}$ and we define $P = 3\hat{J}$. Our goal is now to extract, from the vectors $\tilde{f}_i$ (3), texture features that are the least sensitive as possible to texture sample size. Our approach is based on the study of the most energetic channels i which are considered as the most relevant. This assumption is very common in texture analysis [6] [23] and will be a posteriori justified in the context of features stability. We will compare in the following three features based on energetic channels.

We define the energy of the channel i over the region I as follows:

$$\tilde{E}_i^I = \sum_{[n,m]\in I} \tilde{f}_i^2[n, m]. \tag{4}$$

If we assume $N = 2^m$ and then split the region I into non overlapping subregions with support I_k of size $K \times K$, $K = 2^{m'}$ with $m' < m$, we define the energy of channel i on I_k by:

$$\tilde{E}_i^{I_k} = \sum_{[n,m]\in I_k} \tilde{f}_i^2[n, m]. \tag{5}$$

The test images we consider for the measure of stability are either extracted from Brodatz album [5] (D9, D12, D17, D24, D77, D92, Fabric.0000, Food.0000) or from MeasTex database [25] (Misc.0000, Asphalt.0000, Rock.0004, Grass.0000) (see Figure 1).

A. Stability of the Three Most Energetic Channels

Given the depth of decomposition $\hat{J}$, we determine the three most energetic channel indices i associated with the image of Figure 1. Table 1 displays the three most energetic channel indices. This informs us that, for the textures under consideration, the three most energetic channel indices differ which suggests that these features may be discriminating. We also note that, on the subset of studied textures, the three most energetic channel indices subject to any permutations are still discriminating. The question is to determine if these properties are still valid when the size of the texture sample varies.

We first study whether the order of the three most energetic channel indices is preserved through a variation of texture sample size. Those indices may be interesting features in that they are invariant through a translation of f.

Let $\mathcal{E}_i^{I_k}$ be the indice of the ith most energetic channel on I_k. We compute for various K:

$$S_K = \frac{K^2}{3N^2} \sum_{(I_k, I_{k'}), \#I_k = \#I_{k'} = K^2} \sum_{1 \le i \le 3} (1 - \delta_{\mathcal{E}_i^{I_k} \mathcal{E}_i^{I_{k'}}}) \tag{6}$$

where δ_{kp} is the Kronecker symbol (i.e. $= 1$ if $k = p$ and 0 otherwise) and $\#I_k$ is the size of I_k. A value of S_K close to zero indicates a low variability of the most energetic channel indices at scale K. The value of S_K is displayed in Table 2 for various K and for the reference textures of Figure 1. It clearly indicates a high variability of the order of the three most energetic channel indices with texture sample size. Recalling that

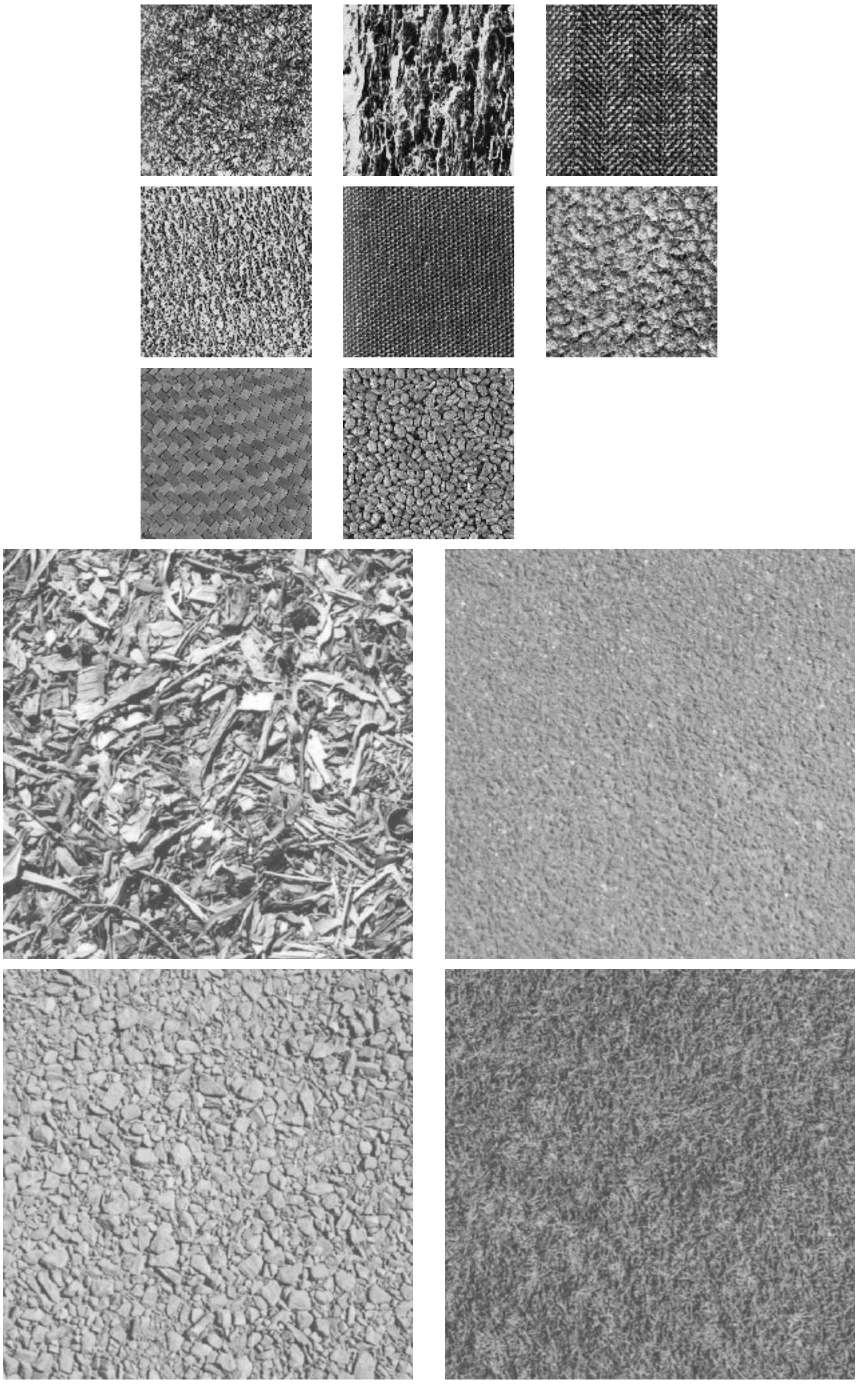

Fig. 1. The first six textures are extracted from Brodatz Database and are 128×128 images. The next two images are 128×128 images and are extracted from MeasTex Database so as the last four which are 512×512 images. These are referred by from left to right and top to bottom: D9,D12,D17,D24,D77,D92,Fabric.0000, Food.0000,Misc.0000,Asphalt.0000,Rock.0004, Grass.0000

	image dimension	dominant channel	$\hat{J}$
D9	128	2-3-1	3
D12	128	3-5-8	5
D17	128	4-3-2	2
D24	128	2-5-4	3
D77	128	3-2-1	2
D92	128	3-6-9	4
Fabric.0000	128	1-2-3	4
Food.0000	128	2-5-3	3
Misc.0000	512	12-9-8	6
Asphalt.0000	512	8-5-9	4
Rock.0004	512	8-11-9	5
Grass.0000	512	8-5-6	5

TABLE I
THE FIRST COLUMN DISPLAYS THE TEXTURES UNDER CONSIDERATION, THE SECOND DISPLAYS THE THREE MOST ENERGETIC CHANNEL INDICES WHILE THE THIRD DISPLAYS THE DEPTH OF DECOMPOSITION $\hat{J}$ ASSOCIATED WITH THE THRESHOLD $R = 0.9$

	K=256	K=128	K=64	K=32	K=16
D9	×	×	0	0.08	0.48
D12	×	×	0.58	0.67	0.74
D17	×	×	0	0.20	0.49
D24	×	×	0.13	0.32	0.40
D77	×	×	0.25	0.31	0.40
D92	×	×	0.36	0.55	0.71
Fabric.0000	×	×	0.25	0.37	0.55
Food.0000	×	×	0	0.48	0.62
Misc.0000	0.46	0.61	0.78	0.84	0.87
Asphalt.0000	0.17	0.30	0.57	0.70	0.79
Rock.0004	0.33	0.40	0.54	0.71	0.81
Grass.0000	0	0.44	0.54	0.73	0.80

TABLE II
VALUE OF S_K FOR THE REFERENCE TEXTURE OF FIGURE 1 FOR $J = \hat{J}$

the three most energetic channel indices are different for every studied texture regardless of their order, at the biggest scale, we check if this property still holds when the size of the texture sample decreases through the computation of

$$\widetilde{S_K} = \frac{K^2}{3N^2} \sum_{(I_k,I_{k'}),\#I_k=\#I_{k'}=K^2} \sum_{1\leq i\leq 3} (1 - \max_{1\leq i'\leq 3} \delta_{\mathcal{E}_i^{I_k}\mathcal{E}_{i'}^{I_{k'}}}). \tag{7}$$

The measure of $\widetilde{S_K}$ (7) on textures of Figure 1 is available in Table 3. The comparison between Table 2 and Table 3 tells us that the subset of indices associated with the three most energetic channels provides more stable features than ordered indices. However, these features are still too sensitive to texture sample size especially for MeasTex textures.

	K=256	K=128	K=64	K=32	K=16
D9	$\times$	$\times$	0	0	0.09
D12	$\times$	$\times$	0.13	0.15	0.31
D17	$\times$	$\times$	0	0	0.01
D24	$\times$	$\times$	0.13	0.16	0.19
D77	$\times$	$\times$	0	0	0.06
D92	$\times$	$\times$	0.13	0.23	0.33
Fabric.0000	$\times$	$\times$	0	0.04	0.14
Food.0000	$\times$	$\times$	0	0.13	0.28
Misc.0000	0.13	0.26	0.41	0.55	0.64
Asphalt.0000	0.17	0.17	0.18	0.29	0.42
Rock.0004	0.13	0.19	0.26	0.39	0.51
Grass.0000	0	0.13	0.20	0.34	0.44

TABLE III

Value of $\widetilde{S_K}$ for the reference texture of Figure 1 for $J = \hat{J}$

B. *Feature Extraction from the Whole Decomposition*

We have seen that the most energetic channel indices are not invariant with texture sample size. This means that selecting a priori some channels on an energy basis for texture segmentation is hazardous. However, we still believe that the most energetic channels are those that contain the useful information. Another way to look at it is to remark that $(\tilde{E}_i)_{i=1,\cdots,P}$ defines a specific direction in R^P which is mostly determined by the most energetic channels. This idea has already been exploited by Unser in [23] through a Karhunen-Loeve transform of the vector of energies. Indeed, the first eigenvector associated with a Karhunen-Loeve transform of the energies defines a direction in R^P which is mainly determined by the most energetic channels. We are going to show that this direction is sufficient for segmentation tasks and that, consequently, it is not essential to go through a Karhunen-Loeve procedure which is computationally demanding.

We want to test the stability of the direction defined by the energies when the size of the texture sample varies. To have a vector of features which is invariant through a translation of f we normalize the energy $\tilde{E}_i$ into

$$\overline{E_i}^{I_k} = \frac{\tilde{E}_i^{I_k}}{\sqrt{\sum_{1\leq i\leq P}\left(\tilde{E}_i^{I_k}\right)^2}}. \tag{8}$$

We denote $\overline{E}^{I_k} = \left(\overline{E_i}^{I_k}\right)_{i=1,\cdots,P}$ and then compute

$$\begin{aligned}\overline{S_K} &= \frac{K^2}{N^2}\sum_{(I_k,I_{k'}),\#I_k=\#I_{k'}=K^2}\left(1-\sum_{1\leq i\leq P}\overline{E_i}^{I_k}\overline{E_i}^{I_{k'}}\right)\\ &= \frac{K^2}{N^2}\sum_{(I_k,I_{k'}),\#I_k=\#I_{k'}=K^2}\left(1-<\overline{E}^{I_k},\overline{E}^{I_{k'}}>\right)\\ &= \frac{K^2}{2N^2}\sum_{(I_k,I_{k'}),\#I_k=\#I_{k'}=K^2}\|\overline{E}^{I_k}-\overline{E}^{I_{k'}}\|_2^2\end{aligned} \tag{9}$$

where $\|.\|_2$ is the Euclidean norm on R^P. We display the results of the computation of $\overline{S_K}$ for various K in Table 4.

The comparison between Tables 2, 3, 4 shows an overall better stability of the normalized energy defined in (8).

	256	128	64	32	16
D9	$\times$	$\times$	0.05	0.10	0.17
D12	$\times$	$\times$	0.09	0.18	0.29
D17	$\times$	$\times$	0.03	0.06	0.13
D24	$\times$	$\times$	0.03	0.08	0.15
D77	$\times$	$\times$	0.03	0.07	0.12
D92	$\times$	$\times$	0.07	0.13	0.21
Fabric.0000	$\times$	$\times$	0.09	0.15	0.21
Food.0000	$\times$	$\times$	0.04	0.09	0.17
Misc.0000	0.06	0.12	0.19	0.28	0.38
Asphalt.0000	0.07	0.09	0.12	0.18	0.25
Rock.0004	0.04	0.09	0.14	0.22	0.32
Grass.0000	0.04	0.07	0.12	0.19	0.27

TABLE IV
Value of $\overline{S_K}$ for the reference textures of Figure 1

As explained in the introduction, we are to build a segmentation algorithm containing a splitting step and a merging step for which the stability of the features with texture sample size is crucial. Therefore, in accordance with the previous study we will use the normalized energy vectors as features.

IV. Unsupervised Texture Segmentation Algorithm

We consider an image f with support $I = [1, N] \times [1, N]$, where N is a power of 2, containing several textures T_l, $l = 1, \cdots, nbtex$ whose number is a priori unknown.

Our algorithm starts with the computation of overcomplete wavelet coefficients $(\tilde{f}_i)_{i=1,\cdots,3\hat{J}}$, $\hat{J}$ being determined given a threshold R. The steps of our segmentation algorithm are as follows. We first get homogeneous samples of textures through a hierarchical splitting. Then, a merging step determines the number of textures and provides an estimation of texture features. Finally, a segmentation step using the feature estimation of the merging step is performed.

We will need in the following a measure between two regions, I_β and $I_{\beta'}$ whose sizes are not necessarily equal. It is done through the quadratic error

$$M(\overline{E}^{I_\beta}, \overline{E}^{I_{\beta'}}) = \frac{1}{2}\|\overline{E}^{I_\beta} - \overline{E}^{I_{\beta'}}\|_2^2 \tag{10}$$

A. Feature and Texture Number Determination Through Split and Merge Procedure

Hierarchical Splitting Step
The procedure is the following:

- *Split I into 4 square regions of equal size (I_1, I_2, I_3, I_4)*
- *Compute $\overline{E}^{I_k}$, $\quad k = 1, \cdots, 4$,*
- *Compute the measure M, defined in (10), between the regions I_k*
- *Given a threshold $\hat{\alpha}$ and a minimum size s_{min} for the sample, the algorithm stops if one of the following conditions is satisfied:*

1.

$$\max_{(I_\beta, I_{\beta'})} M(\overline{E}^{I_\beta}, \overline{E}^{I_{\beta'}}) < \hat{\alpha}$$

2. $\# I$ equals s_{min}^2

otherwise it carries on recursively on every region I_β.

I is all the more split that $\hat{\alpha}$ is small. The goal of the splitting step is to keep homogeneous texture samples of size larger than s^2_{min}. In our application $\hat{\alpha}$ is determined as follows:

$$\hat{\alpha} = \min_{\alpha}\{\alpha, N_\alpha \geq \lambda \left(\frac{N}{s_{min}}\right)^2\}$$

where N_α is the number of unsplit regions at the end of the splitting step considering the threshold α. Note also that $\left(\frac{N}{s_{min}}\right)^2$ is the number of samples of size s_{min} and λ is a tuning parameter whose choice is discussed in the Result section.

Agglomerative Merging Step

The problem is then to merge the regions determined at the end of the splitting step in a way that we get the number of textures *nbtex* and an estimation of the features for each texture.

As two samples of the same texture may be disconnected, we do not incorporate neighbor constraints in our merging criterion. The merging of the regions two by two is done basically by considering the quadratic error (10) between them. To do this we use the following procedure (A denotes the set of regions):

merge(A,s)
finish = false
while finish = false

$$(I_1, I_2) = \underset{I_\beta, I_{\beta'} \in A}{argmin}\, M(\overline{E}^{I_\beta}, \overline{E}^{I_{\beta'}})$$

if $M(\overline{E}^{I_1}, \overline{E}^{I_2}) > s$
finish = true
else
$A = A \setminus \{I_1, I_2\}$ $A = \{A, I_1 \bigcup I_2\}$
compute normalized energies over $I_1 \bigcup I_2$
end if
end while

If we apply the function **merge** iteratively with an increasing threshold s, we notice that some homogeneous regions are detected while some heterogeneous regions remain isolated. To prevent the merging of small heterogeneous regions with other regions, we do away with regions that do not contain a square of size $2s^2_{min}$ when the number of merged regions becomes smaller than a threshold p_1. This threshold p_1 is the maximum number of textures we suppose the image can contain.

The merging algorithm proceeds until the number of merged regions does not change when the threshold s varies. The merged regions are then considered as samples of the different textures of the image and are used to compute the feature estimation. To stop the merging algorithm we put a threshold on the variation of the number of merged regions with s. This threshold is called p_2 and its appropriate value will be determined in the Result section.

The initial threshold s used in the merging step is equal to $\hat{\alpha}$ to merge first the regions that were wrongly split. The following algorithm details the procedure:

function number-and-feature(A,$\hat{\alpha}$)
s=$\hat{\alpha}$
while ($\#A > p1$)
merge(A,s)
$s = s + \Delta s$

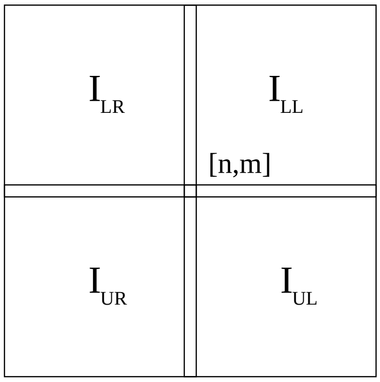

Fig. 2. Regions I_{LR},I_{LL},I_{UR},I_{UL} and point $[n, m]$

end while

for any $I \in A$
  if (no square region of size $2s_{min}^2 \subset I$*)* $A = A \setminus I$

$co = 0$;
while $(co < p2)$
  $size = \#A$
  merge(A,s)
  if $(size = \#A)$ $co = co + \Delta s$ *else* $co = 0$
  $s = s + \Delta s$
end while

$nbtex = \#\ A$

return (nbtex, normalized energy vectors for $I \in A$*)*

The advantage of such a merging algorithm is to do away with small unstable texture samples that may bias the evaluation of the texture features. A second advantage is that it allows the merging of disconnected subimages contrary to classical merging algorithms such as that proposed in [18].

To summarize, at the end of this step we know the number of textures in the image and we have an estimation of their normalized energy vector. We call these estimations $\overline{E}^k, k = 1, \cdots, nbtex$.

B. Final Segmentation Step

The features estimated at the end of the merging step provide an initial guess of the classes for the texture segmentation algorithm that we detail now.

For each point in I, we need to compute the normalized energy vectors over a square window to be able to compare them to the estimated features. However, it is well known that computing energies over a square window is not recommendable near texture boundaries [20]. To reduce the effect of the window we propose to consider, at each point $[n, m]$, the square regions I_{LR}, I_{LL},I_{UR} and I_{UL} such that $[n, m]$ is situated in the lower right corner of I_{LR}, in the lower left corner of I_{LL}, in the upper right corner of I_{UR} and in the upper corner of I_{UL} (see Figure 2).

To assign to each point $[n, m]$ a label we determine the closest reference normalized energy vector of one of the vectors computed over the four regions surrounding the point $[n, m]$. This amounts to apply the procedure:

$$l[n, m] = \underset{k}{argmin} \min_{\beta \in \{LR, UR, UL, LL\}} M(\overline{E}^k, \overline{E}^{I_\beta})$$

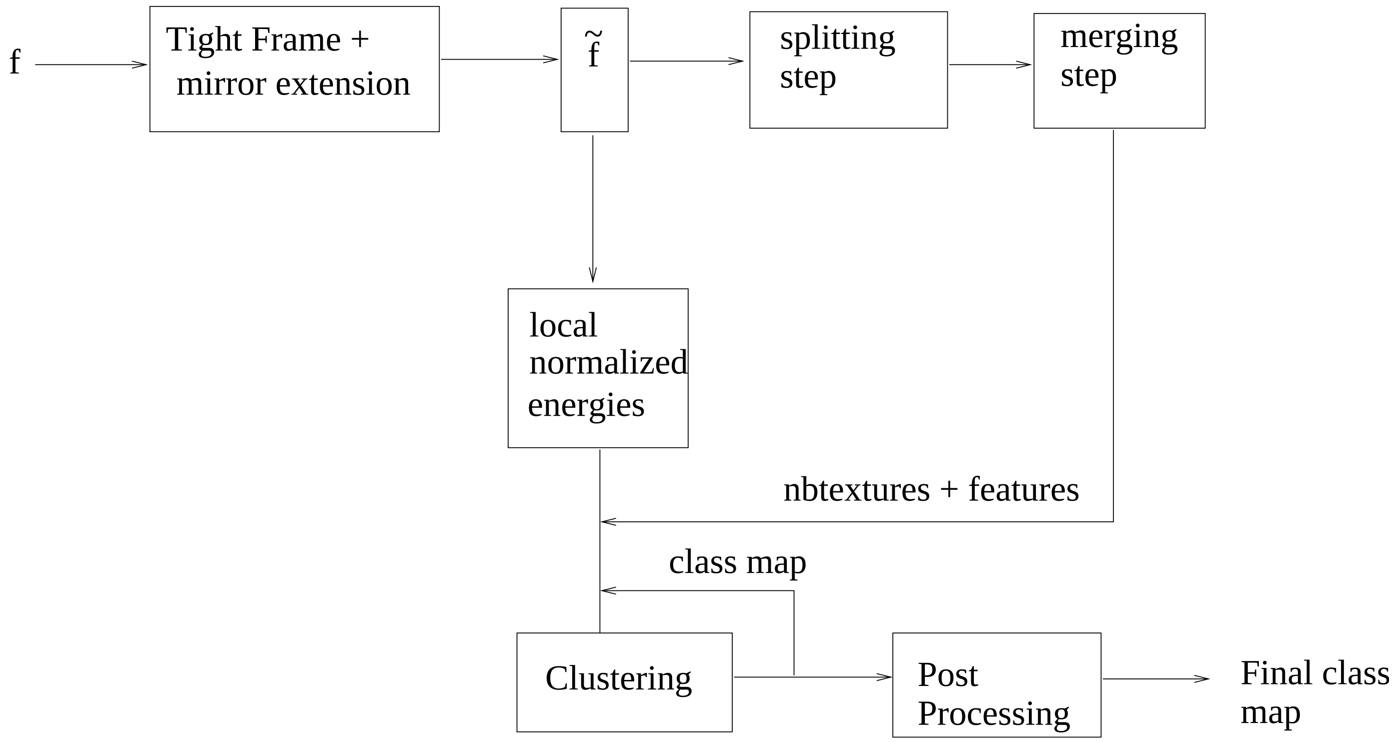

Fig. 3. Block diagram of the algorithm of our method for texture segmentation

where $\overline{E}^k, k = 1, \cdots, nbtex$ arises from step A. We assign to the point $[n, m]$ the label $l[n, m]$. Using the class map l we compute new reference vectors. We then iterate the procedure until there is no change in the classes. To take into account high spatial relationship on the labels of neighboring pixels we finally use a median filter.

The whole segmentation procedure is summarized in the diagram of Figure 3.

To illustrate the effects of the different steps of our algorithm, we display on Figure 4 the results of each step on a two texture image. Figure 4 displays the original image (top left) and the three steps of the segmentation. The first step (top right) splits the image into homogeneous regions and unstable small regions. The second step (bottom left) merges the homogeneous regions obtained in the first step and yields samples of the different textures (equal to two in this example, white and gray pixels regions) on which the features are computed. Dark pixels corresponds to small regions that are left aside during the merging process. The final segmentation step (bottom right) includes a comparison between normalized energy vectors at each point and those computed in the merging step and a filtering procedure using a median filter.

C. Computational cost of the algorithm

To comment on the computational cost of the algorithm we use the diagram of Figure 3. The tight frame decomposition is in $O(N^2 log(N))$, the splitting step, the merging step are both in $O(N^2)$, the constant in front of N^2 is a function of the parameter λ (defined in the splitting step). The bigger λ is, the bigger the computational cost of the splitting and the merging steps are. The computation of the local normalized energies and of the post processing (i.e median filtering) are both in $O(N^2)$ but the constant is larger for the median filtering operation. To summarize, the value of λ monitors the costs of the splitting and merging steps and the size of the median filter for post processing plays central role in the second part of the algorithm.

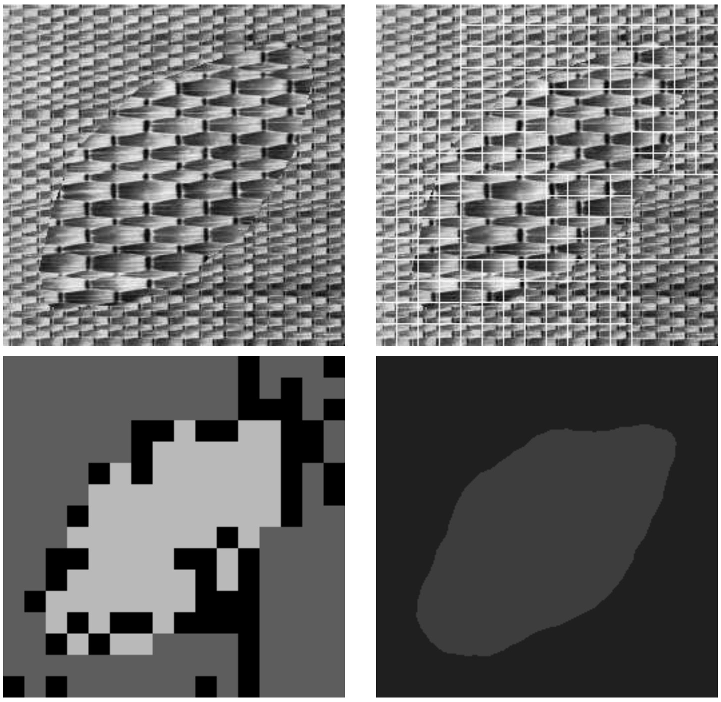

Fig. 4. Segmentation results for texture Nat2: we display the image, the results of the splitting step ($\lambda = 0.7$), the merging step ($p_2 = 0.03$ and $p_1 = 15$, the darker pixels are not used in the estimation of the features) and the segmentation results

V. Result

A. Performance of the Split and Merge Procedure for Computing the Number of Textures

To test the performance of the split and merge procedure, we compute the number of textures found for images of Figure 5 with respect to parameters p_2 and λ with p_1 fixed to 15.

Table 5 displays the number of textures found when p_2 and λ vary. We note that for p_2 in $[0.02, 0.04]$ the true number of textures is found for λ in $[0.5, 0.7]$. This suggests that $p_2 = 0.03$ and $\lambda = 0.7$ are appropriate for texture segmentation tasks. These values are used in the final segmentation step.

B. Final Segmentation Results

The efficacy of the merging step is measured on the one hand by the results of the final segmentation step and, on the other hand, by the number of iterations in the final segmentation step to reach stable classes. On the examples of textured images we studied, the number of iterations never exceeds 10. Furthermore, the feature estimation provided by the merging step limits the possibility for the segmentation algorithm to get trapped in local minima. Such a situation typically appears when one uses the Kmeans algorithm directly on the coefficients of the wavelet filter decomposition.

C. Comparison with other Methods on Reference Images

We now compare our method to other existing methods on textures 11a and D17D55 which are displayed in Figure 5. Referring to the study of stability, we fix the size of the regions I_{UL}, I_{UR}, I_{LL} and I_{LR} to 16×16. We then study the influence of the size of the window used for the median filter on classification results. Table 6 displays the percentage of good classification for textures D17D55 and 11a when the size of the median filter varies. These results suggest that using a larger window for median filtering improves the segmentation results. However, one should bare in mind that this is done to the detriment of the computational cost of the algorithm.

As far as the texture 11a is concerned the results we obtain are of the same quality as that obtained with the best heuristically designed feature extractor reported by Randen in [20]. For texture D17D55 the results

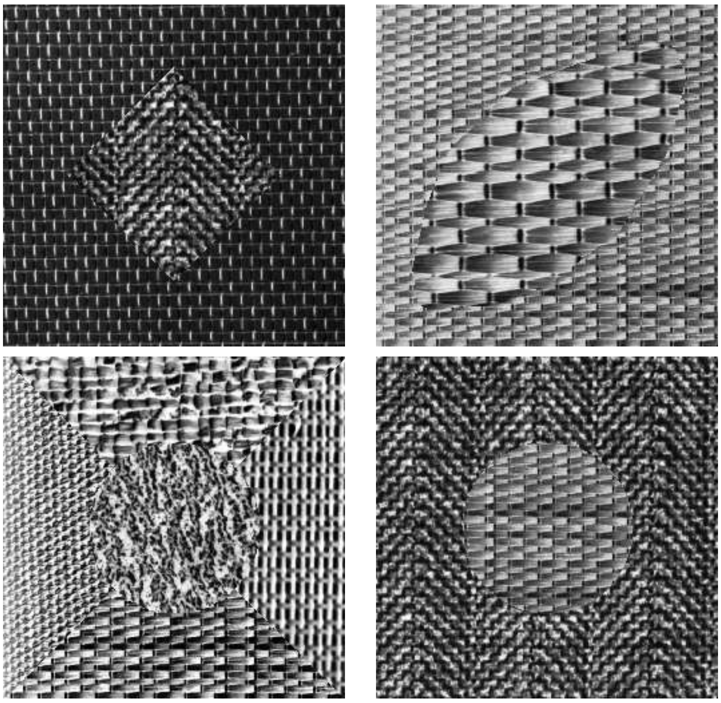

Fig. 5. Textures image : from left to right and top to bottom Nat3, Nat2,11a,D17D55, they are all of size 256×256

	image dimension	$\hat{J}$	p_2=0.02	p_2=0.04	p_2=0.1
Nat3	256	4	2	2	2
Nat2	256	4	2	2	2
11a	256	4	5	5	3
D17D55	256	4	2	2	2

$\lambda = 0.5$, $p_1 = 15$

	p_2=0.02	p_2=0.04	p_2=0.1
Nat3	2	2	2
Nat2	2	2	2
11a	5	5	3
D17D55	2	2	2

TABLE V

Number of textures found for texture Nat3, Nat2, 11a, D17D55, the value for $\hat{J}$ corresponds to $R = 0.9$

Texture	Median filter size	percentage of good classification
11a	33	91,3 %
11a	17	91,2 %
11a	9	90,7 %
D17D55	33	97,8 %
D17D55	17	97,7 %
D17D55	9	97,4 %

TABLE VI

Percentage of good classification with the size of the median filter

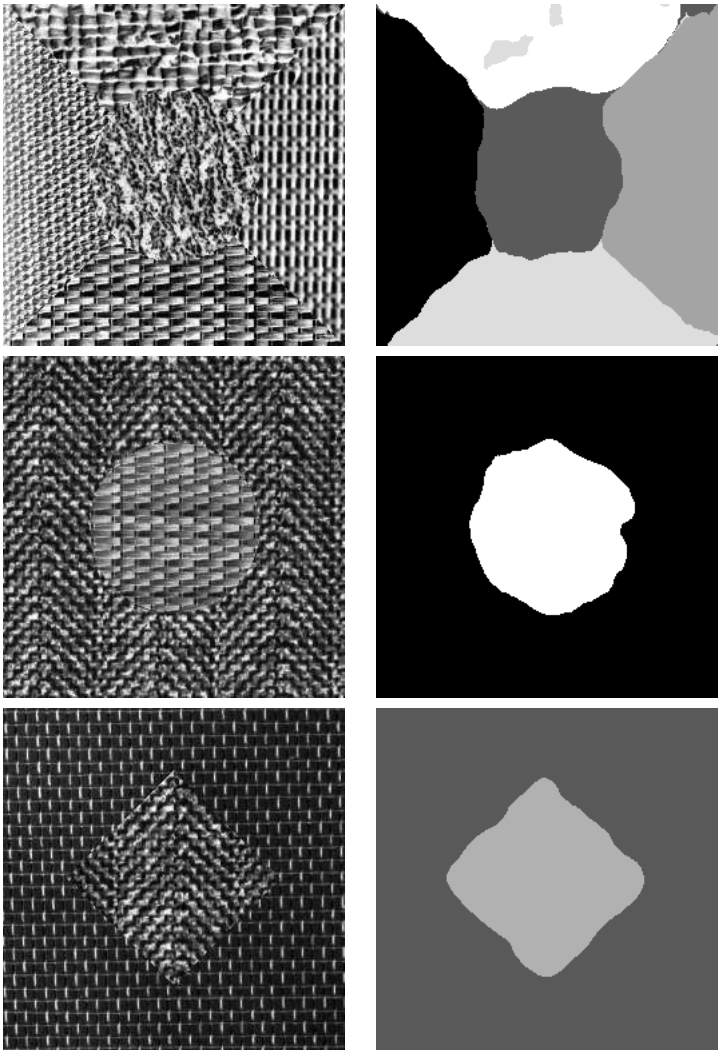

Fig. 6. Segmentation results for texture D17D55, 11a and Nat2

we obtain, 97,8 % of good classification, is a bit worse than the best known method design by Farrokhnia [8] who reports 98,3 % of good classification. However the complexity of the algorithm proposed in [8] is very high and requires an a priori knowledge on the input textures. Acharyya *et al.* [1] also reports slightly better results (98,1 %) but on the other hand assumes the number of textures to be known.

The specificity of our approach is to determine the number of texture in the image in an efficient manner and then to provide segmentation results which are close to the best known results for the two reference image of textures we studied. The segmentation results we obtain on that two reference texture images are available on Figure 6. For texture Nat2 also studied by Acharyya [1] the segmentation results are visually similar.

VI. Conclusion

The scope of this paper was first to study the stability of texture features associated with a tight frame decomposition with texture sample size. The comparison of the features in terms of stability on a benchmark of textures justifies the use of the most energetic channels for the construction of texture features. From this, we derived stable features, the normalized energy vectors, which we use in a new unsupervised texture segmentation algorithm made of three steps: a splitting step, an agglomerative merging step and a final segmentation step.

The specificity of the algorithm is that it allows to determine the number of textures and it estimates texture features through a split and merge procedure. Those features are then used in the segmentation step. Our approach avoids traditional discussions on feature selection. Furthermore, the algorithm is adaptative in the sense that it allows some variability on the channel energetic content. Indeed, we have shown that selecting a priori some channel indices as features is not a good choice due to their instability.

For the reference textures we considered, the results are close to the best known segmentation results. The algorithms that perform better than our contain more a priori knowledge on textures, for instance the number of textures is most often supposed to be known. We have shown that our texture detector works well when the number of textures in the image is moderate, we soon expect to show that this procedure is robust even when the image contains more textures or natural scenes.

VII. Acknowledgments

This work is partially supported by the European project IHP "Breaking complexity" HPRN-CT-2002-00286 and by the "Région Rhône-Alpes".

References

[1] M. Acharyya and M.K. Kundu, *An adaptative approach to unsupervised texture segmentation using M-Band Wavelet transform*, Signal Processing, vol. 81, pp.1337-1356, 2001.

[2] M. Acharyya and M.K. Kundu, *Adaptative basis selection for multi-texture segmentation by M-band wavelet packet frames* Proc. Int. Conf. On Image Processing (ICIP'01), Greece, pp. 622-625, 2001.

[3] R. Azencott, J.P Wang and L. Younes, *Texture classification using windowed Fourier filters*, IEEE Trans. Patt. Anal. Machine Intell., vol. 19, no. 2, pp. 148-153, 1997

[4] A.C. Bovik, M. Clark and W.S. Geisler, *Multichannel texture analysis using localized spatial filters*, IEEE Trans. Patt. Anal. Machine Intell., vol. 12, no. 1 pp. 55-73, 1990.

[5] P. Brodatz, Textures: A photographic album for artists and designers, New York: Dover, 1966.

[6] T. Chang and C.C.J. Kuo, *Texture analysis and classification with tree-structured wavelet transform*, IEEE Trans. Image Processing, vol. 2, no. 4, pp. 429-441, 1993.

[7] J.M. Coggins and A.K. Jain, *A spatial filtering approach to texture analysis* Pattern Recognition Letters, vol. 3, no. 3, pp. 195-203, 1985.

[8] F. Farrokhnia, *Multichannel Filtering Techniques for Texture Segmentation and Surface Quality inspection* PhD thesis, Michigan Sate university, 1990.

[9] C.M. Gray and W.Singer, *Stimulus-specific neural oscillations of cat visual cortex*, Proceedings of the national Academy of Sciences, USA, vol. 86, 1698-1702, 1989.

[10] T. Hofmann, J. Puzicha and J. M. Buhmann, *Unsupervised Texture Segmentation in a Deterministic Annealing Framework* IEEE Trans. Patt. Anal. Machine Intell., vol. 20, no. 8, pp. 803-818, 1998.

[11] A.K. Jain and F. Farrokhnia, *Unsupervised texture segmentation using Gabor filters*, Pattern Recognition, vol. 24, no. 12, pp. 1167-1190, 1991.

[12] A.K. Jain, *Algorithms for clustering data*, Prentice Hall, 1989.

[13] S. Jaffard, Y. Meyer and S. D. Ryan, *Wavelets : Tools for science and technology*, SIAM, 2001.

[14] A. Laine and J. Fan, *Texture classification by wavelet packet signatures*, IEEE Trans. Patt. Anal. Machine Intell., vol. 15, no. 11, pp. 1186-1191, 1993.

[15] A.Laine and J.Fan, *Frame representation for texture segmentation*, IEEE Trans. Image Processing, vol. 5, pp. 771-780, 1996.

[16] S.Mallat, *A wavelet tour of signal processing*, Academic Press, 1998.

[17] A. Mojsilovic, M. V. Popovic and D.M. Rackov, *On the selection of an optimal wavelet basis for texture characterization*, IEEE Trans. Image Processing, vol. 9, no. 12, pp. 2043-2050, 2000.

[18] T. Ojala and M. Pietikäinen, *Unsupervised texture segmentation using feature distribution*, Pattern Recognition, vol. 32, pp. 477-486, 1999.

[19] R. Porter and N. Canagarajah, *A robust automatic scheme for image segmentation using wavelets* IEEE Trans. Image Processing, vol. 5, no. 4, pp. 662-665, 1996.

[20] T. Randen and J.Husoy, *Filtering for texture classification : a comparative study*, IEEE Trans. Patt. Anal. Machine Intell., vol. 21, no. 4, pp. 291-310, 1999.

[21] O. Rioul, *A discrete-time multiresolution theory*, IEEE Trans. Signal Processing, vol. 41, no. 8, pp. 2591-2606, 1993.

[22] A. Teuner, O. Pichler and J. Hosticka, *Unsupervised textures segmentation of images using tuned matched Gabor filters*, IEEE Trans. on Image Processing, vol. 4, no. 6, pp. 863-870, 1995.

[23] M.Unser, *Texture classification and segmentation using wavelet frames*, IEEE Trans. Image Processing, vol. 4, no. 11, pp. 1549-1560, 1995.

[24] M. Vetterli, *Filter bank allowing perfect reconstruction*, Signal Processing, vol. 10, no.3, pp. 219-244, 1986.

[25] VisTex, *Color image database*, http:// www-white.media.mit.edu/vismod/imagery/VisionTexture, MIT Media Lab, 1995.