CCIT Report #570
January 2006

Classification and Segmentation of Textures based on Statistical Models of Wavelet Transforms

R. Mittelman and M. Porat
Department of Electrical Engineering
Technion-Israel Institute of Technology
Haifa 32000, Israel

Abstract

In this work we introduce new multiresolution statistical models for texture classification and segmentation. We develop features for wavelet-based texture classification using statistical modeling based on a Gaussian scale mixture (GSM), and present a new feature weighting scheme suitable for histogram features. Texture segmentation is performed using Markov random fields and variational level-set frameworks. Multi-scale variational level-set texture segmentation is introduced, which in contrast to presently available segmentation schemes, employs a coarse to fine strategy. The new scheme is based on the dual tree complex wavelet transform and hidden Markov trees statistical modeling. The proposed scheme achieves better classification results than those reported in the literature for the same textures and tasks. Classification and segmentation examples are presented and discussed. The segmentation results indicate that the new scheme is superior to presently available methods, in particular when segmenting non-convex shapes, since the coarse to fine strategy facilitates the dependence on the rigidity of the contour. Our conclusion is that the proposed approach could improve presently available tools for classification and segmentation of textures.

Corresponding Author:

Roni Mittelman,
Department of Electrical Engineering,
Technion, Israel
Email: rm@vision.technion.ac.il

1) Introduction

Texture characterization has been an active research area for many years now. Although no clear definition of texture exists, it is clear that any mathematical modeling of textures should account for the unique spatial variability or pattern in the gray level intensities present in the given texture. In [1] four major categories have been identified for the characterization of textures: spectral methods, geometrical methods, model based methods, and signal processing methods.

Statistical methods assume that the unique spatial variability can be characterized by the spatial distribution of the gray level values (such as in the case of the co-occurrence matrix). Model based methods assume that a texture is generated by a certain image model. An example of such an image model is Markov random field (MRF) where the intensity of each pixel is assumed to depend on the intensities of its neighboring pixels alone. Spectral methods characterize textures either in the frequency domain (Fourier transform), or in a combined time frequency domain (multi-resolution representations).

In this work, spectral model-based and statistical methods for texture characterization are employed for texture classification and texture segmentation. Applying a multi-resolution analysis for textures is advantageous as it provides a non-redundant representation [2] which is de-correlated compared to the original image. This is extremely important when considering statistical characterization since as a first order approximation the multiresolution coefficients can be assumed to be independent. Furthermore, pyramid structured multi-resolution representations provide a coarse to fine framework for interpreting images. Early approaches for statistical characterization of the spectral representation assumed that the marginal statistics of the coefficients are Gaussian [3], however followed by Mallat's observation [4] that the coefficients exhibit distinct non-Gaussian statistics, new statistical models were proposed. Mallat suggested the Generalized Gaussian (GG) [4] model and recently the hidden Markov trees (HMTs), which avoid the statistical independence assumption, were suggested [5].

When considering texture classification, further issues related to pattern recognition such as the choice of classifier and feature extraction must be considered. One popular choice when assuming statistical characterization is to use histogram features [6], [7] since such features are very simple to use. Another important issue is the computational complexity of the classification scheme, otherwise the scheme may be impractical.

When considering texture segmentation, prior information about the physical process must be used in order to obtain good results. The most common property which can be enforced using a prior is the spatial continuity of the textures. Two methods to enforce the physical continuity of the textures which are employed in this work, are the MRF and level set frameworks. The Ising model pertaining to the MRF framework, enforces the smoothness of the labels in the field, whereas the level set framework minimizes the length of the contours separating the different texture regions, thus large clusters are formed.

The rest of this work is organized as follows. Section 2 presents a new approach to feature extraction for wavelet based texture classification. Section 3 explores the use of the new features

introduced here to unsupervised texture segmentation using Gaussian Markov random fields (GMRF) and MRF. Section 4 presents a new multi-scale level set supervised texture segmentation scheme which employs a coarse to fine strategy, thus we show that the new scheme may have advantages over presently available level set supervised texture segmentation schemes where the segmentation is performed on a single level. Section 5 concludes this work.

2) Texture Classification

The two basic stages of every texture classification algorithm are the feature extraction and classification stages. The feature extraction stage extracts features of the given texture patch, which are expected to characterize the texture. Subsequently these features are employed in the classifier in order to assign a texture class for the image patch. In order to train the classifier, a training sequence of textures and their given classification is employed.

A well known problem in pattern recognition is the curse of dimensionality which stipulates that the classification performance does not necessarily increase with an increasing number of features. The curse of dimensionality often necessitates a feature selection stage to be used prior to classification in order to select a subset of the features that has the best overall classification performance over the training dataset. However as the number of possible subsets is exponential, performing an exhaustive search is impractical for even a small number of features. There have been different approaches to find a sub-optimal subset of features [8], however such approaches increase the computational complexity considerably. It could be argued that since the feature selection is performed during an off-line training stage, its computational complexity is insignificant, however another drawback of employing a feature selection stage, aside from its high computational complexity, is that adding or removing a single texture class will necessitate a repetition of the feature selection training stage. This may be a serious drawback for example, in content based image retrieval systems where texture features are used to retrieve images. In such a case the texture database is expected to be dynamic. Another approach which tackles the curse of dimensionality with a lesser computational cost compared to feature extraction schemes, is to use feature weighting. The approach taken by feature weighting schemes is to weight each feature according to its importance instead of selecting a subset of all the features. This has the potential to eliminate the effect of features which do not contribute to the classification, while being computationally efficient as it does not require a selection of a subset of features. However feature weighting may not always be a proper substitute to feature selection.

One common approach to classification is to employ a minimum distance classifier, where the features of each texture image patch are compared with the set of signatures computed during the training stage, and the texture class selected is that which gives the minimum distance. When considering non-parametric histogram features, several dissimilarity measures which measure the distance between two PDFs such as the Kullback-Leibler (KL) or the χ^2 dissimilarity measures have

been used. Assume that H_1 and H_2 are two histograms with N bins each, then the KL dissimilarity measure takes the form:

$$KL(H_1, H_2) = \sum_{i=1}^{N} H_1(i) \log \frac{H_1(i)}{H_2(i)}, \tag{2.1}$$

and the χ^2 dissimilarity measure which takes the form:

$$\chi^2(H_1, H_2) = \sum_{i=1}^{N} \frac{(H_1(i) - H_2(i))^2}{H_1(i) + H_2(i)}, \tag{2.2}$$

A major problem of employing either the KL or the χ^2 dissimilarity measures for classification using histograms is that they both necessitate the bins' centers of the histograms to be fixed in advance and therefore introduce adverse effects, such as a strong dependence on the number of bins used. One solution is to use the earth movers distance (EMD) dissimilarity measure [6] however this entails a high computational cost compared to the KL and χ^2 dissimilarity measures. The EMD is formulated as a linear programming optimization problem, which solution can be interpreted as the minimum cost required to transform one histogram to the other, where each histogram can have different bins' centers. Therefore since there is no requirement that both of the histograms have the same bins' centers, the histogram features can characterize each texture more accurately.

The new approach presented here deals with the two problems described previously. In order to avoid a feature selection stage we present a new feature weighting scheme which is adapted to histogram features. In order to solve the histogram scaling problem when using the KL or χ^2 dissimilarity measures, we introduce new features such that their prior scaling is known and identical to every histogram.

2.1 New Features for Texture Classification

The new features are based on histograms of the local second moment estimates of the wavelet transform coefficients. Furthermore it is shown that the bins' centers of the histograms of such features should be logarithmically scaled, thus the major drawback of the KL and χ^2 dissimilarity is avoided since the histograms can be captured efficiently although the bins' centers are fixed in advance. The dissimilarity measure of choice in this work is the χ^2 dissimilarity measure.

2.1.1 The New Features

The new features correspond to the local second moment estimates of the wavelet transform coefficients. It is possible to show that such estimates can be interpreted as the ML estimates of the hidden multipliers of a Gaussian scale mixture (GSM) distribution:

If X is a GSM random vector then its density is given by

$$p(X) = \int p(X \mid z)\, p(z)\, dz, \tag{2.1.1}$$

(2.1.2) $$p(X \mid z) = N(0, zC_u),$$

where C_u is a covariance matrix of a Gaussian random vector U and is independent of z.

Taking the logarithm of (2.1.2) and minimizing for z, one can obtain the ML estimator of the hidden multiplier for each wavelet coefficient:

(2.1.3) $$\hat{z} = \frac{1}{N} X^T C_u^{-1} X,$$

where N is the number of elements in the vector X.

Since $X = \sqrt{z}U$, if we denote the mean value of z by m, then we have $C_x = mC_u$, where C_x is the covariance matrix of the random vector X. Since the wavelet transform coefficients are roughly de-correlated we may write $C_x = \sigma_x^2 I$, where I is the identity matrix, therefore using (4.3.3) the ML estimate of z can be obtained using:

(2.1.4) $$\hat{z} = \frac{m}{N\sigma_x^2} X^T X.$$

Since m/σ_x^2 is just a normalizing constant, we can choose a new matrix $C_u^{'} = \frac{1}{m/\sigma_x^2} C_u$, which we can use in (2.1.3) instead of C_u. Thus the hidden multipliers can be obtained using:

(2.1.5) $$\hat{z} = \frac{1}{N} X^T X.$$

Thus we have shown that the local second moment estimates correspond to the ML estimates of the hidden multipliers of a GSM distribution.

2.1.2 Non-Informative Priors

Using (2.1.5) it is possible to estimate the hidden multiplier of each coefficient of the wavelet transform, using a local region containing it. Subsequently the estimates of the hidden multipliers can be used to construct a histogram which approximates the PDF of the hidden multipliers. One major concern when constructing a histogram is to determine the bins' centers, since this may have a significant effect on the classification performance. If the bins' centers could be chosen arbitrarily then it is possible to employ adaptive binning [9] and thus the distribution can be represented more efficiently. Adaptive binning algorithms for histogram construction are similar to vector quantization schemes for constructing a dictionary. Each training sample is alternately assigned to each bin according to the nearest neighbor criteria, and the bins' centers are updated using the centroid of the cluster assigned to each bin. However since the new approach employs the χ^2 dissimilarity measure, adaptive binning can not be employed and fixed binning must be used since the bins' centers of each of the histograms must be identical. The most common approach to fixed binning is to divide the range linearly, where the number of bins is assumed to be given. If we formulate the problem of finding the bins' centers as finding a prior distribution for the bins' centers, the conventional approach is then

interpreted as choosing a uniform distribution for the prior for bins' centers. This approach assumes no a priori knowledge about the bins' centers. The problem of estimating a prior distribution is encountered in many Bayesian inference problems. A common solution is to use a non-informative prior which has the advantage of being parameter free. One such non-informative prior is known as the Jeffrey's prior [10]. The Jeffrey's non-informative priors are based on the Fisher information given by:

$$I(z) = E\left(-\frac{\partial^2 \log p(X \mid z)}{\partial z^2}\right), \tag{2.1.6}$$

and the corresponding prior distribution is:

$$p(z) \propto \sqrt{I(z)}\,, \tag{2.1.7}$$

where z is the parameter (in our case the hidden multipliers).

The choice of a prior proportional to the Fisher information may be justified intuitively, as the Fisher information $I(z)$ is accepted as an indicator of the amount of information the model has about the parameter z. Therefore it may be argued that values of z for which $I(z)$ is larger should have a higher prior probability.

2.1.3 A non-informative Jeffrey's prior for the Hidden Multipliers

In [11] a non-informative Jeffrey's prior was derived. Using (2.1.2) in (2.1.6) it can be obtained that:

$$-\frac{\partial^2 \log p\,(X \mid z)}{\partial z^2} = \frac{\partial^2}{\partial z^2}\{\frac{1}{2}[N \log\,(z) + \log\left|C_u\right| + \frac{X^T C_u^{-1} X}{z}]\}$$

$$= \frac{-N}{2z^2} + \frac{X^T C_u^{-1} X}{z^3}\,,$$

and since $E\{X^T C_u^{-1} X\} = z \cdot N$, using (2.1.7) we get Jeffrey's non-informative prior:

$$p_z(z) \propto \frac{1}{z}\,, \tag{2.1.8}$$

which is an improper density, but this can be ignored by setting the prior to zero in the interval $(0, z_{\min}]$, where $z_{\min}$ is a small positive constant.

2.1.4 A Prior for the Bins' Centers

We seek a prior $p_{z_0}(z_0)$ for the bins' centers. Clearly larger values of $p_{z_0}(z_0)$ correspond to a denser distribution of bins, and smaller values of $p_{z_0}(z_0)$ correspond to a sparser distribution of bins. We define z to be the hidden multiplier and define the joint distribution:

$$p_{z,z_0}(z, z_0) = p_{z|z_0}(z \mid z_0)\, p_{z_0}(z_0)\,, \tag{2.1.9}$$

and define

$$p(X \mid z, z_0) = p(X \mid z)\delta(z - z_0)\,. \tag{2.1.10}$$

Now we can express equation (2.1.1) in a form which incorporates the bins' centers prior distribution $p_{z_0}(z_0)$:

(2.1.11) $$p(X) = \iint p(X \mid z, z_0) p_{z,z_0}(z, z_0) dz_0 dz ,$$

The formulation in (2.1.11) is equivalent to (2.1.1) since we can choose $p_{z,z_0}(z, z_0) = p(z)$ where $p(z)$ is the same distribution used in (2.1.1). Using (2.1.9) and (2.1.10) in (2.1.11) we have:

(2.1.12) $$p(X) = \int p(X \mid z_0) p_{z|z_0}(z_0 \mid z_0) p_{z_0}(z_0) dz_0 .$$

Clearly we can get a prior for $p_{z_0}(z_0)$ in the similar manner that (2.1.8) was derived. Therefore the prior distribution for z_0 is $p_{z_0}(z_0) \propto \frac{1}{z_0}$. Using the prior for the bins' centers $p_{z_0}(z_0)$ in (4.3.12) and transforming variables we get:

(2.1.13) $$p(X) = \int p(X \mid e^{z'}) p_{z|z_0}(e^{z'} \mid e^{z'}) dz' = \int p(X \mid e^{z'}) p_{z'}(z') dz' ,$$

where $p_{z'}(z') = p_{z|z_0}(e^{z'} \mid e^{z'})$.

Therefore if $p_{z'}(z')$ is to be estimated using the hidden multipliers' samples, it should be scaled logarithmically. It is interesting to note that the formulation in (2.1.1) and (2.1.13) are equivalent but for their scaling.

2.1.5 The New Features as a Pre-Processing Scheme

The new approach to feature extraction presented here can be formulated as a special case of a pre-processing scheme presented in figure 2.1 which was discussed in [12]. If the first non-linear operator used is squaring and the smoothing is obtained using averaging, then the local energy function is obtained during the process of estimating the hidden multipliers. The second non-linearity which is employed in the new approach is logarithmic and could be interpreted as a transformation of the PDF to be defined on a linear scale rather than a logarithmic scale. In our approach the logarithmic non-linearity is used in order to determine the bins' centers, instead of transforming the PDF into linear scaling. The squaring and logarithmic non-linear operators which are used in the new approach are in concurrence with those which have been reported to provide the best classification performance in [13].

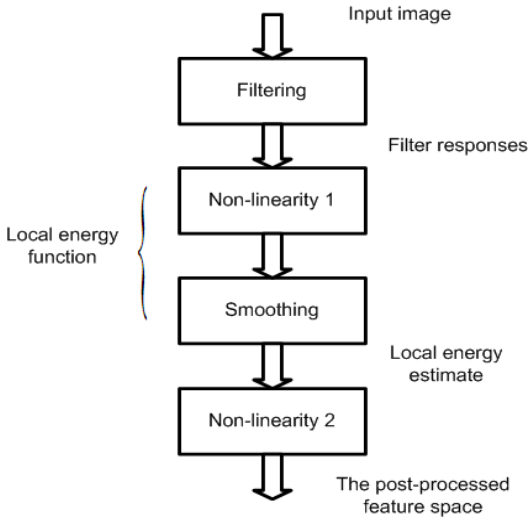


Figure 2.1: A pre-processing scheme

2.2 A Histogram Weighting Scheme

A feature weighting scheme which employed variance features and an Euclidean dissimilarity measure was presented in [14]. It was suggested to weight each feature using the inverse variance of the feature within the training dataset, thus the effect of features which degree of dispersion is relatively small will be enhanced, and the effect of features which degree of dispersion is relatively large will be weakened. The classifier employed was a minimum distance classifier where each sample was tested against each of the possible texture classes and weighted accordingly. When considering histogram features as well as a non-Euclidean dissimilarity measures, some important modifications must be introduced to the scheme presented in [14]. As the histogram features are composed of many bins, each of which can show variation in itself, each bin should be regarded as a different feature and weighted according to its dispersion in the training dataset. Furthermore we consider the χ^2 dissimilarity measure and define a modified variance which generalizes the definition of the variance to different dissimilarity measures which are not Euclidean. Such a feature weighting scheme is especially important in the new approach presented here because of the fully decimated wavelet transform employed, and because of the small texture patches used, which result in a very limited number of samples available to train the histograms. The conventional solution in this case is to use Parzen windows in order to smooth the histograms, however the number of available samples to train the histograms is so small that our experience has shown that smoothing the histograms did not provide an increase in the classification performance.

We notice that the χ^2 dissimilarity measure can be formulated such that it measures the distance between bins instead of histograms [15]. Now suppose that a histogram was constructed for each sample in the training dataset, then the signature associated with a given texture class takes the form:

(2.2.1) $$H^{'}(i)=\frac{1}{T}\sum_{t=1}^{T}H^{(t)}(i),$$

and the modified variance takes the form:

(2.2.2) $$\sigma^2(i)=\frac{1}{T}\sum_{t=1}^{T}\chi^2\left(H^{(t)}(i),H^{'}(i)\right),$$

the minimum distance decision rule then takes the form:

(2.2.3) $$k=\underset{n}{\operatorname{argmin}}\quad\sum_{j=1}^{M}\sum_{i=1}^{N_j}\frac{1}{\sigma^2_{n,j}(i)}\chi^2\left(H^x_j(i),H^{'}_{n,j}(i)\right),$$

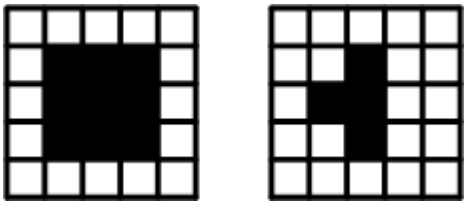
where M denotes the number of feature histograms, N_j denotes the number of bins in histogram with index j, H^x denotes the query's features, $H^{'}_{n,j}$ denotes the signature of the histogram with index j of the texture class with index n.

2.3 The New Classification Scheme

This section presents the new classification scheme which is employed in the new approach. The classifier performs minimum distance classification, where the features are weighted according to their dispersion within the training dataset, using the new feature weighting scheme which is adapted to non-Euclidean dissimilarity measures. Thus the new scheme can avoid a feature selection stage.

2.3.1 Combining Different Feature Types within the Classifier

Another advantage of employing a weighting scheme is that we can employ different feature types within the classifier, since the weighting scheme can make sure no single feature will be dominant because of its higher dynamic range, as all the features are normalized. We have used three types of features in this work. The first two feature types correspond to the local second moment estimates of the wavelet coefficients which were scaled logarithmically. The local regions used to compute the estimates are shown in figure 2.2. The last feature type used is the histogram of the absolute value of the wavelet coefficients which was scaled linearly.



(a) (b)

Figure 2.2: The different regions used to compute the local second moment estimates

2.4 Experimental Results

In order to evaluate the new approach we use an identical experimental setup and tools as those reported in Liu&Wang'03 [16] and show that the classification results of the new scheme are better

than those reported there. The texture dataset which we use consists of 40 textures from the VisTex database available at [17], which is also the dataset used in [16]. The textures are shown in figure 2.3.
The training and test patches were obtained from the top left 256x256 part of each image in [17]. The training and test samples were images of size 32x32, where each texture image was randomly divided into non-overlapping training and test sets with an equal number of texture patches in each set. The texture patches are very small thus the classification task is very difficult, since larger texture patches lead to a more reliable density estimate thus the characterization of each texture is improved. Furthermore several of the textures are very similar, complicating the classification task even more. Each of the experiments was repeated for 100 times to avoid the bias introduced by the different training and test sets. Each texture patch was decomposed with a three level pyramid structured wavelet transform. We used the Daubechies four tap filters [18], where both the wavelet and scaling sub-bands were used. The number of bins used for each of the histograms associated with each of the scales of the logarithmically scaled features was identical all over the scales and orientations, whereas the number of bins used for the linearly scaled histograms was different for each scale, since such a feature is greatly affected by the number of bins used. On the other hand, the logarithmically scaled features were more robust in that respect.
Our experiments have verified that the histograms of the local second moment estimates should be scaled logarithmically, since the correct classification rate for that case was considerably higher compared to using linear scaling. Also the choice of the weights as the modified variance showed considerably improved correct classification rate compared to using the variance as the weight.
Table 4.3 shows the results obtained using various combinations of the features. It can be seen that the weighting scheme improved the correct classification rate all over the cases examined. The best result is obtained when all the features are used together with the weighting scheme. The average correct classification rate obtained in this case was 93.74%, compared to an average correct classification rate of 92.5% which was recently reported in [16], where both the textures and testing procedures used were identical.

Features used	Feature Weighting	Feature Selection	Average [%]	Worst [%]	Best [%]
1	No	No	82.98	81.17	85.08
2	No	No	89.34	87.66	91.17
3	No	No	87.45	85.23	89.30
1+2+3	No	No	89.99	88.28	91.87
1	Yes	No	89.15	87.34	91.25
2	Yes	No	91.63	89.45	93.28
3	Yes	No	92.10	90.63	93.91
1+2+3	Yes	No	93.74	92.03	95
Liu & Wang '03	No	Yes	92.5	N/A	N/A

Table 2.1: Comparison of classification results for 40 textures using the various features used in this work. 1, 2, 3 denote the histogram features of:
1- The absolute value of the wavelet coefficients, 2- The local second moment estimates using the local region shown in figure 2.2a, 3- The local second moment estimates using the local region shown in figure 2.2b.
Liu & Wang '03 denote the results reported in [16] using identical experimental tools.

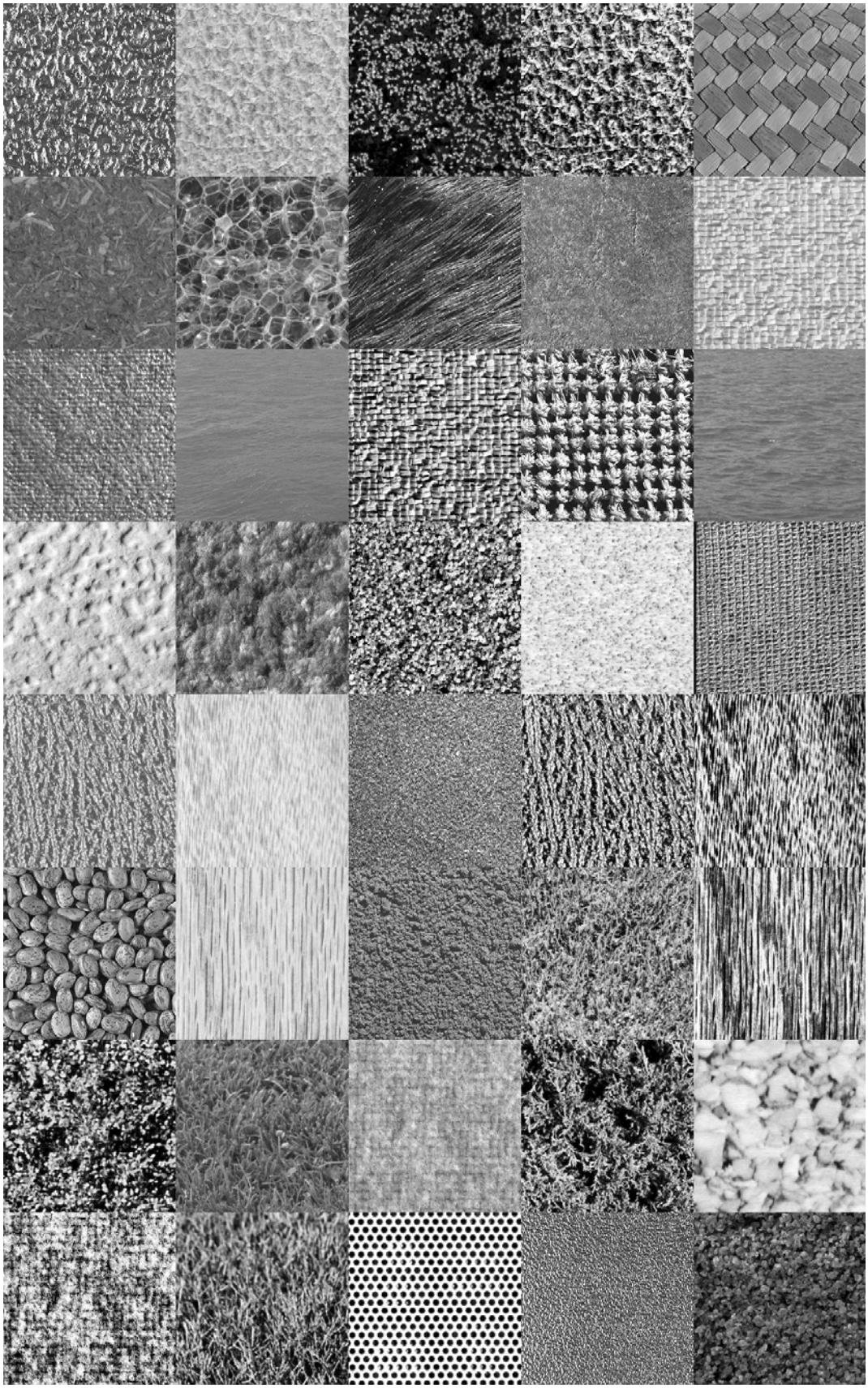


Figure 2.3: The 40 textures used in the classification experiments

3 A New Approach to Multi-Resolution Texture Segmentation Using Gaussian Markov Random Fields

There have been many applications presented in literature in which Gaussian Markov random fields (GMRFs) were employed for modeling of image statistics. GMRF modeling accounts for the correlation structure of an observation field while keeping the derivation tractable. A GMRF model assumes that an observation field is a realization of a multivariate Gaussian with a certain parametric type of covariance matrix which in turn necessitates the marginal distributions to be Gaussian. Therefore GMRF modeling may not be efficiently utilized when the marginal distribution of the observation field does not conform to a Gaussian distribution. When modeling the statistical structure of the wavelet transform coefficients of natural images, the GMRF model is often not applicable since the marginal distributions of the wavelet coefficients of natural images are characterized by a large peak at zero and heavier tails than a Gaussian of the same variance [19].

In this section we exploit the observation that the marginal distribution of the features obtained by the pre-processing scheme discussed in section 2, can be modeled closely using a Gaussian. Thus we argue that the post-processed features are more appropriate to texture segmentation using GMRF compared to multi-resolution representations such as the wavelet transform.

3.1 The Marginal Distributions of the Post-Processed Features

It was reported in [13] where the pre-processing scheme shown in Figure 2.1 was evaluated, that the post-processed features follow a Gaussian distribution. This is also in concurrence with the observation that the pre-processing scheme computes ML estimates of the hidden multipliers which were shown empirically in [20] to be closely modeled by a Gaussian density on a logarithmic scale. It was further noted in [13] that the post-processed feature space tends to transform variance differences into mean differences, thus it is easier to converge to the right parameters when using the post-processed features. Since GMRF modeling is equivalent to assuming a multivariate Gaussian distribution with a certain type of covariance matrix, it lends itself efficiently to modeling the post-processed feature space. Since the marginal distribution is Gaussian, modeling the post-processed feature space using a GMRF is more accurate than modeling other feature spaces such as the Gaussian pyramid or the wavelet transform using a GMRF.

3.2 The Statistical Models Used in the New Scheme

The texture segmentation scheme presented here follows [21] closely with a few major differences described in section 3.5. We consider semi-unsupervised texture segmentation where the number of texture classes is known in advance, and the parameters of the observation field are estimated in an unsupervised manner. The parameters of the label field were selected experimentally. Since we employ here a Bayesian approach to texture segmentation we have to define the PDF both for the label process

and for the observation process. We model the observation process given the texture class as a causal GMRF as it leads to an efficient parameter estimation algorithm. The label process is modeled using a multi-scale Ising MRF model, where the neighborhood system for such a model is shown in figure 3.1. We denote $X^{(n)}$ to be a raster scanned vector of the class labels assigned to level n of the lattice, which can take values from the class label set $\{1,\ldots,M\}$. Its distribution follows a multi-scale Markov random field, where both intra-scale and inter-scale pair-site clique types, denoted C_1 and C_2 respectively, are used. C_1 is the set of all first order neighborhoods in the same level, and C_2 is the set of all the neighborhoods which include a node at level n, and its parent at level $n+1$, and its four sons at level $n-1$ as shown in figure 3.2 (level L is the coarsest level and 1 is the finest level).

Accordingly, the probability mass function for $X^{(n)}$ is given by:

$$p_{X^{(n)}}(x) = \frac{1}{z^{(n)}} \exp\left\{ -\sum_{r,s\in C_1} \beta_1^{(n)}(1-\delta(x_r - x_s)) - \sum_{r,s\in C_2} \beta_2^{(n)}(1-\delta(x_r - x_s)) \right\},$$

where $z^{(n)}$ here is a normalizing constant.

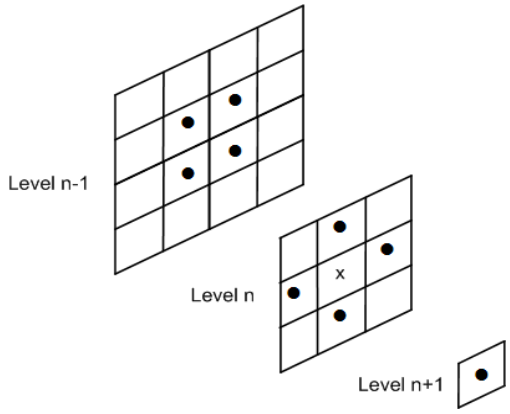


Figure 3.1: The neighbourhood system used for the multi-scale MRF model

We denote $Y^{(n,k)}$ to be a column stacked vector of the observed data field at level n which is assumed to be a causal GMRF model conditioned on the class label field, where the number of prediction coefficients is R. We assume that the observations of each of the multi-resolution's scales and orientations are statistically independent. As was shown in chapter 3 the conditional probability density function of Y given X is:

$$f_{Y|X}(y \mid x) = \prod_{n=1}^{L} \prod_{k=1}^{O(n)} \prod_{s\in S(n,k)} \frac{1}{\sqrt{2\pi\sigma_s^2}} \exp\{-\frac{(\tilde{y}_s^{(n,k)})^2}{2\sigma_s^2}\}$$

(3.2.1)

$$\tilde{y}_s^{(n,k)} = y_s^{(n,k)} - \mu^{(n,k)} + \sum_{r>0} a_r^{(n,k)} (y_{s-r}^{(n,k)} - \mu^{(n,k)}),$$

where $S(n,k)$, $\left\{a_r^{(n,k)}\right\}_{r=1}^{R}$, $\mu^{(n,k)}$ denote the set of all observation nodes, the set of prediction coefficients and class means in level n and orientation k respectively. $O(n)$ and $S(n)$ denote the number of orientations, and label nodes in level n respectively, where we define the coarsest scaling sub-band to be the fourth orientation in level L. Using the Bayes formula the conditional PDF of the class labels given the multi-textured image can be obtained as follows:

$$p_{X|Y}(x \mid y,\theta) = \frac{f_{Y|X}(y \mid x,\theta)\, p_X(x)}{f_Y(y \mid \theta)},$$

therefore,

$$f_{X|Y}\left(x \mid y\right) \propto \prod_{n=1}^{L}\left[\prod_{k=1}^{O(n)}\prod_{s\in S(n,k)}\frac{1}{\sqrt{2\pi\left(\sigma_{x_s}^{(n,k)}\right)^2}}\exp\left\{-\frac{\tilde{y}_s^2}{2\left(\sigma_{x_s}^{(n,k)}\right)^2}\right\}\right]\cdot \tag{3.2.2}$$

$$\sum_{s\in S(n)}\frac{1}{z^{(n)}}\exp\left\{-\sum_{(s,r)\in C_1}\beta_1^{(n)}\left(1-\delta(x_s-x_r)\right)-\sum_{(s,r)\in C_2}\beta_2^{(n)}\left(1-\delta(x_s-x_r)\right)\right\},$$

where $z^{(n)}$ is the partition function of the MRF of level n.

3.3 The Segmentation Scheme

There are two methods considered in literature to perform texture segmentation using such a Bayesian formulation. The first is the MAP estimation of the posterior distribution given by (3.2.2), which is known to be a conservative estimator [21], and leads to many misclassified pixels. The second method used in literature is known as the maximum of posterior marginals (MPM) which minimizes the expected number of miss-classified nodes in the lattice.

The expected number of miss-classified label nodes can be obtained using [22]:

$$E\left[\sum_{n=1}^{L}\sum_{s\in S(n)}\left(1-\delta(x_s-c_s)\right)\mid Y\right] = \sum_{n=1}^{L}\sum_{s\in S(n)}\left(1-p\left(x_s=c_s \mid Y\right)\right)$$

where c_s is the correct classification for node s.

Therefore in order to minimize the expected number of miss-classified label nodes in the lattice, we need to maximize the posterior probability of each label node in the lattice, therefore the classification rule takes the form:

$$X_s = \underset{x_s}{\operatorname{argmax}}\; p\left(x_s \mid Y\right).$$

In order to approximate the posterior marginals it was suggested in [21] to use the fraction of time that the Markov chain with a fixed temperature spends in each label at each node of the label lattice:

$$p_{X_s|Y}(x_s = c \mid y) = \frac{1}{\left(T-\tau\right)}\sum_{t=\tau}^{T}\delta(x_s(t)-c),$$

where T here is the number of iterations used for the Gibbs sampler, and τ is the number of iterations required for the system to be in thermal equilibrium.

3.4 Parameter Estimation

The parameter estimation for the causal GMRF parameters is performed using a modified EM algorithm, where the segmentation and parameter estimation stages are repeated alternately. In each iteration the posterior marginals are computed using the current parameter estimates and the Gibbs sampler [23]. Subsequently the posterior marginals are used to update the parameters.

By minimizing the expectation maximization (EM) Q-function with respect to the parameters $\sigma_c^{(n)}$, $\mu_c^{(n,k)}$, $\left\{a_{c,r}^{(n)}\right\}_{r=1}^{R}$, for each *n=1…L*, *k=1…O(n)*, *c=1…M*, *r=1…R*, the following equations are obtained:

(3.4.1) $$\sum_{s\in S(n,k)} [y_s - \hat{\mu}_c^{(n,k)} + \sum_{q>0} \hat{a}_{c,q}^{(n,k)} (y_{s-q} - \hat{\mu}_c^{(n,k)})] p(x_s = c \mid Y, \theta_{old}) = 0,$$

(3.4.2) $$\sum_{s\in S(n,k)} [y_s - \hat{\mu}_c^{(n,k)} + \sum_{q>0} \hat{a}_{c,q}^{(n,k)} (y_{s-q} - \hat{\mu}_c^{(n,k)})](y_{s-r} - \hat{\mu}_c^{(n,k)}) p(x_s = c \mid Y, \theta_{old}) = 0,$$

(3.4.3) $$\sum_{s\in S(n,k)} [y_s - \hat{\mu}_c^{(n,k)} + \sum_{q>0} \hat{a}_{c,q}^{(n,k)} (y_{s-q} - \hat{\mu}_c^{(n,k)})]^2 p(x_s = c \mid Y, \theta_{old}) - (\sigma_c^{(n,k)})^2 N_c^{(n,k)} = 0.$$

where θ and θ_{old} are the new and old parameter estimates respectively, and where $p\left(x_s = c \mid Y, \theta_{old}\right)$ are the posterior marginals estimated using the Gibbs sampler.

Since equations (3.4.1)-(3.4.3) are not linear we estimate the class means using:

(3.4.4) $$\hat{\mu}_c^{(n,k)} = \frac{1}{N_c^{(n,k)}} \sum_{s\in S(n,k)} y_s p(x_s = c \mid Y = y, \theta_{old}),$$

$$\text{where } N_c^{(n,k)} = \sum_{s\in S(n,k)} p(x_s = c \mid Y = y, \theta_{old}).$$

subsequently the class prediction coefficients are computed using (3.4.2), and the class variance estimates are computed using (3.4.3).

3.5 Parameter Initialization

In order to initialize the parameters we perform first a coarse segmentation using k-means clustering. Subsequently we use 0/1 instead of $p(x_s = c \mid Y, \theta_{old})$ in equations (3.4.4)-(3.4.6) according to the coarse segmentation, and obtain initial estimates for the parameters. The feature vectors for the k-means clustering were obtained by concatenating the logarithmically scaled histograms of the local second moment estimates of the sub-bands of the wavelet transform of 16x16 blocks (unless specified otherwise), as was shown in section 2. The appropriate block for each pixel was the 16x16 block for which the pixel is at the center.

3.6 Comparison to Comer and Delp [21]

Our texture segmentation algorithm differs from [14] in three major details:

1) The observation field in [21] was the Gaussian pyramid representation.
3) Since the filter coefficients used to obtain the Gaussian pyramid representation sum to one, the class means in [21] did not depend on the resolution level.
4) The different levels in the Gaussian pyramid in [21] were not assumed to be statistically independent. The pyramid representation was divided into quad-trees, where each quad-tree was scanned from the coarsest resolution to the finest by a predefined raster scan order.

3.7 Experimental Results

We compared the algorithm presented here when using the feature space of the wavelet transform and the feature space obtained using the pre-processing scheme presented in chapter 4.

3.7.1 Implementation Details

We used three levels both for the wavelet transform and for the Gaussian pyramid. It is important to use linear phase filters, otherwise the texture boundaries in each scale may be translated compared to their original position. The values of β_1 and β_2 were identical for each of the levels, where we used $\beta_1 = 1.6$ and $\beta_2 = 0.8$. We used three prediction coefficients to the GMRF model. In each iteration the of the parameter estimation stage (but for the last iteration) the label field was randomly initialized, as was reported in [24]. The segmentation result is that obtained for the finest resolution level of the label pyramid.

3.7.2 Segmentation Results

Figures 3.2, 3.6 show that generally when the different texture regions are separate then there is no great difference when using the wavelet features or the post-processed features. Figures 3.3-3.5 show that when the texture classes are very similar and are not separate then there is a considerable advantage to using the post-processed features compared to using the wavelet features. This is because the texture classes are very similar, and because the smoothness prior can not mask the inconsistency of the assumed statistical model as efficiently as before since the texture classes are not separate. Therefore the improved characterization of the textures obtained by the Gaussian marginal distributions of the post-processed features is very important in these cases. On the other hand it is evident that the boundaries obtained when using the post-processed features, are less accurate due to the reduced spatial resolution incurred by the pre-processing stage.

(a) (b) (c)

Figure 3.2: Segmentation results: (a) The original image (size 128x128), (b) Segmentation result using the wavelet transform, (c) Segmentation results using the pre-processing scheme.

(a) (b) (c)

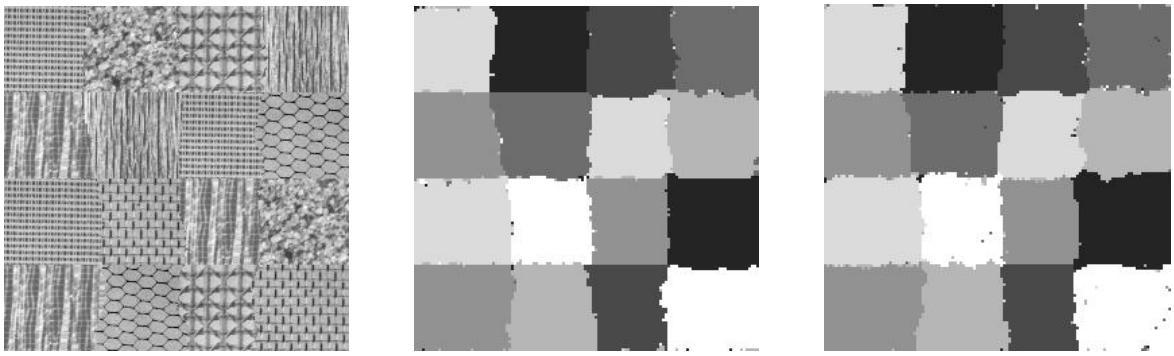
Figure 3.3: Segmentation results: (a) The original image (size 128x128), (b) Segmentation result using the wavelet transform, (c) Segmentation results using the pre-processing scheme.

(a) (b) (c)

Figure 3.4: Segmentation results: (a) The original image (size 128x128), (b) Segmentation result using the wavelet transform, (c) Segmentation results using the pre-processing scheme.

(a) (b) (c)

Figure 3.5: Segmentation results: (a) The original image (size 128x128), (b) Segmentation result using the wavelet transform, (c) Segmentation results using the pre-processing scheme.



(a) (b) (c)

Figure 3.6: Segmentation results (initialization performed using 32x32 blocks): (a) The original image (size 128x128), (b) Segmentation result using the wavelet transform, (c) Segmentation results using the pre-processing scheme.

4 Multiscale Level Set Supervised Texture Segmentation using HMTs

Texture segmentation using a level set approach and statistical modeling of the multi-resolution coefficients has already been considered in [25], [26], [27], [28], however non of these schemes took advantage of the pyramid structure which enables a coarse to fine strategy. In [25] the data term used in the functional, employed an un-decimated wavelet packet transform, which was treated as a vector valued image, where every vector was consisted of all the coefficients at the same spatial location at each sub-band. The coefficients were modeled using a GG statistical model, therefore the data term used for each pixel was the sum of the log likelihoods of all the elements in the vector. This has significant drawbacks since textures which may easily be discriminated at a coarse resolution may be harder to discriminate at a fine resolution.

Image segmentation using HMTs was considered in [29]. The scheme uses the wavelet transform and adopts a Bayesian coarse to fine strategy, where each dyadic square is classified using the likelihoods of the appropriate HMTs and a likelihood term which is based on the previous classification of the dyadic squares performed at coarser scales. The scheme employs a context model in order to model the effect of the coarser scales on the segmentation at finer scales and thus it can capture the physical property of spatial continuity of the textures. On the other hand the approach presented in [29], can not minimize the length of the contours separating the textures which may result in a large misclassification rate.

The scheme presented in this section performs a sequence of level set segmentations, moving from coarse to fine scales, where the data term for each iteration of the level set segmentation scheme is related to the likelihood of the HMT of each dyadic square and to a likelihood derived from the previously classified dyadic squares in coarser scales using a context model. Thus a multi-scale level set segmentation is obtained, benefiting from the advantages of a coarse to fine strategy, and minimizing the contours' length simultaneously.

First we present the statistical characterization used in this paper, the Bayesian segmentation scheme, and the level set segmentation method which is the basis to the proposed multiscale scheme. Subsequently the new level set texture segmentation scheme is described. Finally experimental results are presented.

4.1 Background

4.1.1 The Dual Tree Complex Wavelet Transform (DT-CWT)

Shift variance is considered to be a major weakness of texture characterization as the characterization of the texture is not expected to change under translation. In order to deal with the shift variance of the wavelet transform while avoiding the considerable redundancy induced by the un-decimated wavelet transform, Kingsbury introduced in [30] the DT-CWT. Kingsbury suggested the 1D decomposition shown in figure 4.1, where the upper and lower trees can be considered as the real and imaginary parts respectively of a complex decomposition. Kingsbury defined a transform to be shift invariant if the

reconstruction obtained for each complex coefficient assuming all other coefficients are set to zero is free of aliasing. In such a case the system producing each complex coefficient can be regarded as a linear time invariant system, and therefore shift invariant by definition. Kingsbury found the conditions that the low-pass and high-pass filters in figure 4.1 (*H* and *G* respectively) must satisfy in order to obtain approximate shift invariance, and presented the appropriate filters. Furthermore, it can be shown that the filters satisfying these conditions must discriminate between positive and negative frequencies, therefore the DT-CWT has 6 orientations. An efficient implementation of a 2D DT-CWT was presented in [31].

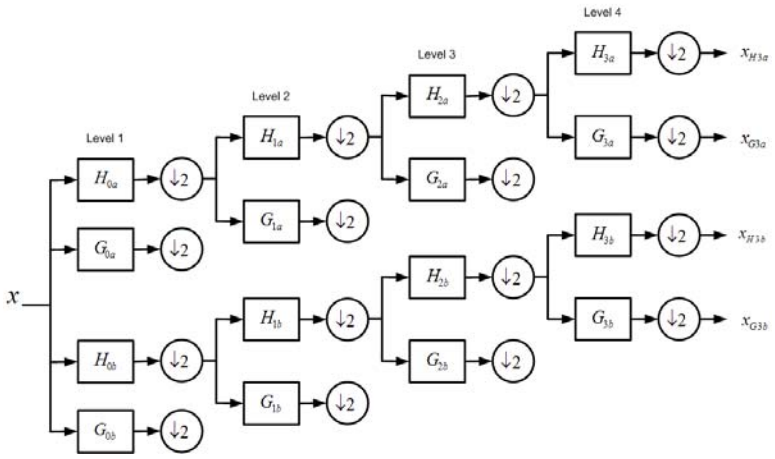


Figure 4.1: The dual tree complex wavelet transform

4.1.2 Hidden Markov Trees (HMT)

HMT is a statistical model which can account both for the non-Gaussian marginal statistics of the wavelet coefficients and for some statistical dependencies between the wavelet coefficients. Figure 44.2 shows the graphical structure of a HMT, where the distribution of each wavelet coefficient is assumed to be a Gaussian which is statistically independent of all other hidden states given its hidden state. The hidden states in the tree follow a tree structured Markov process of a first order, where the transition probabilities are parameterized by the HMT model. The HMT parameters are estimated using an iterative expectation maximization (EM) algorithm. Furthermore, the likelihood of every quadtree is obtained as a by-product of the EM algorithm. Let $f\left(T_i^{j,k} \mid \theta^k\right)$ denote the likelihood of dyadic square *i* in scale *j* and orientation *k*, where θ denotes the set of parameters, then the likelihood of dyadic square *i* in scale *j* can be obtained as the product of the likelihood at every orientation: (4.1.1)

$$f\left(D_i^j \mid \theta\right) = \prod_{k=1}^{O} f\left(T_i^{j,k} \mid \theta^k\right),$$

where *O* denotes the number of orientations.

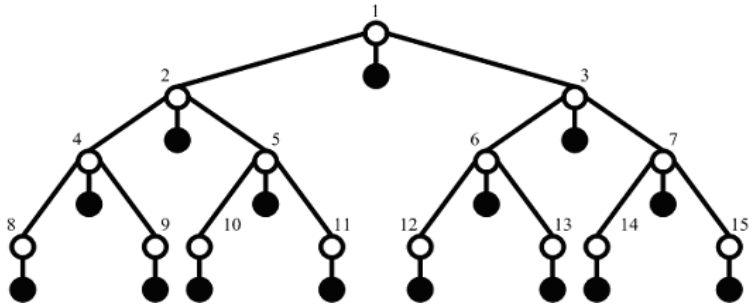


Figure 4.2: The graphical structure of a HMT. The white nodes represent the state variable, and the black nodes represent the wavelet coefficients

4.1.3 Bayesian Segmentation using HMTs

Using the likelihood of the dyadic squares (4.1.1) maximum likelihood (ML) classification of every dyadic square can be obtained, however as the classification of interest is that in the finest resolution, direct ML classification will produce a poor result as it ignores all the data in the coarser resolutions. A solution to this problem was presented in [29] using two assumptions:

1) The classification of each wavelet coefficient depends on the classification obtained at coarser scales alone
2) The classification of each wavelet coefficient depends on the classification obtained at coarser scales through a context variable v_i alone.

Let $c \in 1 \ldots L$ denote the texture class, where *L* denotes the number of texture classes, then under these conditions it was shown in [29] that maximum a posteriori (MAP) classification can be obtained using:

$$\ell_i^j = \underset{j}{\operatorname{argmax}}\, f\left(D_i^j \mid c_i = \ell\right) p\left(c_i = \ell \mid v_i\right), \tag{4.1.2}$$

where $f\left(D_i^j \mid c_i = \ell\right)$ is obtained using (4.1.1), and $p\left(c_i = \ell \mid v_i\right)$ is a parameter which has to be estimated.

The scheme that determines the context variable v_i can be chosen arbitrarily, however in order to obtain good performance it is important to keep its complexity low so that the parameters can be estimated efficiently. According to the context scheme suggested in [29], v_i was a two element vector consisted of the classification label of the parent coefficient, and the majority vote in the neighborhood of the parent. Since the number of labels is finite, the context variable v_i can take a finite number of discrete values and the parameters can be estimated efficiently.

4.1.4 Estimating the Context Model's Parameters

In order to estimate the context model's parameters, an EM algorithm was presented in [29]. The EM algorithm for $p\left(c_i = \ell \mid v_i\right)$ was specified indirectly through $e_{j,m} = p_{c_j}\left(m\right)$ which is the likelihood that the classification of a dyadic square at scale *j* is of class *m*, and $\alpha_{v,m}^j = p\left(v^j \mid c = m\right)$ which is the

likelihood of the context at scale *j* given that the classification of the dyadic square is of class *m*. Initially the parameters $e_{j,m}$, $\alpha_{v,m}^{j}$ must be initialized. Subsequently the EM algorithm then iterates through the expectation and maximization stages:

Expectation: Given the previous parameters' estimates calculate,

$$p\left(c_i = m \mid D_i^j, v_i^j\right) = \frac{e_{j,m}\alpha_{v_i^j,m}^{j} f\left(D_i^j \mid c_i = m\right)}{\sum_{\ell=1}^{L} e_{j,\ell}\alpha_{v_i^j,\ell}^{j} f\left(D_i^j \mid c_i = \ell\right)}. \tag{4.1.3}$$

Maximization: Update the parameters using:

$$e_{j,m} = \frac{1}{2^{DS(j)}} \sum_i p\left(c_i = m \mid D_i^j, v_i^j\right), \tag{4.1.4}$$

$$\alpha_{v,m}^{j} = \frac{1}{DS(j)e_{j,m}} \sum_{\substack{\text{all } i \\ v_i^j = v}} p\left(c_i = m \mid D_i^j, v_i^j\right). \tag{4.1.5}$$

where *DS(j)* denotes the number of dyadic squares in scale *j*.

4.1.5 The Bayesian scheme

The Bayesian texture segmentation scheme using HMTs proceeds from coarse to fine scale. At the coarsest scale considered the classification of each dyadic square is obtained using ML estimation without the context scheme. This is reasonable since at a coarse scale the HMT likelihood may be considered reliable. When moving to finer scales, the context variables are obtained using the context scheme and the appropriate parameters are estimated using (4.1.3)-(4.1.5). Subsequently the classification of each dyadic square is obtained using (4.1.2). The process is repeated until the finest scale is reached, and the segmentation result is obtained for the finest scale.

4.1.6 Level Set Texture Segmentation

This section describes a level set segmentation scheme which was used in [25]. Additionally we limit the case studied for a two class segmentation for simplicity. Let Ω denote a subset of $\mathbb{R}^2$ which is the domain of the image u_0 which satisfies $u_0 : \Omega \to \mathbb{R}$, and let *C* denote the evolving boundary between the textures. The level set segmentation minimizes a functional which is consisted of two terms, the first term is the data term which penalizes the misclassification of texture regions, and the second term penalizes the length of the curve *C*. Let $\ell_k(x,y), k = 1,2$ be a measure for the misclassification at $(x,y) \in \Omega$ such that $\ell_k(x,y)$ is smaller when the correct classification is texture class *k* and vice versa. The functional then takes the form:

$$F(C) = \mu \cdot Length(C) + \lambda_1 \iint_{inside(C)} \ell_1(x,y)\,dxdy + \lambda_2 \iint_{outside(C)} \ell_2(x,y)\,dxdy \tag{4.1.6}$$

where μ, λ_1, λ_2 are constants which determine the rigidity of the contour relative to the importance given to the data term. In order to find the contour *C* which minimizes (4.1.6), a level set function is defined such that:

$$\begin{cases} C = \{(x,y) \in \Omega : \phi(x,y) = 0\} \\ inside(C) = \{(x,y) \in \Omega : \phi(x,y) > 0\} \\ outside(C) = \{(x,y) \in \Omega : \phi(x,y) < 0\} \end{cases} \tag{4.1.7}$$

Using the level set function and the regularized step and impulse functions $\$H_\alpha(z)$, $\delta_\alpha(z)$ respectively, it was shown in [26] that equation (4.1.6) can be written as:

$$\begin{aligned} F(C) = {} & \lambda_1 \iint_\Omega \ell_1(x,y) H_\alpha(\phi(x,y))\,dxdy + \lambda_2 \iint_\Omega \ell_2(x,y)\left(1 - H_\alpha(\phi(x,y))\right)dxdy \\ & + \mu \iint_\Omega \delta_\alpha(x,y)\left|\nabla\phi(x,y)\right|dxdy \end{aligned} \tag{4.1.8}$$

where,

$$H_\alpha(z) = \begin{cases} \frac{1}{2}\left(1 + \frac{z}{\alpha} + \frac{1}{\pi}\sin\frac{\pi z}{\alpha}\right) & \text{if } |z| \le \alpha \\ 1 & \text{if } z \ge \alpha \\ 0 & \text{if } z < -\alpha \end{cases} \tag{4.1.9}$$

$$\delta_\alpha(z) = \begin{cases} \frac{1}{2\alpha}\left(1 + \cos\frac{\pi z}{\alpha}\right) & \text{if } |z| \le \alpha \\ 0 & \text{if } |z| > \alpha \end{cases} \tag{4.1.10}$$

In [32] is was shown that the minimization of (4.1.8) can be obtained by solving the following evolution equation $\phi(t,x,y)$:

$$\frac{\partial\phi}{\partial t} = -\delta_\alpha(\phi)\left[-\mu\,\mathrm{div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right) + \lambda_1\ell_1 - \lambda_2\ell_2\right], \tag{4.1.11}$$

where $\phi(0,x,y) = \phi_0(x,y)$ is the initialization of the level set function $\phi(x,y)$. Equation (4.1.11) is solved numerically where ϕ is discretized using a finite differences scheme [33]. Furthermore the level set function has to be reinitialized periodically [32].

4.2 The Proposed Multiscale Supervised Level Set Texture Segmentation Scheme

In the previous section a level set image segmentation scheme was described, however, there were no details given about the missclassification measure used in the data term of the functional (4.1.6). One approach which has been very popular is to use the minus log likelihood of a certain probability distribution function (PDF) as the penalty for misclassifying an image element. For example in [25] the textures were characterized using an un-decimated wavelet transform and a GG statistical model. The wavelet coefficients were assumed to be statistically independent and each element in the lattice was vector valued, containing all the wavelet coefficients at the same spatial location at all resolutions. The major drawback is that all the coefficients at the same spatial location at every resolution count the

same, whereas there may be a considerable advantages to using a coarse to fine strategy, where large elements are classified using the coarser scales and the small elements are classified using the finer scales.

Motivated by the Bayesian approach to texture segmentation using HMTs, we propose a scheme that performs a sequence of level set segmentations from coarse to fine scales, where the data term at each scale depends both on the segmentation obtained at the coarser scale and on the corresponding dyadic square. At the coarsest scale considered, the data term for the level set segmentation scheme presented in the previous section is obtained as the minus log likelihood of the corresponding HMTs, which can be obtained using (4.1.1). At each finer scale the data term is obtained using:

$$\ell_c(x,y) = -\log\left(f\left(D^j(x,y)\,|\,c\right)\right)p\left(c\,|\,v(x,y)\right), \tag{4.2.1}$$

where $D^j(x,y)$ is the dyadic square at position (x,y) at scale j, and $v(x,y)$ is the context vector which is obtained form the segmentation result obtained at the coarser scale.

Aside from the advantage of employing a coarse to fine strategy, the new scheme also benefits from the use of the HMT statistical model, which can account for some statistical dependencies between the wavelet coefficients, in contrast to simpler models such as the GG. It has been established in [34], [35] that employing the HMT model for modeling the DT-CWT provides superior results to modeling the traditional wavelet transform using the HMT model, thus the DT-CWT is suitably employed in this work. Since each coefficient of the DT-CWT is complex, the HMT models the absolute value of the DT-CWT coefficients.

The segmentation result is that obtained using the finest resolution data. Thus since the finest resolution corresponds to blocks of four pixels each, pixel-wise segmentation cannot be obtained using the new scheme.

4.2.1 The Context Scheme

The context variables in the new scheme are the segmentation results obtained for the parent of each wavelet coefficient. Any dependencies arising from the neighbors of the parent are ignored, as the contour can model the spatial continuity of the different texture regions well enough. The parameters for the context model are estimated recursively using the EM algorithm described by equations (4.1.3-4.1.5), where the parameters are initialized at each scale to those obtained at the previous parameter estimation stage at the coarser scale. At the finest scale where the textures' class probabilities are not reliable, the parameters used are identical to those obtained at the coarser scale.

4.3 Experimental Results

We used a three level transform to decompose all the textured images in the experiments of the new scheme presented here. Since the number of observation nodes in each scale is different, the parameters

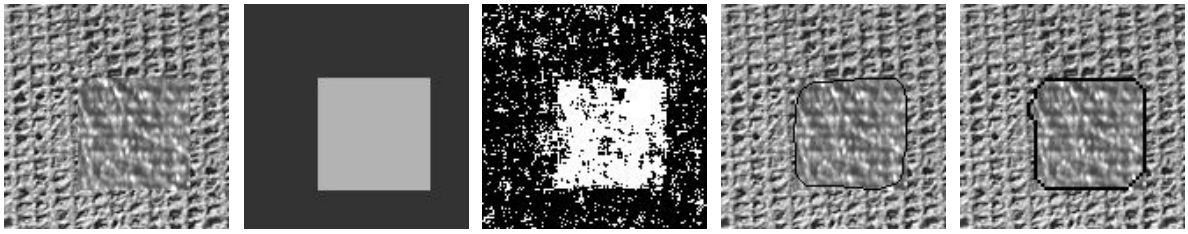
required in the level set functional μ, λ_1, λ_1 may have to be different as well. In the experiments presented in this section we have used the values $\lambda_1 = \lambda_2 = 0.1$ for all the scales, and

$$\mu_k = \text{constant} \cdot \left(\text{Number of Scales } - k + 1 \right),$$

where we used $\text{constant} = 0.00001 \cdot 128^2$ (unless specified otherwise), where $k = 1 \ldots \text{Number of Scales}$, and where 1 is the finest scale.

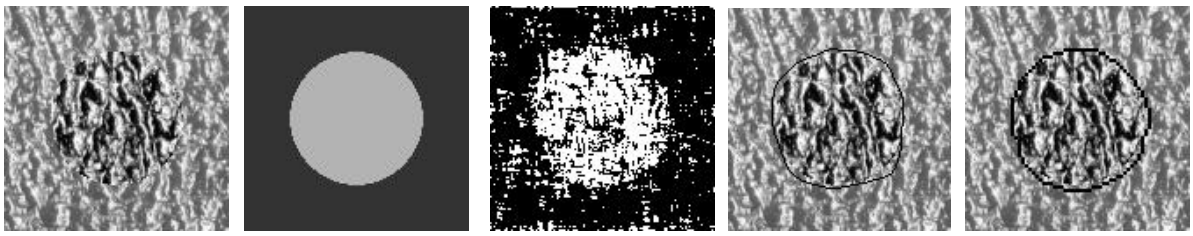
We compare the results of the new scheme to those obtained using a GG model in the data term of the level set segmentation scheme, as was presented in [25]. Although in [25] the textured image was decomposed with an un-decimated wavelet packet transform which decomposes also the high frequencies of each sub-band, we use an un-decimated wavelet transform since our experience has shown that the using the wavelet packet transform does not lead to improved results. It is important to use filters which are linear phase or nearly linear phase, otherwise there may be phase distortions which might cause the texture boundaries to be shifted compared to their original position [36]. In the experiments presented here the symlets 16 tap filters [18] were used for the un-decimated wavelet transform since they are nearly linear phase filters. We used a three level un-decimated transform to decompose the textured images and the constants which we used for the functional were $\lambda_1 = \lambda_2 = 0.5$, and $\mu = 16.384$ (unless specified otherwise).

Since we consider supervised texture segmentation, the parameters for each texture class were obtained by training the models (either HMT or GG) using a training set which is separate from the textured images. The values of all the other parameters were set experimentally to provide the best results. Figures 4.3-4.5 show the segmentation results obtained using the new scheme and using a GG based data term, for various texture classes and shapes. The results suggest that the two schemes are comparable when considering the error rate of the misclassified pixels, however the boundaries in the new scheme seem to be more similar to the boundaries in the original image. This is especially evident in figure 4.5 where the shape of the inner texture is not convex, thus a large value for the parameter μ which sets the rigidity of the contour will hamper the edges and include erroneous pixels, and small value of μ will produce a jagged segmentation due to the overemphasis of the data term in the functional. On the other hand the new scheme eliminates many pixels from consideration at coarser scales, thus it does not require the contour to be very rigid, and it is much simpler to perform segmentation in the finer scales. However since the new scheme can perform segmentation up to a resolution of 2x2 blocks, the scheme using the un-decimated wavelet transform, can apriori obtain a better correct classification rate of the pixels when the data term is very similar to the ground truth segmentation mask. The main conclusion is that the performance of the two schemes are comparable, where the new scheme may provide better results when the inner shape is not convex.



(a) (b) (c) (d) (e)

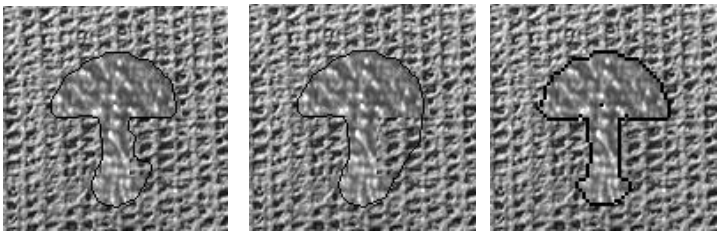
Figure 4.3: Segmentation of synthetic image composed of two textures: (a) the original image (size 128x128), (b) the ground truth, (c) the data term obtained using the GG, (d) the segmentation obtained using the GG data term (error=3.11%), (e) the segmentation obtained using the new scheme (error=3.44%).



(a) (b) (c) (d) (e)

Figure 4.4: Segmentation of synthetic image composed of two textures: (a) the original image (size 128x128), (b) the ground truth, (c) the data term obtained using the GG, (d) the segmentation obtained using the GG data term (error=2.51%), (e) the segmentation obtained using the new scheme (error=1.92%).

(a) (b) (c) (d)



(e) (f) (g)

Figure 4.5: Segmentation of synthetic image composed of two textures: (a) the original image (size 128x128), (b) the ground truth, (c) the data term obtained using the GG (d)-(f) the segmentation obtained using the GG data term for various parameter values ($\mu = 3,\ \lambda_1 = \lambda_2 = 0.5$, error=4), ($\mu = 4,\ \lambda_1 = \lambda_2 = 0.5$, error=3.17), ($\mu = 5,\ \lambda_1 = \lambda_2 = 0.5$, error=6) respectively, (g) the segmentation obtained using the new scheme ($\text{constant} = 0.00004 \cdot 128^2,\ \lambda_1 = \lambda_2 = 0.5$, error=2.91).

5 Summary

New feature statistics for texture classification based on GSM statistical modeling have been introduced. Using the GSM distribution and the Jeffrey's non-informative prior, it has been shown that the histograms of the new features should be scaled logarithmically. Thus the major drawback of the strong dependence on the scaling of the histogram bins is avoided, as the histogram is scaled according to the prior distribution of the features. Additionally, we have presented a new feature weighting scheme which is appropriate to histogram features and accounts for the non-Euclidean dissimilarity measure used in this work. The overall scheme has been shown to obtain better classification results than those reported recently in the literature using the same textures and the same testing procedures. Furthermore the scheme is computationally efficient, avoiding a feature selection stage.

The new features are a special case of a pre-processing scheme for the wavelet coefficients. The feature space was applied to unsupervised texture segmentation using GMRF statistical modeling and MRF Ising model to model the spatial continuity of the textures. It has been shown that such a modeling may be superior to modeling the feature space of the wavelet coefficients using a GMRF, since the marginal distributions of the features obtained by the pre-processing scheme are closely Gaussian in contrast to the marginal distributions of the wavelet coefficients.

A new level set supervised texture segmentation scheme was presented, where in contrast to presently available level-set segmentation schemes, the new scheme employs a coarse to fine strategy. The new scheme employs hidden Markov trees statistical modeling and a context model in order to fuse the segmentation results from different scales. The segmentation in each scale is performed using a level-set scheme in each scale, where the data term fuses the information from coarser scales together with the appropriate HMT likelihood. The results indicate that the new scheme is mostly useful when segmenting non-convex shapes since the coarse to fine strategy facilitates the dependence on the rigidity of the contour. Our conclusion is that the proposed approach could be superior to presently available methods for classification and segmentation of textures.

Bibliography

[1] M. Tuceryan, and A. K. Jain, "Texture analysis", *The handbook of Pattern Recognition and Computer Vision,* 1998.

[2] M. Porat, Y. Zeevi, "The generalized Gabor scheme of image representation in biological and machine vision" *IEEE Trans. on PAMI*, Vol. 10, No. 4, pp. 452-468, July 1988.

[3] M. Porat, and Y. Y. Zeevi, "Localized texture processing in vision: Analysis and synthesis in the Gaborian Space", *IEEE Trans. on Biomedical Engineering*, Vol. 36, No. 1, pp. 115-129, 1989.

[4] S. G. Mallat, "A Theory for multiresolution signal decomposition: The wavelet representation", *IEEE Trans. on PAMI*, Vol. 11, No. 7, pp. 674- 693, 1989.

[5] M. S. Crouse, R. D. Nowak, and R. G. Baraniuk, "Wavelet-based statistical signal processing using hidden Markov models", *IEEE Trans. on Signal Proc.*, Vol. 46, No. 4, pp. 886-902, 1998.

[6] J. Puzicha, J. M. Buhman, Y. Rubner, and C. Tomasi, "Empirical evaluation of dissimilarity measures for color and texture", *ICCV* , pp. 1165-1172, 1999.

[7] E. Hadjidemetriou, M.D. Grossberg, S.K. Nayar, "Multiresolution histograms and their use for recognition", *IEEE Trans. on PAMI*, Vol . 26, No. 7, pp. 831-847, 2004.

[8] A. Jain, and D. Zongker, "Feature selection: Evaluation, application and small sample performance", *IEEE Trans. on PAMI*, Vol. 19, No. 2, pp. 153-158, 1997.

[9] W. K. Leow, and R. Li, "Adaptive binning and dissimilarity measure for image retrieval and classification", *CVPR* 2001.

[10] C.P. Robert, "The Bayesian choice", *Springer-Verlag*,1994.

[11] J. Portilla, V. Strela, M. J Wainwright, and E.P. Simoncelli, "Image denoising using scale mixtures of Gaussians in the wavelet domain", *IEEE Trans. on Image Proc.*, Vol. 12, No. 11, pp. 1338-1351, 2003.

[12] T. Randen, and J. H. Husoy, "Filtering for texture classification: A comparative study", *IEEE Trans. on PAMI*, Vol. 21, No. 4, pp. 291-310, 1999.

[13] M. Unser, and M. Eden, "Nonlinear operators for improving texture segmentation based on features extracted by spatial filtering", *IEEE Trans. on SMC*, Vol. 20, No. 4, pp. 804-815, 1990.

[14] W. Gaohong, and Z. Yujin, "Wavelet transform-based texture classification with feature weighting", *ICIP* , pp. 435-439, 1999.

[15] R. Mittelman, M. Porat, "A new approach to feature extraction for wavelet-based texture classification", *ICIP* 2005.

[16] X. Liu, and D. Wang, "Texture classification using spectral histograms", *IEEE Trans. on Image Proc.*, Vol. 12, No. 6, pp. 661-670, 2003.

[17] http://www-dbv.cs.uni-bonn.de/image/texture.tar.gz

[18] I. Daubechies, "Ten lectures on wavelets", *Society for industrial and applied mathematics*, 1992.

[19] J. Portilla, V. Strela, M. J. Wainwright and E. P. Simoncelli, "Adaptive Wiener denoising using a Gaussian scale mixture model in the wavelet domain", *ICIP*, pp. 37-40, 2001.

[20] J. Portilla, V. Strela, M. J. Wainwright and E. P. Simoncelli, "Adaptive Wiener denoising using a Gaussian scale mixture model in the wavelet domain", *ICIP*, pp. 37-40, 2001.

[21] M. J. Comer and E. J. Delp, "Segmentation of textured images using a multiresolution Gaussian autoregressive model", *IEEE Trans. on Image Proc.*, Vol. 8, No. 3, pp. 408-420, 1999.

[22] R. Mittelman, M. Porat, "A new approach to multiresolution texture segmentation using Gaussian Markov random fields", *Eusipco* 2005.

[23] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images", *IEEE Trans. on PAMI*, Vol. 6, pp. 721-741, 1984.

[24] D. E. Melas and S. P. Wilson, "Double Markov random fields and Bayesian image segmentation", *IEEE Trans. on Signal Proc.*, Vol. 50, No. 2, pp. 357-365, 2002.

[25] J. F. Aujol, G. Aubert, and L. B. Feraud, "Wavelet-based level set evolution for classification of texture images", *IEEE Trans. on Image Proc.*, Vol. 12, No. 12, pp.1634-1641, 2003.

[26] T. F. Chan, L. A. Vese, "Active contours without edges", *IEEE Trans. on Image Proc.*, Vol. 10, No. 2, pp. 266-277, 2001.

[27] M. Heiler, C. Schnorr, "Natural image statistics for image segmentation", *Int. Journal of Computer Vision*, Vol. 63, No. 1, pp. 5-19, 2005.

[28] C. Samson, L. B. Feraud, G. Aubert, and J. Zerubia, " A level set model for image classification", *Int. Journal of Computer Vision*, Vol. 40, Issue 3, 2000.

[29] H. Choi, and R. G. Baraniuk, "Multiscale image segmentation using wavelet-domain hidden Markov models", *IEEE Trans. on Image Proc.*, Vol. 10, No. 9, pp. 1309-1321, 2001.

[30] N. G. Kingsbury, "Complex wavelets for shift invariant analysis and filtering of signals", *Journal of applied computational harmonic analysis*, Vol. 10, No. 3, pp. 234-253, 2001.

[31] N. G. Kingsbury, "Image processing with complex wavelets", *Phil. Trans. Royal Society, London A*, September 1999.

[32] H. K. Zhao, T. Chan, B. Merriman, and S. Osher, “A variational level set approach to mutiphase motion”, *Journal of computational physics* 127, pp. 179-195, 1996.

[33] G. Aubert, and L. Vese, “A variational method in image recovery”, *SIAM Journal of Numerical Analysis*, Vol. 34, No. 5, pp. 1948-1978, 1997.

[34] J. H. Won, K. Pyun, and R. M. Gray, “Hiddn Markov multiresolution texture segmentation using complex wavelets”, *ICT*, 2003.

[35] J. Romberg, H. Choi, R. G. Baraniuk, and N. Kingsbury, “Multiscale classification using complex wavelets and hidden Markov models”, *ICIP* pp. 371-374, 2000.

[36] M. Unser, “Texture classification and segmentation using wavelet frames”, *IEEE Trans. on Image Proc.*, Vol. 4, No. 11, pp. 1549-1560, 1995.