Non-stationary Texture Segmentation
Using an AM-FM Model

M. S. Pattichis [†], C. I. Christodoulou [‡], C. S. Pattichis [‡], and A. C. Bovik [†]
*[†]Laboratory for Vision Systems, University of Texas, Austin, TX 78712-1084
Tel: (512) 471-2887; fax: (512) 471-1225
e-mail: marios@vision.ece.utexas.edu

[‡]Department of Computer Science, University of Cyprus, Kallipoleos 75, Nicosia, Cyprus
Tel: (357)-2-338705; fax: (357) 2-339062
e-mail: pattichi@turing.cs.ucy.ac.cy

Abstract

We present a novel method for segmenting non-stationary textures. Our approach uses a multi-dimensional AM-FM representation for the texture, and provides the FM features to an SOFM-LVQ neural network system that performs the segmentation. For the segmentation, we use the eigenvalues of the instantaneous frequency gradient tensor, and show how these eigenvalues capture the non-stationary structure of a texture. For a woodgrain image, the segmentation results are shown to capture the essential non-stationary nature of the grain.

1 Introduction

Progress in tackling the problem of non-stationary texture segmentation has been hindered by the lack of effective models that capture the fundamental structure of such images ([7], page 622). To capture the non-stationary structure of texture, we use an AM-FM model, and characterize non-stationarity as a multi-dimensional frequency modulation process.

We begin with the AM-FM model. For an image g, we write [1]:

$$g(x_1, x_2) = \sum_n a_n(x_1, x_2) \exp\{j\phi_n(x_1, x_2)\} \quad (1)$$

where a_n denotes a slowly-varying amplitude function, and $\nabla\phi_n$ denotes a slowly-varying instantaneous frequency vector.

To capture the non-stationarity in an image, we develop measures for multi-dimensional frequency modulation. Let the symbol $\mathbf{O}$, $\mathbf{O} \equiv \nabla\phi$ denote the instantaneous frequency vector. Then, the instantaneous frequency gradient tensor is defined to be the matrix $\mathbf{F}$, whose entries are the spatial derivatives of the instantaneous frequency. Using $\mathbf{F}$, we express the relative change in the instantaneous frequency $d\mathbf{O}$, by: $d\mathbf{O} = \mathbf{F}\,d\mathbf{x}$. Using the fact that $\mathbf{F}$ is real-symmetric, we can get a bound for the increase in the magnitude of the instantaneous frequency vector:

$$|\lambda_1| \geq \frac{\|d\mathbf{O}\|}{\|d\mathbf{x}\|} \geq |\lambda_2|, \qquad |\lambda_1| > |\lambda_2| \quad (2)$$

From (2), we see that the maximum change in the magnitude of the instantaneous frequency vector occurs in the direction of the first eigenvector. In contrast, the minimal change in the magnitude of the instantaneous frequency vector occurs in the direction of the second eigenvector. Furthermore, it is easy to show that the divergence of the instantaneous frequency vector is simply the sum of the eigenvalues. Locally, a positive divergence implies that the instantaneous freqeuncy vector field is emanating from a point. In contrast, a negative divergence implies the instantaneous frequency vector field is converging towards a point. We will return to this interpretation in Section 4.

In Section 2, we will briefly describe the AM-FM demodulation algorithm that was used for computing the eigenvalues of the instantaneous frequency gradient tensor. Then, in Section 3, we describe the neural

network system that was used for non-supervised segmentation of the eigenvalues. In Section 4, we apply the algorithm to a non-stationary texture and analyze the results.

2 Feature extraction

In this section, we briefly outline how the eigenvalues are computed. We summarize the AM-FM demodulation algorithm in Figure 1.

The AM-FM demodulation algorithm estimates the instantaneous frequency over a lattice of gabor filters (see [2] for more details). The Teager-Kaiser operator is used to estimate image energy over each channel [6]. Using:

$$\Psi_c \{I\} (\mathbf{x}) \equiv \| \nabla I(\mathbf{x}) \| - I(\mathbf{x}) \nabla^2 I(\mathbf{x}) \qquad (3)$$

the instantaneous frequency vector estimates are given by:

$$\phi_{x_i} \approx \sqrt{\Psi_c \left\{ \frac{\partial I}{\partial x_i} \right\} / \Psi_c \{I\}} \qquad i = 1, 2 \qquad (4)$$

To eliminate sign ambiguities, we select the candidate instantaneous frequency for which the image gives the largest projection (see step 6).

For estimating the components of the instantaneous frequency gradient tensor, we convolve the instantaneous frequency vector estimates with derivatives of gaussians. For example, to estimate $\partial O_i / \partial x_j$, we convolve the ith component O_i with $\partial g / \partial x_j$. The eigenvalues are then computed using the 'eig (.)' function in Matlab.

3 Neural Network Pattern Recognition Technique

A single-layer neural network is used for the segmentation of the woodgrain image. The developed ANN architecture is composed from two input nodes for the two eigenvalues and from three (or five) output nodes for the three (or five) possible classes. The segmentation procedure is implemented in three phases: in the first phase unsupervised learning is applied based on a modified one dimensional self-organizing feature map [4], [5], and competitive learning, in the second phase, in order to improve segmentation performance, the learning vector quantization is applied in a (self) supervised learning manner, and in the third phase the actual segmentation takes place. It should be noted that in each learning phase the input is presented to the network only once, for only one learning epoch.

This makes the segmentation algorithm fast and suitable for real time applications.

First, we describe the first learning phase. The objective of this phase is to provide a first "approximate" quantization of the input space (Voronoi vectors) by adapting the weights vectors of the neurons in the feature map [3]. The algorithm for phase 1 is summarized in Figure 2. After all inputs are presented to the network, the first adaptation of the weights vector is completed and the system proceeds to the second learning phase.

Second, we describe the second learning phase. The task of this phase is to adapt the weights vectors slightly (move Voronoi vectors) in order to improve the segmentation quality. Here, we note that the LVQ is actually a supervised learning technique, i.e. it demands prior knowledge of correctly labelled (classified) inputs. Since such a knowledge is not available, it is assumed that the adaptation carried out during the first learning phase is correct enough and thus the segmented inputs coming in will be correctly classified. Weights adaptation and winner selection is again on-going. In this modified version of LVQ2 the implementation steps are given in Figure 3. After all inputs are presented to the network, the network is trained and the actual segmentation process starts.

Third, we describe the segmentation phase. In the segmentation phase, all the input vectors are classified to one of the output nodes. The implementation steps for the segmenation phase are given in Figure 4.

4 Results and Discussion

In Figure 5, we show segmentation results for a woodgrain image. We describe the results for three and five classes. The original woodgrain image is shown in Figure 5(a). Superimposed on the woodgrain image, we show the instantaneous frequency vectors. By convention, sign ambiguities in the instantaneous frequency (resulting from the fact that: $\cos \left(\Omega^T \mathbf{x} \right) = \cos \left(-\Omega^T \mathbf{x} \right)$), are resolved by choosing the instantaneous frequency vector that points towards an instantaneous frequency of less magnitude).

As shown in Figure 5(b), the segmentation results of Figures 5(e) and 5(f) are mostly influenced by the sign of the divergence of the instantaneous frequency vector field. Next, we examine each of the segmentation classes. For Figure 5(e), we have three classes. The three classes correspond to:

- class 1: λ_1, λ_2 are of comparable magnitude. In Figure 5(e), this class is characterized by the dark regions.

- class 2: $\lambda_1 < 0$, $|\lambda_1| \gg |\lambda_2|$

  In Figure 5(e), this segmentation class is shown as gray. It corresponds to the dark region in Figure 5(b).

- class 3: $\lambda_1 > 0$, $|\lambda_1| \gg |\lambda_2|$

  In Figure 5(e), this segmentation class is shown as white. It corresponds to the white region in Figure 5(b).

We begin with the first class. For horizontal coordinate from 20 to 40, and vertical coordinate from 30 to 70, we have ellipsoidal deformations in the original woodgrain. This is characterized by eigenvalues that are proportional to the grain compression. Since there is significant compression in the woodgrain in all directions, we get significant values for both eigenvalues. Hence, the results for the first class agree with our intuition on what we might expect.

For both of the second and third clases, there is a direction along the image for which the woodgrain gets far more compressed, than any direction orthogonal to it. This is in the direction of the first eigenvector (with largest eigenvalue magnitude). The sign of this eigenvalue also determines the sign of the divergence of the instantaneous frequency vector field. Furthermore, it is evident from Figures 5(b) and 5(e) that only the sign of the divergence was important for the segmentation.

As discussed earlier, the sign of the divergence of the vector field tells us whether the local instantaneous frequency vector field points towards a point, or away from a point. To explain the sign change, we return to the region classified as class 1: of horizontal-coordinates 20 to 40, vertical coordinates 30 to 70. As we move south-east on the image, the grain is getting less and less compressed along one direction, while it becomes more and more compressed along the direction orthogonal to the direction of least compression. Now, the instantaneous frequency vectors still point towards the zero frequency region of the ellipsoidal deformations. This means that the vector field is converging. It has a negative divergence. As we keep moving south-east, the grain stops compressing, but starts decompressing. This causes the instantaneous frequency vectors to change sign (by our convention). Now, the instantaneous frequency vector field is pointing away from the original zero frequency region. The vector field is diverging. It has negative divergence.

If we examine the plots of Figures 5(a) and 5(b) along the boundaries where the divergence field changes sign, we see that the woodgrain is characterized by regions of high grain compression, followed by low grain compression. The high compression regions are described by class 2, for positive eigenvalues, while the low compression regions are described by class 3 for negative eigenvalues.

In Figure 5(f), we have the results from using five segmentation classes. Comparing Figures 5(e) and 5(f), we see that the first three classes of Figure 5(e) also dominate the five-class segmentation. However, the segmentation results of Figure 5(f) show more sensitivity for the second eigenvalue.

Overall, we find the segementation results for the three woodgrain classes to be very meaningful. The segmentation classes characterize the fundamental structure of the woodgrain.

References

[1] A. C. Bovik, N. Gopal, T. Emmoth, and A. Restrepo. Localized measurement of emergent image frequencies by Gabor wavelets. *IEEE Trans. on Information Theory*, 38(2):691–712, Mar. 1992. Special Issue on Wavelet Transforms and Multiresolution Signal Analysis.

[2] J. P. Havlicek. *AM-FM Image Models*. PhD thesis, The University of Texas at Austin, 1996.

[3] S. Haykin. *Neural Networks - A comprehensive foundation*. Macmillan College Publishing Company, 1994.

[4] T. Kohonen. The self-organizing map. *Proc. of the IEEE*, 78(9):1464–1480, 1990.

[5] T. Kohonen. *Self-Organizing Maps*. Springer Verlag, 1995.

[6] P. Maragos and A. C. Bovik. Demodulation of images modeled by amplitude-frequency modulations using multidimensional energy separation. *Proc. IEEE Int. Conf. on Image Processing*, III:421–425, Nov. 1994.

[7] W. K. Pratt. *Digital Image Processing*. John Wiley and Sons, New York, second edition, 1991.

Step 1. Mirror-extend the image $I(\mathbf{x})$ (given in equation (8.3-1), [7])

Step 2. Apply logarithmic point oparator by setting: $I := \log(1 + I)$

Step 3. Remove DC component, and normalize image energy to unity.

Step 4. Compute image energy $A^2 || \widehat{\nabla \phi} ||^2$ over gabor lattice

Step 5. Compute the instantaneous frequency estimates over channels with maximum energy.

Step 6. For each estimate $\left(\hat{\phi}_{x1}, \hat{\phi}_{x2} \right)$, eliminate two-quadrant *ambiguities* by taking

the instantaneous frequency vector with the largest projection onto:

$$\exp\left(-\frac{x^2+y^2}{2\sigma^2} \right) \exp\left[-j2\pi \left(\pm\hat{\phi}_{x1}x_1 + \hat{\phi}_{x2}x_2 \right) \right]$$

Step 7. Estimate the instantaneous frequency gradiet tensor components by differentiating $\widehat{\nabla \phi}$.

Step 8. Estimate the eigenvalues of the instantaneous frequency gradient tensor.

Figure 1. The AM-FM demodulation algorithm.

Step 1. Initialise weights at small random values.

Step 2. Calculate distances between input vector and
weights vectors for each output node k:
$d_k = \sum_{i=1}^{N} \left(x_i - w_{i,k} \right)^2$,   where $N = 2$ and $k = 1, 2, \cdots, M$ M being the number of classes
The output node with minimum distance is the winner.

Step 3. Adapt the weights. The weights for each output
node k and for each i are adapted with
$w_{i,k}(t + 1) = w_{i,k}(t) + h_k \left(x_i - w_{i,k}(t) \right)$
The learning rate is a Gaussian function that gets narrower within time (neighbourhood).
The learning rate is also frequency sensitive per output neuron, which means that it gets
smaller the more often a neuron is selected as a winner.

$$h_k = g\exp\left[-\left(k - k_w \right)^2 t/2 \right]/\sqrt{t_{k_w}}$$

where $0 < g \le 1$, k_w is the winner node, t is the number of iterations and t_{kw} is the
number of times the specific node is selected winner. Setting the initial value of $g = 1$ forces
the network to fast learning even with limited amount of data.

Step 4. Go to Step 2 for all segmented inputs.

Figure 2. The algorithm for Learning Phase 1.

Step 1. Use the values of the weights vectors as obtained from learning phase 1.

Step 2. Present input and calculate distances d_k between input vector x_i and weight vectors $w_{i,k}$ for each output node k as in Step 2 of Figure 2.

The output node with minimum distance d_{k1} is the first winner $k1$ and the output node with the following minimum distance d_{k2} is the second winner $k2$.

Step 3. Adapt weights. The weights for the first and second winner output nodes $k1$, k_2 are adapted with:

$$w_{i,k1}(t+1) = w_{i,k1}(t) + h_{k1}(x_i - w_{i,k1}(t)), \quad w_{i,k2} = w_{i,k2}(t) - 0.1(d_{k1}/d_{k2}) h_{k1}(x_i - w_{i,k2}(t))$$

The learning rate h_{k1} begins from 0.2 and decreases linearly with the number of times $t_{k1,w}$ the specific node $k1$ is selected as first winner:

$$h_{k1} = 0.2 - 0.01 t_{k1,w}$$

If $h_{k1} < 0$, then $h_{k1} = 0$

The factor d_{k1}/d_{k2} is used so as to move the second winner far away if the segmentation boundaries are close enough or little if the segmentation boundaries are far away. In other words the weights vector with the correct label is moved towards the input while the weights vector with incorrect label is moved away from it.

Step 4. Go to Step 2 for all segmented inputs.

Figure 3. The algorithm for Learning Phase 2.

Step 1. Calculate distances between input vector and weight vectors using the d_k expression from Step 2 of Figure 2. The output node kw with minimum distance d_{kw} is the winner and the input is assigned to the class of the winner node.

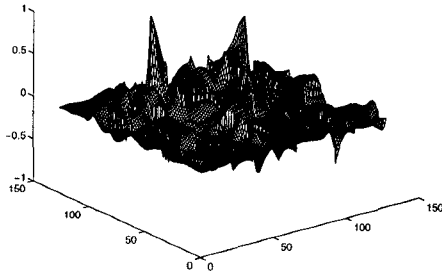**Step 2.** Repeat step 1 for all the inputs.

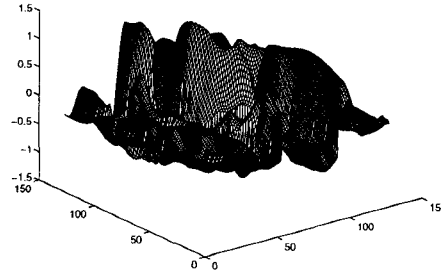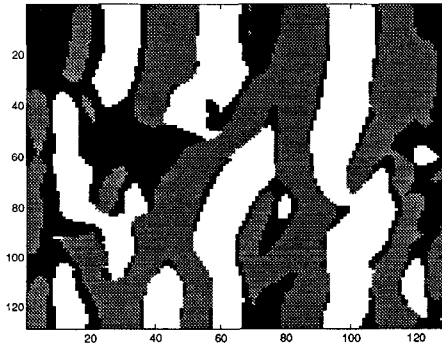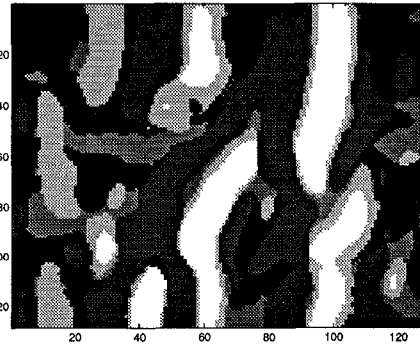**Figure 4. The algorithm for the segmentation phase.**
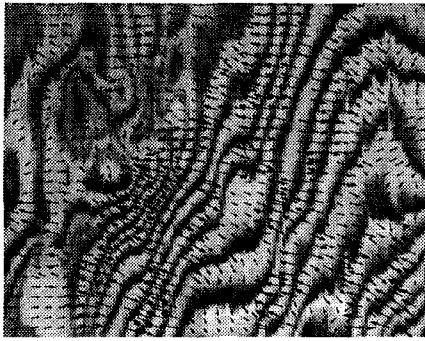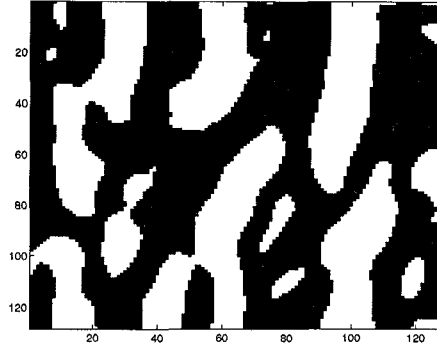
Figure 5. The segmentation of a woodgrain texture. (a) the wood grain image with the instantaneous frequency vectors (vector magnitudes were plotted using $\log(1 + 100 * M)$, where M denotes the vector magnitude). (b) positive region of the divergence of the instantaneous frequency vectors, (positive divergence $\nabla \cdot O > 0$ plotted white), (c) first eigenvalue plot, (d) second eigenvalue plot, (e) segmentation results for three classes, and (f) segmentation results for five classes.