

*Submitted as a REGULAR paper to
the IEEE Transactions on Image Processing
EDICS: IP 1.5 segmentation*

Frame Representations for Texture Segmentation¹

Andrew Laine² and Jian Fan

Center for Computer Vision and Visualization
Computer and Information Sciences Department
Computer Science and Engineering Building, Room 301
University of Florida, Gainesville, FL 32611-2024

Email: laine@cis.ufl.edu, Phone: (904)392-1239

Abstract³

We introduce a novel method of feature extraction for texture segmentation that relies on multi-channel wavelet frames and two-dimensional envelope detection. We described and compared two algorithms for envelope detection: (1) the Hilbert transform and (2) zero-crossings representation. Our study includes the selection of symmetric Quadrature Mirror Filters (QMF) for analysis and an ISODATA clustering algorithm. The reliability of our methods are demonstrated experimentally by quantitative evaluation on both natural and synthetic textures.

Keywords: Feature extraction, multiscale representations, image segmentation, wavelet analysis, computer vision.

¹This work was supported by the National Science Foundation under Grant No. IRI-9111375.

²Please address all correspondence to Andrew Laine.

³Permission to publish this abstract separately is granted.

1. INTRODUCTION

In the field of computer vision, the problem of texture segmentation has been investigated by many researchers through a diversity of approaches. In general, such methods consist of two distinct phases: feature extraction and clustering. Features for texture representation are of crucial importance for accomplishing segmentation[1]. Previous approaches for representing texture features can be divided into two categories [2] [3]: *statistical approaches* and *spatial/spatial-frequency approaches*.

In this paper, we describe a multi-channel approach, closely tied to wavelet multiresolution analysis. The existence of a multi-channel filtering model is evident from studies of the human visual system. Recent psychovisual experiments suggest the presence of a “biological spectrum analyzer” within the human visual system [4]. An early paper by Bovik [5] used complex Gabor filters, where texture features were computed by envelope and phase information extracted from two quadratic components of distinct output channels. In a related paper, A.K.Jain [9] chose real Gabor filters for analysis. Each filtered channel (output) was subjected to a nonlinear transformation, $\psi(t) = \tanh(\alpha t)$. An average absolute derivation was then computed in terms of overlapping windows.

Recent developments in wavelet theory provide an alternative approach through multi-channel filter banks that have several advantages over Gabor filters: (1) Wavelet filters cover *exactly* the frequency domain (complete representation), (2) Signal adaptive decomposition is made possible by the tree structure of multi-channel filters. (3) Fast algorithms are readily available to facilitate computation.

Several recent studies have reported the success of applying wavelet theory to texture analysis[11]. Laine *et al* [10] used wavelet packet signatures for texture classification and achieved perfect classification for 550 samples obtained from 25 distinct natural textures. In addition, M.Unser [12] has reported promising results using wavelets for both classification and segmentation of textures. In this paper, we introduce a feature extraction scheme that relies on wavelet multi-channel analysis and a novel envelope detection algorithm.

The remainder of this paper is organized as follows. Section 2 briefly reviews the framework of discrete wavelet transforms and the discrete wavelet packet transform, including corresponding variations of wavelet frames. Section 3 considers criteria for filter selection. Section 4 describes our multi-channel feature extraction scheme. Next, section 5 describes the ISODATA clustering algorithm used in our texture segmentation experiments. Section 6 presents our experimental results. Finally, Section 7 provides a brief summary and conclusion.

2. MULTI-CHANNEL WAVELET ANALYSIS

The general structure and computational framework of the discrete wavelet transform (DWT) are similar to subband coding systems. The main difference lies in filter design, where wavelet filters are required to be regular [13]. In this study, we considered two methods of multiscale analysis:

- Discrete wavelet transform (DWT) [13][14] [17], which corresponds to an octave-band filter bank.
- Discrete wavelet packet transform (DWPT) [15][16], which corresponds to a general tree-structured filter bank.

In general, the DWPT imposes a more rigorous constraint on a wavelet, that is, the wavelet should be orthonormal. However, the DWPT allows more flexibility by providing an adaptive basis. For a discussion on their continuous counterparts, please see references [13] [12] [16].

Unfortunately both decompositions are not translation invariant (a desirable property for accomplishing texture analysis [12]). A possible solution is to use an overcomplete wavelet decomposition called a discrete wavelet frame (DWF) [12]. In this case, a DWF or DWPF scheme is similar to its DWT and DWPT except that no down sampling occurs between levels. Figure 1 shows a general DWPF as a binary tree for a three level decomposition.

For each case above, filters $H_l(\omega)$ and $G_l(\omega)$ at level l were generated as described in [13] :

$$H_l(\omega) = H_0(2^l\omega) \tag{1}$$

$$G_l(\omega) = G_0(2^l\omega). \tag{2}$$

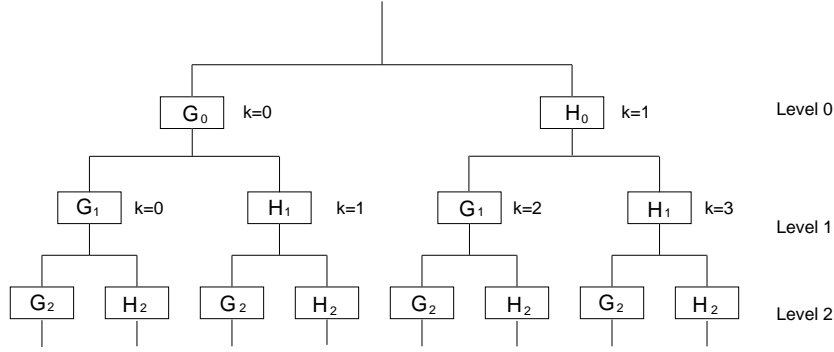


Figure 1: Tree structure for wavelet packet frames and associated indexes.

Let $S_k^l(\omega)$ be the Fourier transform of the input signal at channel k for level l , then

$$S_{2k}^{l+1}(\omega) = G_l(\omega)S_k^l(\omega) \quad (3)$$

$$S_{2k+1}^{l+1}(\omega) = H_l(\omega)S_k^l(\omega). \quad (4)$$

From the filter bank point of view, this is equivalent to a filter bank with channel filters $\{F_k^l(\omega) | 0 \leq k \leq 2^l - 1\}$, where $F_k^l(\omega)$ is defined recursively by the formula

$$F_0^0(\omega) = G_0(\omega), \quad F_1^0(\omega) = H_0(\omega), \quad (5)$$

$$F_{2k}^{l+1}(\omega) = G_{l+1}(\omega)F_k^l(\omega) = G_0(2^{l+1}\omega)F_k^l(\omega), \quad (6)$$

$$F_{2k+1}^{l+1}(\omega) = H_{l+1}(\omega)F_k^l(\omega) = H_0(2^{l+1}\omega)F_k^l(\omega). \quad (7)$$

For images, we simply use a tensor product extension for which the channel filters are written as

$$F_{i \times j}^l(\omega_x, \omega_y) = F_i^l(\omega_x)F_j^l(\omega_y). \quad (8)$$

Such 2-D filters naturally exhibit orientation selectivity. We classified each node in the decomposition tree into four possible categories taking into account orientation:

- The root node is *omni-directional*.
- The node at level l that was last filtered by $G_l(\omega_x)H_l(\omega_y)$ corresponds to *vertical-orientation*. (High-pass filter G_l is applied row-wise and low-pass filter H_l column-wise.)
- The node at level l that was last filtered by $H_l(\omega_x)G_l(\omega_y)$ corresponds to *horizontal-orientation*. (Low-pass filter H_l is applied row-wise and high-pass filter G_l column-wise.)

- The node at level l that was last filtered by $G_l(\omega_x)G_l(\omega_y)$ corresponds to *diagonal-orientation*. (High-pass filter G_l is applied row-wise and high-pass filter G_l column-wise)
- The node at level l that was last filtered by $H_l(\omega_x)H_l(\omega_y)$ has the same orientation as its parent. (Low-pass filter H_l is applied row-wise and low-pass filter H_l column-wise)

Recent results on texture classification have shown that different filters can have considerable impact on system performance [10, 12]. In the next section, we provide some justification and discuss considerations regarding filter selection.

3. FILTER SELECTION

Symmetry, frequency characteristics, and boundary accuracy are important factors in the design of filters for feature extraction. Below we discuss these constraints in terms of system performance.

- **Symmetry.** For accomplishing texture segmentation, accuracy in texture boundary detection is crucially important for reliable performance. In this application, filters with symmetry or antisymmetry are clearly favored. Such filters have a linear phase response, where the delay (shift) caused by such phase factors is predictable. Alternatively, filters with non-linear phase may introduce complicated distortions. Moreover, symmetric or anti-symmetric filters are also advantageous in alleviating boundary effects through simple methods of mirror extension.
- **Frequency characteristics.** One of the desirable frequency characteristic of a filter bank is that every band should exhibit a single pass-band with flat in-band frequency response. Many Quadrature Mirror Filters (QMF) possess this property. For multi-channel decompositions using a QMF, all filters are generated from a prototype filter that satisfies the property:

$$|H_0(\omega)|^2 + |H_0(\omega + \pi)|^2 = 1 \tag{9}$$

$$H_0(0) = 1, \quad H_0(\pi) = 0.$$

Unfortunately, a compactly supported QMF cannot be symmetric or anti-symmetric [17]. Therefore, in this investigation, we selected Lemarié-Battle wavelets, which are symmetric

and quadrature mirror filters (QMF). In our application, the high-pass filter $G_0(\omega)$ was obtained by frequency shifting of $H_0(\omega)$ by

$$\begin{aligned} g_0(n) &= (-1)^n h_0(n), \quad \text{or,} \\ G_0(\omega) &= H_0(\omega + \pi). \end{aligned} \tag{10}$$

In this way, both low-pass and high-pass filters had zero-phase, and thus no space (time) shifting existed after processing (analysis). This construction corresponds to a non-orthogonal wavelet transform, and only retains perfect decomposition-reconstruction properties for a frame-based representation (without downsampling between levels). In this application, however, the loss of orthogonality was not significant, since the transform coefficients were not used directly as features.

- **Time-Frequency localization.** The segmentation problem can be seen as a time-frequency localization problem. Therefore, the characteristics of a filter will affect system performance[5]. The effectiveness of a discrete filter in terms of time-frequency localization can be characterized by the *uncertainty factor*. The uncertainty factor U for a discrete filter can be defined as follows [6]:

$$U = \sigma_n^2 \sigma_\omega^2 \tag{11}$$

$$\sigma_n^2 = \frac{1}{E} \sum_n (n - \bar{n})^2 |h(n)|^2 \tag{12}$$

$$\sigma_\omega^2 = \frac{1}{2\pi E} \int_{-\pi}^{\pi} (\omega - \bar{\omega})^2 |H(e^{j\omega})|^2 d\omega \tag{13}$$

$$E = \sum_n |h(n)|^2 = \frac{1}{2\pi} \int_{-\pi}^{\pi} |H(e^{j\omega})|^2 d\omega \tag{14}$$

$$\bar{n} = \frac{1}{E} \sum_n n |h(n)|^2 \tag{15}$$

$$\bar{\omega} = \frac{1}{2\pi E} \int_{-\pi}^{\pi} \omega |H(e^{j\omega})|^2 d\omega \tag{16}$$

For the discrete case, Liu *et al*[6] proved that $U \geq 0.25$.

Figure 2 shows the uncertainty factor U for the Lemarié-Battle filters with odd order. This result showed that uncertainty factor U for the Lemarié-Battle filters is approximately linear, increasing with order. The minimum ($U = 0.393$) occurs at order 1.

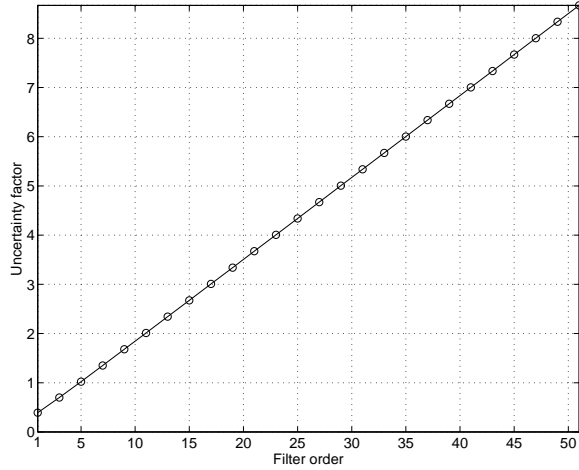


Figure 2: Uncertainty factor of odd-ordered Lemarié filters from 1 to 51.

Table 1: Truncated L1 filter of length 21.

h[0]	5.781633152e-01
h[1]	2.809314643e-01
h[2]	-4.886177424e-02
h[3]	-3.673090668e-02
h[4]	1.200034220e-02
h[5]	7.064417346e-03
h[6]	-2.745880792e-03
h[7]	-1.557014508e-03
h[8]	6.529218131e-04
h[9]	3.617812565e-04
h[10]	-1.586014278e-04

Note: $h[-n] = h[n]$

In consideration of this, we selected an order 1 Lemarié filter. The frequency response for such a filter can be written explicitly as:

$$H(\omega) = \cos^2\left(\frac{\omega}{2}\right) \sqrt{\frac{2 + \cos \omega}{1 + 2 \cos^2 \omega}} \quad (17)$$

$$G(\omega) = H(\omega + \pi) = \sin^2\left(\frac{\omega}{2}\right) \sqrt{\frac{2 - \cos \omega}{1 + 2 \cos^2 \omega}} \quad (18)$$

Figure 3 shows the filter banks constructed from the order 1 Lemarié filter for both DWF and DWPF analysis. In practice, this filter was truncated to finite length. In our study, we used an FIR filter $h_0(n)$ of length 21. The filter coefficients are shown in Table 1.

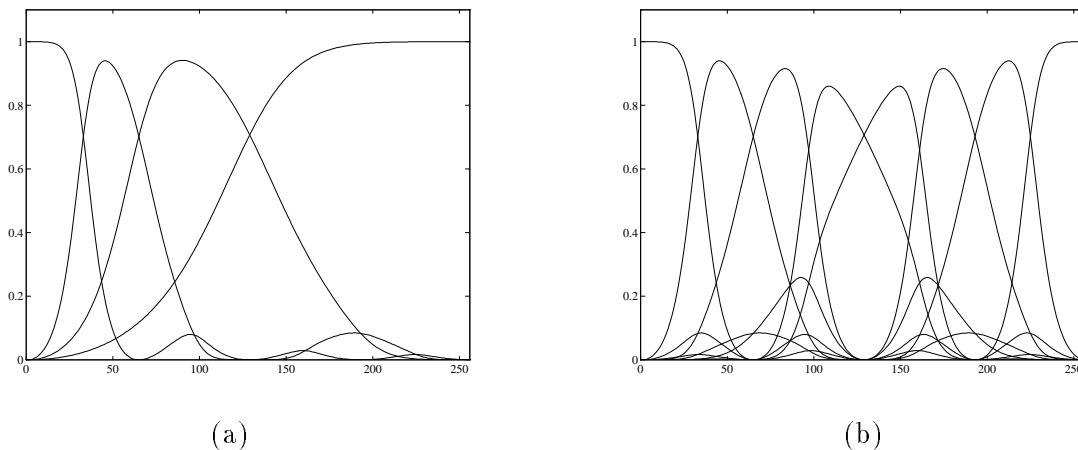


Figure 3: Filter banks constructed using a Lemarié filter of order 1, (a) DWF (b) DWPF.

4. FEATURE EXTRACTION

Feature extraction plays a crucial role in accomplishing reliable segmentation. A 'good' representation (feature) should be consistent among pixels within a class, while as disparate as possible between classes for reliable classification. This means that it should reflect some global view while retaining some discrimination capability at the pixel level. Therefore, the problem in general, is one of spatial-frequency analysis, and a natural application for multiscale wavelet analysis.

In addition, a feature extraction scheme should be able to accommodate a large variety of diverse inputs. This will exclude any fixed windowing schemes, since such methods are limited in adaptability. Next, we present an efficient representation by extending the concepts of a channel envelope to construct a feature set for reliable texture segmentation.

In an earlier study, Bovik [5] used envelopes obtained from channel outputs for texture analysis and segmentation in the setting of Gabor multi-channel filtering. The 2-D complex Gabor filters used by Bovik approximated a pair of conjugate filters. In addition, the approach of squaring followed by iterative low-pass filtering used by Unser [12] can also be interpreted as envelope detection. However, the method of iterative low-pass filtering is not adaptive, and can blur boundaries.

We investigated the following envelope-based feature extraction schemes using real wavelet-based multi-channel methods of analysis. For sake of clarity, we first present each

algorithm for the 1-D case.

1. **Envelope detection by Hilbert transform.** For a narrowband bandpass signal, its envelope can be computed by a corresponding analytical signal. For a signal $x(t)$, the analytic signal is defined by:

$$\tilde{x}(t) = x(t) + j\hat{x}(t) \quad (19)$$

where $\hat{x}(t)$ is the Hilbert transform of $x(t)$:

$$\hat{x}(t) = \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{x(\eta)}{t - \eta} d\eta. \quad (20)$$

The envelope of the original signal $x(t)$ is then simply the modulus of the analytic signal $\tilde{x}(t)$:

$$Env[x(t)] = |\tilde{x}(t)|. \quad (21)$$

The frequency characteristic of the Hilbert transform is then expressed by:

$$H(\omega) = \begin{cases} -j & , \omega \geq 0 \\ j & , \textit{otherwise}. \end{cases} \quad (22)$$

Therefore, the Fourier transform of the analytical signal $\tilde{x}(t)$ is:

$$\tilde{X}(\omega) = \begin{cases} 2X(\omega) & , \omega \geq 0 \\ 0 & , \textit{otherwise}. \end{cases} \quad (23)$$

Although the frequency characteristics corresponding to a noncausal system cannot be exactly realized in practice, for a DWT implemented by an FFT, the analytic signal can be approximately computed by setting the FFT values of the filter within the negative frequency range to zero. Since our implementations used convolution, it was necessary to design an approximate FIR Hilbert transformer.

2. **Envelope estimation by zero crossings.** Clearly, zero-crossing-based representations are adaptive. In this method, the maximum value between two adjacent zero-crossings was found, and assigned to points within the interval. Figure 4 shows an example using the L1 filter described earlier for a three-level DWF decomposition overlaid with detected envelopes.

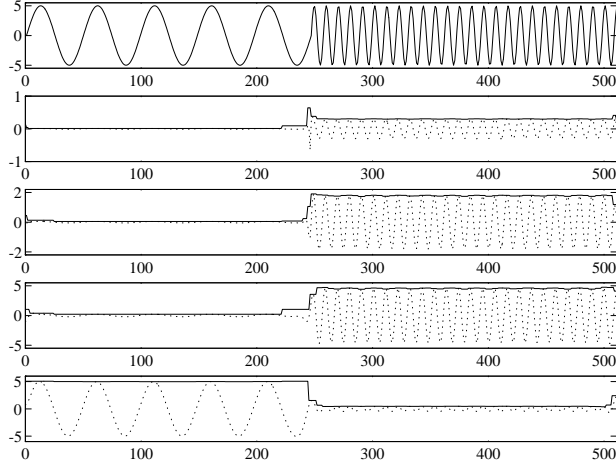


Figure 4: Envelope estimation via zero crossings using an L1 filter.
(solid line: envelope; dotted line: channel output).
Row 1: input signal; Row 2-4: wavelet coefficients for levels 1 through 3;
Row 5: DC coefficients, level 3.

We now extend the above algorithms for the analysis of two-dimensional image signals. In the 2-D frequency domain, a two-dimensional analytic signal can be obtained by setting an appropriate half plane to zero based on its orientation. That is, for a 2-D signal $f(x, y)$, the Fourier transform of an analytic signal $\tilde{f}(x, y)$ is either

$$\tilde{F}(\omega_x, \omega_y) = \begin{cases} 2F(\omega_x, \omega_y) & , \omega_x \geq 0 \\ 0 & , \textit{otherwise} \end{cases}$$

or,

$$\tilde{F}(\omega_x, \omega_y) = \begin{cases} 2F(\omega_x, \omega_y) & , \omega_y \geq 0 \\ 0 & , \textit{otherwise}. \end{cases} \quad (24)$$

For the 2-D filters used in our study, the equivalent complex quadrature filters exhibited the frequency response shown in Figure 5. Note that diagonal components can have an alternate arrangement by zeroing out the left half plane. This separable property allowed us to compute the envelope of a 2-D signal using the 1-D algorithms described earlier in a straight forward way, described below:

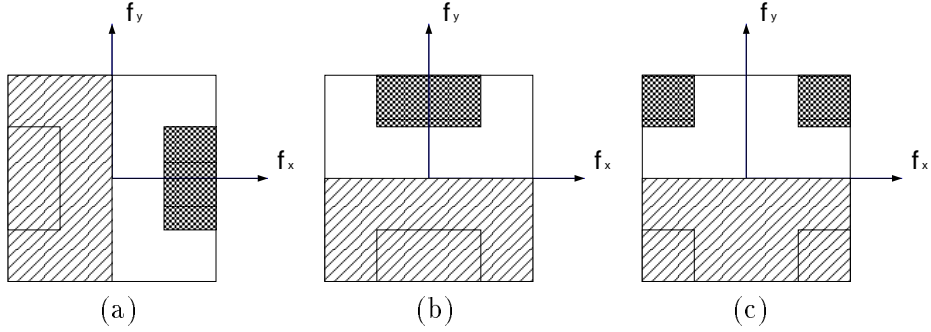


Figure 5: Frequency response of equivalent complex quadrature filters (level 1) for (a) vertical, (b) horizontal, (c) diagonal components, respectively. The diagonal shadowed areas identify the zeroed half planes.

Envelope_2D(W , Orientation)

```

    {  $W$  is a  $N \times N$  array }
begin
    case Orientation of
        Horizontal: apply Envelope_1D column-wise;
        Vertical   : apply Envelope_1D row-wise;
        Diagonal   : apply Envelope_1D column-wise;
    end case;
end Envelope_2D;

```

Note that no operation was applied to the omni-directional component due to its low frequency content. Figure 6 shows sample envelopes extracted by applying the two methods. The 1-D Hilbert transformer was an FIR (63 tap) filter designed using MONARCH [21] DSP software for a convolution based implementation of the DWPF.

At the end of the feature extraction process, we constructed feature vectors for every pixel (point) according to the following definition:

$$V_{i,j} = \{ |\widetilde{W}_{k,i,j}^{L-1}| \mid 0 \leq k \leq (2^{L-1} - 1) \}, \text{ for DWPF} \quad (25)$$

where $|\widetilde{W}_{k,i,j}^{L-1}|$ denotes the envelope value of pixel (i, j) for the k th component at level $L - 1$ in a DWPF tree.

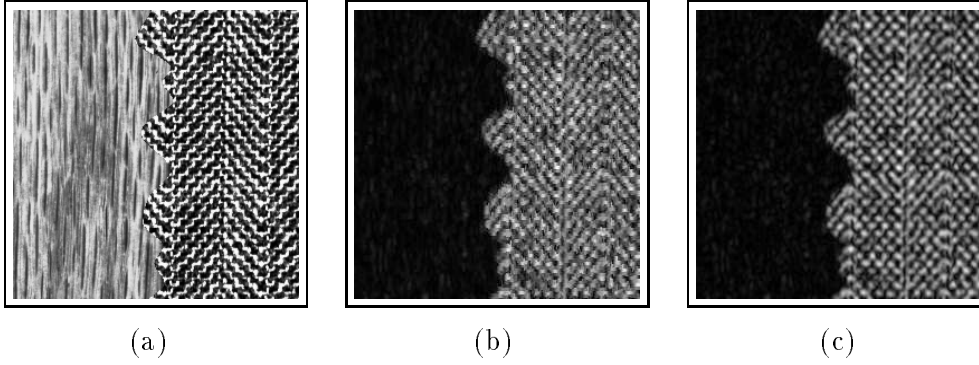


Figure 6: Envelope representations for horizontal components at level 2. (a) Original 256×256 , 8-bit texture image. (b) Envelope coefficients obtained from zero-crossings based algorithm. (c) Envelope coefficients obtained from the Hilbert transform based algorithm.

5. CLUSTERING ALGORITHM

Traditionally, segmentation algorithms accept as input a set of features and output a consistent labeling for each pixel. Fundamentally, this can be considered a multi-dimensional data clustering problem. As pointed out by Haralick [22], at present no general algorithms exist for this problem. Clustering algorithms that have been previously used for texture segmentation can be divided into two categories: *supervised segmentation* and *unsupervised segmentation*.

For practical applications, unsupervised segmentation is often desirable and easy to validate. It is particularly useful in those cases where testing samples are difficult to prepare (making supervised segmentation infeasible). Since our present focus emphasized feature extraction (representation), we used a general *ISODATA* clustering algorithm [8] (unsupervised). An overview of the algorithm is presented below:

Algorithm ISODATA(x, NC)

x : $N \times M$ array of structure containing vector and label fields.

NC : number of classes.

begin

begin (Initialization)

Scan the representation matrix x in raster order. Randomly pick a label from set $\{0, \dots, NC - 1\}$ and assign the label to each pixel.

Compute the class center $\{\vec{C}_k \mid 0 \leq k \leq NC - 1\}$ by calculating the mean vector for each class k .

end

repeat

Rescan the representation matrix x , and assign pixel (i, j) to the class k if the Euclidean distance between the pixel and the class center \vec{C}_k is closest.

Update the class centers $\{\vec{C}_k\}$ by recomputing their mean vectors.

until no change in labeling occurs.

end ISODATA;

This algorithm differs from a K-means algorithm in one important aspect: ISODATA updates class centers after a complete scan of an input feature set while K-means updates in every reassignment. In our experiments, ISODATA outperformed K-means for almost all cases.

This simple algorithm labeled each pixel independently and did not take into account the high correlation between neighboring pixels. Clearly, a more sophisticated algorithm should incorporate some neighborhood constraint into the segmentation process, such as relaxation labeling. For simplicity, we used median filtering in our preliminary experiments to simulate the benefit of a local constraint. In particular, we applied a 9×9 median filter as a post processing step to each initial segmentation.

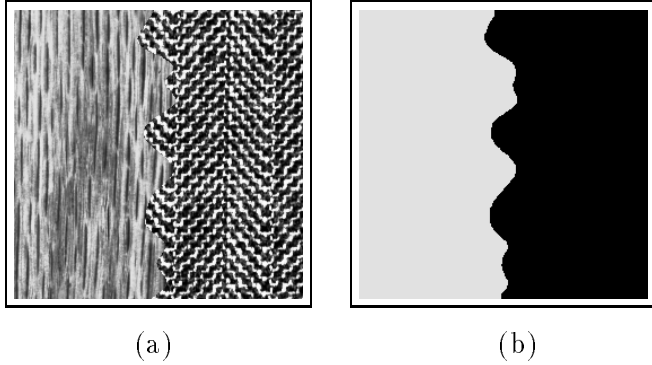


Figure 7: Multi-channel segmentation result No.1: (a) Test image T1 (256×256 , 8-bit) consists of D68, wood grain and D17, herringbone weave. (b) Final segmentation.

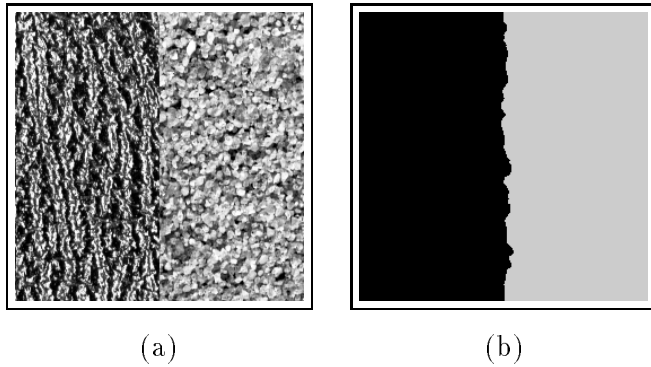


Figure 8: Multi-channel segmentation result No.2: (a) Test image T2 (256×256 , 8-bit) consists of D24, pressed calf leather and D29, beach sand. (b) Final segmentation.

6. EXPERIMENTAL RESULTS FOR TEXTURE SEGMENTATION

To test our segmentation algorithm, we carried out experiments using two distinct families of texture samples:

- **Natural textures.** Here we used textures obtained from the Brodatz album [24]. Each testing sample was histogram equalized so that a segmentation result based only on first order statistics was not possible. Experimental results showing accurate boundary separation are displayed in Figures 7, 8.
- **Synthetic textures.** We also tested the performance of our algorithm on several synthetic images of texture. Figure 9 shows a segmentation result on a Gaussian low-pass texture image [1], and Figure 10 shows a segmentation result on a filtered impulse noise (FIN)

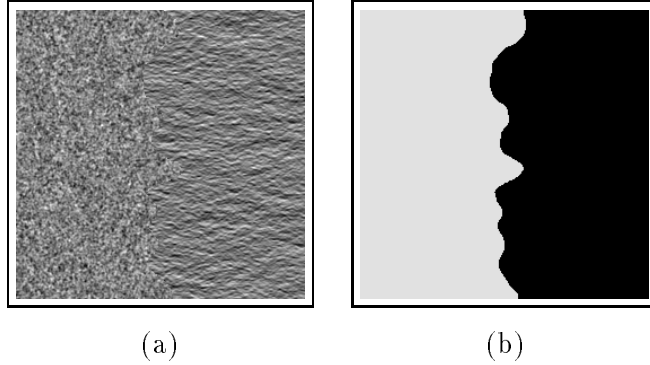


Figure 9: Multi-channel segmentation result No.4: (a) Test image T4 (256×256 , 8-bit): Gaussian LP, left: isotropic $F_c = 0, S_r = 60$; right: non-isotropic $F_c = 0, S_r = 60, \theta_0 = 0, B_0 = 0.175$. (b) Final segmentation.

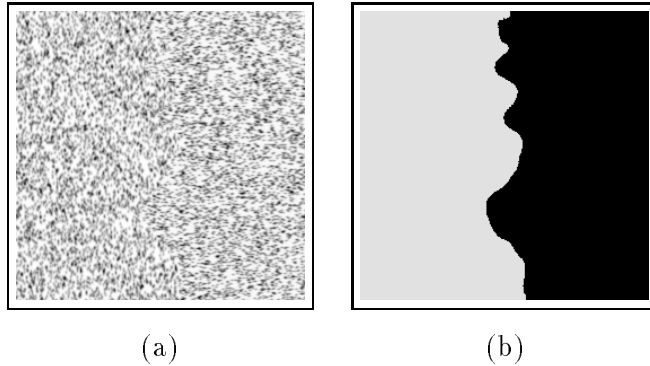


Figure 10: Multi-channel segmentation result No.5: (a) Test image T5 (256×256 , 8-bit): Filtered impulse noise, left: non-isotropic $T = 0.15, S_x = 1.0, S_y = 1.5$; right: non-isotropic $T = 0.15, S_x = 2.0, S_y = 1.0$. (b) Final segmentation.

texture image [1]. Figure 11 shows a remarkable segmentation result on a texture image containing illusory contours.

The accuracy of our segmentation results are summarized in Table 2. Note that the average absolute boundary errors remained less than 3 pixels. This performance is consistent with the difficulty of segmentation perceived by a human observer. Note that boundary errors were dependent on boundary shape in that complex boundaries yielded more variance.

7. CONCLUSIONS

We have described a feature extraction method for texture segmentation that relied upon multiscale wavelet frames representations. Our approach was adaptive in that no windowing was required. Two approaches for accomplishing one-dimensional envelope detection were presented

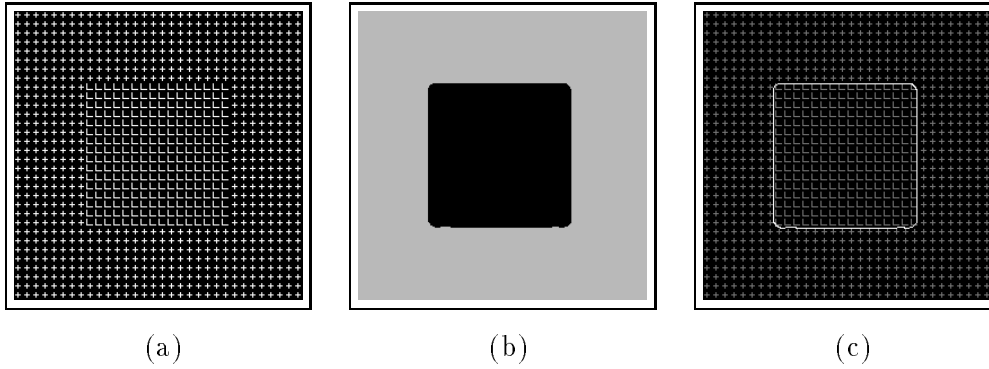


Figure 11: Multi-channel segmentation result No.6: (a) Test image T6 (256×256 , 8-bit). (b) Final segmentation. (c) Detected boundary overlaid with the original image.

Table 2: Boundary accuracy for multi-channel segmentation.

Test image	Maximum ABE	Average ABE	$\pm\sigma$
T1	4.0	1.5	1.0
T2	9.0	2.7	2.1
T4	14.0	2.6	2.1
T5	12.0	2.7	2.3

ABE: Absolute Boundary Error (in pixels).

and formulated for two-dimensional analysis. Performance for reliable texture segmentation was then evaluated. We observed that the algorithm based on zero-crossing detection outperformed the Hilbert transform algorithm. Filters were selected based on optimum criteria. Our results exhibited highly reliable and accurate performance for several distinct texture types including macro-texture and micro-textures.

Future research shall focus on exploiting the multiresolution nature of this representation to increase the accuracy of class boundaries, and finding an adaptive and robust criterion to prune the decomposition tree by selecting specific channels.

References

- [1] J.M.H.DU Buf, M.Kardan and M.Spann. “texture feature performance for image segmentation” *Pattern Recognition*, Vol. 23, 1990, pp. 291-309
- [2] J.Y.Hsiao and A.A.Sawchuk. “supervised textured image segmentation using feature smoothing and probabilistic relaxation techniques” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, 1989, pp. 1279-1292
- [3] T.R.Reed and H. Wechesler. “segmentation of textured images and Gestalt organization using spatial/spatial-frequency representations” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, 1990, pp. 1-12
- [4] S.G.Mallat. “multifrequency channel decompositions of images and wavelet methods” *IEEE Trans. Acoust., Speech., Signal Processing*, Vol.37, 1989, pp. 2091-2110.
- [5] A.C.Bovik. “analysis of multichannel narrowband filters for image texture segmentation”, *IEEE Trans. Signal Processing*, vol.39, pp.2025-2043, Sept. 1991
- [6] Yipeng Liu and A.N. Akansu. “an evaluation of time-frequency localization in transforms and filter banks” *IEEE Conf. on ASSP 1993* pp. 261-263, 1993
- [7] A.C.Bovik, N. Gopal, T.Emmoth, and A. Pestrepo. “localized measurement of emergent image frequencies by Gabor wavelets”, *IEEE Trans. Information Theory*, vol.38, pp.691-712, March 1992
- [8] R.O.Duda and P.E.Hart. “pattern classification and scene analysis” *John Wiley & Sons*, 1973

- [9] A.K.Jain and F.Farrokhnia. “unsupervised texture segmentation using Gabor filters”. *Pattern Recognition*, vol.24, No.12, pp. 1167–1186, 1991.
- [10] A.Laine and J.Fan. “texture classification by wavelet packet signatures” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 15, No. 11, November, 1993.
- [11] A.Laine and J.Fan. “An Adaptive Approach for Texture Segmentation by Multi-channel Wavelet Frames” *SPIE Proceedings on Mathematical Imaging: Wavelet Applications in Signal and Image Processing*, Vol. 2034, pp. 288-299, July 15-16, 1993, San Diego, CA.
- [12] M.Unser. “texture classification and segmentation using wavelets” *preprint*, NIH, 1992
- [13] O.Rioul and M.Vetterli. “wavelet and signal processing” *IEEE Signal Processing Magazine*, Oct. 1991, pp. 14-38
- [14] S.Mallat. “a theory for multiresolution signal decomposition : the wavelet representation”. *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-11, pp. 674–693, 1989.
- [15] M.V.Wickerhauser. “acoustic signal compression with wavelet packets” in *Wavelets: a tutorial in theory and applications*, C.K.Chui (eds), Academic Press, 1992
- [16] R.R.Coifman and M.V.Wickerhauser. “entropy-based algorithms for best basis selection”. *IEEE Trans. Infor. Theory*, vol. 38, pp. 713–718, 1992.
- [17] I.Daubechies. “Ten lectures on wavelets”. *Society for Industrial and Applied Mathematics, Philadelphia, Pennsylvania, 1992*
- [18] S.Mallat and S. Zhong. “characterization of signals from multiscale edges”. *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 14, pp. 710–732, 1992.
- [19] A.Papoulis. “signal analysis” *McGraw-Hill Book Company*, 1977.
- [20] J.Canny. “a computational approach to edge detection”. *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-8, pp. 679–697, 1986.
- [21] *MONARCH digital signal processing software*, The Athena Group, Inc., 3424 N.W. 31st Street, Gainesville, FL 32605, USA
- [22] R.M.Haralick and L.G.Shapiro. “image segmentation techniques” *Comput. Vision Graphics Image Processing* vol. 29, 1985, pp. 100-132

- [23] A.K.Jain and R.C.Dubes. “algorithms for clustering data”. *Prentice Hall, Englewood Cliffs, NJ 07632, 1988*
- [24] P.Brodatz. *Textures-a photographic album for artists and designers*. Dover, 1966.