

16th Conference on Applied Mathematics, Univ. of Central Oklahoma.
Electronic Journal of Differential Equations, Conf. 07, pp. 61–70.
ISSN: 1072-6691. URL: <http://ejde.math.swt.edu> or <http://ejde.math.unt.edu>
<ftp://ejde.math.swt.edu> (login: ftp)

DETERMINATION OF THE NUMBER OF TEXTURE SEGMENTS USING WAVELETS

JOSEPH P. HAVLICEK & PETER C. TAY

ABSTRACT. This paper presents a robust method of determining the number of texture segments in an image. We take an $N \times N$ image and decompose it into $n \times n$ blocks. A three-scale two-dimensional discrete wavelet transform is performed on each $n \times n$ block. For each $n \times n$ block, this transformation produces coefficients for 25 wavelet channels. The energy of each channel is used as a tuple for a vector in the feature space. Nearest neighbor clustering is used to segment the feature space. A measure is defined to determine the “goodness” of the clustering. The optimal number of segments is taken to be the clustering which maximizes our measure.

1. INTRODUCTION

Given a digital image, the segmentation problem is concerned with partitioning the image into several disjoint regions or *segments*. Each region should be homogeneous with respect to some particular properties of interest in the application at hand. Moreover, the segmentation should be such that merging any two segments results in a region that is not homogeneous. By *homogeneous* we mean that, within any segment, pixels should be similar with respect to one or more features including, *e.g.*, brightness, color, texture, motion, *etc.* [1]. Ideally, segmentation should yield a partition of the image into regions that correspond to meaningful image objects. One classic example arises in robot or autonomous vehicle navigation, where the goal of segmentation is to identify objects and boundaries that are useful for constructing a computer model of the vehicle’s immediate surroundings. Another example occurs in automated manufacturing, where the goal may be to identify defective components in a product as it progresses along an assembly line.

Image segmentation is generally classified as a *low-level* or *early* task because it usually precedes and is critical to the success of later high-level image processing and machine vision techniques. Normally, these high-level techniques are concerned with representing, interpreting, and perhaps enhancing the visual information present in an image. Despite its importance, the problem of segmenting a general image in the absence of *a priori* information remains unsolved today. Consequently, segmentation has become one of the most intensely studied problems in image processing

2000 *Mathematics Subject Classification.* 65T60, 42C40.

Key words and phrases. Differential geometry, algebraic geometry.

©2001 Southwest Texas State University.

Published July 20, 2001.

and machine vision. Surveys of many of the sophisticated techniques that have been proposed may be found in [1, 2, 3, 4, 5].

Among classes of segmentation problems, the segmentation of textured images is one of the most difficult [7]. Textured images are those containing objects and regions covered by quasi-periodic or random patterns of surface markings. Wood grains, a leopard's spots, and weave patterns in a fabric are but a few examples of naturally occurring image textures. While the notion of texture is easy to grasp intuitively, it is extremely difficult to quantify. To understand one aspect of why texture segmentation is so difficult, consider that, in the absence of textured regions, one approach to the image segmentation problem that is often effective is to first apply an edge detection technique and then to segment the image along the detected edges. Textured regions typically contain large numbers of edges at a multiplicity of scales, however. This generally causes edge-based segmentation techniques to fail miserably. For textured regions, the notion of homogeneity must be formulated in terms of the local coarseness, granularity, and spatial statistics of the texture patterns. This is a difficult problem in view of the fact that no satisfactory quantitative definition of texture exists at this time.

Numerous texture segmentation techniques have appeared in the recent literature, including, *e.g.*, approaches based on wavelet analysis [7, 20], filter banks [10, 14, 15], deterministic annealing [16], and stochastic models [17, 21, 22, 12, 13]. Recently, we also introduced a technique based on the idea of describing texture in terms of joint nonstationary amplitude and frequency modulations [6, 8]. The techniques we have just mentioned can be divided into three categories: unsupervised, partially unsupervised, and supervised. Unsupervised techniques are those that segment an image without making use of *a priori* information about the number of textures or their properties. Partially unsupervised techniques require some *a priori* information, typically either the number of regions or the region properties. Supervised techniques generally must be given both the number of regions and their properties and may also require guidance from a human operator at various stages in the segmentation procedure. For the general texture segmentation problem, unsupervised algorithms are clearly the most desirable.

When developing an unsupervised texture segmentation algorithm, there is no question that determining the number of texture regions present in an image without *a priori* information is one of the most challenging problems that must be addressed. In this paper we present a new technique for determining the number of texture regions present in an image without *a priori* information. This technique can be combined with existing supervised or partially unsupervised texture segmentation algorithms to create robust, fully unsupervised algorithms. It may also be combined with unsupervised techniques to improve the estimate of the number of regions that are present.

Our approach performs statistical clustering in a feature space of discrete wavelet transform coefficients computed over small disjoint blocks in the image. Each feature vector contains the wavelet coefficients from a single image block. Nearest neighbor clustering is then applied to group the feature vectors into clusters, where each cluster represents a texture segment in the image. A full dendrogram is constructed giving configurations with a number of clusters ranging from just one supercluster all the way up to a number of clusters equal to the number of image blocks over which wavelet coefficients were computed. Given a collection of

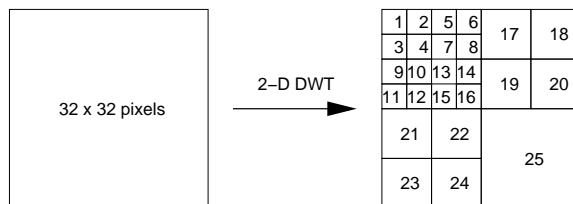


FIGURE 1. Depiction of the 25 channels in the three-level wavelet decomposition.

clustering configurations, the task of choosing which configuration (*i.e.*, how many clusters) is best is known as the *cluster validation problem* [18, 19]; it is a difficult problem that remains unsolved in general [17]. We use a novel validation criterion to select the optimal configuration and then take the number of clusters in this configuration as our estimate for the number of texture segments in the image.

The feature space in which clustering is performed is described in Section 2, while the details of the clustering algorithm and validation criterion are given in Section 3. In Section 4 we present a number of examples where the technique is applied to juxtapositions of textures from the well known Brodatz album [25].

2. FEATURE SPACE

Multiresolution image analysis as described in [9] has been a useful tool in image processing. Since low frequencies dominate virtually all real images, the 2-D wavelet transform’s ability to decompose an image in the low-frequency channel makes it ideal for image analysis [7]. Also for this reason we choose our decomposition to be more dense in the lower spatial frequencies than in the higher frequencies.

For simplicity in exposition, we consider only grayscale images of size 256×256 pixels. The image is partitioned into disjoint 32×32 pixel blocks. Let M be the number of such blocks in the image; for a 256×256 image, $M = 64$. Index the blocks in row major order so that B_1 and B_2 are the first and second leftmost blocks on the first block row, whereas B_M is the rightmost block on the last block row of the image. A three-scale 2-D discrete wavelet transform is applied to each block independently to produce a 25 channel subband decomposition of the block. The block diagram shown in Fig. 1 depicts this decomposition pictorially.

To perform the discrete wavelet decomposition, we use the Daubechies D_4 order-eight wavelet. The coefficients of the low-pass quadrature mirror filter are [24]

$$[-0.0106, 0.0329, 0.0308, -0.1870, -0.0280, 0.6309, 0.7148, 0.2304],$$

while high-pass filter coefficients are given by [24]

$$[-0.2304, 0.7148, -0.6309, -0.0280, 0.1870, 0.0308, -0.0329, -0.0106].$$

We use a separable 2-D discrete wavelet transform implemented by sequentially performing 1-D convolution along each row of a block with the appropriate filter, discarding every other column from the resulting filtered rows (down sampling), and then convolving each of the remaining columns with the appropriate 1-D filter. Finally, every other row is discarded from the resulting filtered columns. The 1-D

convolution operation is defined by

$$y[n] = \sum_{k=1}^{32} x[n-k]h[k], \quad (2.1)$$

where $y[n]$ is the filtered result, $x[n]$ is the row or column of a 32×32 block that is being filtered, and $h[n]$ is the vector of filter coefficients. Edge effects (indices in (2.1) that fall outside the domain of definition of $x[n]$) are handled by reflecting the vector $x[n]$ about its endpoints. We elected to handle edge effects in this way in order to minimize the introduction of frequencies not present in the image during the finite-length convolution operations.

For each $i \in [1, M]$ and each $k \in [1, 25]$, let $e_{i,k}$ denote the average absolute value of the wavelet coefficients in the k^{th} subband of image block B_i . We describe block B_i by constructing a wavelet domain feature vector \mathbf{e}_i according to

$$\mathbf{e}_i = [e_{i,1} \ e_{i,2} \ \dots \ e_{i,25}]^T.$$

Let $\mathcal{F} = \{\mathbf{e}_i : i \in [1, M]\}$. Thus, \mathcal{F} is a 25-D feature space that contains M vectors, each describing one block from the original image.

If clustering were performed on \mathcal{F} alone, there is no guarantee that the resulting clusters would correspond to spatially connected regions in the image. Since such regions are almost always desirable, we augment the feature space by adding two additional dimensions to describe spatial position. This has the effect of enforcing a spatial correspondence constraint on the clusters delivered by the algorithm described in Section 3. Let r_i and c_i denote, respectively, the average row coordinate and average column coordinate for pixels in block B_i . Let $\mathcal{C} = \{[r_i \ c_i]^T : i \in [1, M]\}$. Then \mathcal{C} contains vectors that describe the spatial centroids of the M image blocks B_i . The augmented feature space is given by $\mathcal{F} \times \mathcal{C}$. In this feature space, image block B_i is described by the vector $\mathbf{w}_i = [\mathbf{e}_i^T \ r_i \ c_i]^T$.

For each $k \in [1, 27]$, the collection of the k^{th} entries from all M vectors $\mathbf{w}_i \in \mathcal{F} \times \mathcal{C}$ is called a *feature*. To minimize the possibility that one or a few features with relatively large numerical values might dominate the segmentation procedure, we normalize each feature independently. For feature k , the normalization consists of first computing the sample standard deviation of the feature and then dividing the k^{th} entry of each vector \mathbf{w}_i by this value. We use the notation $\mathcal{F}' \times \mathcal{C}'$ to denote the normalized feature space.

An example of one of the input images we consider appears in Fig. 2(a). This image is a juxtaposition of two textures from the Brodatz album [25]: the texture in the center is called *burlap* and the one in the surround is called *mica*. Hence we refer to this image as *micaburlap*.

For four blocks B_i taken from this image, Fig. 2(b) illustrates the projection of the vector \mathbf{w}_i onto the wavelet subspace \mathcal{F}' . Each graph in Fig. 2(b) depicts the normalized entries $e_{i,k}$, where k is the abscissa. Specifically, the top two graphs show the projections of \mathbf{w}_1 and \mathbf{w}_8 , corresponding to blocks B_1 and B_8 which are located, respectively, in the upper left and right corners of the image. Note that both of these blocks belong to the surrounding *mica* texture. The bottom two graphs in Fig. 2(b) correspond to blocks B_{28} and B_{29} , both of which lie within the center *burlap* texture. In Fig. 2, it is evident that the blocks corresponding to the two different texture regions have noticeably distinct feature vectors, particularly in the fifth through fifteenth coordinates.

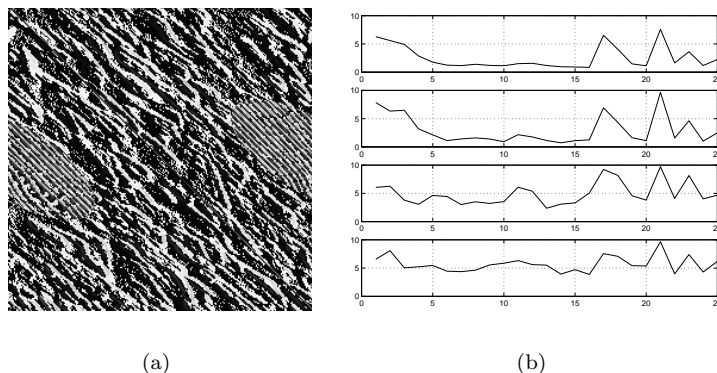


FIGURE 2. (a) Two-texture image *micaburlap*. (b) Plot of 25 entries of normalized feature vectors computed from four blocks of the *micaburlap* image. The top two graphs show feature vectors computed from the upper leftmost and upper rightmost blocks of the surround texture (*mica*), while the bottom two graphs show feature vectors computed from two blocks of the center texture (*burlap*).

3. NEAREST NEIGHBOR CLUSTERING

The well known nearest neighbor clustering (NNC) algorithm is described in [18]. Initially, each of the M feature vectors in $\mathcal{F}' \times \mathcal{C}'$ is considered to be a cluster. The algorithm iterates through $M - 1$ passes. At each pass, the two clusters that are closest to one another are merged together into a single cluster. Thus there are M clusters prior to the first pass and only one cluster remains after pass $M - 1$.

To make the notion of the *closeness* of two clusters precise, we impose the following metric on $\mathcal{F}' \times \mathcal{C}'$:

$$\delta(\mathbf{w}_i, \mathbf{w}_j) = \lambda d(\mathbf{e}_i, \mathbf{e}_j) + (1 - \lambda) d([r_i, c_i]^T, [r_j, c_j]^T), \quad (3.1)$$

where $0 \leq \lambda \leq 1$ and $d(\cdot, \cdot)$ is the usual Euclidean metric. The term λ appearing in (3.1) weights the relative contributions to the metric δ of the wavelet coefficient energies in \mathcal{F}' and the spatial position information in \mathcal{C}' .

In a given pass of the NNC algorithm, let L_j denote the number of feature vectors contained in cluster C_j and induce an arbitrary ordering on these feature vectors so that $C_j = \{\mathbf{w}_{j,1} \ \mathbf{w}_{j,2} \ \dots \ \mathbf{w}_{j,L_j}\}$. We define the distance between clusters C_j and C_k by

$$\Delta(C_j, C_k) = \min_{p \in [1, L_j], q \in [1, L_k]} \delta(\mathbf{w}_{j,p}, \mathbf{w}_{k,q}) \quad (3.2)$$

The intuitive meaning of (3.2) is that the *closeness* of clusters C_j and C_k is defined by the distance between their two nearest elements with respect to the metric δ . In each pass of the NNC algorithm, we merge the two clusters that minimize Δ .

When it terminates after $M - 1$ iterations, the NNC algorithm delivers M cluster configurations $\Gamma_M \dots \Gamma_1$, where k clusters are present in configuration Γ_k . We choose one of these as the final clustering result by applying a validation criterion to quantify the “goodness” of each configuration. Typically, for some K considered to

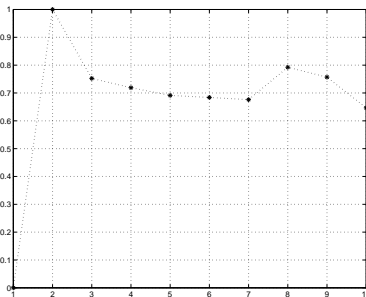


FIGURE 3. Ratio R_k for *micaburlap* image. The ratio is maximized by the correct choice $\mathcal{N} = 2$ for the number of texture segments present in the image.

be the maximum number of segments that might be present in the image, validation is applied only to configurations Γ_k for $k \in [1, K]$.

The validation criterion is applied to configuration Γ_k as follows. Using the distance metric (3.1), we first compute the centroid of each of the clusters in the configuration. Then we compute \overline{C}_k , the average distance between any two distinct centroids. The average within cluster distance \overline{W}_k is the average distance of all the feature vectors in $\mathcal{F}' \times \mathcal{C}'$ to the centroid of their respective clusters. The goodness of Γ_k is then defined by the ratio $R_k = \overline{C}_k / \overline{W}_k$. Ideally, we would like for the average between cluster distance \overline{C}_k to be large and the average within cluster distance \overline{W}_k to be small. Hence our validation criterion selects the configuration Γ_k that maximizes R_k to be the final clustering result. For our estimate \mathcal{N} of the number of texture segments that are present in the image, we use the number of clusters in the final clustering result, which is given by

$$\mathcal{N} = \arg \max_{k \in [1, K]} R_k. \quad (3.3)$$

4. RESULTS

In this section, we present several examples where the algorithm described in Sections 2 and 3 was applied to textured images. Each image was a 256×256 grayscale composition of textures from the Brodatz album [25]. Each image was partitioned into 32×32 blocks B_i , giving $M = 64$. In every case, we took $K = 10$ as an upper bound on the number of texture segments that might be present. The experimentally determined value $\lambda = 0.8$ for the weight in (3.1) was used throughout. The algorithm was implemented using the software environment *Matlab* with the Signal Processing and Wavelet toolboxes.

For the two-texture image *micaburlap* shown in Fig. 2(a), the ratio R_k is graphed as a function of k in Fig. 3. As demonstrated by the figure, R_k is maximized for the correct choice.

Another two-texture example is given in Fig. 4. The image *FlowersStraw* appears in Fig. 4(a). The ratio R_k is plotted in Fig. 4(b), where it is again seen that our technique selects the correct value $\mathcal{N} = 2$ for the number of texture segments that are present.

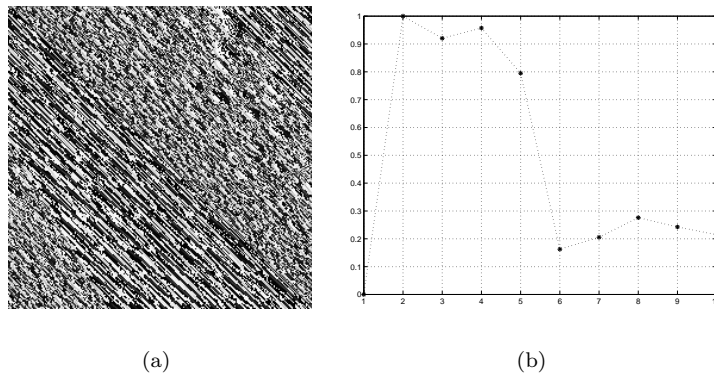


FIGURE 4. (a) Two-texture image *FlowersStraw*. (b) Ratio R_k for the *FlowersStraw* image; the ratio is maximized for the choice $\mathcal{N} = 2$ textured regions.

Two three-texture examples are given in Fig. 5. The image *CorkWoodBurlap* is shown in Fig. 5(a), while the image *BurlapGrassReptile* appears in Fig. 5(c). Corresponding graphs of the ratio R_k produced by the algorithm are shown in Fig. 5(b) and (d). In both cases, the ratio is maximized by the choice $\mathcal{N} = 3$, which agrees with the number of texture segments that are actually present in the images.

Our final two examples are a pair of four-texture examples presented in Fig. 6. The original images are shown in Fig. 6(a) and (c), while corresponding plots of the ratio R_k delivered by our algorithm appear in Fig. 6(b) and (d). Once again, we see that the proposed approach delivers the correct choice $\mathcal{N} = 4$ in both of these cases.

5. SUMMARY

The problem of segmenting an image into several disjoint homogeneous regions that correspond to meaningful objects is fundamental to a variety of applications in image processing and machine vision. Among such problems, the segmentation of textured images is particularly difficult. Of the highest practical interest are fully unsupervised texture segmentation algorithms capable of performing the segmentation task in the absence of *a priori* information on the number of texture segments or their properties. One of the most difficult aspects of developing such algorithms is determining how many segments are actually present.

In this paper, we have presented a robust technique that determines the number of texture segments by performing nearest neighbor clustering in a wavelet domain feature space. The image is partitioned into small disjoint blocks and a three-scale 2-D wavelet transform is applied to decompose each block into 25 wavelet subbands. For each block, a feature vector is constructed by averaging the absolute values of the wavelet coefficients in each of the 25 subbands. Two additional feature space dimensions are added to incorporate spatial position information, effectively enforcing a spatial correspondence constraint on the clustering results.

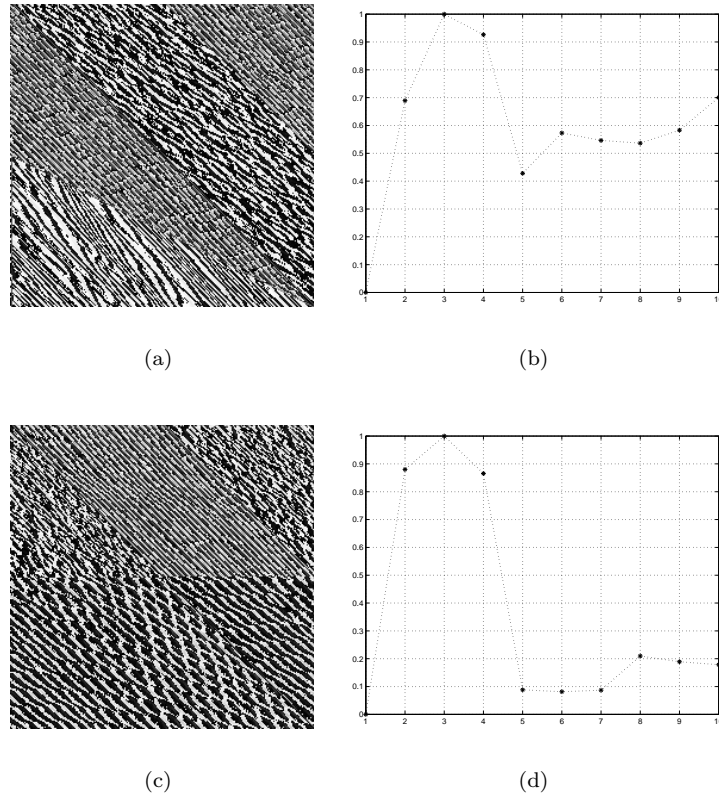


FIGURE 5. Three-texture examples. (a) *CorkWoodBurlap* image. (b) Ratio R_k for the *CorkWoodBurlap* image; the ratio is maximized for the choice $\mathcal{N} = 3$ textured regions. (c) *BurlapGrassReptile* image. (d) Ratio R_k for the *BurlapGrassReptile* image; the ratio is maximized for the choice $\mathcal{N} = 3$ textured regions.

The nearest neighbor algorithm is applied repeatedly to produce multiple clustering configurations, where the number of clusters in each configuration ranges from one up to the number of blocks into which the image was partitioned. By applying a validation criterion based on the ratio of the average between and within cluster distances, one of the configurations is selected as the final clustering result. The estimate for the number of texture regions present in the image is taken equal to the number of clusters in this final result.

Using the Daubechies D_4 wavelet, we demonstrated the technique on a number of two-, three-, and four-texture images. In each case, correct estimates for the number of textured regions present were obtained using an experimentally determined value of $\lambda = 0.8$ for the weight parameter in (3.1). In total, we have applied the technique to 15 images similar to the ones shown in Fig. 2 - Fig. 6. With $\lambda = 0.8$, the algorithm delivered correct results in all but three cases. Of these three cases, one was a four-texture image for which the algorithm estimated $\mathcal{N} = 3$. The other two were five-texture images for which the algorithm estimated $\mathcal{N} = 6$. However, for

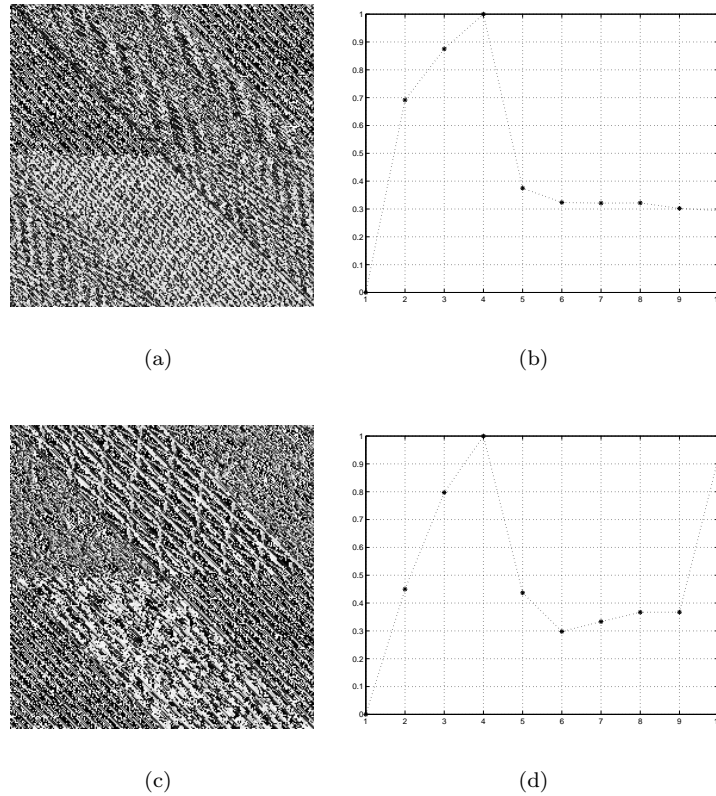


FIGURE 6. Four-texture examples. (a) Original image. (b) Ratio R_k for the image in (a). (c) Original image. (d) Ratio R_k for the image in (c).

both of these five-texture images, the correct result $\mathcal{N} = 5$ was obtained using a value $\lambda = 0.7$ for the weight parameter.

Ideally, one would hope to find a single value for λ that works universally on large classes of images. In our future work, we will continue to fine tune this parameter and will also investigate methods for determining λ dynamically from the data. We also plan to investigate the sensitivity of the algorithm to the particular choice of wavelet. The *Bath Visually Optimal Wavelet*[23] merits significant future study in this regard and may have the potential to lead to segmentations that agree more closely with biological visual perception.

REFERENCES

- [1] S. Lakshmanan, *Statistical methods for image segmentation*, in *Handbook of image and video processing*, A.C. Bovik, ed., Academic Press, San Diego, 2000, pp. 355–365.
- [2] B.S. Manjunath, G.M. Haley, and W.Y. Ma, *Multiband techniques for texture classification and segmentation*, in *Handbook of image and video processing*, A.C. Bovik, ed., Academic Press, San Diego, 2000, pp. 367–382.
- [3] J. Ghosh, *Adaptive and neural methods for image segmentation*, in *Handbook of image and video processing*, A.C. Bovik, ed., Academic Press, San Diego, 2000, pp. 401–414.

- [4] R.M. Haralick, *Image segmentation survey*, in *Fundamentals in computer vision*, O.D. Faugeras, ed., Cambridge Univ. Press, Cambridge, 1983, pp. 209–224.
- [5] T.R. Reed and J.M.H. DuBuf, *A review of recent texture segmentation and feature extraction techniques*, CVGIP: Image Understanding, **57** (1993), 359–372.
- [6] T.B. Yap, T. Tangsukson, P.C. Tay, N.D. Mamuya, and J.P. Havlicek, *Unsupervised texture segmentation using dominant image modulation*, Proc. 34th IEEE Asilomar Conf. Signals, Syst., Comput., Pacific Grove, CA, Oct. 29–31, 2000.
- [7] R. Porter, *A robust automatic clustering scheme for image segmentation using wavelets*, IEEE Trans. Image Proc., **5** (1996), 662–665.
- [8] T. Tangsukson, *AM-FM texture segmentation*, M.S. Thesis, University of Oklahoma, 2000.
- [9] S. G. Mallat, *A theory for multiresolution signal decomposition: the wavelet representation*, IEEE Trans. Pattern Anal. Machine Intell., **11** (1989), 674–693.
- [10] A. C. Bovik, M. Clark, and W. S. Geisler, *Multichannel texture analysis using localized spatial filters*, IEEE Trans. Pattern Anal. Machine Intell., **12** (1990), 55–73.
- [11] K. Etemad and R. Chellappa, *Separability-based multiscale basis selection and feature extraction for signal and image classification*, IEEE Trans. Image Proc., **7** (1998), 1453–1465.
- [12] B.S. Manjunath and R. Chellappa, *Unsupervised texture segmentation using Markov random field models*, IEEE Trans. Pattern Anal. Machine Intell., **13** (1991), 478–482.
- [13] C.S. Won and H. Derin, *Unsupervised segmentation of noisy and textured images using Markov random fields*, CVGIP: Graph. Models and Image Proc., **54** (1992), 308–328.
- [14] D. Dunn and W.E. Higgins, *Optimal Gabor filters for texture segmentation*, IEEE Trans. Image Proc., **4** (1995), 947–964.
- [15] T.P. Weldon and W.E. Higgins, *An algorithm for designing multiple Gabor filters for segmenting multi-textured images*, Proc. IEEE Int’l. Conf. Image Proc., Chicago, IL, Oct. 4–7, 1998.
- [16] T. Hofmann, J. Puzicha, J.M. Buhmann, *Unsupervised texture segmentation in a deterministic annealing framework*, IEEE Trans. Pattern Anal. Machine Intell., **20** (1998), 803–818.
- [17] D. A. Langan, J. W. Modestino, and J. Zhang, *Cluster validation for unsupervised stochastic model-based image segmentation*, IEEE Trans. Image Proc., **7** (1998), 180–195.
- [18] A.K. Jain and R.C. Dubes, *Algorithms for clustering data*, Prentice Hall, Englewood Cliffs, NJ, 1988.
- [19] R.C. Dubes, *How many clusters are best? An experiment*, Pattern Recognit., **20** (1987), 645–663.
- [20] M. Unser, *Texture classification and segmentation using wavelet frames*, IEEE Trans. Image Proc., **4** (1995), 1549–1560.
- [21] J. Chen and A. Kundu, *Unsupervised texture segmentation using multichannel decomposition and hidden Markov models*, IEEE Trans. Image Proc., **4** (1995), 603–619.
- [22] C. Kervrann and F. Heitz, *A Markov random field model-based approach to unsupervised texture segmentation using local and global spatial statistics*, IEEE Trans. Image Proc., **4** (1995), 856–862.
- [23] B. G. Sherlock and D. M. Monro, *On the space of orthonormal wavelets*, IEEE Trans. Signal Proc., **46** (1998), 1716–1720.
- [24] I. Daubechies, *Orthonormal bases of compactly supported wavelets*, Commun. Pure Appl. Math., **41** (1988), 909–966.
- [25] P. Brodatz, *Textures: a photographic album for artists and designers*, Dover, New York, 1966.

SCHOOL OF ELECTRICAL & COMPUTER ENGINEERING, THE UNIVERSITY OF OKLAHOMA, NORMAN, OK 73019-1023, USA

E-mail address, Joseph P. Havlicek: joebob@ou.edu

E-mail address, Peter C. Tay: Peter.C.Tay-1@ou.edu