A Robust Automatic Clustering Scheme for Image Segmentation Using Wavelets

Robert Porter and Nishan Canagarajah

Abstract— **The optimal features with which to discriminate between regions and, thus, segment an image often differ depending on the nature of the image. Many real images are made up of both smooth and textured regions and are best segmented using different features in different areas. In this correspondence, a scheme that automatically selects the optimal features for each pixel using wavelet analysis is proposed, leading to a robust segmentation algorithm. An automatic method for determining the optimal number of regions for segmentation is also developed.**

I. Introduction

Image segmentation is becoming increasingly important in a variety of fields such as video coding, computer vision, and medical imaging [1], [2]. The objective of dividing an image into homogeneous regions remains a challenge, especially when the image is made up of complex textures. A number of approaches have been suggested for this task, including spatial frequency techniques [3], [4] which have proven quite successful. Image segmentation becomes much simpler for images made up of smoother regions, where the use of simple local grey level statistics often suffices. However, many real images are made up of a variety of smooth and textured regions, all of which need to be reliably identified in the segmentation algorithm. In these cases, the existing techniques fail to produce a meaningful segmentation [4], successfully segmenting only the smooth or textured regions, depending on the features used. Therefore, it would clearly be desirable to have some means of feature selection prior to segmentation. In this way, highly textured regions can be segmented using spatial frequency-based features, whereas smooth regions can be segmented using local grey level statistics such as mean and variance. In this correspondence, a scheme that automatically selects the optimal features for each pixel using wavelet analysis is proposed.

Clustering techniques [5] are commonly used for image segmentation in a multidimensional feature space. The widely used *k-means* clustering routine usually requires a threshold in its determination of the optimal number of regions for segmentation (the "true cluster number"). The setting of this threshold is somewhat arbitrary and seldom results in a consistently optimal segmentation. In this correspondence, we also develop an automatic method for determining the true cluster number, which requires no arbitrary threshold and leads to a fully automatic image segmentation algorithm.

The block diagram in Fig. 1 summarizes the main features of the proposed segmentation algorithm.

II. Optimal Feature Selection

The first stage of image segmentation usually involves the development of a feature space. This comprises calculating the values of several features for each pixel (or block of pixels) in the image. Each feature should in some way describe the appearance of the local area surrounding the pixel. These features make up a feature vector for each pixel so that each is represented by a point in a multidimensional feature space. Consequently, if the features used for the image are good descriptors, similar appearing regions in the image will contain pixels whose feature vectors occupy similar positions in the feature space. These are known as clusters, and the purpose of clustering is to identify these clusters and classify the image's pixels accordingly.

Manuscript received June 30, 1995; revised November 3, 1995. The associate editor coordinating the review of this paper and approving it for publication was Prof. Patrick A. Kelly.

The authors are with Image Communications Group, Centre for Communications Research, Department of Electrical and Electronic Engineering, University of Bristol, Bristol, U.K.

Publisher Item Identifier S 1057-7149(96)02755-8.

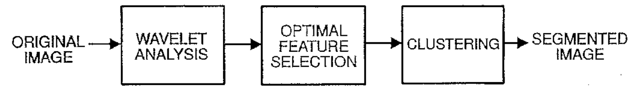

Fig. 1. Block diagram of the proposed segmentation algorithm.

The quality and the accuracy of the segmentation ultimately depends on the type of features used. Therefore, it is very important that these features suitably characterize the aspects of the image on which the segmentation is to be based. For example, images consisting of a number of highly textured regions are best segmented using frequency-based features, whereas images made up of smoother regions can more easily be segmented using local grey level mean and variance as features. Many real images are made up of both types of region and, thus, require different features to be used in different areas of the image. As a result, existing segmentation algorithms have failed to produce meaningful segmentations of many real images. The proposed algorithm involves the use of wavelet analysis to determine the optimal features corresponding to each pixel or block of pixels as described in the next section.

III. Wavelet Analysis of Images

The 2-D wavelet transform [6], [7] is a very popular tool in image processing. Its ability to repeatedly decompose an image in the low-frequency channels makes it ideal for image analysis since the lower frequencies tend to dominate real images. In this correspondence, the wavelet transform is used both to analyze the image prior to segmentation enabling feature selection as well as to provide spatial frequency-based descriptors as features for segmenting textures.

We would expect smooth images in which there are only gradual variations in grey level to be dominated by low spatial frequencies, whereas textured images in which the grey level varies rapidly should be made up of a wide range of frequencies. Smooth and textured images can thus easily be distinguished from each other by examining their wavelet transforms. Fig. 2 shows the 2-D wavelet transform of a smooth image and a textured image (a portion of cloud and canvas, respectively, from Fig. 6(a), both of which have had their means subtracted prior to transformation so that there is no dc component). The 2-D graphs in Fig. 2 represent the magnitudes of the wavelet coefficients across the various frequencies and orientations of the 2-D wavelet transform (with the lowest frequencies displayed closest to the viewer). The smooth image has strong components only in the low frequencies as can be seen in the area around its main peak. However, the textured image has substantial components in a wide frequency/scale spectrum, as expected. In order to extract useful information from this and, hence, discriminate between smooth and textured images, it is useful to group the wavelet coefficients into channels representing the various frequency/scale bands.

A three-level wavelet decomposition of an image results in 10 main wavelet channels, as shown in Fig. 3. The energy of each channel can be calculated by simply finding the mean magnitude of its wavelet

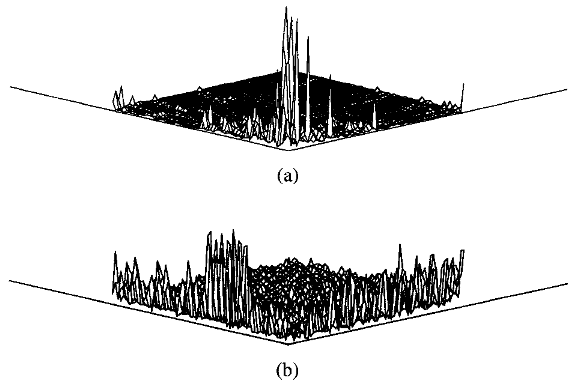

Fig. 2. Typical 2-D wavelet transform of (a) smooth and (b) textured images.

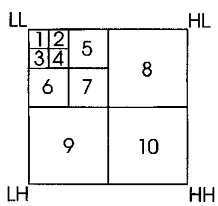

Fig. 3. Ten main channels of a three-level wavelet decomposition of an image.

coefficients. The energy in these channels was observed to vary quite differently, depending on whether the image was smooth or textured. Images in which the grey level varies smoothly are heavily dominated by the low-frequency channels in their wavelet transforms, as we would expect. However, textured images have large energies in both the low and middle frequencies. This is illustrated in Fig. 4(a), which shows the energy in the 10 main wavelet channels for a smooth image and a textured image (cloud and canvas again). These channels can be further grouped into low- (channels 1–4), middle- (channels 5–7), and high-frequency (channels 8–10) bands, as shown in Fig. 4(b). This clearly reflects the difference in frequency distribution between smooth and textured images. Hence, the ratio of the mean energy in the four low-frequency channels (1–4) to the mean energy in the three middle-frequency channels (5–7) is proposed as a criterion for optimal feature selection. If this ratio is above a certain threshold, the pixel or block of pixels is labeled as smooth; otherwise, it is labeled as textured. This ratio is given by

$$R = \frac{e_{C1} + e_{C2} + e_{C3} + e_{C4}}{e_{C5} + e_{C6} + e_{C7}}$$

where R is the ratio, and e_{Cn} is the energy in the nth channel given by

$$e_{Cn} = \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} |x(i,j)|$$

where the channel is of dimensions M by N (usually $M = N$) and x is a wavelet coefficient within the channel. The pixel is labeled as smooth if $R > T$ or textured if $R \leq T$, where T is the threshold.

Different features are then used for segmentation in different areas of the image. Local grey level mean and variance were found to be adequate features for the segmentation of smooth images and were used in the smooth areas of the image. The textured areas of the image require more complex spatial frequency-based features.

It was found that the energy values of the various channels of the 2-D wavelet transform varied considerably between different textures, depending on their dominant orientation and spatial frequencies. The energy values in the 10 channels of the wavelet transform of the local area were therefore used as features for segmentation in the textured areas of the image.

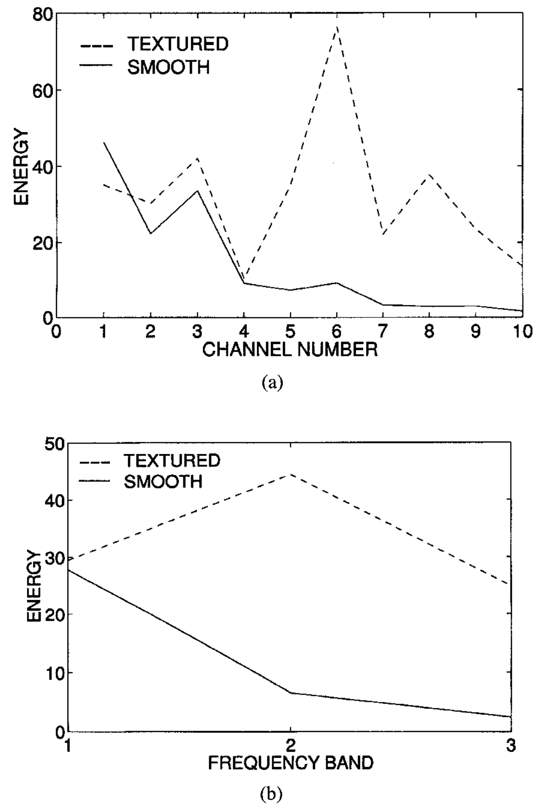

Fig. 4. Example of energy levels in wavelet channels for smooth and textured images: (a) Ten main wavelet channels; (b) grouped into low-, middle-, and high-frequency bands.

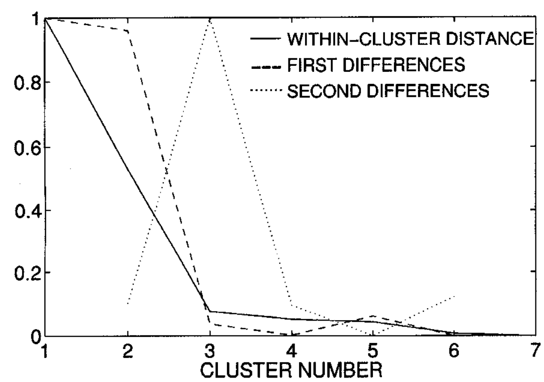

Fig. 5. True cluster number detection: Within-cluster distances (solid line), first difference of within-cluster distances (dashed line), and second difference of within-cluster distances (dotted line) all normalized here to a range of 0–1. The peak of the second differences indicates that the true cluster number is 3.

IV. Image Segmentation by Clustering

The *k-means* clustering technique involves grouping together those pixels in the image whose feature vectors represent points that are close together in the feature space. The final result is a number of clusters K, where each hopefully depicts a perceptually different region in the image. Each cluster can be represented by the mean

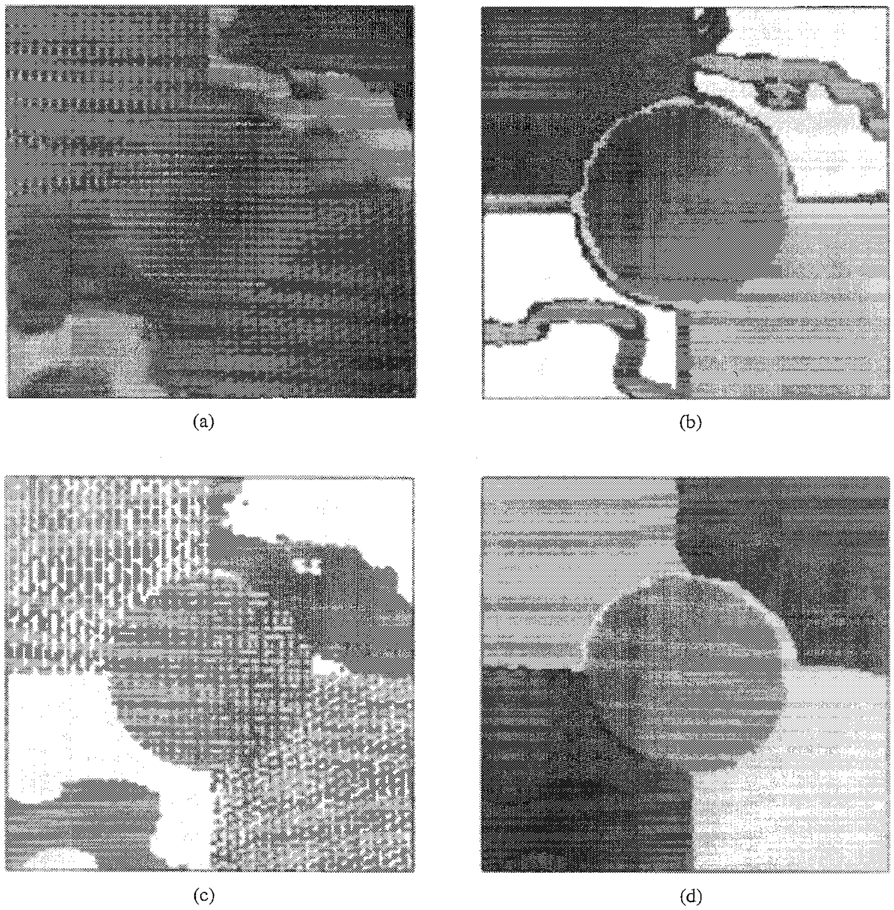

Fig. 6. (a) Original image; (b) segmentation using only wavelet features; (c) segmentation using only local grey level statistics as features; (d) segmentation using the proposed algorithm in which the optimal features are always used in different areas of the image.

feature vector of all its members (pixels or blocks of pixels) known as its cluster center $C_k, k \in \{1, \cdots, K\}$. To obtain good segmentation, the members of each cluster should have feature vectors lying as close as possible to the cluster center, and the cluster centers should be well separated.

Each pixel is labeled as being a member of the cluster center to which it is closest, using the distance function $d(C_k, X)$, where X is the pixel's feature vector. The distance function measures the distance between two points in the feature space and is simply the Euclidean distance here. The cluster centers are then recalculated as the mean feature vector of all their members:

$$C_k = \frac{1}{[g_k]} \sum_{s \in g_k} X_s \qquad k \in \{1, \cdots, K\}$$

where

- g_k kth cluster,
- s member of such a cluster,
- X_s feature vector,
- $[g_k]$ number of members in cluster g_k.

Pixels are then relabeled according to their new nearest cluster center, and the process is repeated until convergence to a stable partition of the feature space has occurred. This gives a segmentation result with a certain number of regions equal to the number of clusters K. The clustering routine is repeated several times: once for every possible number of clusters. The difficulty then lies in determining the optimal number of clusters for the segmentation, which is known as the true cluster number.

V. Automatic True Cluster Number Detection

One possible method for the determination of the true cluster number involves the calculation of the within-cluster distance for each cluster number. This is given by the sum of the distances of every pixel's feature vector from its associated cluster centre:

$$\text{Within-cluster distance, } W_k = \frac{1}{N} \sum_{k=1}^{K} \sum_{s \in g_k} d(C_k, X_s)$$

where N is the total number of members of all clusters.

The higher the value of the within-cluster distance, the less similarity there is between each pixel's feature vector and its associated cluster center; therefore, the segmentation will be worse. An example of how the within-cluster distance varies with cluster number is given in Fig. 5, where the solid line shows the actual within-cluster distances for the clustering of the textured areas of Fig. 6(a) giving the result shown in Fig. 6(d). Initially, the within-cluster distances are very high for only one cluster but decrease quickly as the number of clusters increases until the true cluster number is reached. They subsequently remain approximately constant or decrease very gradually. The true cluster number is thus taken to be the point at which the within-cluster distance becomes approximately constant (three in this case). This is usually done by calculating the differences between adjacent within-cluster distances (dashed line in Fig. 5). When this difference is suitably low, the true cluster number has been found. However, this involves the use of a threshold that is very difficult to set so that the optimal segmentation is always guaranteed. The proposed algorithm requires no threshold and is thus completely automatic.

It can be seen in Fig. 5 that the true cluster number occurs at the point where there is the largest change in gradient in the graph of within-cluster distance against cluster number. Therefore, the second derivative of this curve should exhibit a peak at this point. The true cluster number can thus be detected simply by finding the cluster number at which the maximum value of the second difference of within-cluster distance occurs, as the dotted line in Fig. 5 shows. This method is exceptionally simple and fully automatic and, therefore, never fails to provide a good estimate for the true cluster number with no need for a threshold or supervision.

VI. Results

The proposed algorithm was used to segment Fig. 6(a), which shows an image made up of both smooth and textured regions, taken from the Brodatz album [8]. The textures are oriental straw cloth (D53), French canvas (D21), and cotton canvas (D77), whereas the smooth regions are clouds (D91). This image was segmented using three approaches. The first two approaches used just one set of features (either grey level mean and variance features or wavelet features) for the whole image and performed the segmentation using a single run of the clustering algorithm described above. The third approach used different features in different areas of the image as proposed in Sections II and III and thus employed the clustering algorithm twice: once for each set of features.

The first approach used wavelet features for the entire image and resulted in the segmentation shown in Fig. 6(b). As can be seen, the algorithm successfully distinguished between the three textures but gave meaningless results in the smooth areas. This is because although the wavelet features are very good at differentiating between the very different spatial frequencies and orientations of the various textures, they cannot differentiate between the very similar low spatial frequencies that dominate smooth regions. In addition, the sharp boundaries between the smooth regions give rise to a number of higher frequencies, causing the wavelet transform of such boundary areas to be similar to that of a textured area. The use of wavelet features in these areas thus results in the boundaries between smooth regions being given the same classification as the textured regions during clustering, as can be seen in Fig. 6(b). Therefore, new features must be introduced that can accurately discriminate between different smooth regions without causing significant misclassifications at the boundaries—local grey level mean and variance.

The second approach using only the local grey level statistics as features successfully segmented the cloud from its background but gave meaningless results in the textured areas as shown in Fig. 6(c). This indicates that although these features are suitable for the segmentation of smooth images, they cannot distinguish between textures where the grey levels can vary in very different ways from texture to texture but still have the same local mean and variance. As was seen earlier, wavelet-based features are much better suited to texture segmentation.

The third approach was the proposed algorithm using both types of feature in the relevant areas as described in this correspondence and resulted in a much more meaningful segmentation (Fig. 6(d)). The three textures have been segmented correctly, and the cloud has been discriminated from its background. There are some small boundary errors, but these could be "cleaned up" using some form of postprocessing such as contextual information along with certain criteria on minimum region dimensions. More importantly, the feature selection algorithm had correctly identified the cloud areas as smooth regions, identified the three textures as textured regions, and then segmented the image accordingly. This clearly demonstrates the advantages of the proposed approach, which uses automatic feature selection to determine the optimal features for segmentation in each area of the image.

VII. Conclusion

Image segmentation routines too often concentrate only on texture segmentation or some very application-specific segmentation of smoother regions. Many real images are made up of both smooth and textured regions, and the segmentation technique must therefore incorporate features capable of describing these regions effectively. An algorithm that does this by using wavelet analysis to select the relevant features for each area of the image prior to segmentation has been proposed here. A clustering routine is then invoked to perform the segmentation for the smooth and then the textured areas of the image using different features in each case. One common difficulty with the *k-means* clustering algorithm is the determination of the optimal number of regions for segmentation: the so-called "true cluster number." A completely automatic method for true cluster number estimation using the second derivative of the within-cluster distances has been suggested here in which, unlike existing methods, no threshold settings are required.

References

[1] L. P. Clarke *et al.*, "MRI Segmentation: Methods and applications," *Magn. Resonance Imaging,* vol. 13, no. 3, pp. 343–368, 1995.

[2] M. Kunt, A. Ikonomopoulos, and M. Kocher, "Second-generation image-coding techniques," *Proc. IEEE,* vol. 73, no. 4, pp. 549–574, Apr. 1985.

[3] Y. M. Zhu and R. Goutte, "Analysis and comparison of space/spatial-frequency and multiscale methods for texture segmentation," *Opt. Eng.,* vol. 34, no. 1, pp. 269–282, Jan. 1995.

[4] A. K. Jain and F. Farrokhnia, "Unsupervised texture segmentation using Gabor filters," *Patt. Recogn.,* vol. 24, no. 12, pp. 1167–1186, Dec. 1991.

[5] Y. Hu and T. J. Dennis, "Textured image segmentation by context enhanced clustering," *Proc. Inst. Elec. Eng.—Vision, Image, Signal Processing,* vol. 114, no. 6, pp. 413–421, Dec. 1994.

[6] S. G. Mallat, "Multifrequency channel decompositions of images and wavelet models," *IEEE Trans. Acoustics, Speech, Signal Processing,* vol. 37, no. 12, pp. 2091–2110, Dec. 1989.

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

[8] P. Brodatz, *Textures: A Photographic Album for Artists and Designers.* New York: Dover, 1966.