PERGAMON

Pattern Recognition 32 (1999) 477–486

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

Unsupervised texture segmentation using feature distributions

Timo Ojala, Matti Pietikäinen

Machine Vision and Media Processing Group, Infotech Oulu, University of Oulu, FIN-90570 Oulu, Finland

Received 19 December 1997; in revised form 24 February 1998

Abstract

This paper presents an unsupervised texture segmentation method, which uses distributions of local binary patterns and pattern contrasts for measuring the similarity of adjacent image regions during the segmentation process. Non-parametric log-likelihood test, the *G* statistic, is engaged as a pseudo-metric for comparing feature distributions. A region-based algorithm is developed for coarse image segmentation and a pixelwise classification scheme for improving localization of region boundaries. The performance of the method is evaluated with various types of test images. © 1999 Pattern Recognition Society. Published by Elsevier Science Ltd. All rights reserved.

Keywords: Texture segmentation; Feature distribution; *G* statistic; Spatial operator; Local binary pattern; Contrast

1. Introduction

Segmentation of an image into differently textured regions is a difficult problem. Usually, one does not know *a priori* what types of textures exist in an image, how many textures there are, and what regions have which textures [1]. In order to distinguish reliably between two textures relatively large samples of them must be examined, i.e. relatively large blocks of the image. But a large block is unlikely to be entirely contained in a homogeneously textured region and it becomes difficult to correctly determine the boundaries between regions.

Many different approaches to image and texture segmentation have been proposed [2–4]. Segmentation methods are usually classified as region-based, boundary-based or as a hybrid of the two. The segmentation can be supervised or unsupervised. In unsupervised segmentation no *a priori* information about the textures present in the image is available. This makes it is a very challenging research problem in which only limited success has been achieved so far. Early methods proposed for unsupervised region-based texture segmentation include approaches based on split-and-merge methods [5], pyramid node linking [6], selective feature smoothing with clustering [7], and a quadtree method combining statistical and spatial information [8]. Examples of more recent approaches are methods based on local linear transforms and multiresolution feature extraction [9], feature smoothing and probabilistic relaxation [10], autoregressive models [11,12], Markov random field models [13–16], multichannel filtering [17–19], neural network-based generalization of the multichannel approach [20], wavelets [21,22], fractal dimension [23], and hidden Markov models [24]. A method for unsupervised segmentation of color textures using Markov random fields and a split-and-merge type algorithm was proposed by Panjwani and Healey [25].

Some of the existing methods perform reasonably well for a small set of fine-grained texture mosaics, but they usually need some prior knowledge of the image contents to achieve satisfactory results, like the number of textures or regions. The choice of proper parameters for different types of images may also be difficult and the methods typically perform poorly for natural images containing nonuniform textures. Unsupervised segmentation of images containing texture primitives at very different scales may even be unrealistic, because it is hard to discriminate small image regions from large texture primitives without any prior knowledge.

The choice of highly discriminating texture features is the most important factor for a success in texture segmentation, but this has been neglected in most earlier approaches. The features should easily discriminate various types of textures and the window size used for computing textural features should be small enough to be useful for small image regions and to provide small error rates at region boundaries.

Our recent studies show that excellent texture discrimination can be obtained with local texture operators and nonparametric statistical discrimination of sample and prototype distributions. Texture classification results obtained by using distributions of local binary patterns (LBP) or gray-scale differences have been better than those obtained with the existing methods [26–29]. Our method can be easily generalized to utilize multiple texture features, multiscale information, color features and combinations of multiple features using the new multichannel approach to texture description [29].

This paper presents an efficient method for unsupervised texture segmentation based on texture description with feature distributions. A region-based algorithm is developed for coarse image segmentation and a pixelwise classification scheme for improving the localization of region boundaries.

2. Texture description

The texture contents of an image region are characterized by the joint distribution of local binary pattern (LBP) and contrast (C) features [27]. The original 3×3 neighborhood (Fig. 1a) is thresholded by the value of the center pixel. The values of the pixels in the thresholded neighborhood (Fig. 1b) are multiplied by the binomial weights given to the corresponding pixels (Fig. 1c) and obtained values (Fig. 1d) are summed for the LBP number (169) of this texture unit. By definition LBP is invariant to any monotonic gray-scale transformation. LBP describes the spatial structure of the local texture, but it does not address the contrast of the texture. For this purpose we combine LBP with a simple contrast measure C, which is the difference between the average gray-level of those pixels which have value 1 and those which have value 0 (Fig. 1b).

The LBP/C distribution is approximated by a discrete two-dimensional histogram of size $256 \times b$, where b is the number of bins for C. Choosing b is a trade-off between the discriminative power and the stability of the texture transform. If b is too small, the histogram will lack resolution and feature C will add very little discriminative information to the process. However, since the image region contains a finite number of pixels, it does not make sense to go to the other extreme, for then the histogram becomes sparse and unstable. Based on the results of our past texture classification experiments with the LBP/C transform, we chose to use 8 bins, although we expect to achieve comparable results with 4 or 16 bins as well. See Ref. [27] for a detailed description of the mapping from the continuous C space to the discrete bin index.

A log-likelihood-ratio, the G statistic [30], is used as a pseudo-metric for comparing LBP/C distributions. The value of the G statistic indicates the probability that the two sample distributions come from the same population: the higher the value, the lower the probability that the two samples are from the same population. We measured the similarity of two histograms with a two-way test of interaction or heterogeneity:

$$G = 2\left[\sum_{s,m}\sum_{i=1}^{n} f_i \log f_i\right] - \left[\sum_{s,m}\left(\sum_{i=1}^{n} f_i\right)\log\left(\sum_{i=1}^{n} f_i\right)\right] - \left[\sum_{i=1}^{n}\left(\sum_{s,m} f_i\right)\log\left(\sum_{s,m} f_i\right)\right] + \left[\left(\sum_{s,m}\sum_{i=m}^{n} f_i\right)\log\left(\sum_{s,m}\sum_{i=1}^{n} f_i\right)\right]\Bigg) \quad (1)$$

where s, m are the two sample histograms, n is the number of bins and f_i is the frequency at bin i. The more alike the histograms s and m are, the smaller is the value of G.

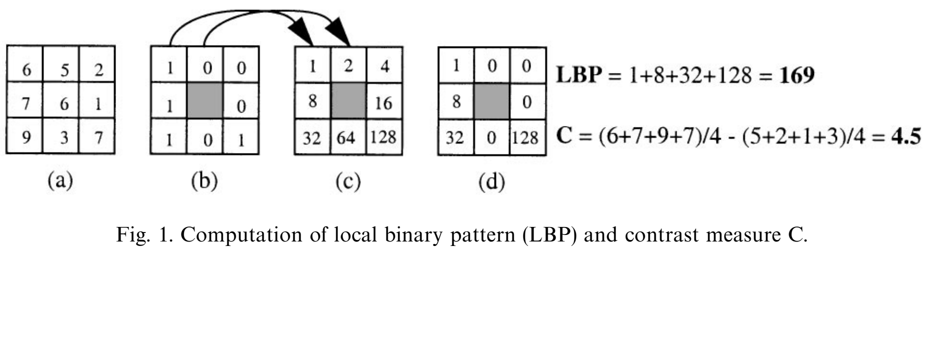

Fig. 1. Computation of local binary pattern (LBP) and contrast measure C.

3. Segmentation algorithm

The segmentation method consists of three phases: hierarchical splitting, agglomerative merging and pixelwise classification. First, hierarchical splitting is used to divide the image into regions of roughly uniform texture. Then, agglomerative merging procedure merges similar adjacent regions until a stopping criterion is met. At this point we have obtained rough estimates of the different textured regions present in the image and complete the analysis by a pixelwise classification to improve the localization. Fig. 2 illustrates the progress of the segmentation algorithm on a 512×512 mosaic containing five different Brodatz [31] textures.

3.1. Hierarchical splitting

A necessary prerequisite for the agglomerative merging to be successful is that the individual image regions are uniform in texture. For this purpose we apply the hierarchical splitting algorithm, which recursively splits the original image into square blocks of varying size. The decision whether a block is split to four subblocks is based on a uniformity test. We measure the six pairwise G distances between the LBP/C histograms of the four subblocks. If we denote the largest of the six G values by $G_{\max}$ and the smallest by $G_{\min}$, the block is found to be nonuniform and is thus split further into four subblocks, if a measure of relative dissimilarity within region is greater than a threshold

$$R = \frac{G_{\max}}{G_{\min}} > X. \tag{2}$$

Regarding the proper choice of X, one should rather choose a too small value for X instead of a too large one. It is better to split too much than too little, for the following agglomerative merging procedure is able to correct errors, where an uniform block of a single texture has been needlessly split. But error recovery is not possible, if segments containing several textures are assumed to be uniform. Threshold X was experimentally set to value 1.2. We computed $G_{\max}$ and $G_{\min}$ [Eq. (2)] for numerous blocks with varying texture contents, and a 20% difference generally indicated a deviation in the local texture.

To begin with, we divide the image into rectangular blocks of size $S_{\max}$. If we applied the uniformity test on arbitrarily large image segments, we could fail to detect small texture patches and end up treating regions containing several textures as uniform. The next step is to use the uniformity test. If a block does not satisfy the test, it is divided into four subblocks. This procedure is repeated recursively on each subblock until a predetermined minimum block size $S_{\min}$ is reached. It is necessary to set a minimum limit for the block size, for the block has to contain a sufficient number of pixels for the LBP/C histogram to be reliable. Since it is fundamental to make reliable merges in the early stages of the merging process, we decided to use the relatively large value of 16 for $S_{\min}$. Comparable results were obtained with value 8, whereas histograms of 4×4 blocks turned out to be too noisy in some cases. The choice of $S_{\max}$ is less crucial, and we chose to use 64. Fig. 2b illustrates the result of the hierarchical splitting algorithm. As expected, the splitting goes deepest around the texture boundaries.

Note that the hierarchical splitting phase is not mandatory, but we could skip it by dividing the input image directly to blocks of size $S_{\min}$ and the successive agglomerative merging phase would still succeed. This is particularly true for easier problems of homogeneous and clearly distinct textures. However, our experiments have shown that finding larger areas of uniform texture with the hierarchical splitting method improves the convergence of the agglomerative merging algorithm.

3.2. Agglomerative merging

Once the image has been split into blocks of roughly uniform texture, we apply an agglomerative merging

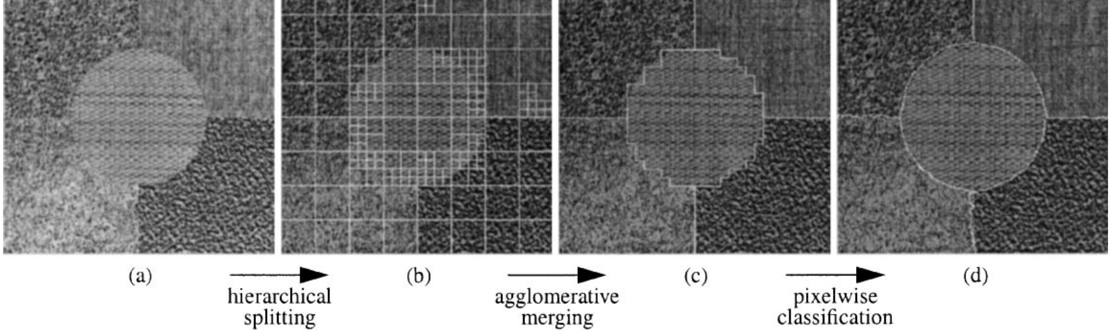

Fig. 2. Texture mosaic #1; the main sequence of the proposed segmentation algorithm.

procedure, which merges similar adjacent regions until a stopping criterion is satisfied. At a particular stage of the merging, we merge that pair of adjacent segments, which has the smallest merger importance (MI) value. MI is defined as

$$MI = p \times G, \tag{3}$$

where p is the number of pixels in the smaller of the two regions and G is the distance measure defined in Eq. (1). In other words, at each step the procedure chooses that merger of all possible mergers, which introduces the smallest change in the segmented image. Once the pair of adjacent segments with the smallest MI value has been found, the regions are merged and the two respective LBP/C histograms are summed to be the histogram of the new image region. Before moving to the next merger we compute the G distances between the new region and all adjacent regions to it. Merging is allowed to proceed until the stopping rule

$$MIR = \frac{MI_{\text{cur}}}{MI_{\text{max}}} > Y \tag{4}$$

triggers. Merging is halted if MIR, the ratio of MI_{cur}, merger importance for the current best merge, and MI_{max}, the largest merger importance of all preceding mergers, exceeds a preset threshold Y. In theory, it is possible that the very first merges have a zero MI value (i.e. there are adjacent regions with identical LBP/C histograms), which would lead to a premature termination of the agglomerative merging phase. To prevent this the stopping rule is not evaluated for the first 10% of all possible merges.

The value of threshold Y was determined experimentally. We applied the algorithm to numerous texture images and examined the values of MIR during the merging process. The conclusion was that for well-defined homogeneous textures MIR values up to 1.5 or 1.6 were still acceptable, while values over 2.0 generally were due to a perceivable difference in the texture contents of the adjacent regions of the current best merge. Based on this observation we chose value 2.0 for Y. In more general terms, threshold Y can be interpreted as the scale of texture differences we want to discriminate, and thus the value of Y may be a very subjective decision. This is particularly the case with outdoor scenes and irregular textures, where the number of distinct regions in the segmentation result strongly correlates with the threshold.

Fig. 2c shows the result of the agglomerative merging phase after 174 merges. The MIR of the 175th merge (MIR_{stop}) is 9.5 and the merging is halted. The highest MIR value up to that point (MIR_{hi}) had been 1.2 (Fig. 3). The relationship between MIR_{stop}, MIR_{hi} and threshold Y reflects the reliability of the result of the agglomerative merging phase. The very large value of MIR_{stop} and very

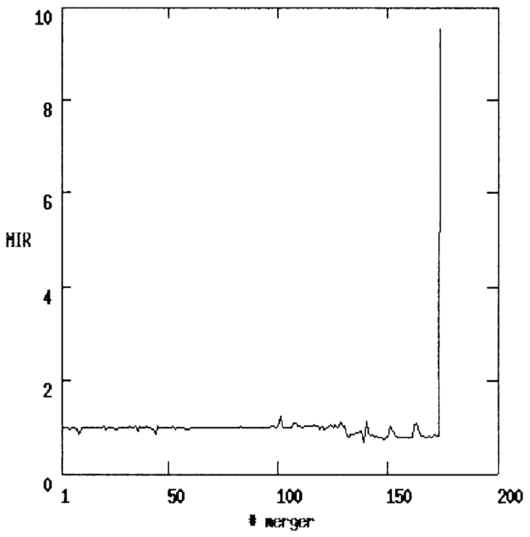

Fig. 3. Plot of MIR.

small value of MIR_{hi} underline the easiness with which the rough estimate of the texture regions is obtained for mosaic #1. Note that the segmentation error of 1.4% after the agglomerative clustering phase (ERR_{a}) is somewhat biased in this problem, for the horizontal and vertical texture boundaries are accidentally aligned with the initial blocks.

3.3. Pixelwise classification

To improve the localization of the boundaries a simple pixelwise classification algorithm is used. If the hierarchical splitting and agglomerative merging phases have succeeded, we have obtained quite reliable estimates of the different textured regions present in the image. Treating the LBP/C histograms of the image segments as our texture models we switch into a texture classification mode. If an image pixel is on the boundary of at least two distinct textures (i.e. the pixel is 4-connected to at least one pixel with a different label), we place a discrete disc with radius r on the pixel and compute the LBP/C histogram over the disc. We compute the G distances between the histogram of the disc and the models of those regions, which are 4-connected to the pixel in question. We relabel the pixel, if the label of the nearest model is different from the current label of the pixel and there is at least one 4-connected adjacent pixel with the tentative new label. The latter condition improves smooth adaption of texture boundaries and decreases the probability of small holes occurring inside the regions. If the pixel is relabeled, i.e. it is moved from an image segment to the adjacent segment, we update the corresponding texture models accordingly, hence the texture models become more accurate during the process. Only those pixels at which the disc is entirely inside the image are examined, hence the final segmentation result will contain a border of r pixels wide.

In the next scan over the image we only check the neighborhoods of those pixels, which were relabeled in the previous sweep. The process of pixelwise classification continues until no pixels are relabeled or maximum number of sweeps is reached. This is set to be two times $S_{\min}$, based on the reasoning that the boundary estimate of the agglomerative merging phase can be at most this far away from the "true" texture boundary. Setting an upper limit for the number of iterations ensures that the process will not wander around endlessly, if the disc is not able to capture enough information of the local texture to be stable. According to our experiments the algorithm generally converges quickly with homogeneous textures, whereas with locally stochastic natural scenes maximum number of sweeps may be consumed. We did not apply any postprocessing method to improve the final segmentation result, e.g. by smoothing the texture boundaries or removing small regions as many existing algorithms do.

The relationship between the radius r of the disc and the final segmentation result is obvious. A very small disc is unstable, producing ragged texture boundaries and holes inside regions, whereas a very large disc is stable and produces smooth boundaries, but may fail in locating the boundaries accurately. We used the final segmentation error as a guide-line in choosing the value of r. Mosaics #4 (Fig. 7) and #5 (Fig. 8) were processed with the pixelwise classification algorithm, with r ranging from 1 to 20. The segmentation errors are plotted as a function of r in Fig. 4. As expected, the error first decreases with increasing disc size, reaches the minimum and then slowly increases as the disc becomes too large to locate the boundaries accurately. Based on this result we chose value 11 for radius r.

Fig. 2d shows the final segmentation result after the pixelwise classification phase, where 16 sweeps were needed. The final segmentation error (ERR_p), computed over the area processed by the disc which excludes the border of r pixels, is 1.7%.

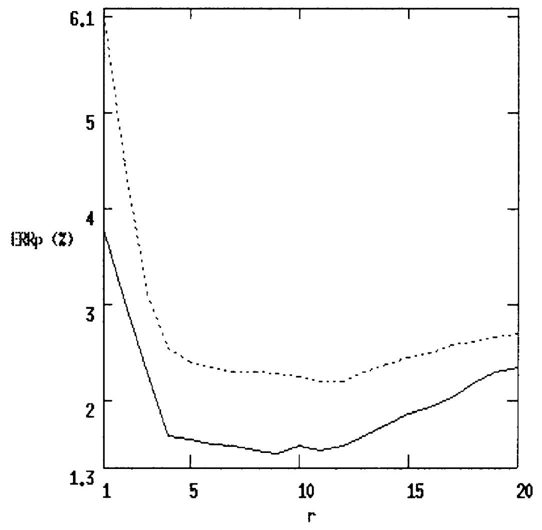

Fig. 4. Final segmentation error as a function of the radius r of the disc. The solid line corresponds to mosaic #5 and the dashed line corresponds to mosaic #6.

Table 1
Parameters describing segmentation of mosaics #1 to #5

Mosaic	MIR_{stop}	MIR_{hi}	ERR_a(%)	ERR_p(%)	Sweeps
#1	9.5	1.2	1.4	1.7	16
#2	5.2	1.6	4.2	1.2	23
#3	8.1	1.2	4.6	1.9	13
#4	4.1	1.3	2.0	1.4	9
#5	2.8	1.2	7.8	2.1	24

4. Experimental results

The segmentation results for four additional texture mosaics and two natural scenes are presented. The same set of parameter values was used for all texture mosaics to demonstrate the robustness of the approach: $b = 8$, $S_{\max} = 64$, $S_{\min} = 16$, $X = 1.2$, $Y = 2.0$, and $r = 11$. In each case we provide the original image, the rough segmentation result after the agglomerative merging phase and the final segmentation result after the pixelwise classification phase. The segmentation results are superpositioned on the original image. Parameters describing segmentation of texture mosaics are given in Table 1.

Mosaic #2 (Fig. 5a) is a 512 × 512 image containing four textures made by a GMRF process and a circle of painted surface in the middle [32]. The more difficult nature of this problem shows in the values of MIR_{stop} (5.2) and MIR_{hi} (1.6), which are clearly closer to threshold Y than what was the case with mosaic #1. Nevertheless, the rough segmentation result (Fig. 5b) with segmentation error of 4.2% is quite decent. The final segmentation result (Fig. 5c) after 23 sweeps with segmentation error of 1.2% is excellent.

Mosaic #3 (Fig. 6a) is a 512 × 512 image with a background made by a GMRF process and four distinct regions; the square and the circle are painted surfaces with different surface roughnesses and the ellipse and the triangle are made by a fractal process [32]. As we can see from the values of MIR_{stop} (8.1) and MIR_{hi} (1.2), the rough estimate (Fig. 6b) of the texture regions is obtained relatively easily. The final segmentation result (Fig. 6c) after 13 sweeps contains only 1.9% misclassified pixels.

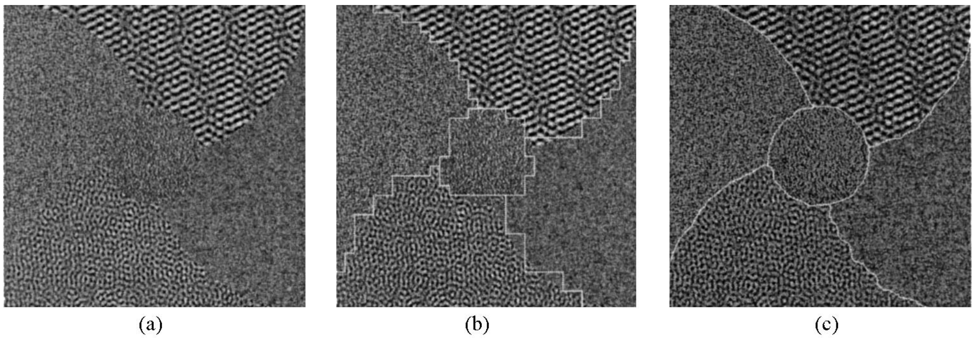

Fig. 5. Texture mosaic #2.

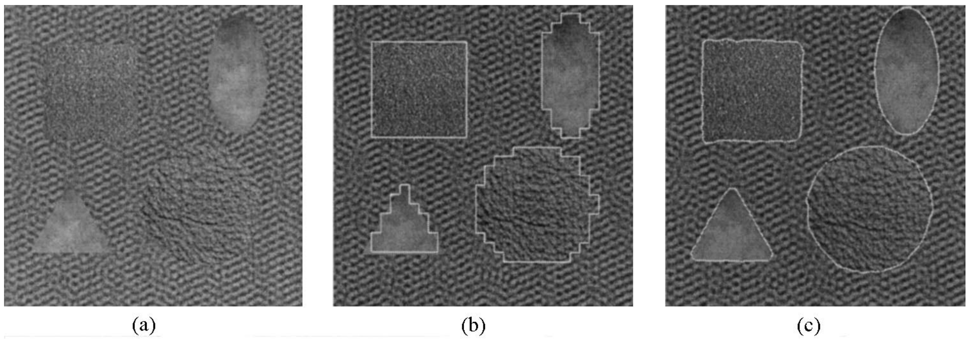

Fig. 6. Texture mosaic #3.

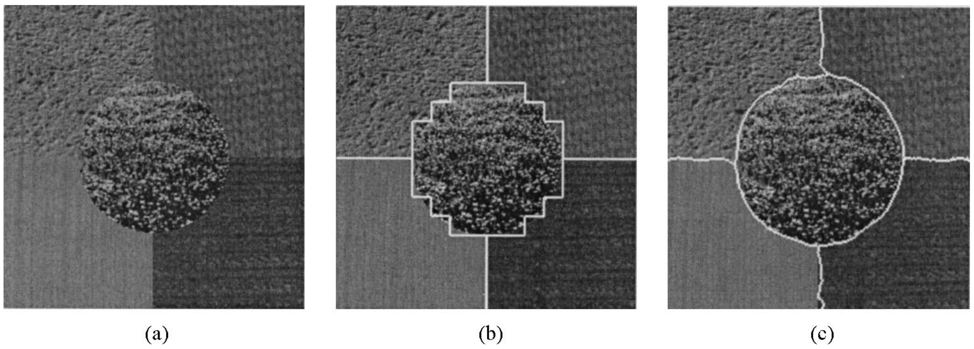

Fig. 7. Texture mosaic #4.

Mosaics #4 (Fig. 7a) and #5 (Fig. 8a) are composed of textures taken from outdoor scenes [20]. In their study, Jain and Karu tackled the problem of texture segmentation with a neural network generalization of the traditional multichannel filtering method, using various filter banks for feature extraction. For mosaic #4, which is 256×256 pixels in size, they obtained a segmentation error of 3.3% with learned masks in unsupervised mode. Their method was not strictly unsupervised, though, because the number of clusters was manually set to five. Our method achieves a smaller misclassification error of 1.4% (Fig. 7c). The difference between MIR_{stop} (4.1) and MIR_{hi} (1.3) is considerable, which reflects the reliability of the analysis.

For mosaic #5, which is 384×384 pixels in size, Jain and Karu reported a labeling error of 6% with Laws' filters in supervised mode. Our unsupervised method gives a clearly better segmentation result of 2.1%. Note that the pixelwise classification clearly improves the result of the agglomerative merging phase (7.8%). The

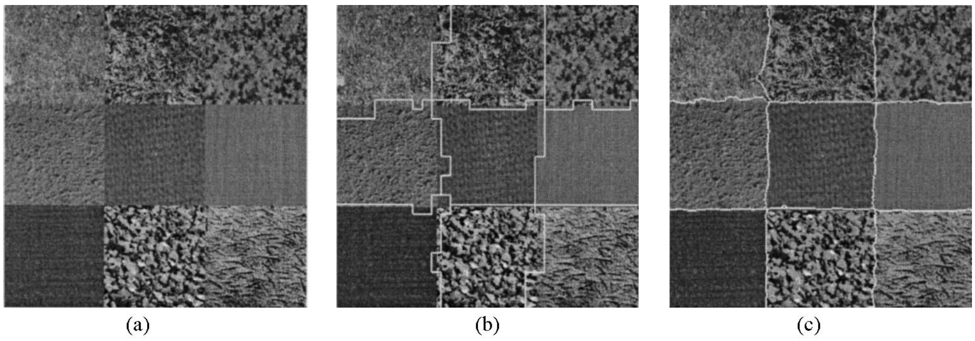
Fig. 8. Texture mosaic #5.

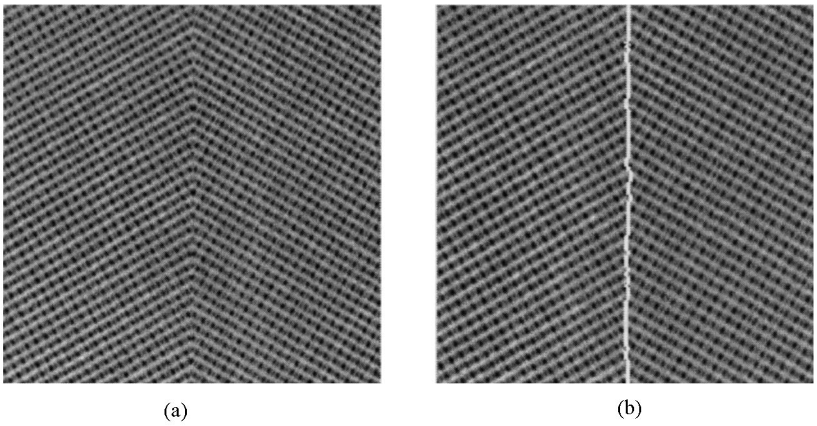
Fig. 9. Segmentation of a mosaic which contains two different orientations of a directional texture.

difference between MIR_{stop} (2.8) and MIR_{hi} (1.2) is still noticeable, but by far the smallest in the three cases, reflecting the inherent difficulty of this problem.

LBP/C transform is by definition rotation-variant. Fig. 9 demonstrates how the segmentation algorithm works in the case of an edge between two different orientations of a directional texture. The original texture (D21 from the Brodatz album) was rotated 30° in both clockwise and counterclockwise direction using cubic interpolation, and the rotated textures were merged into the 200×200 mosaic shown in Fig. 9a. This size guarantees that the edge between the two orientations is not accidentally aligned with the initial blocks which could bias the result. The same set of parameter values was used as with texture mosaics #1 to #5. The final segmentation result in Fig. 9b contains 66 mislabelled pixels along the edge.

We also applied the texture segmentation method to natural scenes. The scenes were originally in RGB format [25], but we converted them to gray-level intensity images. As an example, scene #1 (Fig. 10a) is a 384×384 image of rocks in the sea and scene #2 (Fig. 11a) is a 192×192 image of a beach, water and foliage. As we can observe from the image, the textures of natural scenes are generally more nonuniform than the homogeneous textures of the test mosaics. Also, in natural scenes adjacent textured regions are not necessarily separated by well-defined boundaries, but the spatial pattern smoothly changes from one texture to another. Further, we have to observe the infinite scale of texture differences present in natural scenes; choosing the right scale is a very subjective matter. For these reasons there is often no 'correct' segmentation for a natural scene, as is the case with texture mosaics.

The parameters X and Y primarily control the scale of texture differences that will be detected. With values $X = 1.1$ and $Y = 1.5$ the rough segmentation results after the agglomerative merging phase are presented in Figs. 10b and 11b, and the final segmentation results are shown in Figs. 10c and 11c, respectively. If we decreased Y further, the segmentation result would contain an increasing number of regions. The invariance of the LBP/C transform to average gray-level shows in the bottom part of the image, where the sea is interpreted as a single region despite the shadows.

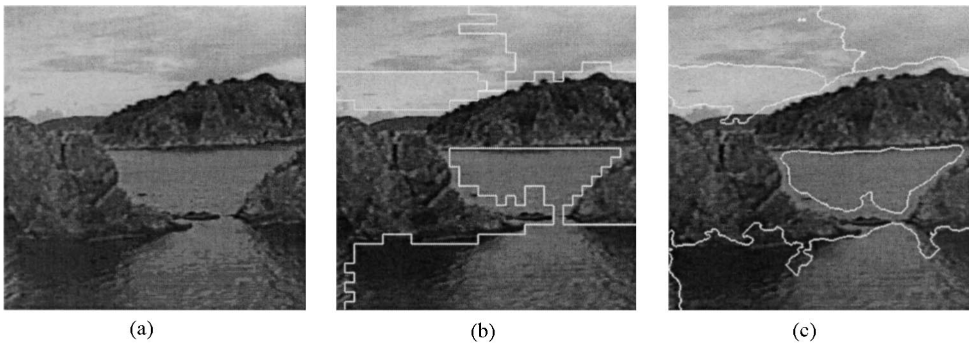

Fig. 10. Natural scene #1.

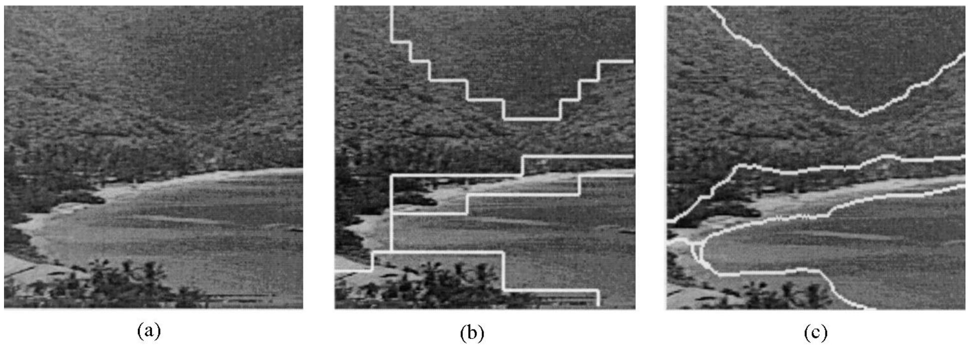

Fig. 11. Natural scene #2.

The results obtained for these natural scenes are very satisfactory, considering that important color or gray scale information is not utilized in the segmentation.

5. Discussion

In the presented method texture is described by joint occurrences of LBP and C. Obvious generalizations are to use other texture features or feature domains (e.g. color) and scale. Although LBP/C is a very powerful texture transform, we expect to achieve better results by combining a larger number of features in the analysis. Other powerful texture measures, like distributions based on gray-level difference histograms or co-occurrence matrices, can be easily incorporated into our algorithm. In Pietikäinen, Nieminen, Marszalec and Ojala [33] we demonstrated that a method based on comparison of feature distributions can be used for high-accuracy color measurements. This suggests that distributions of color features could be easily used to find small color differences between neighboring regions in segmentation. Color features should make our method efficient for segmenting images containing color textures, like the original color images used in the experiments [25]. Further, we could consider a particular feature at multiple scales, by straightforwardly computing the desired feature for suitably symmetrical discrete neighborhoods of any size, such as disks or boxes of odd or even size. A simple way to define a "multiresolution" LBP would be to choose the eight neighbors of the center pixel from the corresponding positions in different neighborhoods (3×3, 5×5, 7×7, etc.).

The remaining question is how to combine the multiple feature channels obtained with several features and/or scales. We can hardly expect to reliably estimate joint distributions for a large number of features. Also, multidimensional histograms with large numbers of bins are very computationally intensive and consume very much memory. An alternative is to use an approximation with marginal distributions and to employ each independent feature separately, as a 1-D histogram, to compute a similarity score such as G for each feature, and then integrate individual scores into an aggregate similarity score. This approach has given very promising results in our texture classification experiments [29] and more recently in color classification as well [33]. Combining it with a carefully chosen set of non-redundant complementary features we expect to improve the performance of our segmentation method considerably. In a similar way, joint pairs of features, like LBP/C, can be combined with other single features or feature pairs. It would also be possible to use single features or joint features one by one, by e.g. first comparing the uniformity of regions with respect to texture and then with respect to color.

The histogram comparison approach based on the *G* test could be replaced with some other related method, like histogram intersection [34] or a statistical chi-square test. According to our experience the choice of proper texture measures is usually a much more important factor in texture discrimination than the particular method used for histogram comparison [29]. However, it would be interesting to study the performances of different types of approaches in the case of very small image windows.

In recent texture classification studies [35], we have compared the performance of LBP and other operators to that of GMRF and Gabor energy features. The image data included both Brodatz textures and the many different texture images available at MeasTex [36] which is an independent texture classification algorithm evaluation site accessible in WWW. LBP did well in these classification experiments, hence it should be suitable for texture segmentation as well.

6. Conclusion

We proposed a solution to unsupervised texture segmentation, in which a method based on comparison of feature distributions is used to find homogeneously textured image regions and to localize boundaries between regions. Texture information is measured with a method based on local binary patterns and contrast (LBP/C) that we have recently developed. A region-based algorithm is developed for coarse image segmentation and a pixelwise classification scheme for improving the localization of region boundaries.

The method performed very well in experiments. It is not sensitive to the selection of parameter values, does not require any prior knowledge about the number of textures or regions in the image, and seems to provide significantly better results than existing unsupervised texture segmentation approaches. The method can be easily generalized, e.g. to utilize other texture features, multiscale information, color features, and combinations of multiple features.

Acknowledgements

The financial support provided by the Academy of Finland, the Technology Development Center of Finland, and the Graduate School in Electronics, Telecommunications and Automation is gratefully acknowledged. The authors also wish to thank following persons for providing images used in this study; Richard C. Dubes, Anil K. Jain, John Lees and Kalle Karu from the Michigan State University and Glenn Healey and David Slater from the University of California at Irvine.

References

[1] M. Tuceryan, A.K. Jain, Texture analysis, in Handbook of Pattern Recognition and Computer Vision, C.H. Chen, L. F. Pau, P.S.P. Wang, eds., Ch. 2.1, pp. 235–276. World Scientific, Singapore, 1993, pp. 235–276.
[2] R.M. Haralick, L.G. Shapiro, Image segmentation techniques, Computer Vision Graphics, Image Processing 29 (1985) 100–132.
[3] T.R. Reed, J.M.H. Du Buf, A review of recent texture segmentation, feature extraction techniques, CVGIP Image Understanding 57 (1993) 359–372.
[4] N.R. Pal, S.K. Pal, A review on image segmentation techniques, Pattern Recognition 26 (1993) 1277–1294.
[5] P.C. Chen, T. Pavlidis, Segmentation by texture using a co-occurrence matrix and a split-and-merge algorithm, Comput. Graphics Image Processing 10 (1979) 172–182.
[6] M. Pietikäinen, A. Rosenfeld, Image segmentation by texture using pyramid node linking, IEEE Trans. Systems Man Cybernet. 11 (1981) 822–825.
[7] L.S. Davis, A. Mitiche, Mites: A model driven, iterative texture segmentation algorithm, Comput. Graphics Image Processing 19 (1982) 95–110.
[8] M. Spann, R. Wilson, A quad-tree approach to image segmentation which combines statistical and spatial information, Pattern Recognition 18 (1985) 257–269.
[9] M. Unser, M. Eden, Multiresolution feature extraction and selection for texture segmentation, IEEE Trans. Pattern Anal. Machine Intelligence 11 (1989) 717–728.
[10] J.Y. Hsiao, A.A. Sawchuk, Unsupervised texture image segmentation using feature smoothing and probabilistic relaxation techniques, Comput. Vision Graphics Image Processing 48 (1989) 1–21.
[11] A. Khotanzad, A. Bouarfa, A parallel non-parametric clustering algorithm with application to image segmentation, Proc. 22nd Asilomar Conf. on Signals, Systems, and Computers, pp. 305–309, Pacific Grove, CA 1988.
[12] J. Mao, A.K. Jain, Texture classification and segmentation using multiresolution simultaneous autoregressive models, Pattern Recognition 25 (1992) 173–188.
[13] J.F. Silverman, D.B. Cooper, Bayesian clustering for unsupervised estimation of surface and texture models, IEEE Trans. Pattern Anal. Machine Intelligence 10 (1988) 482–495.
[14] C. Bouman, B. Liu, Multiple resolution segmentation of textured images, IEEE Trans. Pattern Anal. Machine Intelligence 13 (1991) 99–113.
[15] B.S. Manjunath, R. Chellappa, Unsupervised texture segmentation using Markov random field models, IEEE Trans. on Pattern Anal. Machine Intelligence 13 (1991) 478–482.
[16] F.S. Cohen, Z. Fan, Maximum likelihood unsupervised textured image segmentation, CVGIP: Graphical Models and Image Processing 54 (1992) 239–251.

[17] A. Bovik, M. Clark, W.S. Geisler, Multichannel texture analysis using localized spatial filters, IEEE Trans. Pattern Anal. and Machine Intelligence 12 (1990) 55–73.

[18] A.K. Jain, F. Farrokhnia, Unsupervised texture segmentation using Gabor filters, Pattern Recognition 24 (1991) 1167–1186.

[19] H. Greenspan, R. Goodman, R. Chellappa, C.H. Anderson, Learning texture discrimination rules in a multiresolution system, IEEE Trans. Pattern Anal. Machine Intelligence 16 (1994) 894–901.

[20] A.K. Jain, K. Karu, Learning texture discrimination masks, IEEE Trans. Pattern Anal. Machine Intelligence 18 (1996) 195–205.

[21] M. Unser, Texture classification and segmentation using wavelet frames, IEEE Trans. Image Processing 4 (1995) 1549–1560.

[22] A. Laine, J. Fan, Frame representations for texture segmentation, IEEE Trans. Image Processing 5 (1996) 771–780.

[23] B.B. Chaudhuri, N. Sarkar, Texture segmentation using fractal dimension, IEEE Trans. Pattern Anal. Machine Intelligence 17 (1995) 72–77.

[24] J.-L. Chen, A. Kundu, Unsupervised texture segmentation using multichannel decomposition and hidden Markov models, IEEE Trans. Image Processing 4 (1995) 603–619.

[25] D.K. Panjwani, G. Healey, Markov random field models for unsupervised segmentation of textured color images, IEEE Trans. Pattern Anal. Machine Intelligence 17 (1995) 939–954.

[26] D. Harwood, T. Ojala, M. Pietikäinen, S. Kelman, L.S. Davis, Texture classification by center-symmetric auto-correlation, using Kullback discrimination of distributions, Pattern Recognition Lett. 16 (1995) 1–10.

[27] T. Ojala, M. Pietikäinen, D. Harwood, A comparative study of texture measures with classification based on feature distributions, Pattern Recognition 29 (1996) 51–59.

[28] T. Ojala, M. Pietikäinen, J. Nisula, Determining composition of grain mixtures by texture classification based on feature distributions, Int. J. Pattern Recognition Artif. Intelligence 10 (1996) 73–82.

[29] T. Ojala, Multichannel approach to texture description with feature distributions, Technical Report CAR-TR-846, Center for Automation Research, University of Maryland, 1996.

[30] R.R. Sokal, F.J. Rohlf, Introduction to Biostatistics, 2nd edn. W.H. Freeman and Co, New York, 1987.

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

[32] P.P. Ohanian, R.C. Dubes, Performance evaluation for four classes of textural features, Pattern Recognition 25 (1992) 819–833.

[33] M. Pietikäinen, S. Nieminen, E. Marszalec, T. Ojala, Accurate color discrimination with classification based on feature distributions, Proc. 13th Int. Conf. on Pattern Recognition, Vol. 3, pp. 833–838, Vienna, Austria, 1996.

[34] M. Swain, D. Ballard, Color indexing, Int. J. Computer Vision 7 (1991) 11–32.

[35] T. Ojala, Nonparametric texture analysis using spatial operators, with applications in visual inspection, Ph.D. thesis, Acta Univ. Oul. C 105, University of Oulu, Finland, 1997.

[36] G. Smith, I. Burns, MeasTex Image Texture Database and Test Suite, CSSIP, University of Queensland, Australia. http://www.cssip.elec.uq.edu.au/·guy/meastex/home.html.

About the Author—TIMO OJALA received the M.S. degree in Electrical Engineering with honors from the University of Oulu, Finland, in 1992, and the Dr. Tech. degree from the same university in 1997. He is a member of Machine Vision and Media Processing Group at the University of Oulu. From 1996 to 1997 he was visiting the University of Maryland Institute for Advanced Computer Studies (UMIACS). His research interests includes pattern recognition, texture analysis and object-oriented software design.

About the Author—MATTI PIETIKÄINEN received his Doctor of Technology degree in Electrical Engineering from the University of Oulu, Finland, in 1982. Currently he is Professor of Information Technology, Scientific Director of Infotech Oulu—a center for information technology research, and Head of Machine Vision and Media Processing Group, at the University of Oulu. From 1980 to 1981 and from 1984 to 1985 he was visiting the Computer Vision Laboratory at the University of Maryland, USA. His research interests include machine vision, document analysis, and their applications. He has authored about 100 papers in journals, books and conferences. He is the editor (with L.F. Pau) of the book "Machine Vision for Advanced Production", published by World Scientific in 1996. Prod. Pietikäinen is Fellow of International Association for Pattern Recognition (IAPR) and Senior Member of IEEE, and serves as Member of the Governing Board of IAPR and Chairman of IAPR Education Committee. He also serves on program committees of several international conferences.