Unsupervised Texture Segmentation Using Feature Distributions

Timo Ojala and Matti Pietikäinen
Machine Vision and Media Processing Group, Infotech Oulu
University of Oulu, FIN-90570 Oulu, Finland
ojala@ee.oulu.fi, mkp@ee.oulu.fi

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. Nonparametric 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. The same set of parameter values is used in all the experiments with texture mosaics in order to demonstrate the robustness of our approach.

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 texture segmentation have been proposed. Segmentation methods are usually classified as region-based, boundary-based or as a hybrid of the two. For surveys of image and texture segmentation techniques, see [2,3]. 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. Examples of different approaches to unsupervised segmentation are presented, e.g., in [4-10].

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 [11-14]. 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 presented in [14].

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 [12]. The original 3x3 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).

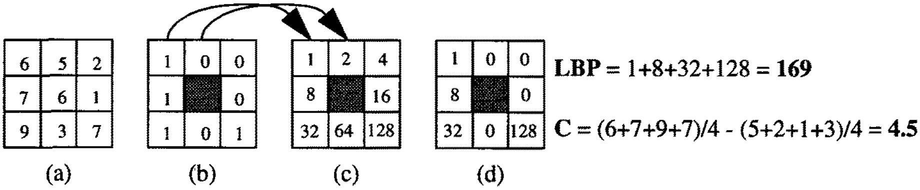

Fig. 1. Computation of Local Binary Pattern (LBP) and contrast measure C.

The LBP/C distribution is approximated by a discrete two dimensional histogram of size 256x*b*, where *b* is the number of bins for C. A log-likelihood-ratio, the *G* statistic [15], 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 independence:

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

where *s, m* are the two sample histograms, *n* is the number of bins and f_i is the frequency at bin *i*. See [15] for a detailed derivation of the formula.

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 512x512 mosaic containing five different Brodatz [16] textures.

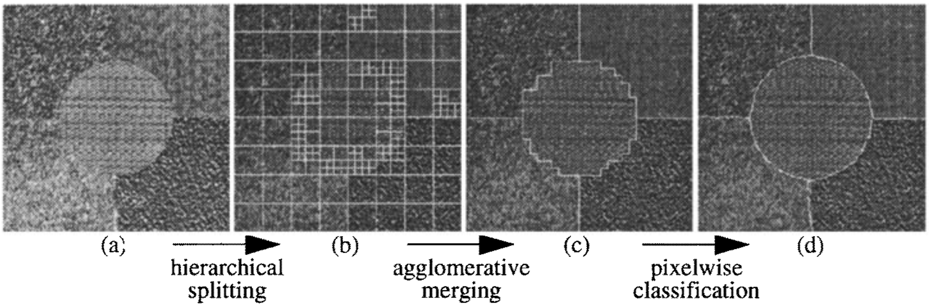

Fig. 2. Texture mosaic #1; the main sequence of the proposed segmentation algorithm. MIR_{stop}**=9.5,** MIR_{hi}**=1.2,** ERR_a**=1.4%,** ERR_p**=1.7%, sweeps=16.**

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

$$R = \frac{G_{max}}{G_{min}} > X \qquad (2)$$

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.

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.

Fig. 2b illustrates the result of the hierarchical splitting algorithm with X=1.2, S_{max}=64 and S_{min}=16. 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 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

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

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.

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. 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 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 a 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 post-processing method to improve the final segmentation result, e.g. by smoothing the texture boundaries or removing small regions as many existing algorithms do.

Fig. 2d demonstrates the final segmentation result after the pixelwise classification phase. Disc with radius of 11 pixels was used and 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%.

4 Experimental Results

Next, we present some quantitative results obtained with the method. The segmentation results for two additional texture mosaics and a natural scene 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. See [17] for results for additional images and for a detailed discussion on parameter selection. For each mosaic 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.

Mosaic #2 (Fig. 3a) is a 512x512 image containing four textures made by a GMRF process and a circle of painted surface in the middle [18]. 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. 3b) with segmentation error of 4.2% is quite decent. The final segmentation result (Fig. 3c) after 23 sweeps with segmentation error of 1.2% is excellent.

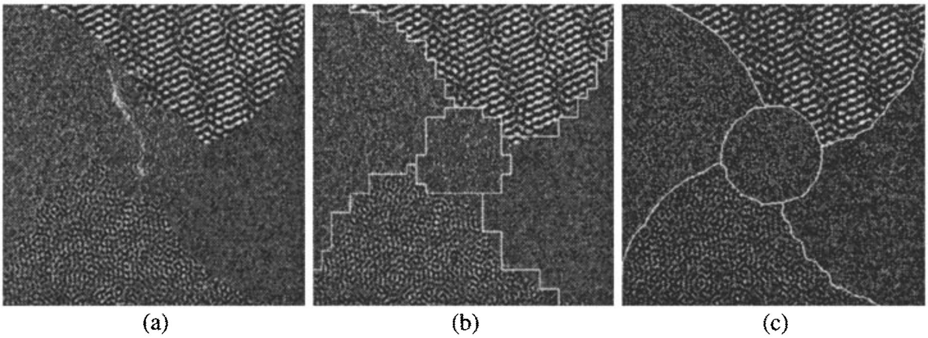

Fig. 3. Texture mosaic #2. MIR_{stop}=5.2, MIR_{hi}=1.6, ERR_a=4.2%, ERR_p=1.2%, sweeps=23.

Mosaic #3 (Fig. 4a) is 384x384 pixels in size and it is composed of textures of outdoor scenes [10]. 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 this mosaic Jain and Karu reported 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 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.

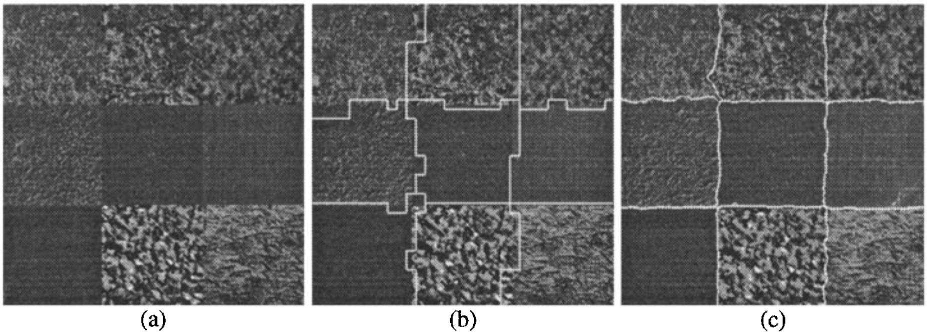

Fig. 4. Texture mosaic #3. MIR_{stop}=2.8, MIR_{hi}=1.2, ERR_a=7.8%, ERR_p=2.1%, sweeps=24.

We also applied the texture segmentation method to natural scenes. The scenes were originally in RGB format [9], but we converted them to gray level intensity images. As an example, scene #1 (Fig. 5a) is a 384 x 384 image of rocks in the sea. 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.

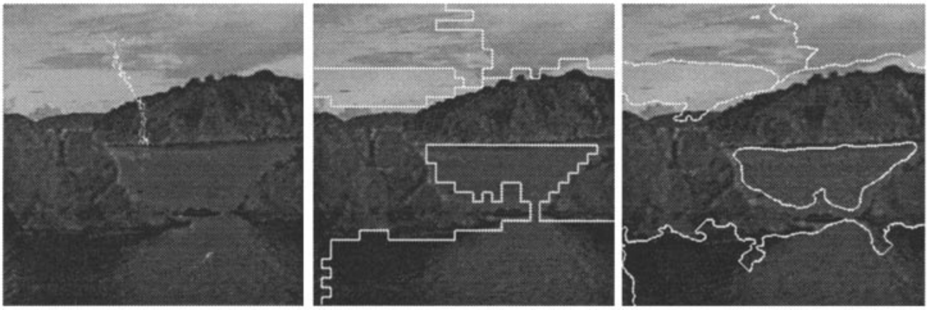

Fig. 5. Natural scene #1.

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 result after the agglomerative merging phase is presented in Fig. 5b and the final segmentation result is shown in Fig. 5c. 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. The obtained result is very satisfactory, considering that important color or gray scale information is not utilized in the segmentation.

5 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 [14].

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 and A.K. Jain, "Texture analysis," Chapter 2.1 in *Handbook of Pattern Recognition and Computer Vision*, C.H. Chen, L.F. Pau, P.S.P. Wang (eds.), World Scientific, Singapore, pp. 235-276, 1993.

[2] R.M. Haralick and L.G. Shapiro, "Image segmentation techniques," *Computer Vision, Graphics, and Image Processing*, vol. 29, pp. 100-132, 1985.

[3] T.R. Reed and J.M.H. Du Buf, "A review of recent texture segmentation and feature extraction techniques," *CVGIP Image Understanding*, vol. 57, pp. 359-372, 1993.

[4] M. Pietikäinen and A. Rosenfeld, "Image segmentation by texture using pyramid node linking," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 11, pp. 822-825, 1981.

[5] A.K. Jain and F. Farrokhnia, "Unsupervised texture segmentation using Gabor filters," *Pattern Recognition*, vol. 24, pp. 1167-1186, 1991.

[6] B.S. Manjunath and R. Chellappa, "Unsupervised texture segmentation using Markov random field models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, pp. 478-482, 1991.

[7] F.S. Cohen and Z. Fan, "Maximum likelihood unsupervised textured image segmentation," *CVGIP: Graphical Models and Image Processing*, vol. 54, pp. 239-251, 1992.

[8] M. Unser, "Texture classification and segmentation using wavelet frames," *IEEE Transactions on Image Processing*, vol. 4, pp. 1549-1560, 1995.

[9] D.K. Panjwani and G. Healey, "Markov random field models for unsupervised segmentation of textured color images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, pp. 939-954, 1995.

[10] A.K. Jain and K. Karu, "Learning texture discrimination masks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, pp. 195-205, 1996.

[11] D. Harwood, T. Ojala, M. Pietikäinen, S. Kelman and L.S. Davis, "Texture classification by center-symmetric auto-correlation, using Kullback discrimination of distributions," *Pattern Recognition Letters*, vol. 16, pp. 1-10, 1995.

[12] T. Ojala, M. Pietikäinen and D. Harwood, "A comparative study of texture measures with classification based on feature distributions," *Pattern Recognition*, vol. 29, pp. 51-59, 1996.

[13] T. Ojala, M. Pietikäinen and J. Nisula, "Determining composition of grain mixtures by texture classification based on feature distributions," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 10, pp. 73-82, 1996.

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

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

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

[17] T. Ojala and M. Pietikäinen, "Unsupervised texture segmentation using feature distributions", Technical Report CAR-TR-837, Center for Automation Research, University of Maryland, 1996.

[18] P.P. Ohanian and R.C. Dubes, "Performance evaluation for four classes of textural features," *Pattern Recognition*, vol. 25, pp. 819-833, 1992.