Image Segmentation by Texture Using Pyramid Node Linking

MATTI PIETIKÄINEN, MEMBER, IEEE, AND AZRIEL ROSENFELD, FELLOW, IEEE

Abstract—**In a "pyramid" of successively reduced-resolution versions of an image, by linking nodes representing image blocks to nodes representing nearby larger blocks that most closely resemble them, trees can be constructed (defined by the links) representing homogeneous parts of the input image. This approach is applied to segmenting an image on the basis of texture. It is started from an initial decomposition of the image into small blocks (e.g., 8 by 8); a textural property is computed for each block, yielding an array of property values; a "pyramid" is built of reduced-resolution versions of this array; and the node linking process is applied to this pyramid. The resulting trees define a segmentation of the original image into unions of the small blocks. This segmentation is similar to that obtained by minimum-error thresholding of the textural property values. Substantially better results are obtained when this "bottom-up" block linking process is preceded by a "top-down" process in which large homogeneous blocks are linked to all of their subblocks; the bottom-up linking is then used only for the blocks that were not linked by the top-down process.**

I. Introduction

Segmentation of an image into differently textured regions is a relatively difficult problem [1]. In order to distinguish reliably between two textures, we must examine relatively large samples of them, 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.

Chen and Pavlidis [2] have investigated a solution to the block size problem based on the use of a "pyramid" of successively reduced-resolution versions of the given image. If the image is 2^n by 2^n, the successive layers of the pyramid are, e.g., 2^{n-1} by 2^{n-1}, 2^{n-2} by $2^{n-2}, \cdots, 2$ by 2, 1 by 1. The elements of the array at layer k (with the original image being layer 0) thus represent image blocks of size 2^k by 2^k, and the size of the array is 2^{n-k} by 2^{n-k}. We assume here, for simplicity, that the elements in each layer correspond to nonoverlapping 2 by 2 blocks of elements in the layer below. (Other ways of constructing pyramids, based on overlapping blocks, are also possible, as will be seen below.) Thus each 2^k by 2^k block is the union of four 2^{k-1} by 2^{k-1} blocks, which are its four quadrants. For each block we can compute any desired textural property, or a set of such properties; see [1] for a review of textural properties. We can now define a top-down segmentation of the image into unions of blocks, based on the values of these properties, as follows. Starting from the top of the pyramid (a single node corresponding to the entire 2^n by 2^n image), we compare the property value(s) for each block with the values for its quadrants. If the values are sufficiently similar, we leave the block intact; if not, we split it into quadrants and repeat the process for each quadrant. When this process is complete, each block that remains unsplit should be contained in a homogeneously textured region. Moreover, the maximal connected sets of blocks that have similar textural properties should correspond to the homogeneously textured connected components of the image. Note that we can use a special case of this method to segment an image into connected regions of different average gray level by simply using average gray level as the "textural property."

Recently, a different pyramid-based method of segmenting an image was proposed by Burt *et al.* [3]–[5]. It makes use of a pyramid defined by overlapping blocks, e.g., the elements at each level correspond to 4 by 4 blocks of elements at the level below, where these blocks overlap by 50 percent both horizontally and vertically; the levels thus shrink by powers of 2 just as in the nonoverlapped case. Thus an element of level k has 16 "sons" at level $k-1$, and it is easily verified that this implies that an element at level $k-1$ has four "fathers" at level k. Initially, we associate property values with the elements at each level by simple averaging the values of the 16 "underlying" elements at the level below. We then define "links" between elements at successive levels based on the similarity of their values; e.g. [3], we link each element to that one of its four "fathers" which is most similar to it. (For variations on this idea see [4], [5].) We now recompute each element's value by averaging the values of only those of its sons that are linked to it (if any). This causes the similarities to change, so we may need to change some of the links; we then recompute the values again and repeat the process. The links tend to stabilize after a few iterations. If we trace them up to a level near the top of the pyramid (e.g., the 2 by 2 level), they define *trees* of linked image blocks. The sets of pixels at the leaves of such a tree constitute a homogeneous subpopulation of image pixels (but not necessarily a connected region!), so that the trees define a segmentation of the image into (at most four) subsets.

In the experiments described in [3]–[5], the property used was simply (average) gray level, so that the images were segmented into subsets having different average gray levels. This paper investigates a generalization of the "pyramid linking" approach of [3]–[5] which makes use of textural properties. Since such properties are not meaningful for single pixels, we begin with a fixed partition of the image into small blocks (e.g., 8 by 8), and compute a textural property for each block; this yields a 2^{n-3} by 2^{n-3} array of property values, which we use as input to the pyramid linking process. The trees defined by pyramid linking thus have 8 by 8 blocks, rather than single pixels, as their leaves, and the original image is segmented into unions of such blocks.

Since textural properties measured on 8 by 8 blocks are quite noisy, the pyramid linking process will not always yield a segmentation into the desired regions; for example, a block near the border of a region whose property value is close to that of the neighboring region may become linked to that region, and clusters of nearby blocks interior to a region whose property values differ from that of the region may support one another and become linked to a different subtree. In [6] it was found that smoothing the array of textural property values, e.g., by median filtering, greatly improves texture classification performance; note that a process such as median filtering tends to smooth the values within a homogeneous region without blurring them across region boundaries. Property value smoothing is also used in this correspondence to produce more reliable values, thus improving the results of the linking process.

Considerable further improvement is obtain by combining the "bottom-up" linking process described above with a "top-down" process similar to that used by Chen and Pavlidis. Here blocks judged to be homogeneous are linked to all of their subblocks (i.e., the links are created top down), and bottom-up linking is used only for those blocks that are left unlinked by the top-down process. This process will be described in further detail in Section IV.

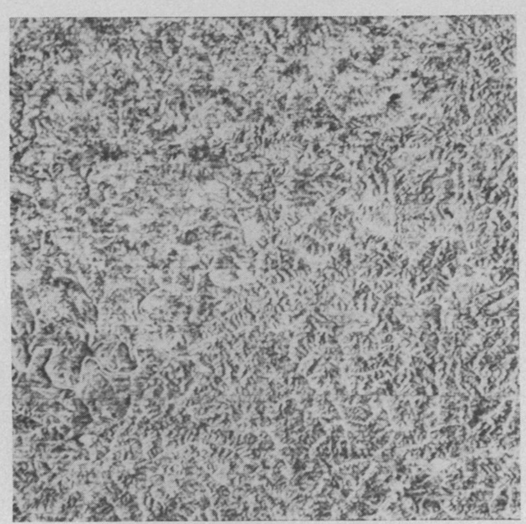
In Sections III and IV of this correspondence, the pyramid linking approach is applied to the two 512 by 512 test images shown in Fig. 1. These images are composed of the geological terrain textures used in earlier studies of texture classification [6],

Manuscript received April 2, 1981; revised August 10, 1981. This work was supported by the U.S. Air Force Office of Scientific Research under Grant AFOSR-77-3271.

M. Pietikäinen is with the Computer Vision Laboratory, Computer Science Center, University of Maryland, College Park, MD 20742, on leave from the Department of Electrical Engineering, University of Oulu, Finland.

A. Rosenfeld is with the Computer Vision Laboratory, Computer Science Center, University of Maryland, College Park, MD 20742.

0018-9472/81/1200-0822$00.75



(a)

(b)

Fig. 1. Test images.

[7]; Fig. 1(a) is Mississippian limestone and shale above the 45° diagonal and lower Pennsylvanian shale below it (labeled M/L), while Fig. 1(b) is lower Pennsylvanian shale above and Pennsylvanian sandstone and shale below (labeled L/P).

II. Texture Features and Feature Arrays

The texture feature used was the second-order gray level statistic "CONTRAST," which is the moment of inertia of the cooccurrence matrix about its main diagonal [1]. Cooccurrences were tabulated for a 1-pixel displacement in the horizontal direction. This feature was chosen because it performed quite well in the texture feature studies of Weszka *et al*. [7], and it is also computationally cheap, since it can be computed from a difference histogram rather than from a cooccurrence matrix [7]. Many other texture features could have been used, but we restricted ourselves to one feature because our primary interest was in the relative performance of pyramid linking schemes in comparison with standard methods.

The features were computed for nonoverlapping small windows (blocks) of the image. The sizes of these windows were 8 by 8 or 16 by 16 pixels. The size of the resulting feature array was 64 by 64 or 32 by 32. For example, if we compute the features for a 512 by 512 image in 8 by 8 blocks, the size of the feature array is 64 by 64. In the computation of these "CONTRAST" feature arrays we used a fast algorithm which reduced the computation time drastically compared to the conventional method. Instead of tabulating the cooccurence matrices for each of the 4096 (or 1024) blocks and deriving the "CONTRAST" features from these matrices, we derived the features from a difference histogram (in effect) by simply summing the squared differences of those pairs of pixels which had the required displacement. With this approach the whole feature array was computed during one image scan.

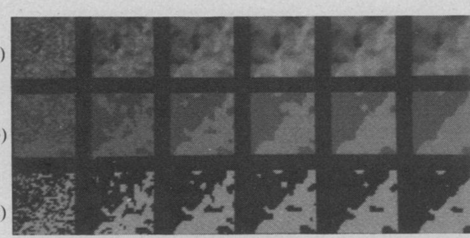


Fig. 2. (a) Feature arrays using 16 by 16 windows (after 0–5 iterations of median filtering). (b) Pyramid segmentation results. (c) Minimum error thresholding results.

Prior to pyramid segmentation, the feature values were scaled to make them suitable for the pyramid algorithms, which were designed to operate on input data in the range 0–63. Also, because texture features measured over small windows are unreliable, smoothing was applied to the feature arrays. The smoothing method used was median filtering (value replaced by the median of the feature values in the neighborhood), which was found in [6] to be effective for texture feature value smoothing. In the present studies we applied 0–5 iterations of median filtering (using a 3 by 3 pixel neighborhood) to the feature arrays and then we scaled these arrays linearly to have values ranging between zero and 63.

III. Experiments Using Iterative Bottom-up Linking

In all segmentation experiments, we used ten iterations in the pyramid node linking computations, although in most cases the segmentation converged earlier to a stable state. In the pyramid initialization, the methods with unweighted averaging of 16 or 4 sons were used. Forced linking was performed on one pyramid level at a time, and the segmentation was forced to give just two classes. These and other modifications of the original pyramid process are described in [4] and [5].

The effect of median filtering prior to segmentation is illustrated in Fig. 2 for the image M/L. Figure 2(a) shows the median filtered 32 by 32 pixel feature arrays after 0–5 iterations of median filtering. The pyramid segmentation results for these six cases are presented in Fig. 2(b). For comparison, Fig. 2(c) shows the corresponding segmentations using a minimum error thresholding method (the threshold that gives the minimum number of misclassified pixels is used to segment the feature array into two classes). It can be seen that the median filtering effectively reduces the error rate and that the results for these two segmentation methods (pyramid node linking and minimum-error thresholding) are quite similar. The selection of the minimum error threshold is very difficult for the images with zero to one iterations of median filtering, because the feature value histograms are not bimodal in these cases.

Figs. 3 and 4 illustrate the use of 4 and 16 sons in pyramid initialization for the 64 by 64 feature arrays M/L and L/P. Figs. 3(a) and 4(a) are the median filtered arrays after five iterations, and 3(d) and 4(d) are after three iterations of median filtering. In Fig. 3(b), 4(b), 3(e), and 4(e) are the corresponding segmentations using 4-son initialization, while in Figs. 3(c), 4(c), 3(f), and 4(f) are the results for 16 sons. 16-son initialization gave slightly better results for these noisy feature arrays, while for less noisy gray level images the 4-son initialization appears to be preferable [4].

To make the evaluation of the results easier, error rates were computed for each case. The error rate is defined to be the percentage of misclassifications for the unmixed windows in the

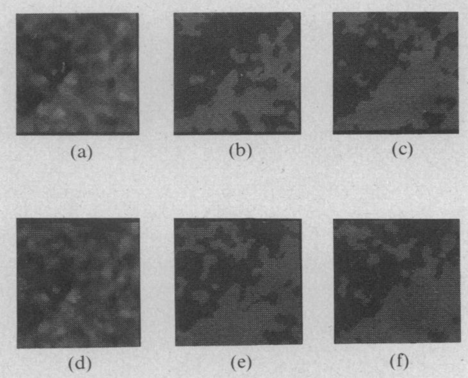


Fig. 3. Pyramid segmentation results for image in Fig. 1(a) using 8 by 8 windows. (a) (d) Feature array after 5(3) iterations of median filtering. (b) (e) Pyramid segmentations using 4-son initialization. (c) (f) Same using 16-son initialization.

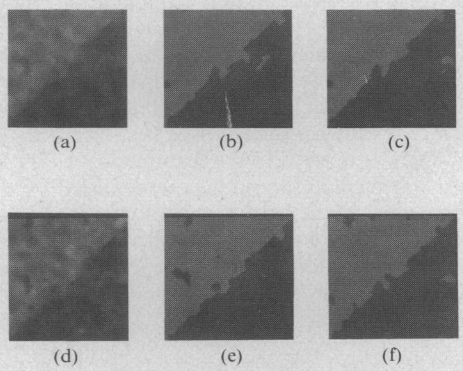


Fig. 4. Analogous to Fig. 3 for image in Fig. 1(b).

original image [6]. The error rate is based on the unmixed windows since the mixed windows (on the diagonal) always have 50 percent error.

In Table I are presented the error rates for 64 by 64 feature arrays derived from M/L and L/P and for a 32 by 32 array derived from M/L using 16 by 16 windows. In each case 0–5 iterations of median filtering were used before segmentation. Error rates for minimum error thresholding, for pyramid segmentation with 16-son initialization, and for the top-down/bottom-up linking method (described in Section IV) are shown. It can be seen that the error rates for bottom-up pyramid segmentation are very close to the error rates for minimum error thresholding. The minimum error thresholds were found empirically by looking for a threshold which gives the minimum error rate. It was found, however, that these thresholds can be derived automatically with fairly good accuracy by Gaussian filtering to feature value histograms obtained from properly selected training samples.

To reduce the effects of some very high feature values in some of the feature arrays, experiments were conducted in which the feature values were truncated by setting the values above a threshold equal to the value of the threshold. After this, the arrays were again linearly scaled. The results were slightly better using this method. This suggests that it is desirable to use some kind of nonlinear scaling of features, if there are feature values that are too dominating even after median filtering. It was also found that reduction of the gray level range of the original image prior to feature value computation did not have much effect on the segmentation results. When 32 or 16 gray levels were used instead of 64, the error rates were only slightly higher.

IV. Experiments Using Noniterative Top-down/Bottom-up Linking

The top-down phase of this new linking method resembles the split-and-merge algorithm used by Chen and Pavlidis in [9]. But instead of using a quadtree data structure, split-and-link operations are done in the pyramid structure. The following steps are used in this segmentation approach.

1) Initialize the node values of the pyramid by block averaging of each node's four sons.

2) Start linking at a specified level k. Find the minimum and maximum values of each node's four sons (at level $k-1$). If the difference between the maximum and minimum values is less than a selected threshold, link all four sons to their father, and go to level $k-1$. At this level, link all four sons (at level $k-2$) to those nodes which are linked to their fathers, i.e., those which belong to uniform blocks at level k. For the remaining nodes, apply the same test that was applied at level k, and link a node's sons to it if their range of values is below the threshold.

3) Link each unlinked node to one of its four fathers (closest in value). Do this at all levels starting from level 0. This process is done only once, rather than being iterated as in [3]–[5]. The resulting tree defines the final segmentation of the image.

For all test images the top-down linking was done from level 4 to level 1. The selection of the threshold value for block uniformity testing was done empirically. The same threshold value was used at each level. Because the error rates seemed not to be sensitive to changes in this threshold value, it should not be difficult to find the value automatically.

The error rates obtained by the top-down/bottom-up linking method are also shown in Table I. It can be seen that these error rates are much lower than the results for bottom-up linking and for minimum error thresholding. The results are quite good even without using median filtering.

Fig. 5 shows the best results for the 64 by 64 feature arrays. Fig. 5(a) and (b) show the M/L and L/P feature arrays after five iterations of median filtering, and Fig. 5(c) and (d) show the corresponding segmentation results. Fig. 6 shows the segmentation results for the same feature arrays without median filtering.

Fig. 7 shows the results for the 32 by 32 feature array M/L (features computed in 16 by 16 blocks). Fig. 7(a) shows the feature arrays after 0–5 iterations and Fig. 7(b) shows the segmentation results.

The results obtained by top-down/bottom-up linking are very good. It is evident that in order to get good segmentation results for texture images, we should use global information obtained from the upper pyramid levels to guide the segmentation at lower levels. If we use only bottom-up linking, the feature arrays are too noisy for good segmentation.

Many variations on the top-down/bottom-up linking method are possible, but the exploration of these variations is beyond the scope of the present study. Further studies in this area are planned.

V. Conclusion

This study shows that the pyramid node linking method can be successfully applied to segmentation by texture. By using iterative feature value smoothing prior to segmentation, quite small windows can be used for texture feature computation. This means that the dividing line between two texture types can be found with reasonable accuracy.

The accuracy of segmentation obtained by the basic bottom-up linking approach is comparable to the accuracy obtained by minimum error thresholding of the feature array. The advantage is that we need not look at the feature value histogram. Determining the appropriate threshold (or thresholds) from the histogram is often very difficult.

A great improvement in segmentation accuracy can be obtained by using a top-down/bottom-up linking method. In this approach, global information obtained from upper pyramid levels is used to locate large homogeneous areas, while more accurate boundary information about these areas is obtained by linking nodes on lower levels to the nodes representing these major areas.

Acknowledgment

The help of D. Lloyd Chesley in preparing this correspondence is gratefully acknowledged.

TABLE I
ERROR RATES
(PERCENT)

Image	Iterations of Median Filtering	Minimum Error Thresholding	Bottom-Up Linking	Bidirectional Linking
M/L (Fig. 1(a))	0	39.4	37.8	18.0
	1	30.7	27.7	10.7
	2	26.5	24.0	8.6
	3	24.3	19.4	8.5
	4	22.8	23.7	8.3
	5	22.0	21.0	8.0
L/P (Fig. 1(b))	0	25.7	25.0	2.9
	1	10.7	9.9	2.2
	2	6.8	5.2	2.2
	3	5.2	4.6	1.6
	4	4.3	6.3	2.0
	5	3.5	3.3	1.6
M/L (Fig. 1(a)) using 16 by 16 windows	0	33.7	36.0	14.7
	1	19.1	19.8	5.5
	2	13.7	11.1	6.0
	3	10.8	12.2	5.2
	4	8.8	8.0	2.6
	5	7.7	6.4	2.8

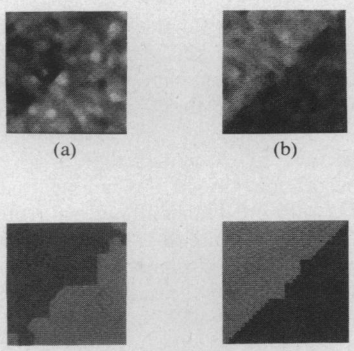


Fig. 5. Bidirectional linking results. (a) (b) Feature arrays for Fig. 1(a) and (b) after five iterations of median filtering. (c) (d) Segmentations.

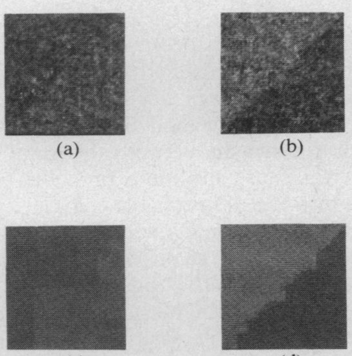


Fig. 6. Analogous to Fig. 5 but without median filtering.

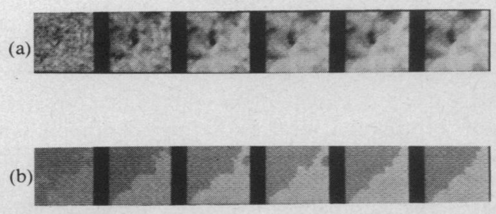


Fig. 7. Bidirectional results using 16 by 16 windows and 0–5 iterations of median filtering. (a) Feature arrays. (b) Segmentations.

REFERENCES

[1] R. M. Haralick, "Statistical and structural approaches to texture," *Proc. IEEE*, vol. 67, pp. 786–804, 1979.

[2] P. C. Chen and T. Pavlidis, "Segmentation by texture using a co-occurence matrix and a split-and-merge algorithm," *Comput. Graphics Image Processing*, vol. 10, pp. 172–182, 1979.

[3] P. Burt, T. H. Hong, and A. Rosenfeld, "Segmentation and estimation of image region properties through cooperative hierarchical computation," Tech. rep. TR-927, Computer Science Center, Univ. Maryland, College Park, 1980; and *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-11, no. 12, pp. 802–809, 1981.

[4] T. Silberberg, S. Peleg, and A. Rosenfeld, "Multi-resolution pixel linking for image smoothing and segmentation," Tech. rep. TR-977, Computer Science Center, Univ. Maryland, College Park, 1980.

[5] K. Narayanan, S. Peleg, A. Rosenfeld, and T. Silberberg, "Iterative image smoothing and segmentation by weighted pyramid linking," Tech. rep. TR-989, Computer Science Center, Univ. of Maryland, College Park, 1980.

[6] T. H. Hong, A. Wu, and A. Rosenfeld, "Feature value smoothing as an aid in texture analysis," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-10, pp. 519–524, 1980.

[7] J. Weszka, C. Dyer, and A. Rosenfeld, "A comparative study of texture measures for terrain classification," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-6, pp. 269–285, 1976.