

Nested Partitions Using Texture Segmentation

V. Lakshmanan
Coop. Inst. of Meso. Met. Studies
University of Oklahoma &
National Severe Storms Laboratory
lakshman@ou.edu

Victor E. DeBrunner
School of Electrical
and Computer Engineering
U. Oklahoma, Norman, OK
vdebrunn@ou.edu

R. Rabin
National Severe Storms Laboratory &
University of Wisconsin, Madison, WI
rabin@nssl.noaa.gov

Abstract

A multi-step method of partitioning the pixels of an image such that the partitions at one step are wholly nested inside the partitions of the next step is described, i.e. we describe an agglomerative, hierarchical segmentation technique that uses texture information to perform the segmentation. The image is requantized using K-Means clustering. Then, clusters are expanded using region growing and morphological processing. This provides the most detailed level of segmentation. The next coarser segmentation levels are obtained by steadily relaxing the inter-cluster distance between the clusters that is allowed by the morphological processing. Results are demonstrated on real-world images and swathes of Brodatz textures.

1 Introduction

Image segmentation is the process of partitioning the pixels of an image into discrete components or partitions. It is possible to devise a multiscale approach where the partitions in one scale are nested within the partitions of a coarser scale [16]. The watershed segmentation approach of [24] provides a way to test the saliency of contours and thus provides a way to do nest the partitions. Watershed segmentation, however, requires that the measurement space be ordered so as to arrange the pixels in the order of increasing “elevation”. When segmenting using texture, the measurement space consists of texture vectors and therefore can not be ordered without reducing its dimensionality.

In this paper, we describe a technique of obtaining nested partitions when using texture to do the segmentation. There have been many techniques proposed to perform multiscale

texture segmentation (for example: [2, 22]). However, these techniques all deal with multiscale inputs, not with the multiscale outputs required for nested partitions. After the multiple scale inputs are obtained, usually through filtering the input image with a filter bank, these inputs are segmented independently. The output partitions at each of the scales are not easily related to each other. Adaptive schemes are often used for this purpose.

A segmentation approach that provides nested partitions is useful for such applications as object tracking and pattern matching. When tracking a scene whose objects change with time (such as severe weather signatures on remote sensing imagery), the coarser partitions remain almost unchanged, but the more detailed signatures that are embedded in these coarse partitions change significantly. If the segmentation scheme can provide nested partitions, then the coarser partitions can be used for tracking, and detailed partitions embedded in them can be advected accordingly. In pattern matching, it is often easier to identify objects when the object can be identified at different scales (windows embedded in a building, for example). The scale of the building can be used to identify all the buildings in an image and then, the positions of the windows can be used to identify a particular building.

The rest of this paper is organized as follows: Section 2 describes the technique of obtaining nested partitions using texture information. Section 3 shows the result of this algorithm on various real-world scenes and swathes of Brodatz textures.

2 Method

A vector of measurements taken in the neighborhood of a pixel is associated with that pixel. This vector of measure-

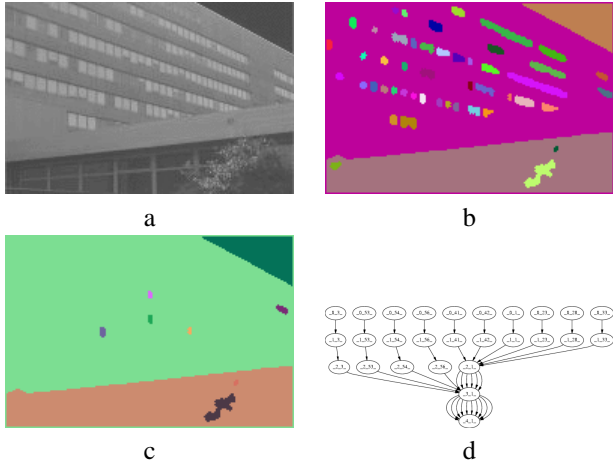


Figure 1. Hierarchical segmentation: Left to right, top to bottom: (a) A photograph of a building from the U. Groningen database [31] (b) Most detailed segmentation, using the multiscale segmentation algorithm described in this paper, colored such that each component is a different color. (c) A coarser level of segmentation. (d) A hierarchical tree representation of the segmentation results. Regions at the top of the graph (the results of more detailed segmentation) are contained within the regions at lower levels in the graph.

ments serves as a descriptor of texture. Suggested descriptors include hidden Markov models [15], Markov Random Fields [5], image moments [25], co-occurrence [11] and correlation matrices [7] and filtering methods [17, 9, 32]. There exists no consensus as to which of these approaches provides the optimal texture vector. Havlicek [12] points out that several approaches ([19, 10, 23], for example) developed as a way to emulate the human visual system. The statistical approach, also referred to as the stochastic approach, assumes that texture is characterized by the gray value pattern in a neighborhood surrounding the pixel [18]. Local coherence and orientation estimates [28], Gabor filters banks [30], statistics of Gabor coefficients [33, 27], amplitude envelopes of band-pass filters [3, 20] and multiple components' frequency estimates [12, 13] have been used successfully. In the absence of any strong consensus in the literature over the best texture to use, we used local neighborhood statistics (mean, variance, coefficient of variance) for all the images discussed in Section 3. The technique described works for any choice of texture vector.

Using the texture vectors associated with every pixel in the image, the images were requantized to a fixed number of levels using K-Means clustering [6, 26, 29, 8]. It should

be emphasized that this fixed number of levels (“K” in the K-means clustering) is not the number of regions in the resulting segmentation. It is the number of levels into which the image is requantized. The requantization is an iterative process that makes use of K-Means clustering to partition the image values into the K bins.

The measurement space (the gray level of the images) was divided up into K equal intervals and each pixel was initially assigned to the interval in which its gray level value lay. A Markov assumption, that a pixel belongs to the same interval as its neighbors, was imposed. In each iteration, the best label for each pixel in the image was chosen based on a cost factor that incorporated two measures. The first measure is the Euclidean distance, $d_m(k)$, between the texture vector at that pixel and the cluster mean of the candidate k , given by: $d_m(k) = \| \mu_k^n - T_{xy} \|$ where μ_k^n is the cluster mean of the k^{th} cluster at the n^{th} iteration and T_{xy} the texture vector at the pixel (x, y) . The second measure is a contiguity measure, $d_c(k)$, that measures the number of neighbors whose labels differed from the candidate label k . We can formally express the distance $d_c(k)$ as: $d_c(k) = \sum_{ij \in N_{xy}} (1 - \delta(S_{ij}^n - k))$ where S_{ij}^n is the label of the pixel (i, j) at the n^{th} iteration and N_{xy} is the set of 8-neighbors of the pixel (x, y) . Then the choice of the label for the pixel (x, y) in the $(n + 1)^{th}$ iteration, S_{xy}^{n+1} , is given by the label $k \in S_{N_{xy}}^n$ for which the energy, $E(k)$, given by $E(k) = \lambda d_m(k) + (1 - \lambda) d_c(k)$ $0 \leq \lambda \leq 1$ is minimum. We used $\lambda = 0.6$ for all the images, finding that any value of λ between 0.2 and 0.8 gave similar results. The candidates that were considered were the labels at the n^{th} iteration of the pixels within the 8-neighborhood of (x, y) . At the end of each iteration, the cluster attributes (the μ_k^n 's) were updated based on all the pixels that were labeled as belonging to the cluster at that time.

At this point, the image has been requantized, but the quantization has taken the spatial arrangement of pixel values into account. A region growing algorithm is employed to build a set of connected regions, where each region consists of 8-connected pixels that belong to the same K-Means cluster. If a connected region is too small, then its cluster mean (the mean of the texture vectors at each pixel in the region) is compared to the cluster means of the adjoining regions and the small region is merged with the closest mean. This process is repeated until the regions are such that all cluster means have reliable statistics. In practice, we considered a region too small if it had less than 10 contributing textural measurements. Usually, the number of texture measurements is the number of pixels in the region. However, if there are 4 independent image values for a single pixel (as in multi-channel satellite imagery), then the threshold of 10 contributing textural measurements may be met by a region of just 3 pixels.

The result of the K-Means segmentation, region growing

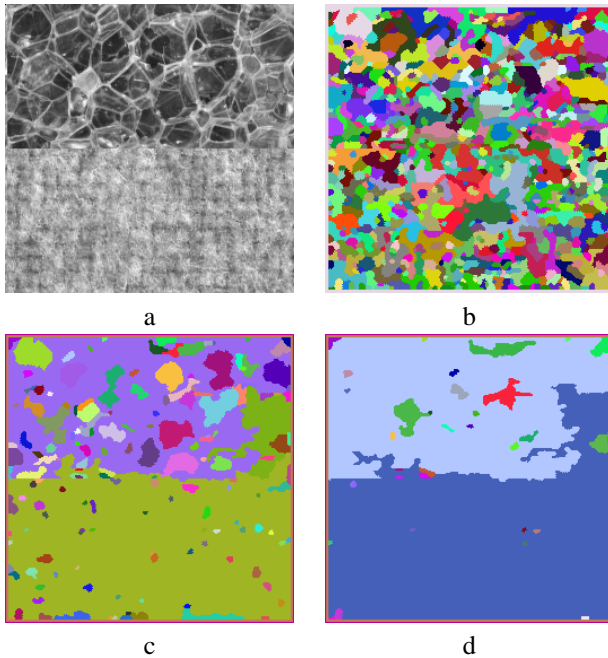


Figure 2. (a) Image comprised of Brodatz [4] textures D112 and D19. The result of segmentation using the method of this paper with $K=16$ at various scales. (b) most detailed (c) second most detailed (d) coarse – this is actually the most accurate.

and region merge steps is the most detailed segmentation of the image (See Figure 1b and Figure 2b). From this point onwards, we work exclusively in the domain of the segmented regions. The inter-cluster distances of all adjacent clusters (or regions) in the image are computed. A threshold is set such that half the pairs fall below this threshold. An iterative region merging is carried out whereby if a pair of clusters differ by less than this threshold, they are merged. More or less than half the clusters in the image may get merged because the cluster means are updated at the end of each merge, resulting in a different number of pairs which are closer than the threshold. The region merges are stopped when none of the resulting pairs of adjoining regions are closer than the threshold. The segmentation result at this point is the next coarser segmentation.

Because the results of segmentation at the second stage are formed by region merges only, every region in the coarse segmentation completely contains one or more regions in the detailed segmentation. Thus, there is a hierarchy of containment between the segmented results at these two scales. The inter-cluster distance threshold is relaxed steadily, set at each iteration to be of a value such that half the cluster pairs are closer to each other than the threshold. This pro-

cess is repeated until the segmented results are stable. The result of the segmentation at each stage gives one level of the hierarchical tree (see Figure 1d).

3 Results

The hierarchical segmentation approach described in this paper was implemented on a natural scene from the University of Groningen database described in [31]. The results are shown in Figure 1. The nested partitions are shown as a dendrogram in Figure 1d. Notice that in the coarser segmentations, the components in the detailed segmentation that corresponded to the windows are subsumed into the building itself, so that the three main regions are those corresponding to the building, its entry and to the sky.

The aerial photograph of San Francisco [14] has been segmented and the results shown in Figure 3. Unlike the study [14] from which this image was taken, we obtained these results without any a priori assumption of the number of regions in the image.

An image consisting of a Brodatz swathe was segmented and the different scale segmentations that result are shown in Figure 2.

An infrared satellite weather image is segmented using various texture segmentation techniques. Watershed segmentation, a scalar technique that provides nested partitions is also used for comparison in Figure 4.

For results on a wider variety of images, including medical and weather imagery, a quantitative measure of the segmentation accuracy as well as a fuller description of the technique, the interested reader is directed to [21].

References

- [1] J. Blum and J. Rosenblat. *Probability and Statistics*. W.B. Saunders Company, 1972.
- [2] C. Bouman and B. Liu. Multiple resolution segmentation of textured images. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, PAMI-13(2):99–113, 1991.
- [3] A. Bovik. Analysis of multichannel narrow-band filters for image texture segmentation. *IEEE Trans. Signal Proc.*, 39(9):2025–2043, Sep. 1991.
- [4] P. Brodatz. *Textures: A Photographic Album for Artists and Designers*. Dover Publishing Co., Toronto, 1966.
- [5] R. Chellappa. Two dimensional discrete gaussian markov random field models for image processing. In L. Kanal and A. Rosenfeld, editors, *Progress in Pattern Recognition*, pages 79–112. North Holland, Amsterdam, 1985.
- [6] C. Chen, J. Luo, and K. Parker. Image segmentation via adaptive k-mean clustering and knowledge-based morphological operations with biomedical applications. *IEEE Trans. on Image Processing*, 7(12):1673–1683, Dec 1998.
- [7] P. Chen and T. Pavlidis. Segmentation by texture using correlation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 5(7):64–69, Jan. 1983.

- [8] H. Derin and H. Elliot. Modeling and segmentation of noisy and textured images using gibbs random fields. *IEEE Trans. Pattern Anal. Mach. Intell.*, 9:39–55, 1987.
- [9] D. Dunn and W. E. Higgins. Optimal Gabor filters for texture segmentation. *IEEE Trans. Image Proc.*, 4(7):947–964, July 1995.
- [10] D. Gabor. Theory of communication. *J. Inst. Elec. Eng. London*, 93(III):429–457, 1946.
- [11] R. Haralick. Statistical and structural approaches to texture. *Proc. IEEE*, 67(5):786–804, May 1979.
- [12] J. Havlicek. The evolution of modern texture processing. *Elektrik*, 5(1):1–28, 1997.
- [13] J. Havlicek, D. Harding, and A. Bovik. The multi-component AM-FM image representation. *IEEE Trans. Image Processing*, 5(6):1094–1100, 06 1996.
- [14] T. Hofmann, J. Puzicha, and J. Buhmann. A deterministic annealing framework for unsupervised texture segmentation. Technical Report IAI-TR-96-2, Institut für Informatik III, U. Bonn, 1996. <http://www-dbv.cs.uni-bonn.de/image/example4.html>.
- [15] X. Huang, Y. Akiri, and M. Jack. *Hidden Markov Models for Speech Recognition*. Edinburgh Univ. Press, Edinburgh, 1990.
- [16] A. Jain and R. Dubes. *Algorithms for Clustering Data*. Prentice Hall, Englewood Cliffs, New Jersey, 1988.
- [17] A. Jain and F. Farroknia. Unsupervised texture segmentation using Gabor filters. *Pattern Recognition*, 24(12):1167–1186, 1991.
- [18] A. Jain and K. Karu. Learning texture discrimination masks. *IEEE Trans. on Pattern Anal. and Machine Intell.*, 18:195–205, 1996.
- [19] B. Julesz. Experiments in the visual perception of texture. *Sci. Amer.*, 232:84–92, 1975.
- [20] M. Kass and A. Witkin. Analyzing oriented patterns. *Comput. Vision, Graphics, Image Proc.*, 37:362–385, 1987.
- [21] V. Lakshmanan. *A Hierarchical, Multiscale Texture Segmentation Algorithm for Real-World Scenes*. PhD thesis, U. Oklahoma, Norman, OK, 2001.
- [22] J. Liu and Y. Yang. Multiresolution color image segmentation. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 16(7):689–700, July 1994.
- [23] S. Marcelja. Mathematical description of the responses of simple cortical cells. *J. Opt. Soc. Am.*, 70(11):1297–1300, 1982.
- [24] L. Najman and M. Schmitt. Geodesic saliency of watershed contours and hierarchical segmentation. *IEEE Trans. Patt. Anal. and Mach. Intell.*, 18:1163–1173, 1996.
- [25] C. Nikias. Higher order spectral analysis. In S. Haykin, editor, *Advances in Spectrum Analysis and Array Processing*, pages 326–365. Prentice Hall, Englewood Cliffs, NJ, 1991.
- [26] T. Pappas. An adaptive clustering algorithm for image segmentation. *IEEE Trans. Signal Processing*, 40:901–914, 1992.
- [27] M. Porat and Y. Zeevi. Localized texture processing in vision: Analysis and synthesis in the Gaborian space. *IEEE Trans. Biomed. Engg.*, 36(1):115–129, Jan. 1988.
- [28] A. Rao and B. Schunck. Computing oriented texture fields. *Proc. IEEE Comput. Soc. Conf. Comput. Vision Patt. Recog.*, pages 61–68, Jun 1989.

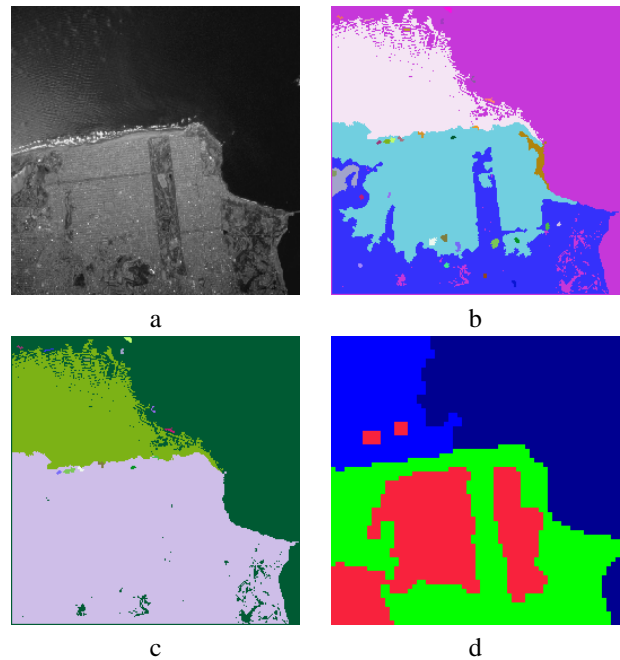


Figure 3. Left to right, top to bottom: (a) An aerial image of San Francisco. (b) The most detailed segmentation of the image using the method of this paper. (c) Segmentation at a coarser level using the method of this paper. There is no a priori assumption of the number of regions in the image in the method described in this paper. (d) The result of segmentation using the method of [14] (a Gabor filtering and clustering based-method, from which this image was taken) assuming that there are four clusters.

- [29] J. Theiler and G. Gisler. A contiguity-enhanced K-Means clustering algorithm for unsupervised multispectral image segmentation. In *Proc. SPIE*, volume 3159, pages 108–118, 1997.
- [30] M. Turner. Texture discrimination by Gabor functions. *Biol. Cybern.*, 55(2):71–82, 1986.
- [31] J. van Hateren and A. van der Schaar. Independent component filters of natural images compared with simple cells in primary visual cortex. In *Proc. R. Soc. Lond.*, volume B 265, pages 359–366, 1998.
- [32] T. P. Weldon and W. E. Higgins. An algorithm for designing multiple Gabor filters for segmenting multi-textured images. In *Proc. IEEE Int'l. Conf. Image Proc.*, Chicago, IL, October 4-7 1998.
- [33] Y. Zeevi and M. Porat. Combined frequency-position scheme of image representation in vision. *J. Opt. Soc. Am.*, 1(12):1248, Dec 1984.

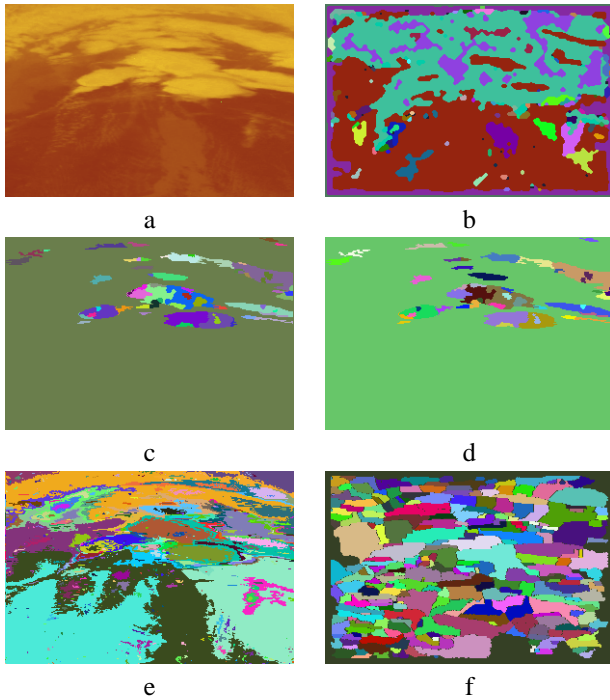


Figure 4. Segmenting an infrared satellite weather image. (a) The infrared image being segmented. Notice the various storms at the top of the image. The darker areas in the bottom correspond to ground. (b) The result of segmenting the image using the Markov Random Field (MRF) approach of [1]. There is no detail – it is effectively a binary segmentation. (c) The result of segmenting the image using the method of this paper (the most detailed scale). Notice the fine detail within the clouds. (d) The next higher scale of segmentation using the method of this paper. The strong storm cells being significantly colder are retained – the large cloud masses are merged. (e) Simply separating the image into contiguous bands of 1Kelvin . There is a lot of detail, but no organization. This is what you get using hierarchical thresholds. (f) Using the watershed segmentation approach of [24]. Because of the textural nature of the data, the watershed algorithm has very poor performance.