Finding Boundaries in Natural Images: A New Method Using Point Descriptors and Area Completion

Serge Belongie and Jitendra Malik

Department of Electrical Engineering and Computer Sciences
University of California at Berkeley, Berkeley, CA 94720, USA
{sjb,malik}@cs.berkeley.edu

Abstract. We develop an approach to image segmentation for natural scenes containing image texture. One general methodology which shows promise for solving this problem is to characterize textured regions via their responses to a set of filters. However, this approach brings with it many open questions, including how to combine texture and intensity information into a common descriptor and how to deal with the fact that filter responses inside textured regions are generally spatially inhomogeneous. Our goal in this paper is to introduce two new ideas which address these open questions and to demonstrate the application of these ideas to the segmentation of natural images. The first idea consists of a novel means of describing points in natural images and an associated distance function for comparing these descriptors. This distance function is aided in textured regions by the use of the second idea, a new process introduced here which we have termed *area completion.* Experimental segmentation results which incorporate our proposed approach into the *Normalized Cut* framework of Shi and Malik are provided for a variety of natural images.

1 Introduction

In this paper we study image segmentation, defined to be the process of partitioning the image into regions of coherent color, brightness, and texture. In each region there may be smooth variation in the attributes due to shading and texture gradients; segmentation should be tolerant to such variation.

There are several reasons why a satisfactory solution to image segmentation for natural scenes has remained elusive. Perhaps the foremost of these is image texture. Edge detection applied to natural images usually results in a tangled web of edges, leaving the grouping problem to be addressed by processes that might operate on such a representation. Approaches based on finding texture boundaries will be reviewed in more detail later, but it is fair to say that such algorithms have been demonstrated largely on synthetic collages of Brodatz texture patches.

The boundaries that we wish to find are those that separate coherent regions from one another. These are, in general, different from the boundaries that separate an object from its background, or from another object, since objects are often made up of several coherent regions. To find an object boundary, one must make use of a fairly specific model which encodes the knowledge of what components, coherent or not, constitute the object.

By no means does this imply, however, that boundaries of coherent regions are easy to find. The difficulties in this task principally arise from the need to quantify the notion of perceptual coherence for image texture. One general methodology which shows promise for solving this problem is to characterize textured regions via their responses to a set of filters. However, as discussed in the next section, this approach brings with it many open questions, including how to combine texture and intensity information into a common descriptor, and how to deal with the fact that filter responses inside textured regions are generally spatially inhomogeneous. Our goal in this paper is to introduce two new ideas. The first idea consists of a method of describing texture via the filter outputs computed at a "point" and a means of including intensity and/or color information in the same point descriptor. The second idea addresses the problem of inhomogeneity of the filter responses in textured regions through the use of *area completion*, which is analogous to contour completion.

The organization of this paper is as follows. In section 2 we discuss other relevant work in the area of texture segmentation in relation to our approach. In section 3 we introduce a novel means of describing "points" in natural images and an associated distance function for comparing these descriptors. The process of area completion is then discussed in section 4. In order to apply this new distance function to the problem of finding boundaries, we need a grouping criterion. For this purpose we appeal to the *Normalized Cut* approach of Shi and Malik [25], discussed in 5. Experimental results using Normalized Cuts for region segmentation and edge detection are presented in Sections 5.1 and 5.2. We conclude and discuss future work in section 6.

2 Related Work

Since the early 1980s, many approaches have been proposed in the literature which employ *filter-based* descriptions of texture [17,26,15,21,22,4,3]. By the term *filter-based* we mean that the fundamental representation for a pixel in an image includes not only its intensity or color information, but also the inner product of the neighborhood centered on that pixel with a set of filters tuned to various orientations and spatial frequencies.

As discussed for example in [16,19], vectors of filter responses have many appealing properties, including relationships to physiological findings in the primate visual system [8] and to the basic mathematical notion of a Taylor series expansion.

As rich as the information provided by these filter response vectors may be, a number of complications arise when one tries to use them for segmenting regions in natural images. Unlike intensity and color descriptors for a smooth image region, filter response vectors in a patch of coherent texture cannot be expected to simply pool together into a tight cluster in feature space, where the only variance is due to noise and measurement error. Rather, such a pool of feature vectors is bound to contain representatives from many different distinctive points throughout what may be called an "element" of the texture, or "texel." For example, assuming the filtering scale is not excessively large, filter responses computed at the center of a polka dot are quite different from those computed at the edge of a polka dot. (This is illustrated in Figure 2.)

Another way of saying this is that it is intrinsic to the nature of filter-based approaches that the filter responses will vary from one point to the next inside a region of coherent texture. The responses will of course vary from one point to the next inside incoherent regions, as well; what makes these two situations different, of course, is whether or not the variation exhibits a certain regularity.

Pervasive throughout the literature has been the use of profuse spatial averaging, which serves in one way or another to beat down the feature vectors until they become homogeneous [15, 21, 6]. Although the use of quadrature filter pairs [17, 11] allows for certain kinds of spatial texture variation to appear as a phase factor, this component is often discarded in favor of the quadrature energy [17, 3, 12] . Other approaches [20] seek to select a local scale at each pixel as a preprocessing step, so as to disallow the very extraction of any feature vectors which may be spatially inhomogeneous.

The price paid for these averaging approaches is that the richness of the descriptors is lost. Less damaging in this sense is the use of histogramming inside local patches [13, 28, 23] which does a much better job of preserving the local empirical feature distribution, though local spatial information is discarded for the sake of obtaining spatial homogeneity. Whether histogramming or simply averaging, however, all of these methods rely on the use of local windows or *patches*, which are naturally quite problematic around region boundaries.

It is worth noting that the ill effects of the use of patches are unlikely to arise in segmentation experiments on Brodatz mosaics [3, 23, 15]. The piecewise-coherent nature of such images can artificially induce clustering behavior in the feature space since, due to a lack of shading and perspective effects, there is not enough real-world variation present to challenge the cluster decision regions. This is particularly true when the texture cut-outs in the mosaic are squarish, lacking narrow or irregular regions, especially when the pixel coordinates are thrown into the clustering machinery.

One of the points we wish to put forward in this work is that profuse averaging of texture descriptors is both damaging and unnecessary. It is our view that a filter-based approach can succeed without appealing to unduly large scales if one properly exploits the behavior of feature response vectors in regions of coherent texture. We feel that the proper domain for a filter-based approach is that of describing *points*. Here we use the term *point* in the sense used by Koenderink and Van Doorn in [18] and [19], i.e., a small Gaussian point spread, at a sub-texel scale. The concept of *area completion* introduced in this paper is a step toward what we believe to be a proper treatment of point descriptors and thereby of filter-based texture segmentation in general.

A challenge which is met along the way is the problem of how to encode intensity alongside the filter responses. In the following section, we argue against appending intensity onto the filter response vector as a single scalar value; though it may seem intuitive since intensity is captured by the "zero-frequency" filter, it is in fact on a completely different scale from tuned texture filters. In order to be placed on equal footing with the texture filters, it must be represented in a compatible way. In section 3.2 we present one such solution.

3 Representing Texture and Intensity in the Same Feature Vector

In this section, we introduce a new method for describing and comparing points in an image. As discussed in the preceding section, we use the term "point" to refer to a small local neighborhood. Our approach is motivated by what one is likely to find at a randomly selected point in a natural image. The filter set described below allows us to characterize oriented edge or bar fragments, small spots, and some simple junction types; this accounts for a large share of the local structure one can expect to find in natural images. When such structure is absent, however, the best descriptor for a point is simply its color, or for grayscale images, its intensity.

Since there exists a continuum across these different point types, it is important that the *point descriptor* gracefully adapt to whichever form of description is most appropriate. The intensity encoding we propose in section 3.2, together with the normalization step described in the following section, represents one means of obtaining a point descriptor with this desirable property. We will also see that the format of the point descriptor described here admits the definition of a simple distance measure, described in 3.4, which allows us to measure similarity in a consistent manner for both textured and non-textured points.

3.1 Linear Filters for Texture Description

The filter set used in our experiments, depicted in Figure 1, is composed of zero-mean difference of Gaussian (DOG) and difference of offset Gaussian (DOOG) kernels. This choice of filters is similar to those used in [21], except that we use only one scale and we include odd-symmetric (edge-sensitive) oriented kernels. Each filter is divided by its L_1 norm in order to equalize the dynamic range across the filter set.

The vector of filter responses to an image I is defined as

$$\mathbf{u}_{tex} = (f_1 * I, f_2 * I, \ldots, f_{N_F} * I)^T$$

where in our case $N_F = 14$. Note that relative to the dimensions of our test images, which are 128×192, all of the filters reside at a fine scale.

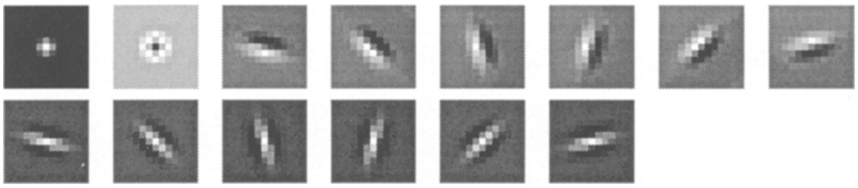

Fig. 1. Impulse responses of the filters f_1 through f_{14}, numbered left to right, top to bottom. Each kernel, which is of size 13×13, has been divided by its L_1 norm. f_1 are f_2 are DOG2 and DOG1 filters, respectively; f_3 through f_8 are DOOG1 filters; f_9 through f_{14} are DOOG2 filters. The oriented kernels are spaced at 30° increments, and have been offset by 15° to make quantization error more evenly distributed across the different orientations.

3.2 Intensity Value Encoding

In order to represent intensity alongside the components of the filter response vector, we make use of a "soft binning" of the intensity axis. Specifically, we encode the intensity I by the values taken on by N_I equally spaced Gaussians $g_k(I)$ on the interval $[0, 1]$.

The functions $g_k(I)$ are given by

$$g_k(I) = Ce^{-(I-\mu_k)^2/2\alpha^2}, \qquad k = 1, \ldots, N_I$$

where

$$\mu_k = \frac{k-1}{N_I - 1}, \qquad \alpha^2 = \frac{1}{2N_I^2}$$

The variance is chosen so as to make the sum of the N_I Gaussians approximately uniform on $[0, 1]$. The constant C is set to 0.1 so as to approximately match the dynamic range of the texture components.

The intensity feature vector is defined as

$$\mathbf{u}_{int} = (g_1(I), g_2(I), \ldots, g_{N_I}(I))^T$$

In our experiments we use $N_I = 10$. Thus, for a value of $I = 0.5$, the $k = 5$ and $k = 6$ components will "fire" strongly while the rest of the components will be quite small.

This representation is an example of a *value encoding*, as opposed to a *variable encoding*, in the sense discussed in [2]. In this manner, an intensity of 0.1 is held in as high a regard as an intensity of 0.9; each has its own place on the number line, and the former should not be viewed as a weak or low-magnitude version of the latter. Value encodings in this sense have been widely accepted in the context of filter orientation and scale, with various filters occupying different points on the frequency plane. It is natural therefore that a similar encoding be used for points on the intensity axis. An immediate benefit of this "equal-footing" representation is the appealing interpretation it lends to the normalization of the combined texture/intensity feature vector; this is discussed next.

3.3 Normalization

The complete feature vector at pixel i and the normalized feature vector are defined as

$$\mathbf{u}_i = (\mathbf{u}_{tex,i}^T, \mathbf{u}_{int,i}^T)^T, \qquad \hat{\mathbf{u}}_i = \frac{\mathbf{u}_i}{\|\mathbf{u}_i\|_2}$$

respectively. Since the L_2 norm of $\mathbf{u}_{int}$ is approximately equal to a constant, which we will refer to as β, we can also express $\hat{\mathbf{u}}_i$ as

$$\hat{\mathbf{u}}_i = \frac{\mathbf{u}_i}{\sqrt{\|\mathbf{u}_{tex,i}\|_2^2 + \beta}}$$

Thus we may think of the normalization step as a form of gain control, which diminishes the contribution of the intensity components when there is a lot of activity in the texture components. This can be related to the model of local gain control in primary visual cortex simple cells discussed in [5].

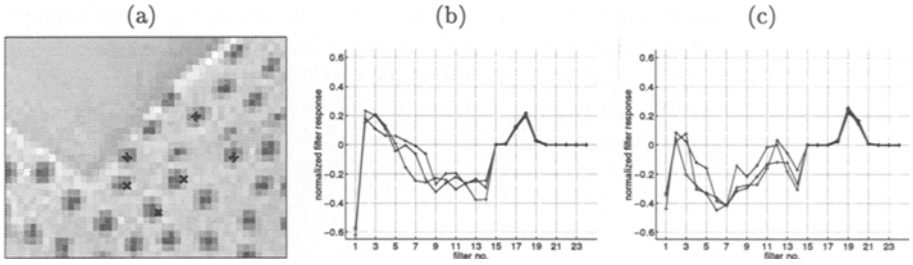

Fig. 2. Examples of normalized feature vectors for selected points in (a). Feature vectors from spot centers, marked by +'s, and spot edges, marked by x's, are plotted in (b) and (c), respectively. Components 1-14 correspond to the texture filter responses, where the numbering is consistent with that in Figure 1. The remaining 10 components correspond to the intensity value encoding.

3.4 Distance Measure

We define the distance between two normalized feature vectors $\hat{\mathbf{u}}_i$ and $\hat{\mathbf{u}}_j$ as

$$d^2(\hat{\mathbf{u}}_i, \hat{\mathbf{u}}_j) = (\hat{\mathbf{u}}_i - \hat{\mathbf{u}}_j)^T \Sigma^{-1} (\hat{\mathbf{u}}_i - \hat{\mathbf{u}}_j) \tag{1}$$

where the covariance matrix is given by

$$\Sigma = \mathrm{diag}(\underbrace{1, \ldots, 1}_{N_F}, \underbrace{\sigma_I^2, \ldots, \sigma_I^2}_{N_I})$$

The value of σ_I^2 represents the tolerance allowed on differences in the intensity components relative to those in the texture components. Its value was chosen empirically according to the following heuristic: distances inside regions possessing perceptually equivalent coherence should be as similar as possible, whether those regions be uniformly gray or uniformly textured. By observing distances within uniform regions of "clear sky" and "zebra stripes," to name two examples, we arrived at the value $\sigma_I^2 = 0.01$.

4 Area Completion

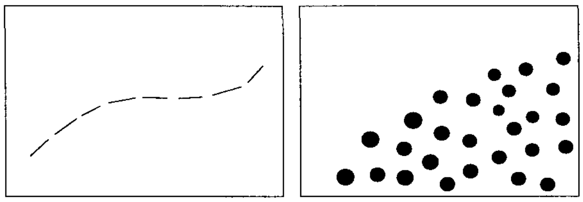

Fig. 3. Comparison of contour completion and area completion. Shown on the left is a set of disconnected contour fragments. These are readily grouped into a single smooth contour. In this process, the gaps, which locally bear no resemblance to the fragments, are assimilated into the final grouping when the "sum of the parts" is brought to bear. With area completion, the basic idea is the same. This is illustrated using the stylized polka dot texture boundary shown on the right. The grouping bridges the gaps between the dots and the area is completed.

The basic motivation behind area completion comes from the related process of contour completion. The latter problem is illustrated in Figure 3, left, by a set of disconnected contour fragments. We readily group the fragments together into a single contour [27]. In this grouping process, one implicitly absorbs the gaps between the fragments into the final smooth, continuous contour: the contour is completed. Conversely, we might conceive that the grouping takes place among the gaps, which then assimilate the fragments into the final contour. Of course the former interpretation is more readily apparent, but the latter highlights an important fact. This fact is that two kinds of elements, which locally bear negligible similarity, can become members of the same group, and thereby similar to one another, when the "sum of the parts" is taken into account.

The process of area completion logically extends this idea to 2D patterns, or in our context, to textures. The idea is the same: if two points are highly similar in terms of proximity and similarity, then they and that which is between them most likely belong together, as well. Figure 3, right, illustrates this idea with a stylized polka dot texture boundary.

Both contour completion and area completion rely on the statistics of the real world. In the case of contour completion, the relevant statistics pertain to the fact that object *boundaries* tend to be smooth and continuous. In the case of area completion, we appeal to the tendency for object *surfaces* to be smooth and continuous.

We shall operationalize this idea shortly. We wish to emphasize that the approach described here represents a departure from the practice of studying texture via large patches centered on pixels. Rather, we advocate an approach based on local computation of similarity, and the judicious propagation of this similarity along lines between matching points according to the process of area completion.

4.1 Explanation of Algorithm

The area completion algorithm begins with the computation of connection weights between pixel pairs in the $M \times N$ image I. A "weight" in this context refers to a monotonically decreasing function of $d(\hat{\mathbf{u}}_i, \hat{\mathbf{u}}_j)$, such as a Gaussian. (A specific choice of weighting function is given in Equation (3).)

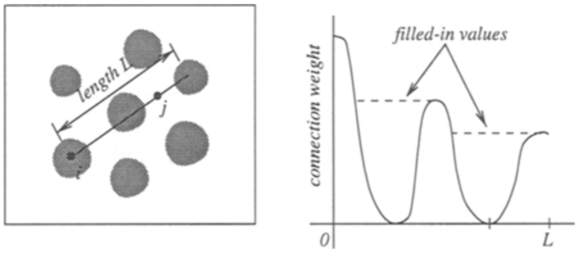

Fig. 4. Illustration of area completion. On the left, two pixels i and j in a textured image patch are depicted along with a line segment of length L starting at pixel i and passing through pixel j. A profile of hypothetical connection weights between pixel i and the pixels along this line segment is plotted on the right. The position of pixel j is indicated by a tick mark on the abscissa. Initially, the connection weight from pixel j to pixel i is zero. After area completion, this weight is updated to the maximum weight found along the line segment between j and the point at L. As indicated by the dashed lines, this procedure is equivalent to a "flood fill" operation on the original solid curve.

The area completion step is illustrated in Figure 4. Consider a line segment of length L extending from pixel i through pixel j. The solid curve in the plot on the right illustrates a hypothetical profile of connection weights for pixels along the line segment to pixel i. As indicated in the plot, the original connection weight from pixel j to pixel i is zero.

The connection weight from pixel j to pixel i after area completion is shown by the dashed curve: it is equal to the maximum over the weights between pixel j and the pixel at the far end of the line segment. This process is illustrated for a patch of real texture in Figure 5.

Let the matrix $\tilde{\mathbf{W}}$ denote the set of connection weights over all i and j after the area completion step. In general, $\tilde{\mathbf{W}}$ will be asymmetric. This arises from the fact that different points within a texture have varying amounts of distinctiveness. For example, the connection weight from one spot center to another spot center in Figure 4 is likely to be greater than the weights between pixels located elsewhere.

$\tilde{\mathbf{W}}$ can be made symmetric in a number of ways, including reassigning it as the maximum or minimum of $\tilde{\mathbf{W}}$ and $\tilde{\mathbf{W}}^T$, computed component-wise. In general, it is most appropriate to use the maximum since this choice automatically favors the weights contributed by the most distinctive elements.

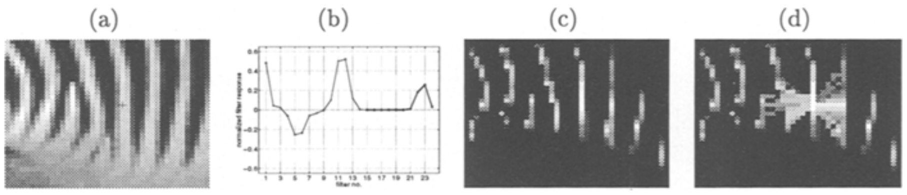

Fig. 5. Illustration of the area completion step applied to a patch of zebra texture. The point descriptor (feature vector) for the pixel marked in (a) is plotted in (b). Components 1-14 correspond to the texture filter responses, where the numbering is consistent with that in Figure 1. The remaining 10 components correspond to the intensity value encoding. The raw connection weights from each pixel in the image to the pixel marked in (a) are shown in (c), where the pixel intensity is proportional to the weight strength. The point descriptor for the marked pixel is highly similar to those found along other vertical white stripes, but is highly dissimilar to those found elsewhere. The connection weights to the marked pixel after area completion (out to a radius of $L = 10$) are shown in (d).

5 Normalized Cuts

The normalized cut approach frames the task of image segmentation as a graph partitioning problem, wherein each vertex represents a pixel and each edge represents a connection weight between two pixels. The term *normalized cut* actually refers to a criterion for measuring the goodness of a putative graph partition. This criterion, abbreviated as *NCut*, is related to the graph-theoretic notion of a *cut*. It is shown in [25] that, in the context of perceptual grouping, the minimum-*NCut* criterion leads to partitions which are preferable to those offered by the standard minimum-*cut* criterion.

Consider the graph $\mathbf{G} = (V, E)$ with vertices V, edges E, and weighted adjacency matrix $\mathbf{W}$. We may think of the weights W_{ij} which comprise $\mathbf{W}$ as a measure of similarity between pixels (i.e. vertices) i and j. Since the weights are symmetric, we say that $\mathbf{G}$ is an *undirected graph*. Let A and B represent two disjoint sets on $\mathbf{G}$, $A \cup B = V$, $A \cap B = \emptyset$, obtained by removing the edges connecting the two sets. These sets may represent, for example, pixels on the tiger skin and pixels on the pond background, respectively, in the tiger image in Figure 8. The normalized cut for (A, B) is a function of the *cut* between A and B and the *association* between A and B, as given by the expression

$$NCut(A,B) = \frac{cut(A,B)}{assoc(A,V)} + \frac{cut(A,B)}{assoc(B,V)}$$

where

$$cut(A,B) = \sum_{i \in A, j \in B} W_{ij}$$

and

$$assoc(A,V) = \sum_{i \in A, j \in V} W_{ij}, \qquad assoc(B,V) = \sum_{i \in B, j \in V} W_{ij}$$

As shown in [25], an approximation to the partition that minimizes the normalized cut criterion can be found by solving

$$(\mathbf{D} - \mathbf{W})\mathbf{v}_k = \lambda_k \mathbf{D}\mathbf{v}_k \tag{2}$$

for $\mathbf{v}_2$, the generalized eigenvector corresponding to the second smallest eigenvalue λ_2, where $\mathbf{D}$ is the diagonal matrix of total connection weights given by $D_{ii} = \sum_j W_{ij}$.

Since $\mathbf{v}_2$ is real-valued, it represents a soft assignment of pixels to the sets A and B. Similarly, successive eigenvectors suggest further partitioning of A and B into smaller groups. To transform this soft information into a more final result, one can proceed in two general directions: that of recursive thresholding to produce regions, and that of combining information from across the eigenvectors to produce edges.

5.1 Region Segmentation

The recursive thresholding procedure used in [25] proceeds as follows. First, 10 equally spaced values are considered between the maximum and minimum values in $\mathbf{v}_2$. Next, $\mathbf{v}_2$ is split into two groups using each threshold, thus resulting in 10 different choices for the sets A and B. The winner is then declared as the one for which $NCut(A, B)$ is a minimum. If K eigenvectors are found, then the above thresholding procedure can be run recursively within each partition until the $NCut$ value exceeds a threshold.[1]

In our implementation of the above described algorithm, we have used the following Gaussian weighting function,

$$W_{ij} = \begin{cases} e^{-d^2(\hat{\mathbf{u}}_i, \hat{\mathbf{u}}_j)/2} & \text{if } \|\mathbf{x}_i - \mathbf{x}_j\| < R \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

[1] Note that the maximum *NCut* value is 2, since in Equation (5), *cut* is always less than or equal to *assoc*.

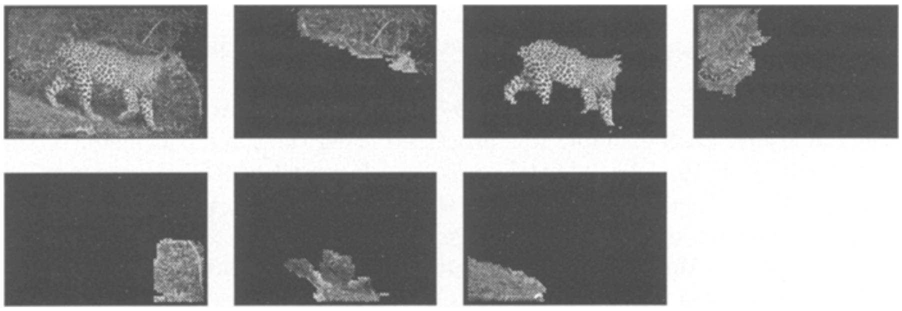

Fig. 6. Segmentation of leopard scene. Please refer to text for parameter settings.

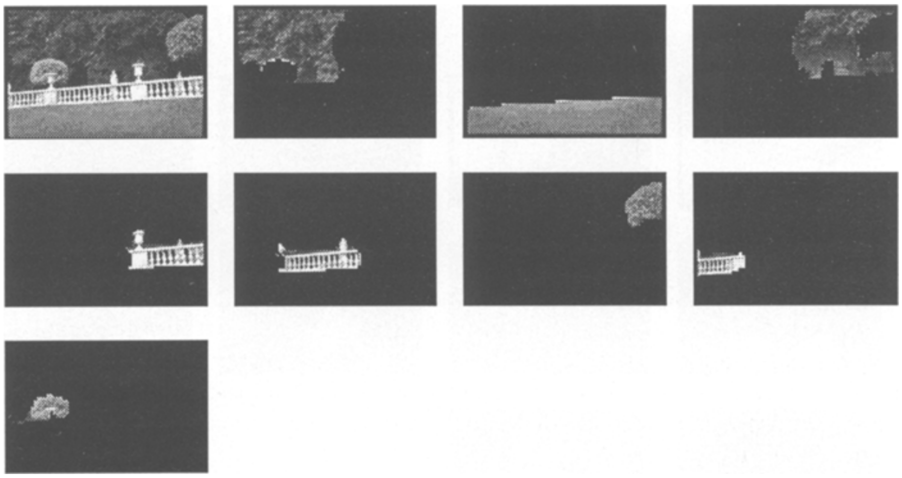

Fig. 7. Segmentation of landscape, with parameter settings as in Figure 6.

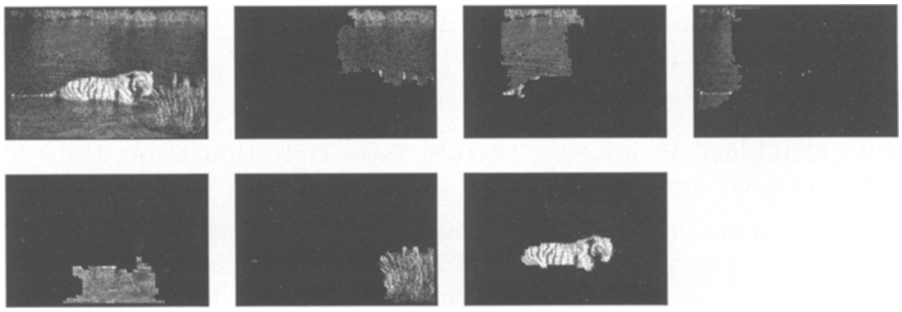

Fig. 8. Segmentation of tiger scene, with parameter settings as in Figure 6.

where $d^2(\hat{\mathbf{u}}_i, \hat{\mathbf{u}}_j)$ is as given in Equation (1), and $\mathbf{x}_k$ represents the coordinates of pixel k. Note that $\mathbf{W}$ will be sparse as a result of the distance threshold at radius R.

Our results are shown in Figures 6–10. The test images are all grayscale, of size 128×192. Each example was processed using the same parameter settings of $L = 10$, $R = 16$, $K = 14$, and an *NCut* threshold of 0.07. In order to save memory, the connection weights were subsampled by a factor of 3 with respect to each dimension of the original image.

We used the Matlab function eigs.m to solve the eigenvalue problem using the area-completed weights $\tilde{\mathbf{W}}$ in place of $\mathbf{W}$ in Equation (2).[2]

In each figure, the original image is shown at the top left, followed by a set of masked-out partitions found using the technique described above. The segmented regions are sorted in descending order according to area; segments comprising less than 2% of the total possible area are not shown.

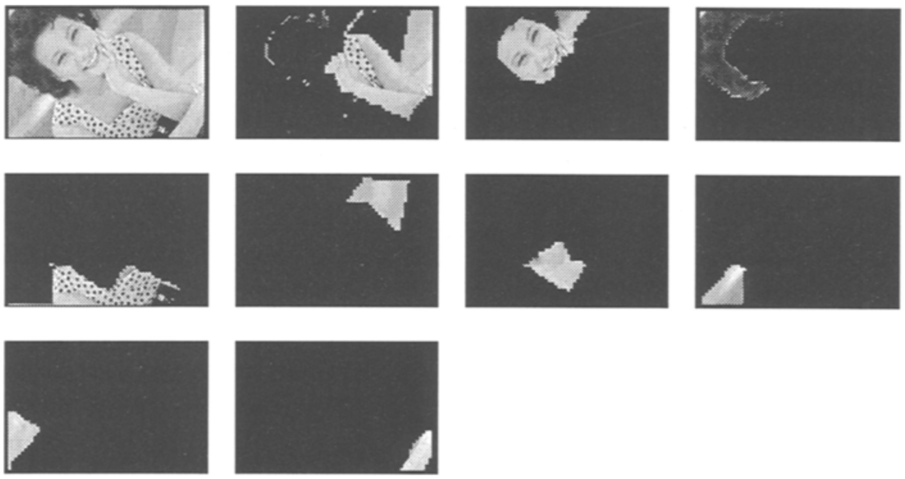

Fig. 9. Segmentation of person with parameter settings as in Figure 6.

5.2 Combined Brightness/Texture Edge Detection Using the Inter-Group Distance

As discussed in the introduction to this section, the Normalized Cut algorithm requires the solution of a generalized eigensystem involving the weighted adjacency matrix. In this section, we consider a physical interpretation of this

[2] `eigs.m` uses Arnoldi iteration, which reduces to Lanczos iteration since $\tilde{\mathbf{W}}$ is symmetric. The processing time for this operation is approximately 2 minutes on an HP J200/9000 using a convergence tolerance of `1e-10`.

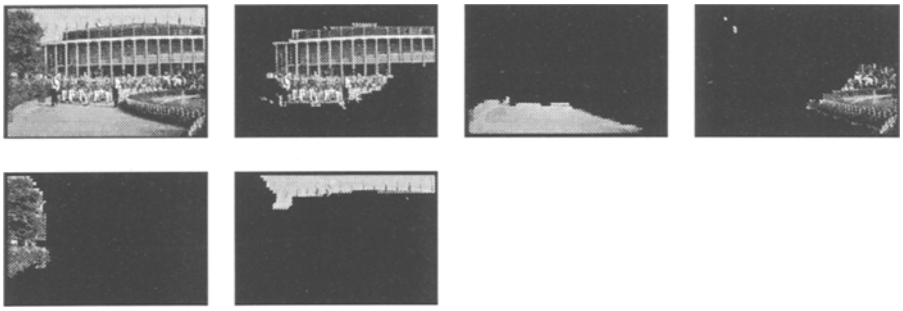

Fig. 10. Segmentation of building scene with parameter settings as in Figure 6.

eigensystem which leads to a new measure of "edginess" that we call *inter-group distance.*

The inter-group distance idea arises from the connection between generalized eigensystems and the analysis of mechanical vibrations in mass-spring systems[7, 10]. One can readily verify that the symmetric positive semidefinite matrix $(\mathbf{D} - \mathbf{W})$, known in graph theory as the *Laplacian* of the graph $\mathbf{G}$, corresponds to the *stiffness matrix* while the diagonal positive semidefinite matrix $\mathbf{D}$ represents a *mass matrix.* These matrices are typically denoted by $\mathbf{K}$ and $\mathbf{M}$, respectively, and appear in the equations of motion as

$$\mathbf{M}\ddot{\mathbf{x}}(t) = -\mathbf{K}\mathbf{x}(t)$$

If we assume a solution of the form $\mathbf{x}(t) = \mathbf{v}_k \cos(\omega_k t + \phi)$, we obtain the following generalized eigenvalue problem for the time-independent part,

$$\mathbf{K}\mathbf{v}_k = \omega_k^2 \mathbf{M}\mathbf{v}_k$$

in analogy to Equation (2).

The intuition is that each pixel represents a mass and each connection weight represents a Hooke spring constant. If the system is shaken, tightly connected groups of pixels will tend to shake together.

In light of this connection, the generalized eigenvectors in Equation (2) represent normal modes of vibration of an equivalent mass-spring system based on the pairwise pixel similarities.[3] When the system is excited, the resulting motion may be expressed as a superposition of the modes, with each mode weighted by a sinusoidal time-dependent term times some constant,

$$\mathbf{x}(t) = \sum_k \alpha_k \mathbf{v}_k \cos(\omega_k t + \phi)$$

where ω_k is equal to $\sqrt{\lambda_k}$ in Equation (2). A few eigenvectors for the test image of Figure 7 are shown in Figure 11, together with a snapshot of $\mathbf{x}(t)$.

[3] Note that since we assume free boundary conditions around the edges of the image, we ignore the first mode since it corresponds to uniform translation.

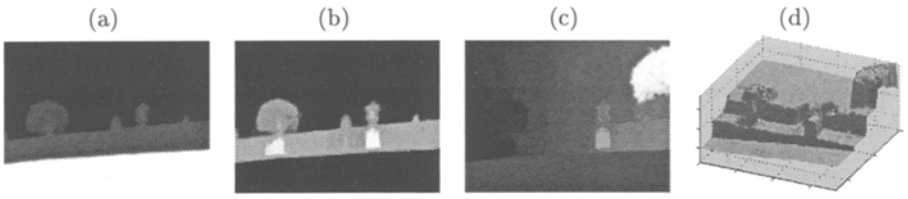

Fig. 11. Three generalized eigenvectors ($\mathbf{v}_2$, $\mathbf{v}_3$ and $\mathbf{v}_4$) for the landscape test image are shown in (a)-(c), reshaped as images. As an illustration of the connection between Normalized Cuts and the analysis of mass-spring systems, a superposition of the modes at an arbitrary time instant is shown in (d) as a surface plot.

The inter-group distance emerges when one considers the maximum extension of a spring over all time. In order to quantify this, we must specify a value for the α_k's that are used to combine the modes. For this purpose, we appeal to the *equipartition theorem* [24] which states that if a system described by classical statistical mechanics is in equilibrium, then it has equal energy in each mode. Since the energy of the kth mode [10] is given by

$$E_k = \frac{1}{2}\alpha_k^2\omega_k^2$$

we set α_k equal to $1/\omega_k$ for $k = 2, \ldots, K$.

Given this choice of α_k, we may define the inter-group distance between two pixels i and j as

$$d_{IG}(i,j) = \sum_{k=2}^{K} \frac{1}{\sqrt{\lambda_k}} |\mathbf{v}_k^i - \mathbf{v}_k^j| \tag{4}$$

As a simple illustration of the inter-group distance, we have shown in Figure 12 the average inter-group distance from each pixel to its closest four neighbors for the test images of the preceding section. Noting the similarity to conventional edge gradient images, one may employ contour closure techniques such as [9, 14, 1] to cut salient regions out of such a representation.

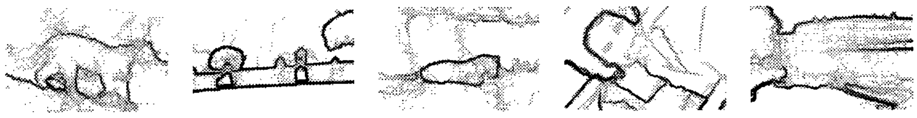

Fig. 12. Illustration of the inter-group distance from each pixel to its closest four neighbors.

6 Conclusion

In this work we have presented two new ideas which allow one to compute meaningful distances in feature space between points in natural images. The first idea consists of a new point descriptor which represents texture and intensity information in a compatible manner. The second idea is a process which is analogous to contour completion for textured regions which we have termed *area completion.* In order to demonstrate the effectiveness of these ideas, we performed several experiments using our new feature distance in the Normalized Cut framework.

In developing our proposed technique, we never appeal to the use of large amounts of spatial averaging to eliminate the inhomogeneities that occur in textured regions. As our method is based on the use of highly local point descriptors, the problem of patches straddling texture boundaries does not arise.

It is important to note that our experiments have been performed on natural images, rather than synthetic mosaics, and that the parameter settings are the same from image to image. Our plans for future work include further investigation of uses of the inter-group distance and the incorporation of color information into the point descriptor.

Acknowledgements

We would like to thank Thomas Leung and Jianbo Shi for their valuable input related to this work. We also wish to thank W. Kahan for numerous helpful discussions on the material discussed in Section 5.2. This work was supported by an NSF Digital Library Grant (IRI 94-11334), (ARO) DAAH04-96-1-0341, and an NSF Graduate Fellowship and U.C. Berkeley Chancellor's Opportunity Predoctoral Fellowship for Serge Belongie.

References

1. T.D. Alter. *The Role of Saliency and Error Propagation in Visual Object Recognition.* PhD Thesis, MIT, 1995.
2. D.H. Ballard. Cortical connections and parallel processing: Structure and function. *The Behavioral and Brain Sciences*, 9:67–120, 1986.
3. J. Bigün and J.M. Hans du Buf. N-folded symmetries by complex moments in gabor space and their application to unsupervised texture segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 16(1):80–87, 1994.
4. A. Bovik, M. Clark, and W.S. Geisler. Multichannel texture analysis using localized spatial filters. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12(1):55–73, 1990.
5. M. Carandini and D.J. Heeger. Summation and division by neurons in primate visual cortex. *Science*, 264:1333–1336, 1994.
6. S. Casadei, S. Mitter, and P. Perona. Boundary detection in piecewise homogeneous images. In *Proc. 2^{nd} Europ. Conf. Comput. Vision, G. Sandini (Ed.), LNCS-Series Vol. 588, Springer-Verlag,* pages 174–183, 1992.
7. J.W. Demmel. *Applied Numerical Linear Algebra.* SIAM, 1997.
8. R. DeValois and K. DeValois. *Spatial Vision.* Oxford University Press, 1988.

9. J.H. Elder and S.W. Zucker. Computing Contour Closure. *Fourth European Conf. Computer Vision*, 1996, Cambridge, England.
10. J.N. Franklin. *Matrix Theory.* Prentice-Hall, 1968.
11. W. Freeman and E. Adelson. The design and use of steerable filters. *IEEE Trans. Pattern Anal. Mach. Intell.*, 13:891–906, 1991.
12. H. Greenspan, S. Belongie, P. Perona, and R. Goodman. Rotation-Invariant Texture Recognition Using a Steerable Pyramid. *12th Int. Conf. Patt. Rec.*, 1994.
13. D.J. Heeger and J.R. Bergen. Pyramid-based texture analysis/synthesis. *Computer Graphics: SIGGRAPH*, pages 229–238, 1995.
14. D.W. Jacobs. Finding Salient Convex Groups. In I.J. Cox, P. Hansen, and B Julesz, eds., *Partitioning Data Sets,* vol. 19 of *DIMACS (Series in Discrete Mathematics and Theoretical Computer Science).* 1995.
15. A.K. Jain and F. Farrokhnia. Unsupervised texture segmentation using gabor filters. *Pattern Recognition*, 24(12):1167–1186, 1991.
16. D. Jones and J. Malik. Computational framework for determining stereo correspondence from a set of linear spatial filters. *Image and Vision Computing*, 10(10), 1992.
17. H. Knutsson and G.H. Granlund. Texture analysis using two-dimensional quadrature filters. In *Workshop on Computer Architecture for Pattern Analysis and Image Database Management*, pages 206–213. IEEE Computer Society, 1983.
18. J.J. Koenderink. Operational significance of receptive field assemblies. *Biol. Cybern.*, 58:163–171, 1988.
19. J.J. Koenderink and A.J. van Doorn. Representation of local geometry in the visual system. *Biol. Cybern.*, 55:367–375, 1987.
20. T. Lindeberg. *Scale-Space Theory in Computer Vision.* Kluwer, 1994.
21. J. Malik and P. Perona. Preattentive texture discrimination with early vision mechanisms. *J. Opt. Soc. Am. A*, 7(5):923–932, 1990.
22. B.S. Manjunath and W.Y. Ma. Texture features for browsing and retrieval of image data. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18(8):837–842, 1996.
23. J. Puzicha, T. Hofmann, and J.M. Buhmann. Non-parametric similarity measures for unsupervised texture segmentation and image retrieval. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 267–272, 1997.
24. F. Reif. *Statistical Physics.* McGraw-Hill, 1965.
25. J. Shi and J. Malik. Normalized cuts and image segmentation. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 731–737, 1997.
26. M.R. Turner. Texture discrimination by gabor functions. *Biol. Cybern.*, 55:71–82, 1986.
27. M. Wertheimer. Laws of organization in perceptual forms(partial translation). In W.B. Ellis, editor, *A Sourcebook of Gestalt Psycychology*, pages 71–88. Harcourt, Brace and Company, 1938.
28. S.C. Zhu, Y. Wu, and D. Mumford. Frame: Filters, random fields, and minimax entropy. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 686–693, 1996.