Biomedical Image Segmentation Using Multiscale Orientation Fields

Kah-Chan Low1 and James M. Coggins1,2
1Computer Science Department
University of North Carolina at Chapel Hill
and
2Center of Excellence in Space Data and Information Science
NASA Goddard Space Flight Center, Code 630.5

Abstract

We present an algorithm for labeling image regions based on pixel-level statistical pattern recognition. The structure of multiscale regions about each pixel is measured using isotropic Gaussian filters and by a Multiscale Orientation Field [1]. We create a redundant feature space representing several aspects of image structure across scale, orientation, and space. Our segmentation algorithm decides membership of pixels in regions using simple statistical pattern recognition methods such as distance measurement and thresholding. Feature vectors are examined locally to determine region membership; the features incorporate multiscale image structure information. Results of multiscale image segmentations on biomedical images are presented.

1. Biomedical Image Segmentation Objectives

The process of labeling pixels as members of different regions is called image segmentation. Automatic procedures for image segmentation often attempt to identify semantically meaningful objects directly from images. In biomedical images, however, this task is very difficult because objects of interest have unpredictable appearances due to differences in imaging technique, individual biological differences, and distortions induced by disease processes. Furthermore, objects of interest may appear at different spatial scales, or a single object may contain components at different spatial scales. Sometimes, the object of interest is a clinical abstraction that is not reflected in the image at all, such as a radiation treatment volume. Identification of semantically meaningful objects requires extensive knowledge of expected forms, plausible variations of forms, and the purpose of the identification task. Representation and integration of these forms of knowledge is far beyond the state of the art.

Our objective is not to label objects, but instead to label sensible image regions from which objects can be composed using tools such as the Image Hierarchy Editor [2]. A segmentation is considered good if its regions are visually sensible, each region contains pixels from only one visible object, and visible objects are composed of few regions. The algorithm produces a syntactic analysis of an image that allows a human to inject semantics into the representation later.

2. Multiscale Segmentation by Pattern Recognition

Since regions of interest occur at many levels of scale, and since a pixel might belong to different regions when examined at different scales, we seek a segmentation algorithm that can produce labelings emphasizing different scales. Furthermore, our algorithm should label as a single region components at different scales of a single sensible region.

Some multiscale segmentation methods attempt to find and exploit a single "best" scale for interpreting each pixel. Actually, there often exists either no best scale or several appropriate scales at which an acceptable labeling of a pixel can be inferred. Therefore, we will assign labels by examining the image structure representation in a uniform, wholistic fashion across space and scale simultaneously.

Our approach involves two main steps. First, we compute a multiscale representation of image structure consisting of the outputs of a series of isotropic Gaussian filters and a Multiscale Orientation Field [1]. This provides at each pixel a series of scalars and a series of vectors describing different structural aspects of multiscale neighborhoods of each pixel. In the second step, we grow regions by thresholding and then grouping pixels that satisfy certain region consistency criteria. The groupings thus identified are the segmented regions. Additional segmentations are produced by adjusting the consistency criterion to emphasize different ranges of scale.

We do not attempt to determine a single "right" segmentation; instead, our algorithm creates a family of plausible segmentations that can be merged into a final product by interaction with a human expert. This strategy allows the human to retain control of the decision-making process while allowing the computer to perform the bookkeeping and display enhancement tasks that are its strength. The segmentation provides a language for human-computer interaction based on regions rather than on pixels and edge segments that are more numerous and difficult for human manipulation. The final segmentations may be used for graphical display, for volume visualization, or for object measurement. This paper describes the segmentation
algorithm and demonstrates results obtained on biomedical images.

3. Segmentation Strategy

The control strategy for our segmentation algorithm is illustrated in Figure 1. Based on the representations of image structure provided by multiscale isotropic Gaussians and the Multiscale Orientation field, three region masks are created and then merged into a segmented image. A region mask is a mapping of pixels to region labels. We implement the region masks as images where the intensity at a pixel is a number identifying the region to which the pixel belongs. The region masks are created by thresholding feature images computed as described below from the image structure representations.

Representing image structure

The isotropic Gaussian filters used in this study have variances given by

$$\sigma^2_i = 2^i, \text{ for } i=0,...,9.$$  

The same isotropic Gaussians are used to create the scale-limited orientation filters that are used to compute the Multiscale Orientation Field. Thus, for each pixel, we have ten scalar values from the isotropic Gaussians and ten vector values from the Multiscale Orientation Field to characterize image structure.

The output of an isotropic Gaussian filter is a weighted average of intensities in a circular neighborhood of a pixel. Variations in this average intensity in multiscale neighborhoods of a pixel are captured by a series of Gaussian filters at different scales. The pattern of responses at a particular location to a series of Gaussian filters provides information about the scale of the region containing the pixel and how the intensity of a neighborhood of the pixel compares to the average intensity of the next-larger scale neighborhood. Thus, we are interested in the output value of the Gaussian filters at all locations, not just at edge locations. This approach treats the responses of the Gaussians as multiscale, adaptive intensity normalizations for image regions.

The Multiscale Orientation Field represents the orientation and eccentricity of the energy (deviation from regional average intensity) in the multiscale neighborhoods of each pixel [1].

![Figure 1: Block diagram of the segmentation procedure described in this paper.](image)

Computing Intermediate Features

The multiscale information provided by the image structure representations must be reduced to a form suitable for making decisions about the region membership of pixels. We perform this reduction by computing two intermediate features that capture particular aspects of region structure relevant to the segmentation problem. Most important, our intermediate features incorporate structural information from across a range of scales. This provides one form of redundancy that yields more robust region inferences. For instance, incorporating information at medium to large scales improves the resulting inferences' robustness against noise. Another form of redundancy.
arises from using the intermediate features to form three separate inferences of region structure that are combined to create a final segmentation. Thus, we have three independent, redundant structural inference mechanisms that reinforce each other to yield good region segmentations.

Both intermediate features consist of sums over a small neighborhood of a particular measure of dissimilarity between pixels. Regions are defined by labeling pixels with particular patterns of variation across this small neighborhood. These features are computed on small spatial neighborhoods, giving a sense of spatial coherence, and the distance measures are computed from across scale, giving a measure of structural coherence across scale.

The first intermediate feature uses the isotropic Gaussian filtered images to identify bright intensity ridges and dark intensity valleys by mapping local ridges or valleys to a relatively narrow range of grey levels. Afterwards, simple thresholdings can be conveniently applied to distinguish these features from the rest of the image. The feature we compute, then, is the sum of the distances between the current pixel and its spatially connected neighbors,

$$S_k(x) = \sum_{y \in N(x)} d_k(x,y)$$  

(1)

where x and y are pixel locations expressed as vectors, N(x) is the set of neighboring pixels of x and dk(x,y) is a measure of distance between pixels as represented in the feature space defined by the isotropic Gaussian filters, as follows:

$$d_k(x,y) = \sum_{i=0}^{D-1} w_{k,i} (G_i(x) - G_i(y))$$  

(2)

where D is the number of scales, Gi(x) is the output of the i-th isotropic Gaussian filter at location x, and wk,i is a weighting factor that determines the contribution of the i-th scale to the distance computation.

<table>
<thead>
<tr>
<th>i</th>
<th>1</th>
<th>1</th>
<th>2</th>
<th>2</th>
<th>8</th>
<th>8</th>
<th>16</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wi</td>
<td>5</td>
<td>15</td>
<td>15</td>
<td>10</td>
<td>10</td>
<td>5</td>
<td>5</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 1 the set of weightings used

The subscript k in formulas 1 and 2 selects a set of weighting factors that determine how much each scale contributes to the computation of the feature. The image structure representation is computed across all scales, so the weighting factors determine the scale of attention used to decide on region membership. Different weighting schemes can result in different segmentations. Table 1 shows the relative weighting factors we have used to compute distances between pixels in this paper. This set of weights emphasizes small-scale structure but contains significant contributions from larger scales. Inclusion of information at larger scales in this manner expands the scale range of intensity ridges or valleys that can be identified through Sk(x).

The second intermediate feature measures differences in both the isotropic Gaussians and in the Multiscale Orientation Field that represent the texture of the neighborhood about each pixel. The vector field, besides being critical to texture representation, helps to cement large regions together that would be fragmented by the Gaussian filters alone. We define the feature as follows:

$$T_k(x) = \sum_{y \in N(x)} t_k(x,y)$$  

(3)

and the distance measure tk is defined as

$$t_k(x,y) = g \sum_{i=1}^{D} w_{k,i} (G_i(x) - G_i(y))^2 +$$

$$\sum_{i=1}^{D} \frac{v_{k,i}}{||V_i(x) - V_i(y)||}$$  

(4)

where the first term is similar to equation 2 except that the difference is squared, Vi(x) is the MOF vector at location x and scale i, ||v|| denotes the length of a vector, and G and V are scalar weights on the Gaussian and vector terms, respectively. The second term is the sum of lengths of the difference vectors between the corresponding scales of the MOF at pixels x and y.

This use of absolute differences in scale and orientation to measure properties of neighborhoods about every pixel is similar to the approach used for representing visual texture by Coggins [3,4]. Those studies characterized texture as local-average energy, where energy refers to intensity deviation from the regional mean. The multiscale Gaussians perform an adaptive intensity normalization across multiscale regions about every pixel.

Regions of coherent texture are perceived as homogeneous segments not because of their uniformity in intensities nor due to geometrical or structural properties, for they are simply large regions with ill-defined boundaries. Given the importance we attach to orientations across scale space as a textural measure, we have decided to segment large sized regions in an image using information present in the MOF’s described in [1]. The orientation information across scale creates coherence across the textured region that holds the region
together when otherwise low-contrast intensity deviations in the Gaussian-filtered images would break the region apart into low-contrast blobs.

In earlier experiments [5], we have established that in the case of an ideal step edge, MOF vectors in the neighborhood but on different sides of the step edge have phase angles that are perpendicular to each other. In addition, vectors near the step edge have larger magnitudes than those that are farther away from it. The same trends can also be observed for non-ideal edges. In general, vectors having small magnitudes correspond to pixels situated far away from the edges of a large region. Large differences between phase angles of these pixels should not preclude us from classifying them as belonging to the same region. The margins of allowable differences shrink, however, as the vectors lengthen. With these observations in mind, we conclude that vector subtraction provides us with a desirable quantification of pixel textural differences.

Creating Region Masks by Thresholding

Three region masks are created from the intermediate features $S_k(x)$ and $T_k(x)$. These three region masks are then merged to create a segmentation of the image. To create each region mask, we apply a threshold to one of the intermediate feature images and label contiguous groups of pixels selected by the thresholding operation as different regions.

The first region mask, $R_1(x)$ is created by marking contiguous pixels that are above threshold $\tau_1$ in intermediate feature $S_k(x)$. These regions correspond to intensity ridges in the image. The second feature mask, $R_2(x)$ is created by marking pixels that are below threshold $\tau_2$ in the same feature image $S_k(x)$, where $\tau_2<\tau_1$. These regions correspond to intensity valleys in the image. The third region mask $R_3(x)$ is created by marking pixels that are below threshold $\tau_3$ in intermediate feature $T_k(x)$. These regions correspond to areas of the image having similar visual texture.

Every pixel that satisfies the thresholding criterion receives a region label in each of the three masks. At present these thresholds are obtained manually by examination of the histogram of the intermediate feature values over the image and placing the threshold at a knee of the frequency curve. The heuristic used to select the thresholds is to select pixels not only for their contrast but also for their density in the tail of the observed distribution of feature values by moving the threshold from an extreme of a distribution toward the mean until the first significant change (10%) in the distribution is encountered. Table 2 shows the thresholds used in this study.

<table>
<thead>
<tr>
<th>Image</th>
<th>$\tau_1$</th>
<th>$F(S; \tau_1)$</th>
<th>$\tau_2$</th>
<th>$F(S; \tau_2)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>figure 2</td>
<td>178</td>
<td>81%</td>
<td>150</td>
<td>13%</td>
</tr>
<tr>
<td>figure 3</td>
<td>260</td>
<td>85%</td>
<td>185</td>
<td>10%</td>
</tr>
<tr>
<td>figure 4</td>
<td>278</td>
<td>70%</td>
<td>245</td>
<td>27%</td>
</tr>
</tbody>
</table>

$F(S;\tau)$ : intensity cumulative percentage of $S_k(x)$

$S(x)$ is normalized to 1-500 in grey levels

Table 2 Threshold selections for $S(x)$

Combining Region Masks to Create a Segmentation

To combine the regions from the three region masks into a single labeling, a priority ordering is assigned in which labels in mask $R_1(x)$ are considered first, then mask $R_3(x)$, then mask $R_2(x)$. The rationale for this labeling is that the bright regions marked in mask $R_1(x)$ are among the brightest and most important ridges in the image, regions in $R_3(x)$ form background regions defined by textural properties, and regions defined in $R_2(x)$ are redundant with the $R_1(x)$ regions but defining dark intensity valleys. The three masks provide a redundant check on the validity of the regions. A light region of low contrast that is missed in mask $R_1(x)$ can be picked up by the reinforcing evidence given for the region's validity in masks $R_2(x)$ and $R_3(x)$.

4. Results and evaluation

The algorithm described in Section 3 uses sophisticated representations of multiscale image structure, but very simple, naive processing of those representations. We have not yet attempted to optimize or enhance the procedures or thresholds, or even to automatically compute the thresholds. Instead, our current objective is to explore the feature space defined by the isotropic Gaussians and the Multiscale Orientation Field to determine whether that feature space is capturing the aspects of image structure that we consider important and that can lead to good image segmentations. During this exploration, we apply trivial decision rules for assigning region labels so that the success or failure of the labeling can be attributed to the feature space and not to some complex analysis technique.

Results of the algorithm are illustrated in Figures 2-4. Figure 2a is part of a sagittal MRI of a human head, Figure 3a is a photograph of a normal optic disc, and Figure 4a is a photomicrograph of a cross section of a nerve fiber stained so that the myelin sheaths are dark. These images provide a range of interesting segmentation
Spatially connected group of pixels with the same non-zero (dark) grey levels define a region

They all contain bright intensity ridges and/or plateaus, large textured background regions, ambiguous structures which can be classified as foreground or background depending on one’s viewing context, and dark structures which help to define certain bright structures. While these images share the above attributes, each image provides these features in a different form. The bright structures in Figure 3a are well-defined, narrow intensity ridges while those in Figure 4a are intensity plateaus. Background regions in Figure 3a are grey and grainy; those in figure 2a are homogeneous dark patches; and those in figure 4a are homogeneous light patches. Further discussion on how image differences affect the resulting segmentations is given below.

The MRI image

Figure 2b indicates the three individual regions* corresponding to the three lobes of cerebral cortex in figure 2a. Structurally, parts of the lobes are intensity ridges, while other parts are intensity plateaus. It is therefore not surprising that these regions are indeed derived from $S_k(x)$. Even though different parts of the lobes exist at different scales, the algorithm manages to pick up the lobes in complete pieces. The benefit of incorporating weighted information from non-peak scales is apparent here. Using information from only a single scale would have resulted in increased fragmentation.

Figure 2c shows the regions corresponding to the scalp and a lump of tissue at the posterior end of the buccal cavity. The scalp is a pronounced intensity ridge and it is derived from $S_k(x)$. The intensity structure associated with the lump of tissue is less pronounced. $S_k(x)$ alone does not segment this area as a single region, but combined with $T_k(x)$ this region holds together. The lump is flanked on the left by the void of buccal cavity, which is well-defined and labelled in region map $R_3(x)$ (associated with $T_k(x)$). The lump is also otherwise surrounded by intensity valleys, which are labelled in $R_2(x)$. Thus, $R_2(x)$ and $R_3(x)$ help to define the lump region even though $R_1(x)$ fails to identify it. Thus, we have three different sources of information mutually providing redundancy checks against errors in individual sources.

Figure 2d shows the segmented regions associated with background areas and intensity valleys. Figure 2e shows a section where our procedure gives mediocre results. If we take all the labelled segments in the figure together, they do define the corpus collosium-cerebellum region. However, there are many fragments and most of the fragments do not correspond to any particular sensible structure within the corpus collosium-cerebellum region. The entire region is simply too large to be picked up in one piece by $S_k(x)$ with the
weighting scheme we are using (recall that we emphasize small scale values). The region is in fact segmented in one piece in mask $R_3(x)$ but since $R_1(x)$ has higher priority in determining the final segmentation, the $R_1(x)$ result is selected. Further research into devising rules that are more flexible is certainly warranted.

**The retinal photograph**

Figure 3b shows the regions corresponding to the bright blood vessels. Since they are pronounced intensity ridges, they are readily picked up from $S_k(x)$. Note that the forked blood vessels in the lower right hand quadrant of the image are identified in spite of the fact that the intensity decreases as it progresses towards the center of the dark optic disc.

Figures 3c shows the segmentation for the retinal background regions. These regions are somewhat more fragmented than the background regions found in figure 2a because the corresponding regions in the original image are much noisier than any background regions found in figure 2a. This suggests that superimposition of $R_1(x)$, $R_2(x)$, and $R_3(x)$ should be done using a more sophisticated algorithm that merges smaller fragments with nearby statistically similar ones to reduce fragmentation. Another possibility is to rely on the multiscale hierarchy to tie the fragments into coherent, sensible regions. This might be accomplished by computing and later merging segmentations created using weighting schemes that emphasize larger scales.

**The nerve fiber cross-section**
In this image the important regions (dark myelin sheath bands) are darker than the background. We therefore reverse the role of region masks $R_1(x)$ and $R_2(x)$ when we combine the region masks into a segmented image. Figure 4b shows the contribution of region mask $R_2(x)$ -- segments that correspond to the circular bands of myelin sheaths. Small bright plateaus picked up in mask $R_1(x)$ help to define the interior of myelin bands as indicated in Figure 4c. $T_k(x)$ helps to define relatively few of the regions in this image. The regions indicated in figure 4d are a few of those regions whose definition can be ascribed to $T_k(x)$. Most of the intercellular white regions are defined in mask $R_1(x)$. This is not surprising since the intercellular regions are intensity ridges flanked by intensity valleys (dark myelin sheath bands), as shown in figure 4e.

5. Conclusion

The formal techniques employed in our algorithm employ the statistical pattern recognition paradigm. The processing used in the entire process are also characterized by their simplicity (or simple-mindedness). Because of this attribute, each of them, while quite effective in extracting image features for which they are designed, do produce some segmentation errors in image areas that they are not equipped to handle. However, when their resulting and somewhat erroneous segmentations are combined in a simple manner, unexpectedly good primitive region segmentations are obtained. This is especially surprising, given the apparent simplicity of the techniques involved.

References


