

# Anti-Geometric Diffusion for Adaptive Thresholding and Fast Segmentation

Siddharth Manay, *Student Member, IEEE*, and Anthony Yezzi, *Member, IEEE*

**Abstract**—In this paper, we utilize an anisotropic diffusion model, which we call the anti-geometric heat flow, for adaptive thresholding of bimodal images and for segmentation of more general greyscale images. In a departure from most anisotropic diffusion techniques, we select the local diffusion direction that smears edges in the image rather than seeking to preserve them. In this manner, we are rapidly able to detect and discriminate between entire image regions that lie nearby, but on opposite sides, of a prominent edge. The detection of such regions occurs during the diffusion process rather than afterward, thereby side-stepping the most notorious problem associated with diffusion methods, namely, “When should you stop diffusing?” We initially outline a procedure for adaptive thresholding, but ultimately show how this model may be used in a region splitting procedure which, when combined with energy based region merging procedures, provides a general framework for image segmentation. We discuss a fast implementation of one such framework and demonstrate its effectiveness in segmenting medical, military, and scene imagery.

## I. INTRODUCTION

**T**HRESHOLDING and segmentation constitute two of the most widely used techniques in image processing and low-level vision, sometimes as end goals by themselves, and sometimes as early steps for higher level vision. The goal of thresholding is to classify image pixels into one of two categories (e.g., foreground and background). Segmentation generalizes this by allowing for an arbitrary number of categories. We will begin this paper by considering the problem of *adaptive thresholding* but ultimately shift our attention to the more general problem of segmentation.

The most straight-forward approach to thresholding is to pick a fixed greyscale value (the threshold) and classify each image pixel by checking whether it lies above or below this value. This technique works reasonably well when the image intensities are distributed bimodally or when the region of interest exhibits intensities consistently above or below all other pixel intensities in the image.

One of the earliest analytical methods for choosing a threshold value, proposed by Chow and Kaneko [1], involved fitting a pair of Gaussian curves to the image histogram (which is assumed to be bimodal) and then choosing the threshold to minimize the probability of misclassification. Extensions and less computationally intensive alternatives were later presented in [2]–[5]. While proposing their method, Chow and Kaneko

noted that applying a single threshold to the entire image was not effective for spatially varying backgrounds. For such cases they proposed dividing the original image into subregions, finding the best local thresholds for each subregion, and then interpolating between these local threshold values to construct a global *thresholding surface* or *adaptive threshold*.

A variety of techniques have been proposed for adaptive thresholding. Yanowitz and Bruckstein [6] obtain an adaptive threshold by noting that pixel intensities near the transitions between foreground and background (edge pixels), in a smoothed image, serve as the best local thresholds. They locate such pixels by checking for large gradients and interpolate the greyscale values of these pixels to form the thresholding surface. There have, of course, been many other interesting approaches to adaptive thresholding which lie beyond the scope of our discussion. See, for example, [7]–[12] for further examples, more extensive discussions, and additional references.

Our model is motivated by the observation of Yanowitz and Bruckstein [6] that information about the best local threshold is to be found near image edges (transitions between foreground and background) *on a smoothed version of the image*. The smoothing step is crucial since the greyscale values on either side of a sharp edge form very poor thresholds when compared to the “average” greyscale value encountered in the middle of a smoothed edge. The question is: “What is the best way to smooth?” In this paper, we suggest that the answer to this question is to use a diffusion model that is specifically designed to spread an edge apart in order to most quickly and effectively draw out this local discriminating information.

In particular, we will utilize an anisotropic diffusion model which, in contrast to earlier anisotropic diffusion models, diffuses specifically *across* image edges as opposed to *along* image edges. The latter behavior, exhibited by the well known *geometric heat flow*, is desirable for image denoising, but the former behavior is better suited to adaptive thresholding and segmentation since it has this desired effect of “spreading out” the edge information as fast as possible. Since the model we use diffuses orthogonally to the geometric heat flow, we refer to it as the *anti-geometric heat flow* and exploit its behavior to detect and classify pixels near image edges quickly and accurately. Remaining pixels (far away from edges) may be classified either by additional diffusion, interpolation, or region growing (as in Intensity Gradient Based Thresholding [8]).

Not only does this diffusion allow us to quickly detect pixels in the vicinity of an edge, but it also discriminates between regions on opposite sides of an edge. This enjoys many advantages over more traditional edge detectors that are designed to capture edge points directly, including better localization (since we are

Manuscript received June 29, 2001; revised March 12, 2003. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Jezekiel Ben-Arie.

The authors are with the School of Electrical and Computer Engineering, Georgia Institute of Technology Atlanta, GA 30332 USA (e-mail: smanay, ece98@gtalumni.org; ayezzi@ece.gatech.edu).

Digital Object Identifier 10.1109/TIP.2003.818039

not trying to assign pixel locations to edges that almost always lie *between* pixels), less sensitivity to noise (since we are diffusing the data instead of differentiating it), and better connectivity (since edges are now given by the boundaries between detected regions). However, since edges are represented indirectly in this framework, the applications which benefit most from this model are those which utilize region representations as opposed to direct edge representations. Adaptive thresholding and segmentation constitute two such applications.

The remainder of this paper is organized as follows. In Section II we discuss the diffusion model that will be utilized for adaptive thresholding and segmentation. In Section III we then outline how this diffusion process can be used to classify pixels into foreground and background regions for adaptive thresholding, or into arbitrary region types for segmentation. An important note to make ahead of time about this approach is that classifications will be made *during the diffusion process* as opposed to afterwards, thus avoiding the notorious problem associated with most diffusion methods, namely, “When should you stop the diffusion?” In the framework to be discussed, this question is naturally answered by “When enough pixels have been classified.” In Section IV we shift our attention to the more general problem of image segmentation and show how this diffusion model may be used in conjunction with a simple, fast region merging algorithm to segment very general classes of greyscale imagery (i.e., not just bimodal images consisting of foreground and background regions); Section V discusses the design of a fast implementation of these methods. In Section VI we compare this method to another segmentation method on synthetic images before showing experimental results on a variety of real images in Section VII. Finally, we discuss current and future research on these methods in Section VIII before concluding in Section IX.

## II. ANTI-GEOMETRIC DIFFUSION

In this section we discuss how diffusion may be used for adaptive thresholding and propose a diffusion model which seems to be ideally suited for this purpose.

A standard adaptive thresholding technique is to form a *threshold surface* over the domain of an image and then classify image pixels based upon whether their values lie above or below this surface. A straight-forward method for constructing the threshold surface is simply to blur the image with a Gaussian low-pass filter. This is equivalent to diffusing the image via the linear heat equation, giving rise to a family of threshold surfaces which comprise a well known scale space [13]. Consequently, choosing any particular threshold surface from this continuum imposes a certain scale on the features that are captured in the resulting binarized image. Near an image edge, a local average (fine scale) will yield an effective threshold, whereas away from an edge, a more global average (coarse scale) is necessary. It is not always clear which scale to choose.

Before addressing the ambiguity of scale (see Section III), we note that it is natural to generalize this basic thresholding algorithm by using anisotropic diffusion. Anisotropic diffusion possesses the advantage of allowing local control over the diffusion direction. This is particularly important where salient image features are concerned. When the preservation of sharp edges is im-

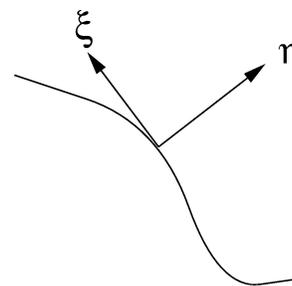


Fig. 1. Normal and tangent directions to a level curve.

portant (as in image denoising), it is natural to consider models which diffuse along, but not across, the edge directions.

Typically, edge directions are related to the tangents of the iso-intensity contours (level curves or level sets) of an image  $I$ . Let  $\eta$  denote the direction normal to the level curve through a given point (the gradient direction), and let  $\xi$  denote the tangent direction (see Fig. 1). We may write these directions in terms of the first derivatives of the image  $I_x$  and  $I_y$  as

$$\eta = \frac{(I_x, I_y)}{\sqrt{I_x^2 + I_y^2}}, \quad \xi = \frac{(-I_y, I_x)}{\sqrt{I_x^2 + I_y^2}}. \quad (1)$$

Since  $\eta$  and  $\xi$  constitute orthogonal directions, we may express the rotationally invariant Laplacian operator as the sum of the second order spatial derivatives  $I_{\eta\eta}$  and  $I_{\xi\xi}$  in these directions and write the linear heat equation as

$$\frac{\partial I}{\partial t} = \nabla \cdot (\nabla I) = I_{\xi\xi} + I_{\eta\eta}. \quad (2)$$

This decomposition of the linear heat equation has been considered in many earlier works on anisotropic diffusion (see [14]–[19] for a few examples).

Omitting the normal diffusion while keeping the tangential diffusion yields the well known *geometric heat flow*, which diffuses along the boundaries of image features but not across them. It derives its name from the fact that, under this flow, the level curves of the image evolve in the normal direction in proportion to their curvature. This model is well known for its ability to denoise images while maintaining sharp edges and is therefore widely used for image enhancement and smoothing. For more extensive discussions of the many properties of this flow and related flows see [20], [14], [18], [21]–[24].

The very property which makes the *geometric heat flow* powerful for image denoising (i.e., its ability to preserve edges in the image) makes it a poor flow for constructing adaptive thresholding surfaces, in which case we actually want to *smear* the image edges. If, instead, we omit the tangential diffusion and keep the normal diffusion, we obtain the complementary diffusion model, which we will refer to as the *anti-geometric heat flow*<sup>1</sup>

$$\frac{\partial I}{\partial t} = \frac{I_x^2 I_{xx} + 2I_x I_y I_{xy} + I_y^2 I_{yy}}{I_x^2 + I_y^2} \quad (3)$$

<sup>1</sup>The term *nongeometric* is used by Evans in a paper [25] outlining some mathematical properties of this and other flows related to the  $p$ -Laplacian.

in which diffusion occurs deliberately across the boundaries of image features. This is precisely what we want to occur when constructing an adaptive thresholding surface. By omitting the tangential *geometric* component of the diffusion, we also avoid the shrinkage of the isointensity contours that occurs in both the geometric and the linear heat flow. Intuitively, the family of isointensity contours which run through a given edge are spread apart, while the shapes of those that remain near the original edge location are less distorted than they would be under the curvature-based shrinkage that would be induced by the discarded tangential diffusion. Since diffusion in the direction orthogonal to the level sets of an image can be useful in any dimension (for instance, we present a volumetric segmentation example later on in this paper using the three dimensional version of this diffusion) we note that the general form of the anti-geometric heat flow is given by

$$\frac{\partial I}{\partial t} = \left( \frac{\nabla I}{\|\nabla I\|} \right)^T \nabla^2 I \left( \frac{\nabla I}{\|\nabla I\|} \right) \quad (4)$$

where  $\nabla^2 I$  denotes the Hessian of  $I$ . Experiments contrasting the the anti-geometric, geometric, and isotropic flows are demonstrated in Section VI-A .

#### A. Remark

As anisotropic diffusion and the use of partial differential equations for image denoising has become a rather mature field, there are many models, other than the geometric heat flow, that have been considered for image smoothing and denoising (such as the seminal work of Perona and Malik [26]). This includes hyperbolic PDEs which form the basis of continuous morphology (as in the early work of Kimia *et al.* [27] and the recent work of Meyer and Maragos [28]) as well as parabolic PDEs, like those considered in this work, which underlie most diffusion models. While a complete discussion of such previous work is well beyond the scope of this paper, we do wish to point out that the trend in this body of work has generally been to *preserve features* in the image while denoising or otherwise simplifying the image data. We go against this trend by considering a model which deliberately smears features in order to quickly reveal their locations within the image. In short, our model is designed to *detect features* in a manner which allows us to quickly construct actual segmentations of the image (these steps will be discussed in the sections that follow) rather than to preserve features and eventually obtain a pseudo-segmentation<sup>2</sup>.

### III. THRESHOLDING VIA DIFFUSION

In this section we discuss how to use the anti-geometric diffusion process described in Section II for adaptive thresholding. One straight-forward idea is simply to use an anti-geometrically diffused version of the original image as a thresholding surface and then classify the image pixels accordingly. This approach, however, suffers from an arbitrary choice of diffusion time (which relates to a particular scale).

Rather than using a traditional thresholding surface, we instead seek to classify pixels *during the diffusion process*. Early

in the diffusion process, only the intensity values of pixels near object boundaries change significantly, allowing for a confident classification of such pixels as locally light or dark based upon their diffusion behavior. As the diffusion proceeds, its more global averaging effects allow us to classify pixels further away. Unfortunately, if we wait long enough for diffused intensities of pixels far away from object boundaries to differ from their original intensities enough to yield a confident classification, then diffused intensities near boundaries of smaller features may switch from being brighter than their original intensities to darker than the original intensities (or vice-versa) due to the more global averaging effect of prolonged diffusion. In other words, the global effect of prolonged diffusion helps for classifying pixels far away from important image features, but could lead to less accurate classification of nearby pixels. If, however, we classify a given pixel *as soon as* its classification becomes unambiguous (i.e., the diffused and original intensities differ significantly) and *maintain* this classification as the diffusion proceeds, then we may run the diffusion as long as necessary to classify pixels far away from region boundaries without worrying about consistency problems for pixels that have already been classified.

In this manner, we are no longer utilizing a single thresholding surface, but an *entire family* of thresholding surfaces generated by our anisotropic diffusion model. This method is effective because pixels in regions with high detail (i.e., high spatial variance) change intensity relatively quickly during diffusion; their intensities differ by large amounts in short periods of time, and are therefore thresholded at a fine scale. Pixels in low-detail regions change intensity slowly, are thresholded at a much later time, and are therefore classified at a coarser scale.

#### A. Classification Criteria

A pixel's net intensity change is not the only criterion that may be used to decide when to classify it during the diffusion process. If an image contains edges that are not of uniform contrast, then the main benefit of adaptive thresholding, namely avoiding a single constant threshold for the entire image, is counteracted in this scheme by requiring a single constant change in diffused intensity (another threshold) to decide when and how a pixel should be classified. In cases where a single value will not suffice near all image edges<sup>3</sup> a better criterion would be to check whether a pixel's diffusing intensity value is consistently increasing or decreasing. In other words, rather than waiting for a fixed change in the diffused intensity to decide whether a pixel is a light pixel near the boundary of a dark region or vice-versa, one waits for a fixed period of time in which the diffusion is monotonic at that pixel location.

The advantage of thresholding based on monotonicity is that this criterion is not sensitive to the magnitude of the intensity changes but is nevertheless robust to noise. To see this, imagine an isolated bright pixel of noise on a dark background but near the boundary of a large bright region. The pixel's diffusing intensity value will initially decrease by a huge amount as the noise is smoothed away. Soon, however, the pixel's intensity will start increasing due to the diffusion of the large bright re-

<sup>2</sup>We use the term pseudo-segmentations to refer to segmentation-like effect which comes from running an edge-preserving diffusion long enough.

<sup>3</sup>The danger of merely choosing a tiny jump that is small enough to work for even the faintest image edges is sensitivity to noise.

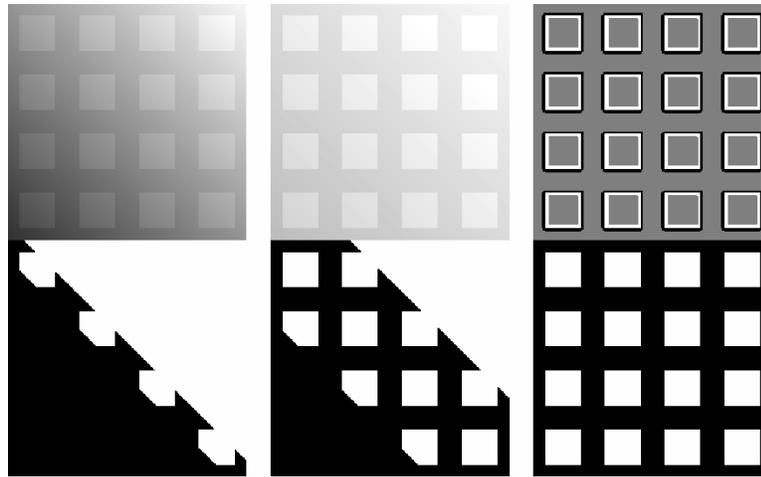


Fig. 2. *Thresholding via anti-geometric diffusion.* LEFT COLUMNS: [top] two different shaded images, [bottom] thresholded images obtained with a fixed threshold. RIGHT COLUMN: [top] partial thresholding of either image using anti-geometric diffusion, [bottom] final thresholded image by extending the classifications obtained above.

gion nearby. Although the latter increase in intensity may be much less than the initial decrease in intensity, the initial decrease happens quickly whereas the latter increase may continue steadily for a long time. Thus, classification based upon absolute intensity changes would erroneously classify the noise as part of a bright region whereas classification based upon monotonicity would correctly classify the noise as part of a dark region.

Finally, pixel classification need not be based upon just one criterion. A combination of monotonicity requirements and absolute change in intensity requirements can yield added robustness in the presence of noise.

### B. Robust Edge-Detection

The fact that pixels near edges are classified quickly during the anti-geometric diffusion process means that a short application of this diffusion/classification procedure essentially captures most, if not all, of the interesting edges and therefore acts as an edge-detector. However, unlike more traditional edge detectors, which typically apply gradient-based operators to an image in order to locate edges directly, we apply smoothing operators to the image, which have better robustness to noise compared to gradient-based operators, in order to locate regions on opposite sides of edges and thereby capture edges indirectly. Two additional advantages to this type of edge detection are that edge points are properly modeled as lying *between pixels* and that we generally obtain better connectivity since we are capturing entire regions (with connected boundaries) as opposed to point measurements which later have to be linked together.

A deeper insight into the edge-detecting behavior of this diffusion/classification procedure may be obtained by first considering the *linear* heat flow. Zero-crossings of the Laplacian of the image data are often used as indicators of an edge, and in the linear heat flow, it is precisely the sign of Laplacian that governs the diffusion direction. Therefore, near a zero-crossing, the diffusion will increase intensities on one side and decrease intensities on the other side of the associated edge. The problem with using the Laplacian, which corresponds to the instantaneous effect of the linear heat flow, is its sensitivity to noise as a second order differential operator. A standard remedy to elim-

inate the resulting number of spurious falsely detected edges is to convolve the image with a Gaussian kernel before finding the Laplacian zero crossings (the famous *Laplacian of Gaussian* operator). This, of course, is also related to the linear heat flow since running the linear heat flow for time  $t$  is equivalent to convolving with a Gaussian of variance  $\sigma^2 = 2t$  (see [29]). The main disadvantage of using the linear heat flow for this purpose is that it distorts the shape of the Laplacian zero-crossings due to the curvature based shrinkage of the image level sets. This effect, which leads to bad edge localization, is exclusively due to the geometric component of the linear heat flow, which is discarded in the anti-geometric model.

### C. Extending Early Classifications

We may capitalize on this edge-detecting behavior by noting that early in the diffusion/classification process, whatever regions remain unclassified are likely to be homogeneous. Confident classifications of such regions may therefore be extrapolated from their boundaries (i.e., neighboring pixels that were classified during the early stages of diffusion). We may therefore speed up the classification process by only running the anti-geometric diffusion for a short amount of time or until a given percentage of the pixels are classified, and then extending these classifications to the remaining pixels. For adaptive thresholding applications, these classifications are binary (white/black, 0/1, or foreground/background). A simple way to extend these binary classifications into the unclassified regions is by region growing: unclassified pixels adjacent to white pixels become white pixels while those adjacent to black pixels become black pixels (repeating as necessary until all pixels are classified).

This method is illustrated on a synthetic image in Fig. 2. The first two columns show a shaded image of 16 squares on the top followed by a thresholded version on the bottom (classified via a fixed threshold). The shading is less severe in the second column, allowing a fixed threshold to capture more squares than in the first column. In both cases, however, it is impossible to extract all 16 squares with a simple threshold. The top of the third column shows bright and dark pixels (shown in white and



Fig. 3. Presegmentation via anti-geometric diffusion. LEFT: Original Cardiac MRI. MIDDLE: Image oversegmented with a short run of diffusion and low thresholding tolerance. RIGHT: Image undersegmented with a long run of diffusion and high thresholding tolerance.

black) on opposite sides of the square boundaries as detected by a very small amount of anti-geometric diffusion (this output is the same for both the mildly and severely shaded images). These classifications are then extended by simple region growing to obtain the final image on the bottom of the third column.

For the more general problem of segmentation (where we assume more than just two region types), we propose a very different procedure to extend early classifications to remaining unclassified pixels. In fact, for segmentation, we do not regard pixels away from edges that were not “detected” during the initial diffusion/classification step as unclassified, but instead we regard such pixels as belonging to yet another class.

#### IV. A METHOD FOR FAST SEGMENTATION

In this section we outline a procedure that uses anti-geometric diffusion in conjunction with region merging for fast segmentation of greyscale images that cannot be easily classified into merely foreground and background (and therefore are not appropriate candidates for adaptive thresholding).

The procedure begins in the same manner discussed in Section III-B, namely, we run anti-geometric diffusion for a short amount of time and then classify pixels into three different categories: (1) pixels whose intensities diffused significantly (and/or monotonically) upward, (2) pixels whose intensities diffused significantly (and/or monotonically) downward, and (3) pixels which did not diffuse significantly. However, instead of considering pixels in categories (1) and (2) as classified and pixels in category (3) as unclassified, we regard all pixels as “equally” classified. We do so by giving each connected region  $R_i$  (regardless of which category (1)-(3) it initially belonged to) a unique label  $i$ . We now have an over-segmentation of the image. In other words, this use of anti-geometric diffusion essentially acts as a region splitting operator, breaking up the initial image domain into smaller regions.

Note, that the degree of oversegmentation depends greatly upon the specific parameters (such as minimum net intensity change or monotonicity requirements) used for classification into categories (1) and (2). Parameters requiring only minor changes during diffusion for classification into categories (1) or (2) result in rapid segmentation of many small-scale features including both desired features and noise. Parameters requiring more significant diffusion behaviors prior to classification re-

duce the effect of oversegmentation but with two prices. First, classification of a given percentage of image pixels requires more diffusion (and therefore more time and computation) when more restrictive classification conditions are imposed. Second, less prominent features may not be detected, causing an undersegmentation of some regions in the image. Fig. 3 demonstrates the extremes of oversegmentation and undersegmentation by utilizing drastically different classification parameters on a cardiac MRI image (regions in category (1) are shown in white, regions in category (2) are shown in black, and regions in category (3) are shown in gray).

However, because the regions in categories (1) and (2) lie on opposite sides of image edges and because regions in categories (3) are unlikely to contain or border any edges, we should be able, through strategic grouping and merging of these regions, to obtain decent coarser segmentations of the image (thereby addressing the problem of oversegmentation). We suggest using energy based region merging criteria as suggested both in [30] and [31].

If we approximate the image data  $I$  within each region  $R_i$  by its mean<sup>4</sup>  $\mu_i$ , then we may measure the total squared error  $E$  between our piecewise constant segmentation and the original image

$$E = \sum_i E_i \quad \text{where} \quad E_i = \sum_{R_i} (I - \mu_i)^2. \quad (5)$$

This total error will generally increase (but never decrease) if we merge any two regions  $R_i$  and  $R_j$  into a single new region  $R_{ij} = R_i \cup R_j$  and replace the old means  $\mu_i$  and  $\mu_j$  by the new mean  $\mu_{ij}$  of  $I$  over the combined regions. The nonnegative change in the error  $\Delta E_{ij}$  is given by

$$\Delta E_{ij} = E_{ij} - E_i - E_j, \quad \text{where} \quad E_{ij} = \sum_{R_{ij}} (I - \mu_{ij})^2. \quad (6)$$

A sensible criterion is then to merge the regions  $R_i$  and  $R_j$  which yield the smallest increase  $\Delta E_{ij}$  in the total squared error. Once this is done, we have reduced the number of regions in our segmentation by one. We may then repeat this procedure as many times as necessary until the desired number of regions

<sup>4</sup>This piecewise constant approximation is naturally coupled to our diffusion model as the heat flow drives initial data toward its mean.

is obtained, or until we detect a significant jump in the total squared error. While it might seem at first glance that the number of pairs to check is quite large, it is really not as bad as it seems. The reason is that we need not consider all possible pairs but only regions which are neighbors (so that merging them eliminates a connected region from the segmentation). This reduces the complexity of the search from  $O(n^2)$  to  $O(n)$  where  $n$  is the total number of regions.

Undersegmentation cannot be remedied by region merging but can be counteracted by further splitting of initially detected regions. In particular, we propose applying anti-geometric diffusion exclusively within regions  $R_i$  exhibiting large squared errors  $E_i$  (since such regions are likely to constitute the undersegmented regions) in order to refine the classification of pixels within these regions. This time features may be detected that were missed the first time around due to interference from structures in nearby regions.

After such refined splitting, the image is probably oversegmented again and requires another merging step to reduce the number of regions. Since the merging procedure is designed to merge those regions that contribute to the squared error the *least* (whereas the splitting procedure is designed to further partition regions which contribute to the squared error the *most*), the following merging step will not merely “re-assemble” the regions that were just further split using the diffusion model. This intuition leads us to an algorithm (based on those discussed in [30]) that integrates global splitting, merging, and local splitting in an iterative manner to find more accurate segmentations of an image. The overall algorithm is summarized as follows.

1. Use anti-geometric diffusion to split the entire image domain into smaller regions.
2. Merge pairs of regions in the image until there are  $r$  (a desired number chosen in advance) regions left or until the total squared error becomes too high.
3. Split the region with the highest squared error using anti-geometric diffusion exclusively within this region.
4. Repeat steps 2 and 3 until the total squared error converges (for example, when  $\Delta E/E$  becomes small).

This iterative algorithm is demonstrated in Fig. 4. We use the undersegmentation shown in Fig. 3 as a starting point to demonstrate the robustness of this method, even though starting with the oversegmentation would yield convergence in fewer iterations with better results. The middle image in each row shows the result of merging the current segmentation down to 20 regions (the constant greyscale value displayed within each region corresponds to the mean value of the data within that region). The middle images are followed, in the right column, by images which display the value of the normalized squared error within each of the 20 regions. The region with the highest squared error (shown in white) is then split via additional anti-geometric diffusion, to yield the new segmentation on the left of the next row (below).

The iterative approach has several distinct advantages. First, iterative splitting and merging provides strong robustness to choices of classification parameters. Second, applying the diffusion within a specific region forces the model to detect “soft” edges or fine scale features that may have been missed by the global diffusion.

## V. IMPLEMENTATION

In this section we discuss the implementation of the models discussed in the previous sections of the paper. We pay special attention to implementation details that will result in fast algorithms.

### A. Implementation of Diffusion Based Splitting Algorithm

The numerical implementation of anti-geometric diffusion consists of two steps per iteration of diffusion. First, the image values at each pixel location  $[i, j]$  are actually diffused for one timestep to yield new diffused image values. Second, pixels which meet specified classification criteria are classified according to their diffusion behavior. Once a pixel has been classified, its classification is maintained. The algorithm iterates until a stop condition is met; the authors’ preferred stop condition is that a chosen percentage of the pixels in the image have been classified.

The discrete implementation of each anti-geometric diffusion step utilizes a standard forward Euler step

$$I[i, j, t + \Delta t] = I[i, j, t] + \Delta t I_{\eta\eta}[i, j, t] \quad (7)$$

where

$$I_{\eta\eta} = \frac{A + 2B + C}{I_x^2[i, j, t] + I_y^2[i, j, t]}, \quad \text{where}$$

$$A = I_x^2[i, j, t] I_{xx}[i, j, t],$$

$$B = I_x[i, j, t] I_y[i, j, t] I_{xy}[i, j, t], \quad \text{and}$$

$$C = I_y^2[i, j, t] I_{yy}[i, j, t]. \quad (8)$$

Note that because the diffusion is well-posed, there is no need to formulate generalized solutions that require special finite differencing schemes. Standard central differences may be employed to compute the spatial derivatives in (8) needed for the Euler step (7) with stable results (just as they may be used in the discrete implementations of both the linear and geometric heat flows)

$$I_x[i, j, t] = \frac{I[i + 1, j, t] - I[i - 1, j, t]}{2\Delta x},$$

likewise for  $I_y[i, j, t]$ ,

$$I_{xx}[i, j, t] = \frac{I[i + 1, j, t] - 2I[i, j, t] + I[i - 1, j, t]}{(\Delta x)^2},$$

likewise for  $I_{yy}[i, j, t]$ , and

$$I_{xy}[i, j, t] = \frac{A - B}{4\Delta x \Delta y}, \quad \text{where}$$

$$A = I[i + 1, j + 1, t] + I[i - 1, j - 1, t], \quad \text{and}$$

$$B = I[i + 1, j - 1, t] + I[i - 1, j + 1, t].$$

The stability condition for choosing  $\Delta t$  is similar to that for the geometric heat flow; namely,  $\Delta t$  should be chosen such that  $\Delta t < .5(\Delta x)^2$ .

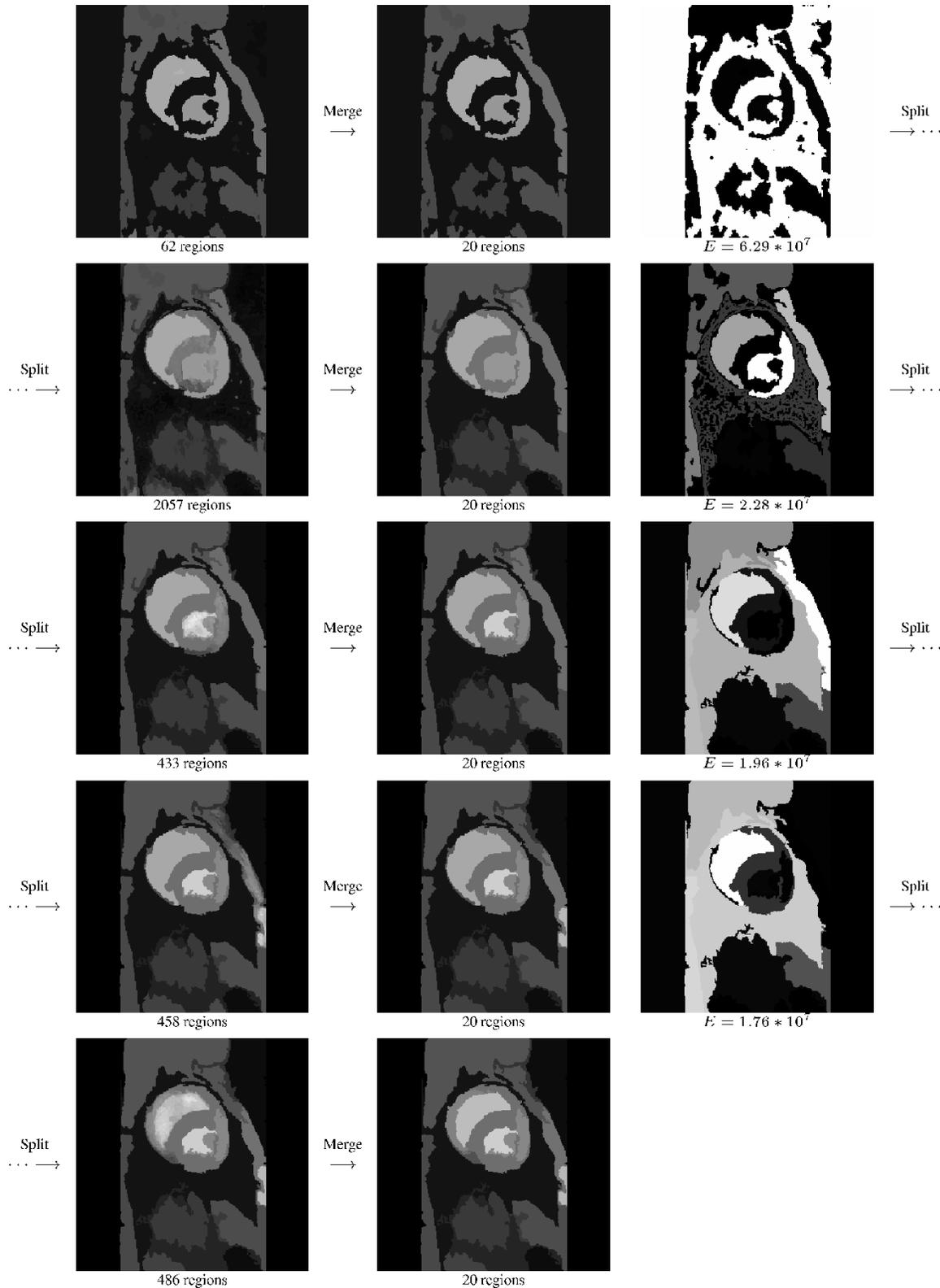


Fig. 4. Six iterations of splitting and merging of cardiac MRI (Fig. 3); each row is one iteration. [LEFT] Image after splitting. [MIDDLE] Image after merging. [RIGHT] Merged image with regions colored by  $E_i$ . Note that the white region (highest  $E_i$ ) at the end of each row is the region split at the beginning of the next row.

After each diffusion step, we check each unclassified pixel to see whether or not it may be classified. Various classification criteria are discussed in Section III. The simplest criterion is to check whether a pixel's original intensity value lies above or

below the diffused value by a given tolerance (in this manner, the diffused image is used like an adaptive thresholding surface). Another criterion is to track the length of time a pixel's diffused value changes monotonically. If this time exceeds a pre-deter-

mined threshold, the pixel can be robustly classified. Further, these two criterion can be combined. (In our experiments, we combine the two criteria in a linear discriminant as discussed in Section VI.)

It is important to reiterate that not every pixel gets classified at the same time. Some are classified earlier in the diffusion than others depending upon the nature of the data, the behavior of the diffusion, and the exact classification criteria employed. Once a pixel is classified, it continues to diffuse, but is never reclassified. To avoid re-checking or re-classifying a pixel, the algorithm maintains a list of unclassified pixels. When a pixel is classified, it is removed from the list; the algorithm only attempts to classify pixels remaining on the list.

### B. Fast Implementation of Region Merging

In this section we outline one possible implementation of region merging, paying careful attention to details which minimize the computational expense of this portion of the overall segmentation algorithm. Recall that the region merging step is performed after pixels have been classified according to the behavior under anti-geometric diffusion. This step is only necessary for segmentation, not adaptive thresholding for which the classifications obtained are already sufficient to construct the desired binary image. The overall region merging procedure consists of the following steps.

1. Initially represent the image data within each connected region  $R_i$  of identically classified (or "unclassified") pixels by the mean intensity  $\mu_i$  of the original image data within that region. Then determine the squared error  $E_i$  between this representation and the original image data for that region.
2. For each pair of adjacent regions  $R_i$  and  $R_j$ , calculate  $\Delta E_{ij}$  (the increase in the total squared error) that would result from merging  $R_i$  and  $R_j$  into a single new region  $R_{ij}$ .
3. Merge the pair which contributes the smallest total error increase  $\Delta E_{ij}$ .
4. Repeat steps 2 and 3 as desired.

To save time during the merging process, the image data should be stored in a data-structure that allows for the fast calculation of  $\Delta E_{ij}$ . The authors' preferred datastructure is a graph, where each node corresponds to a region in the image and a connection between two nodes indicates that the corresponding regions are adjacent in the image. Each node of the graph contains a linked list of all the pixels in the region. An array of such nodes, where each node is a datastructure containing the statistics described below and a list of adjacent nodes (i.e., the connections of the graph), represents the entire graph in a manner that allows random access of the nodes. The graph is constructed while performing an initial connected component analysis of the classified and unclassified pixels. Once the graph has been constructed, searching for the pair

of adjacent regions is a matter of visiting each node and calculating  $\Delta E_{ij}$  for each of its adjacent nodes.

The key to implementing the region merging procedure quickly is to realize that one need not scan through the image data over and over again to compute  $\Delta E_{ij}$  for each pair of adjacent regions  $R_i$  and  $R_j$ . Only one initial scan through the image data is needed during the construction of the initial graph structure to compute a set of statistics  $A_i$ ,  $S_i$ ,  $Q_i$ ,  $\mu_i$ , and  $E_i$  for each initial region  $R_i$ .

- $A_i$ : number of pixels (area) within region  $R_i$ ;
- $S_i$ : sum of image intensities within region  $R_i$ ;
- $Q_i$ : sum of squared image intensities within region  $R_i$ ;
- $\mu_i = S_i/A_i$ : mean image intensity within region  $R_i$ ;
- $E_i = Q_i - 2\mu_i S_i + \mu_i^2 A_i$ : squared error for region  $R_i$ .

From then on, the statistics of a proposed region  $R_{ij} = R_i \cup R_j$  may be immediately computed using the statistics of  $R_i$  and  $R_j$  without revisiting the image data by using the following relationships.

- $A_{ij} = A_i + A_j$ ;
- $S_{ij} = S_i + S_j$ ;
- $Q_{ij} = Q_i + Q_j$ ;
- $\mu_{ij} = S_{ij}/A_{ij}$ ;
- $E_{ij} = Q_{ij} - 2\mu_{ij} S_{ij} + \mu_{ij}^2 A_{ij}$ ;
- $\Delta E_{ij} = S_i \mu_j + S_j \mu_i - S_{ij} \mu_{ij}$ .

Technically, we see from this last relationship that only  $A_i$  and  $S_i$  (which automatically gives  $\mu_i$ ) need to be computed and updated for each region since these are the only quantities needed to evaluate the *change* in total squared error. However, it is also useful to know the actual value of the total squared error (to decide if additional splitting of regions is necessary or to decide when to stop merging regions for example), and thus one might as well keep track of  $Q_i$  and  $E_i$  since the extra computational expense is negligible.

The full merging procedure, if carefully implemented as described here, is  $O(n^2)$  assuming we continue merging until we are back to a single region (of course one should stop earlier than this). If we start with every pixel in an image representing a separate region, then this is an expensive procedure on a standard  $256 \times 256$  image which would contain over 65 000 regions. This is why using anti-geometric diffusion for an initial region splitting allows us to perform the merging very quickly. Even if the output of this region splitting technique yields 1000 individual regions, we are still considering an operation count on the order of 1 million (which only takes a second or two on most computers today) compared to an operation count on the order of 4 billion if we start with each image pixel: a full three orders of magnitude worse!

## VI. COMPARISONS

In this section we compare anti-geometric diffusion to several other diffusion models and a nondiffusion-based splitting model.

### A. Comparison of Diffusion Models

In Fig. 5 we illustrate the contrasting behaviors of the linear, geometric, and anti-geometric heat flows on a cardiac MR image (using equal diffusion times in each case). In the top row

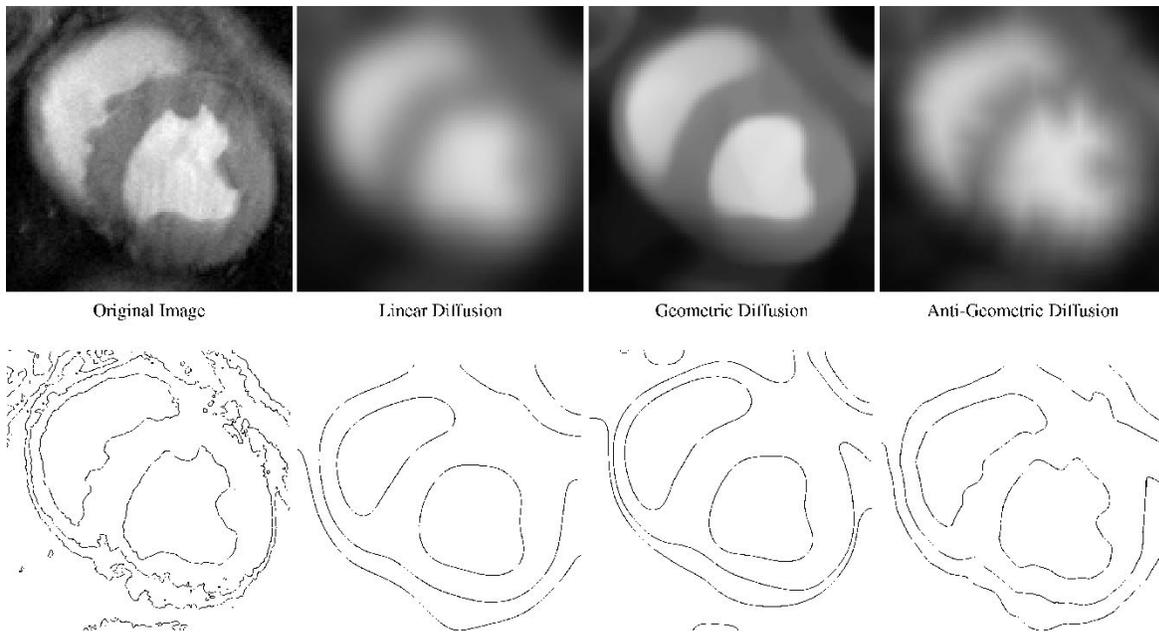


Fig. 5. Comparison of Linear, Geometric, and Anti-Geometric Diffusion. [TOP ROW] Original and diffused versions of a cardiac MR image. [BOTTOM ROW] Some level sets of the original and diffused images.

we see that both the linear (center left) and anti-geometric (far right) diffusions result in significant blurring of the edges while the geometric diffusion (center right) causes very little blurring of the prominent edges. Note, though, that the geometric diffusion caused a significant distortion in the *shape* of the edges even though the edges were not severely blurred as they were by the anti-geometric diffusion. This shape distortion is due to the curvature based shrinking of the image level sets incurred by the geometric heat flow. By comparing level sets of the geometric and anti-geometrically diffused images with those of the original image (more precisely, the level sets whose values lie close to the midpoint between the intensities on opposite sides of the prominent edges) we see in the bottom row that the anti-geometric diffusion has done a better job of preserving the *shape* of prominent image structures (particularly the two bright blood pools) despite the fact that it has significantly blurred the edges. This can be observed even in the top row by noting that it is still possible to make out the original shape of the endocardium in the left ventricle (the boundary of the lower white region) even though the boundary has been significantly blurred. In both the geometric and linear cases, the original shape has been lost (particularly the three concavities at 12, 3, and 6 o'clock).

Obviously, the anti-geometric flow is a much better model for smearing than the geometric heat flow. We also see from the above discussion that there are reasons for favoring it over the linear heat flow as well. We further illustrate the advantage of the anti-geometric flow over the linear heat flow in Fig. 6 by partially thresholding a synthetic volumetric image of an elongated black ellipsoid on a white background (its central cross section is shown on the left) using adaptive thresholding surfaces generated by both the linear (middle) and anti-geometric (right) heat flows (equal diffusion times). The thresholded images show pixels whose intensities diffused upward by more than ten greyscale values (original values were 0 inside the el-



Fig. 6. Anti-Geometric and Linear Diffusion for Adaptive Thresholding. [LEFT] Cross-section of binary ellipsoid image. [MIDDLE] Image partially thresholded using the linear heat flow. [RIGHT] Image partially thresholded using the anti-geometric heat flow. Pixels that diffused significantly upward (ellipse interior) are labeled black, pixels that diffused significantly downward (ellipse exterior) are labeled white. The regions that did not diffuse significantly (and therefore cannot be confidently classified) are labeled grey. See the text for more details about this figure.

lipse and 255 outside the ellipse) as black and pixels that diffused downward more than ten values as white (other pixels are shown in gray). Both diffusions smear the bright intensities from outside the original ellipsoid into the interior of the ellipsoid, allowing for equally easy classification of the interior pixels (black), whereas the anti-geometric diffusion does a better job of smearing dark intensities from inside the original ellipsoid into the exterior of the ellipsoid (particularly near the highly curved endpoints) allowing for much easier classification of exterior pixels (white). This is due to the fact that the linear heat flow has both a geometric and an anti-geometric component and the inward shrinking of the ellipse boundary incurred by the geometric component counteracts much of the outward smearing incurred by the anti-geometric component.

### B. Comparison of Segmentation Methods

In this subsection we demonstrate the effectiveness of the proposed classification method on a set of synthetic grayscale images with added noise.

To validate the segmentation method, anti-geometric and isotropic diffusion are each used in conjunction with region merging to segment  $128 \times 128$  synthetic images of squares of decreasing size from added Gaussian noise of increasing

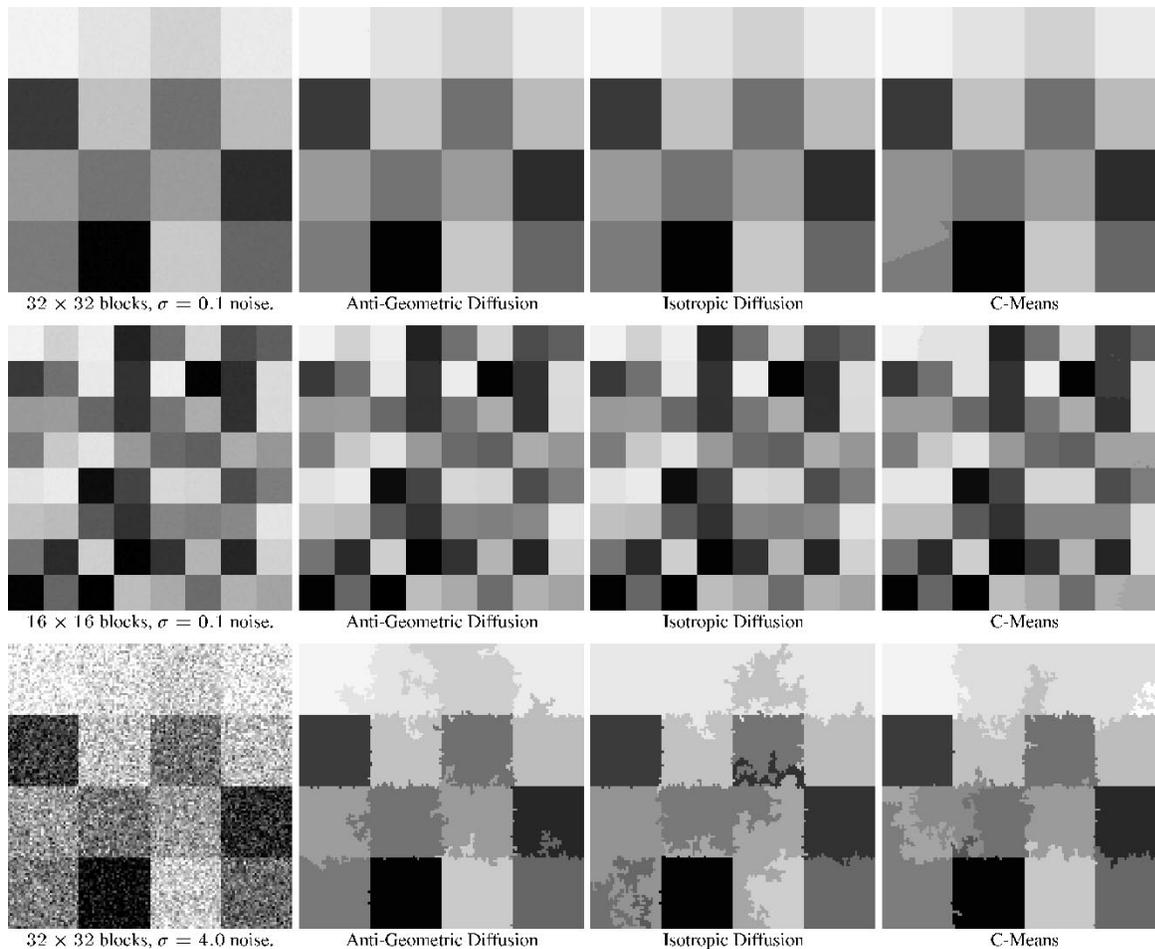


Fig. 7. Synthetic grayscale images segmented using anti-geometric diffusion, isotropic diffusion, and C-Means clustering. Note: Image values have been normalized to increase contrast.

variance. For comparison, we use the C-Means algorithm to cluster pixels based on a feature vector that includes intensity value, and  $x$  coordinate, and  $y$  coordinate. The intensity value is weighted by an empirically chosen factor ranging from 10 to 50 based on the noise level of the image. Though we have defined a region to include a set of (4-connected) pixels, C-Means may classify nonadjacent pixels as belonging to the same cluster, resulting in over-segmentations similar to splitting via diffusion. Again we employ region merging to compensate for this effect.

Fig. 7 shows several examples of images from this experiment. In each row, we first show the original grayscale image, followed by the image segmented via iterative anti-geometric diffusion and region merging, iterative isotropic diffusion and region merging, and finally C-Means clustering and region merging. For diffusion-based splitting, the images are diffused until 30% of the pixels are classified using the linear discriminant criteria  $\Delta I + ct_{monotone} > T$  with the coefficient  $c = 0.3$  and a threshold  $T = 1.0$ . (For the first iteration, these parameters may be different to account for the more global influences in the diffusion.) Region merging is used to reduce the number of regions; the final number of regions is chosen by counting the number of regions in the original image. The process is then iterated until total squared energy decreases by less than 99.99% per iteration. For C-Means clustering, the

image pixels are grouped into either 30, 50, or 200 clusters, based on the noise level of the image. The parameters for both methods are chosen so that the resulting oversegmentations contain similar numbers of regions, and never more than 5000 regions. In the first row of the figure, all three methods are competitive due to the large object size and the low noise. However, note in the second row of the figure that the C-Means method undersegments many of the adjacent objects with similar intensities (for instance, see the left and right borders of the image). In the noisy image in the third row, the differences are more pronounced. Even though edge localization is difficult due to the noise, anti-geometric diffusion is better able to segment each of the 16 regions.

The graphs in Fig. 8 shows the percentage of undersegmented objects in the above experiment as a function of the size of the objects and noise variance for the three segmentation methods used in this validation. We define an undersegmentation as the omission of a border between two regions, resulting in one less region that is expected. Since the expected number of regions is fixed, each undersegmentation is paired with an oversegmentation (the insertion of an extra border resulting in one extra region). Graph (a) demonstrates that region splitting via diffusion is robust to varying object sizes, even when the objects have an area of 4 pixels. Further, anti-geometric diffusion can more consistently localize edges when compared to isotropic diffusion.

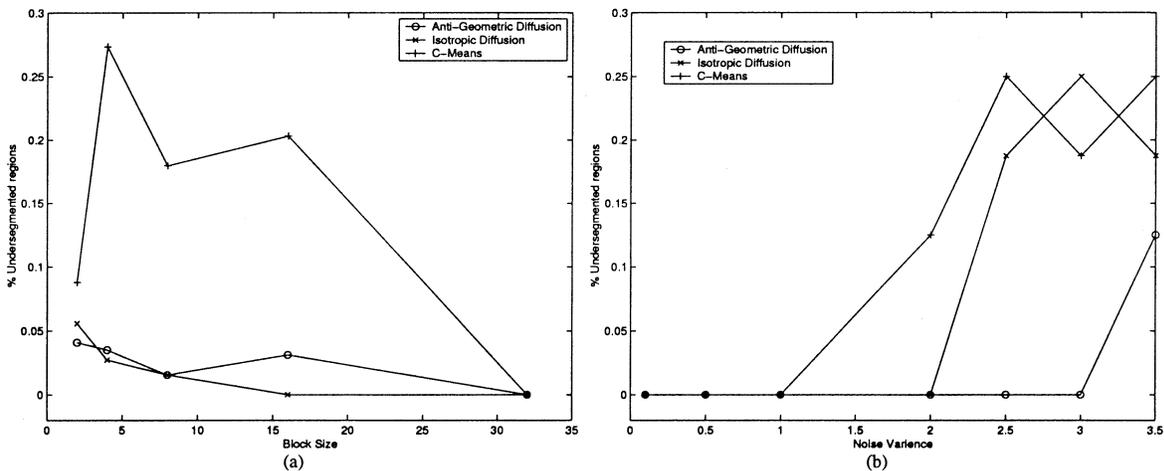


Fig. 8. Graph of undersegmentations per region as a function of (a) increasing block size and (b) increasing noise variance.

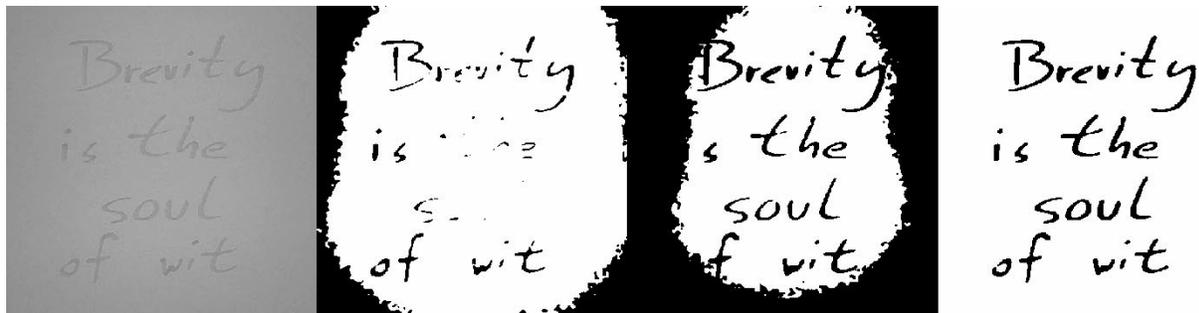


Fig. 9. Text image [LEFT] thresholded via two different fixed thresholds [MIDDLE] and via anti-geometric diffusion [RIGHT].

Anti-geometric diffusion as a splitting operator consistently distinguished between regions even when high noise levels make edge localization difficult, even when using isotropic diffusion, as shown in graph (b). For all these experiments, the average execution time for the diffusion and region merging algorithms was 40.4 seconds when using anti-geometric diffusion and 36.4 seconds when using isotropic diffusion. This is more than four times faster than an average time of 166.2 seconds when using the C-Means algorithm and region merging. These experiments were simulated using C++ on a Pentium IV 900 MHz processor running the Linux operating system.

Lastly, we note that it may be possible to achieve better results using the C-Means algorithm if we were to include it in an iterative framework, as we have for the diffusion-based splitting operators. However, due to the high execution times of the C-Means algorithm, an iterative approach would be far less practical.

## VII. APPLICATIONS AND SIMULATION

In this section we demonstrate the use of anti-geometric diffusion for classification of regions in grayscale images, starting with a low-contrast and heavily shaded image of handwritten text shown in Fig. 9. As seen in the two middle images, the shading makes it impossible to completely separate the text from the background using flat thresholds, while adaptive thresholding using anti-geometric diffusion successfully separates the two regions, as seen on the right.

In Fig. 10, the 2-D anti-geometric heat flow (3) is used in conjunction with region merging to segment five different grayscale images. The original images are shown on the left followed by three segmentations ranging from large to medium to small numbers of regions (i.e., fine to medium to coarse scale), each time displaying the mean image intensity within every region.

In Fig. 11 we compare the segmentation of the bone CT from Fig. 10 with a segmentation using a C-Means clustering algorithm as the splitting operator, as in Section VI. In this example, the two methods give similar segmentations, although anti-geometric diffusion localizes the edges better (for instance, the edges of the thin, bright bone cross sections in the center of the image). Further, the diffusion-based method runs in far less time (79 seconds for the diffusion-based method compared to 494 seconds for C-Means clustering).

In Fig. 12, the 3-D anti-geometric heat flow (4) in conjunction with region merging to segment a volumetric cardiac MRI sequence (two spatial dimensions and one time dimension). Both the diffusion and the region merging were performed fully in three dimensions. The results, however, are shown as a series of 2-D images. Each row represents one particular slice in time. The original image is on the left, followed by two segmentations. The first (middle) segmentation is obtained automatically by merging down to 10 volumetric regions. The second (right) segmentation is obtained by user-interactive splitting and merging of the middle result down to 7 regions. (For all the examples, the final number of regions is chosen empirically.)

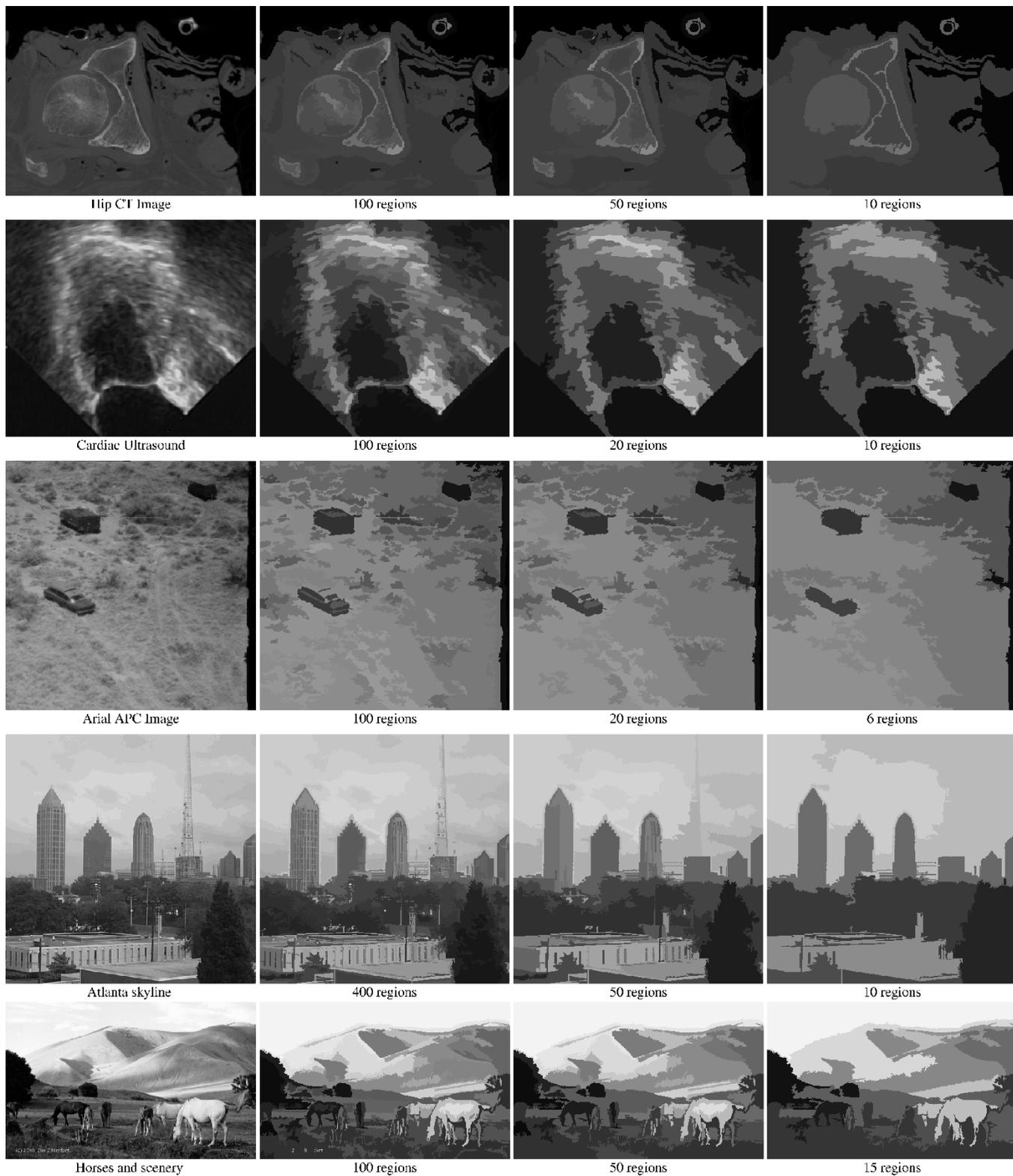


Fig. 10. Greyscale images segmented using anti-geometric diffusion and region merging.

The user-interactive stage demonstrated in Fig. 12 involves a user clicking with the mouse on a particular region to be further split or by clicking on a pair of regions to be merged. Such interaction is helpful for clinical segmentation because merging regions according to their squared error does not always correspond to the segmentation desired by a trained cardiologist.

### VIII. CURRENT AND FUTURE RESEARCH

The fact that piecewise constant segmentations governed by minimal total squared error do not always represent segmentations perceived by the human visual system motivates one of current research directions. Representing the image data within

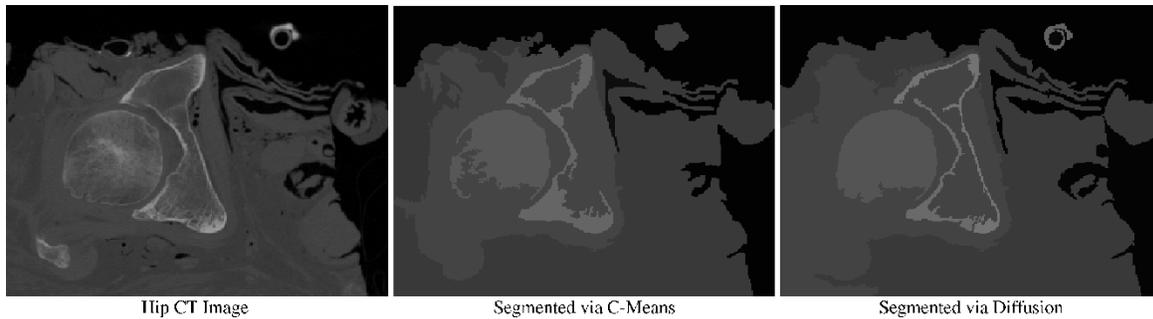


Fig. 11. Grayscale image of a bone CT segmented using two methods. The C-Means algorithm executed in 494 seconds while the Diffusion-based method took 79 seconds.

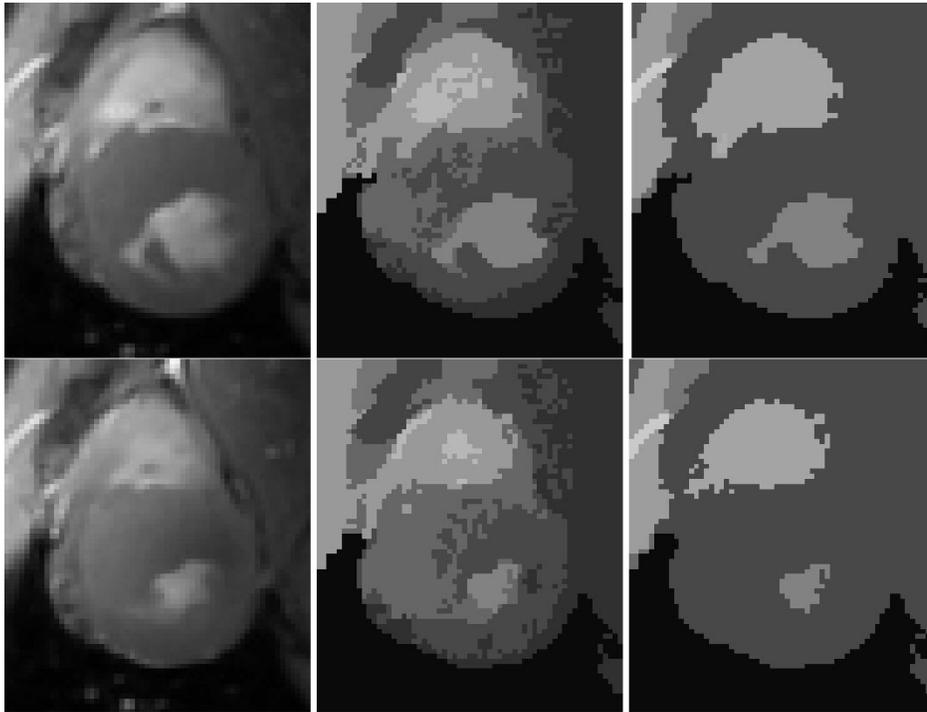


Fig. 12. Two frames of a time series cardiac MR Image segmented using anti-geometric diffusion and region merging. [LEFT] Original image. [MIDDLE] Automatically segmented image (10 regions) using anti-geometric diffusion and region merging. [RIGHT] Segmented image (7 regions) after user directed splitting and merging corrections to previous segmentation.

each region by a linear or smooth function rather than a constant would yield a much more flexible segmentation strategy. In this manner, segmentations which resemble those obtained using the Mumford-Shah [32] functional may be obtained much more quickly when compared to the numerical methods used to obtain true solutions of this functional (including the recent curve evolution approaches proposed in [33], [34] based upon the level set methods of Osher and Sethian [35]). We are currently exploring these more general piecewise representations of the image data.

One point to make in comparing piecewise smooth versions of our segmentation procedure with Mumford-Shah segmentation is that we are able to avoid the penalty on the measure of the region boundaries used in [32]. In our scheme, scale is related to the final number of regions in the segmentation rather than the lengths of the region boundaries. This facilitates more accurate detection of object boundaries that contain corners or other fine scale details.

We further note that total squared error is not the only criterion one could use for region merging. In fact, it is not even necessary to use an energy based criterion. Zhu and Yuille [31], for example, who utilize an energy based upon Minimum Description Length in their region competition algorithm, point out the popular Fisher's test [36], a statistically based region merging criterion.

Finally, we are working to generalize this model for vector-valued (specifically color) images. The classification cues taken from the diffusion behavior of each pixel (monotonicity for example) are currently based upon the well-ordered property of scalar image intensities. Vector spaces lack this property; so a different method of classification will have to be developed if a diffusion model is to be used for segmenting vector valued images.

## IX. CONCLUSION

We have outlined a novel method for adaptive thresholding and segmentation using an anti-geometric diffusion model

to capture regions on opposite sides of prominent edges in grayscale images. In contrast to more traditional anisotropic diffusion techniques, we abandon the goal of preserving edges but seek instead to maximally smear them in order to detect their adjacent neighborhoods. Doing the detection *during* the diffusion avoids the ambiguity of deciding when to stop before utilizing the processed image. Since this diffusion/classification procedure acts as a region splitting operator, we saw that it could be used together with region merging to obtain fast piecewise constant segmentations. We discussed the efficient implementation of these methods, and have shown their application to a variety of images, including both 2-D and 3-D examples.

## REFERENCES

- [1] C. K. Chow and T. Kaneko, "Automatic boundary detection of left ventricle from cineangiograms," *Comput. Biomed. Res.*, vol. 5, pp. 338–410, 1972.
- [2] Y. Nakagawa and A. Rosenfeld, "Some experiments on variable thresholding," *Pattern Recognit.*, vol. 11, no. 3, pp. 191–204, 1979.
- [3] N. Otsu, "A threshold selection method from gray level histogram," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-8, pp. 62–66, 1979.
- [4] J. Kittler and J. Illingworth, "Minimum error thresholding," *Pattern Recognit.*, vol. 19, no. 1, pp. 41–47, 1986.
- [5] D. Grennhill and E. R. Davies, "A new approach to the determination of unbiased thresholds for image segmentation," in *Proc. Int. Conf. Image Processing Applications*, 1995, pp. 519–523.
- [6] S. D. Yanowitz and A. M. Bruckstein, "A new method for image segmentation," *Comput. Vis., Graph., Image Process.*, vol. 46, no. 1, pp. 82–95, 1989.
- [7] F. H. Y. Chan, F. K. Lam, and H. Zhu, "Adaptive thresholding by variational method," *IEEE Trans. Image Processing*, vol. 7, pp. 468–473, Mar. 1998.
- [8] J. R. Parker, "Gray level thresholding in badly illuminated images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 13, pp. 813–819, Aug. 1991.
- [9] R. M. Haralick and L. B. Shaprio, "Image segmentation techniques," *Comput. Vis., Graph., Image Process.*, vol. 29, pp. 100–132, 1983.
- [10] N. R. Pal and S. K. Pal, "A review on image segmentation techniques," *Pattern. Recognit.*, vol. 26, pp. 1227–1249, 1993.
- [11] J. Sauvola, T. Seppänen, S. Haapakoski, and M. Pietikäinen, "Adaptive document binarization," in *Proc. Int. Conf. Document Analysis and Recognition*, vol. 1, 1997, pp. 147–152.
- [12] O. D. Trier and T. Taxt, "Evaluation of binarization methods for document images," *IEEE Trans. Pattern. Anal. Machine Intell.*, vol. 17, pp. 312–315, Mar. 1995.
- [13] A. P. Witkin, "Scale space filtering," in *Proc. Int. Conf. Artificial Intelligence*, 1983, pp. 1019–1023.
- [14] L. Alvarez, P. L. Lions, and J. M. Morel, "Image selective smoothing and edge detection by nonlinear diffusion—II," *SIAM J. Numer. Anal.*, vol. 29, pp. 845–866, 1992.
- [15] R. Carmona and S. Zhong, "Adaptive smoothing respecting feature directions," *IEEE Trans. Image Processing*, vol. 7, no. 3, pp. 353–358, 1998.
- [16] G. Unal, H. Krim, and A. Yezzi, "Feature-preserving flows: A stochastic differential equations's view," in *Proc. Int. Conf. Image Proc. 2000*, vol. 1, Sept. 2000, pp. 896–899.
- [17] R. Whitaker and S. Pizer, "A multi-scale approach to nonuniform diffusion," *Comput. Vis., Graph., Image Process. Image Understand.*, vol. 57, no. 1, pp. 99–110, Jan. 1993.
- [18] B. B. Kimia and K. Siddiqi, "Geometric heat equation and nonlinear diffusion of shapes and images," in *Proc. Comput. Vis. Pattern Recognit.*, 1994.
- [19] Y. L. You, W. X. A. Tannenbaum, and M. Kaveh, "Behavioral analysis of anisotropic diffusion in image processing," *IEEE Trans. Image Processing*, vol. 5, pp. 1539–1553, 1996.
- [20] L. Alvarez, F. Guichard, P. L. Lions, and J. M. Morel, "Axioms and fundamental equations of image processing," *Arch. Ration. Mech. Anal.*, vol. 123, 1993.
- [21] B. B. Kimia, A. Tannenbaum, and S. W. Zucker, "On the evolution of curves via a function of curvature, I the classical case," *J. Math., Anal., App.*, vol. 163, pp. 438–458, 1992.
- [22] R. Malladi and J. Sethian, "A unified approach to noise removal, image enhancement, and shape recovery," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 5, pp. 1554–1568, 1996.
- [23] G. Sapiro and A. Tannenbaum, "On invariant curve evolution and image analysis," *Indiana Univ. J. Math.*, vol. 42, 1993.
- [24] ———, "Affine invariant scale-space," *Int. J. Comput. Vis.*, vol. 11, pp. 25–44, 1993.
- [25] L. C. Evans, "Estimates for smooth absolutely minimizing Lipschitz extensions," *Electron. J. Diff. Eq.*, vol. 1993, no. 3, pp. 1–9, Oct. 1993.
- [26] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, pp. 629–639, July 1990.
- [27] B. B. Kimia, A. Tannenbaum, and S. W. Zucker, "Shapes, shocks, and deformations I: The components of two-dimensional shape and the reaction-diffusion space," *Int. J. Comput. Vis.*, vol. 15, pp. 189–224, 1995.
- [28] F. Meyer and P. Maragos, "Nonlinear scale-space representation with morphological levelings," *J. Vis. Commun. Image Represent.*, vol. 11, pp. 245–265, 2000.
- [29] R. Hummel, "Representations based on zero-crossings in scale space," in *Proc. Comput. Vis. Pattern Recognit.*, 1986, pp. 204–209.
- [30] J. M. Morel and S. Solimini, *Variational Methods in Image Segmentation*. Boston, MA: Birkhauser, 1995.
- [31] S. Zhu and A. Yuille, "Region competition: Unifying snakes, region growing, and Bayes/MDL for multiband image segmentation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, pp. 884–900, Sept. 1996.
- [32] D. Mumford and J. Shah, "Optimal approximations by piecewise smooth functions and associated variational problems," *Commun. Pure Appl. Math.*, vol. 42, no. 4, 1989.
- [33] T. Chan and L. Vese, "A Level Set Algorithm for Minimizing the Mumford-Shah Functional in Image Processing," UCLA, Los Angeles, CA, CAM rep., 2000.
- [34] A. Tsai, A. Yezzi, and A. Willsky, "A curve evolution approach to smoothing and segmentation using the Mumford-Shah functional," in *Proc. Comput. Vis. Pattern Recognit.*, 2000.
- [35] S. Osher and J. Setbian, "Fronts propagating with curvature dependent speed: Algorithms based on Hamilton-Jacobi formulations," *J. Comput. Phys.*, vol. 79, pp. 12–49, 1988.
- [36] T. Philips, A. Rosenfeld, and A. Sher, "O(log n) bimodality analysis," *Pat. Rec.*, vol. 22, pp. 741–746, 1989.



**Siddharth Manay** (S'03) received the Ph.D. degree in 2003 from the School of Electrical Engineering, Georgia Institute of Technology, Atlanta.

He is now a Postdoctoral Fellow with the Vision Lab at the Computer Science Department, University of California at Los Angeles. His research interests are in the field of computer vision, with a particular focus on partial differential equations. He is currently investigating accurate schemes for computation of distance functions and invariant descriptions of shapes.



**Anthony Yezzi** (M'99) was born in Gainesville, FL, in 1972 and received the Ph.D. degree in 1997 through the Department of Electrical Engineering at the University of Minnesota, Minneapolis.

After completing a postdoctoral research position in the Laboratory for Information and Decision Systems (LIDS) at Massachusetts Institute of Technology, he began his current position in 1999 at Georgia Institute of Technology as an Assistant Professor of electrical and computer engineering with an adjunct appointment in biomedical engineering. He has also

consulted for a number of medical imaging companies including GE, Picker, and VTI. His research lies primarily within the fields of image processing and computer vision. He has worked and continues to work on a variety of problems within these fields including image denoising, edge-detection, segmentation and grouping, shape analysis, multiframe stereo reconstruction, tracking, and registration. Much of his work is motivated by and directed towards problems in medical imaging. Two central themes of his research in general are curve/surface evolution theory from differential geometry and partial differential equations.