FLUX MAXIMIZING GEOMETRIC FLOWS FOR 2D AND 3D BLOOD VESSEL SEGMENTATION

Oleksandr Vasilevskiy

Department of Computer Science
McGill University, Montréal

July 2001

A Thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfilment of the requirements for the degree of
Master of Science

0-612-77062-1

ABSTRACT

The visualization of two-dimensional and three-dimensional vascular structures is of significant interest in image-guided surgery. This assists clinicians in pre-operation planning, real-time operating room decision making, and post-operation monitoring. In order to achieve this goal, vascular networks must first be segmented from intensity data such as CT or MR angiography images. A framework for addressing this problem is the use of geometric flows where a curve (in two dimensions) or a surface (in three dimensions) is evolved under constraints from image forces so that it clings to features of interest in an intensity image. Recent variations on this theme take into account properties of enclosed regions and textures and allow for multiple curves or surfaces to be simultaneously represented. However, it is not clear how to apply these techniques to images of low contrast elongated structures, such as blood vessels. To address this problem we derive the gradient flow which maximizes the rate of increase of flux of an auxiliary vector field through a curve or surface. The calculation leads to a simple and elegant interpretation which is essentially parameter free. We illustrate its advantages with level-set based segmentations of 2D and 3D angiography images of blood vessels.

RÉSUMÉ

La visualisation des structures vasculaires en deux et trois dimensions est d'un intérêt important pour la chirurgie guidée par image. Elle aide le chirurgien dans le planning pre-opération, la prise de decision en temps réel au bloc opératoire, le contrôle post-opération.

Pour pouvoir atteindre ce but, les réseaux vasculaires doivent être segmentés à partir d'images telles que les angiographies CT ou RM. Une méthode pour résoudre ce problème est l'utilisation des flux géométriques où une courbe (en deux dimensions) ou une surface (trois dimensions) évolue sous des contraintes de forces de l'image afin qu'elle s'accroche aux structures intéressantes de cette image. Les variations récentes sur ce thème prennent en compte des propriétés de régions closes et de textures et permettent de représenter plusieurs surfaces et courbes simultanément.

Cependant, il est difficile d'appliquer ces techniques a des images de structures allongées de faible contraste, tel que les vaisseaux sanguins. Pour résoudre ce problème, nous dérivons le flux du gradiant maximisant le taux de croissance du flux d'un champ vectoriel auxilliaire passant au travers d'une courbe ou d'une surface. Cette formulation mène à une interprétation élégante qui est essentiellement libre de tout paramètre. Nous illustrons les avantages de cette méthode avec des segmentations d'angiographie 2D et 3D.

ACKNOWLEDGEMENTS

I would like to thank my research supervisor, Professor Kaleem Siddiqi, who guided my research and inspired my interest in the beautiful subject of geometric flows. His assistance and support has gone far beyond the call of duty both inside and outside the research lab. The results shown in this thesis were obtained in collaboration with him and are published in [**52, 53**].

I thank Sylvain Bouix for his helpful suggestions and readiness to help. He also translated the Abstract into French. I would like to thank all my teachers both from Odessa and Montreal.

This thesis would not be possible without the support and encouragement of my family. Thanks go to my wife Milena, for her care, support and help.

Funding for my research was provided by a National Science and Engineering Research Council postgraduate scholarship.

This thesis is dedicated to my father, Dimitri Vasilevskiy.

TABLE OF CONTENTS

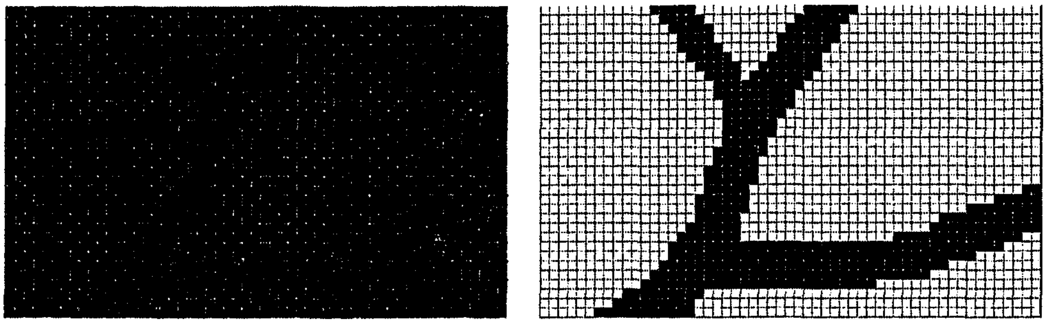
LIST OF FIGURES

CHAPTER 1

Introduction

The accurate visualization and quantification of the human vasculature is an important prerequisite for a number of clinical procedures including the diagnosis of several diseases, surgical planning and navigation during surgery. A common approach is to employ maximum intensity projection where three-dimensional data such as Computed Tomography (CT) or Magnetic Resonance Angiography (MRA) is projected onto a 2D plane by choosing the maximal value along each projection ray. A serious drawback in this approach is that vascular structures may overlap with non-vascular ones and thin or low contrast vessels may not be visible. Thus, it is desirable to segment the vasculature before it is visualized. In the context of this thesis we shall refer to *segmentation* as the process of labeling a location in a 2D or 3D intensity image as a "vessel" or a "non-vessel" point, as illustrated in Figure 1.1. After the data has been segmented, the visualization and quantification of complex human vasculature is greatly simplified.

A naive approach to the segmentation problem is that of thresholding the original angiography data, i.e., all voxels above a certain value are marked as vessels, while the rest are labeled as background. However, due to image acquisition noise, partial volume effects and patient motion between different scans, the intensity data is usually non-homogeneous, particularly for vessels of different sizes. As a result, a conservative



(a) Input image

(b) Segmented vessels

FIGURE 1.1. Illustration of the segmentation process.

high threshold will not capture many of the thin vessels, while a low threshold will incorrectly label many voxels.

A brute force strategy to obtain the vascular network is to employ manual segmentation, where an expert simply labels voxels corresponding to vessels. Several specialized programs have also been developed to assist a neurosurgeon and facilitate and speed up the manual segmentation process. The main drawback of this approach is that a significant amount of expert user time is required. Furthermore, such segmentation is also prone to human error. This suggests a need for robust automated methods of vascular segmentation.

1. Problem Statement

The problem that we wish to solve is that of the accurate estimation of the positions of blood vessels given the two-dimensional or three-dimensional angiography data. Specifically, we wish to separate all points of the initial data into two classes, one containing only points that belong to vessels and the other containing the remaining points. The algorithm must be able to detect vessels of variable sizes and contrast. It should also require minimal user interaction. This leads to the constraint that the number of input parameters should be small. We also assume that no a priori information about the localization of blood vessels is available. In this thesis we present a theoretical solution to this problem and illustrate its performance on a number of 2D and 3D medical images.

2. Approach

Several approaches have been proposed to address the problem of blood vessel segmentation. Among them are multi-scale methods that involve convolving the data with Gaussian filters at multiple scales and analyzing the eigenvalues of the image of a Hessian matrix at each voxel in the data to determine the local shape of structures [**26, 25, 16, 23, 30, 40**], a multi-scale algorithm that detect vessels based on the assumption that their centerlines appear as ridges in intensity images [**3**], a statistical method that models physical properties of blood with a mixture of probability distributions [**49**], an anisotropic diffusion approach which seeks to reduce image noise without blurring vessel boundaries [**24**], a morphological reconstruction method that uses the concept of simple points [**13**], and geometric methods where vessels are detected by evolving an initial estimate represented by 1D curves in a 3D space to the true structures [**31, 33, 32**]. The method presented here is closest in spirit to the last class of methods above, in that it is based on a curve/surface evolution framework. Whereas many segmentation algorithms have been developed using this framework, most will fail to capture thin elongated low contrast structures, such as blood vessels.

The method we propose is motivated by the observation that in the vicinity of the boundaries of the vessels, the gradient vector field of the intensity image should be locally orthogonal to them. Thus, a natural principle to use towards the recovery of these boundaries is to maximize the flux of the gradient vector field through an evolving curve (in 2D) or surface (in 3D).

3. Imaging Modalities

There are several types of imaging methods for obtaining three-dimensional data that can later be processed for vessel segmentation. The most widely used techniques are Computed Tomography (CT) Angiography and Magnetic Resonance Angiography

(MRA). While these imaging methods attempt to allow for a safe and rapid procedure for obtaining a high-contrast, high-resolution image of vascular structures, each has its own limitations.

Magnetic Resonance Imaging (MRI) of the blood vessels is called Magnetic Resonance Angiography (MRA). MRA uses the natural magnetic properties of the hydrogen atoms in our bodies to create stacks of parallel cross-sectional images which show the vessels (more precisely: the flowing blood) as bright spots and lines, surrounded by dark stationary tissue. The brightness of the vessels in MRA images is proportional to the speed of the blood. Hence, thick vessels where the blood flows faster tend to appear brighter than thinner vessels. Fortunately, MRA data can be acquired on standard MRI scanners which are available at many major hospitals. The acquisition of MRA data is painless and typically takes less than an hour to perform. With few exceptions, there are no known risks to an MRA examination. MRA does not involve the use of X-rays.

The main alternative to MRA is Computed Tomography (CT) angiography. In this examination, x-ray contrast material is injected directly into the vessel being studied through a catheter. CT scanners consist of an X-ray source, an X-ray detector, a mechanical platform to rotate the specimen at various angles and elevations, and a computer to produce CT images from the collected data. CT scanners typically produce multiple two-dimensional images each representing cross sections through the specimen (tomographs). The tomographs are stacked to create a three-dimensional volume. Tomographic images are generated by collecting one-dimensional X-ray signals of an object at many angles at one cross section. From these one-dimensional X-ray signals, a cross sectional image is reconstructed representing the X-ray attenuation properties of this slice. To create a 3-D volume data set of the specimen, the process is repeated at various cross sections. In this manner, multiple tomographs are stacked to produce volumetric data. A similar Computed Rotational Angiography (CRA) method produces 3-D images by acquiring projection radiographs from many angles around the patient, followed by reconstruction using the CT algorithm. X-ray angiograms generally are more accurate and may provide detail that cannot be seen routinely in MRA. On the other hand, MRA does not usually involve the use of needles and catheters, or the administration of X-ray contrast material or preparation, and in many cases is sufficient for the problem at hand.

In this thesis we illustrate the advantages of the proposed blood vessel segmentation technique on both MRA and CRA images.

4. Applications

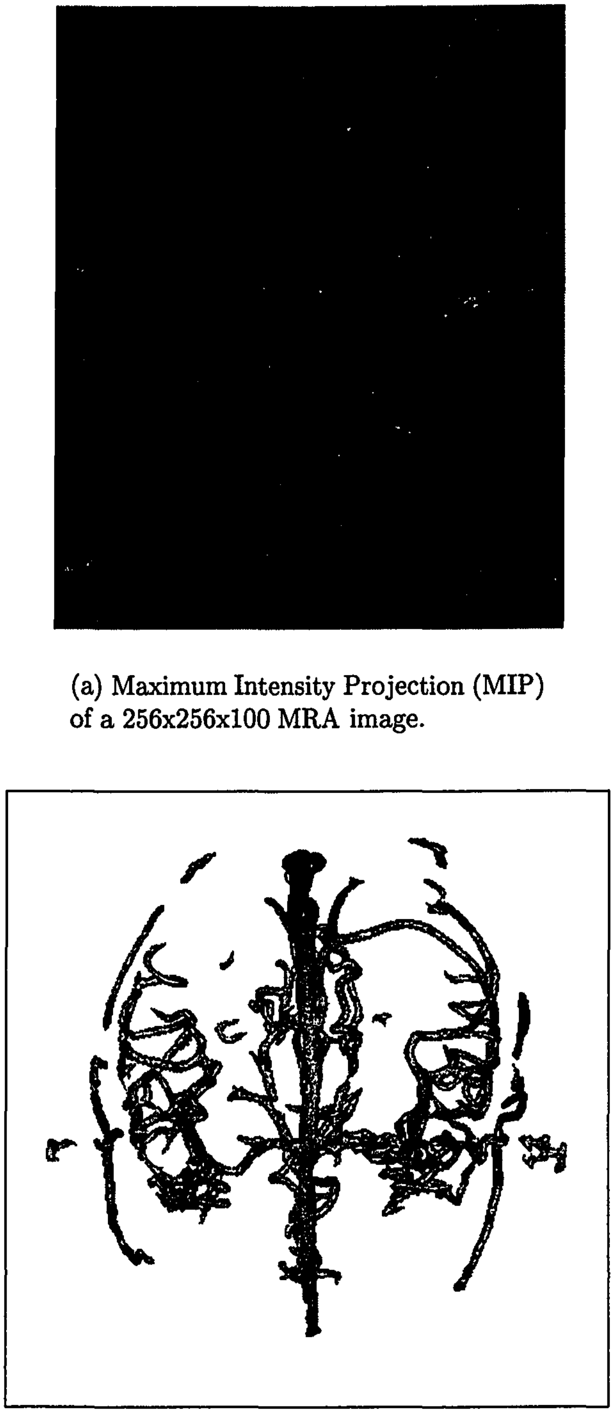
Three-dimensional images of blood vessels are usually visualized using volume rendering techniques such as maximum intensity projection (Figure 1.2(a)) or simply by showing a collection of 2D slices. The method presented here provides accurate information on the location and size of blood vessels in 3D images. Such quantitative information can later be used in conjunction with a surface rendering technique for more accurate visualization of the vessel architecture (Figure 1.2(b)).

Such enhanced visualization of the vascular network of different organs (the brain, the liver, the lungs) can help physicians in performing a number of different clinical operations. Interventional procedures such as the placement of a prosthesis in order to prevent aneurysm rupture or a bypass operation, require precise knowledge of the three-dimensional vessel architecture.

In minimally invasive neurological surgery it is also extremely important to know the location of the blood vessels. Typically, a small hole is drilled in the scalp of the patient and a long needle is inserted to get access to the area being operated on, such as a tumor. As the needle is being inserted it is crucial not to damage any vessels since it could potentially induce a stroke. Since the surgeon is not able to see the needle directly, all navigation is guided by images and the accurate visualization of blood vessels is of utmost importance.

5. Contributions

This thesis makes the following specific contributions:



(a) Maximum Intensity Projection (MIP) of a 256x256x100 MRA image.

(b) Vasculature segmented using the proposed flux maximizing flows, rendered with VTK.

FIGURE 1.2. Visualization techniques.

- We have derived the geometric flows that maximize the flux of an auxiliary vector field through an evolving curve (in 2D) or surface (in 3D).
- As a corollary, we have derived the flows that maximize the circulation of an auxiliary vector field along an evolving curve (2D).
- We have proved that the form of flux maximizing flows remains unchanged in 2D and in 3D.
- We have shown how these flux maximizing flows can be effectively applied for segmentation of blood vessels from angiography images.
- We have demonstrated the performance of our segmentation algorithm on several 2D and 3D medical images and have compared the results with the simple thresholding.

6. Outline

This thesis presents a novel method for the segmentation of tubular structures from intensity images. While the method works both in two and three dimensions for any kind of image containing bright tubular structures, the main application we have focused on is the segmentation of blood vessels from 3D MR or CT angiography images. Both the theoretical foundation as well as the experimental results of the method are presented. The thesis is organized as follows. Chapter 2 presents a general discussion of existing methods for blood vessel segmentation. The theoretical core of our method is presented in Chapter 3. Chapter 4 describes implementational details of our method including the level set technique. Finally, the experimental results presented in Chapter 5 demonstrate the performance of the method in detecting vessels of various sizes and contrast. Chapter 6 concludes with an overview of the proposed method and a discussion of possible directions for future work.

CHAPTER 2

Previous Work

This chapter briefly covers previous work on the problem of segmentation of tubular structures and blood vessels in particular. Several approaches have been proposed to address the problem of blood vessel segmentation. Among them we will discuss methods that try to estimate the local shape of a structure in the image by analyzing the eigenvectors and associated eigenvalues of the Hessian matrix constructed at each voxel [26, 25, 16, 23, 30, 40]. Here, to address the problem of detecting vessels of various sizes, the image is first convolved with Gaussian filters of varying radii. In such methods a vessel is typically assumed to be cylindrical in shape, with a Gaussian cross-sectional profile. Next, we shall review the approach of Aylward *et al.* which detects vessels based on the assumption that their centerlines appear as ridges in intensity images [3]. We will also discuss a statistical method that models physical properties of blood with a Gaussian mixture model [49], an anisotropic diffusion approach which smoothes along the vessel direction but not perpendicular to it [24], a region growing technique that reconstructs the vessel system of the liver based on the idea of simple points [13], and a curve evolution method where vessels are approximated by one-dimensional curves in three-dimensional space. In the latter technique, an initial estimate obtained by thresholding the original image is evolved towards the true structures [31, 33, 32].

One goal of this thesis is to show how our proposed method for blood vessel segmentation successfully unites the strengths of several of the approaches considered in this chapter, while minimizing the effects of their inherent difficulties. In Chapter 5 we also point how some of these methods could be incorporated in our proposed flux maximizing flow algorithm.

1. Multi-scale Methods Based on the Eigenvalue Analysis

Multi-scale filtering for estimating the local structure of an image has been a popular technique for the segmentation of curvilinear structures such as blood vessels [**26, 25, 16, 23, 30, 40**]. The general approach of multi-scale methods is to choose a range of scales and to compute a response of a filter at each scale. All the responses are then combined to get a single multi-scale response which contains the sought features.

We now discuss algorithms that apply this theory to the specific problem of detecting elongated cylindrical structures. Several stages are typically involved. First, a family of images is obtained by convolving the original image with a Gaussian filter G_σ with the standard deviation σ for $\overline{x} \in \Re^3$

$$G_\sigma(\overline{x}) = \frac{1}{(\sqrt{2\pi}\sigma)^3} e^{\frac{-<\overline{x}\cdot\overline{x}>}{2\sigma^2}}$$

for several σ between σ_{min} and σ_{max}.

Then, for each convolved image I_σ at each point $\overline{x} = (x, y, z)$, the Hessian matrix is computed. The Hessian matrix H is a matrix composed of second partial derivatives of the image

$$H(I(\overline{x})) = \begin{bmatrix} I_{xx}(\overline{x}) & I_{xy}(\overline{x}) & I_{xz}(\overline{x}) \\ I_{yx}(\overline{x}) & I_{yy}(\overline{x}) & I_{yz}(\overline{x}) \\ I_{zx}(\overline{x}) & I_{zy}(\overline{x}) & I_{zz}(\overline{x}) \end{bmatrix}.$$

Next, the eigenvalues of the Hessian H are computed and used to estimate the local structure of the image. Let $\lambda_1, \lambda_2, \lambda_3$ be the eigenvalues of H with $|\lambda_1| \geq |\lambda_2| \geq |\lambda_3|$, and $\overline{v_1}, \overline{v_2}, \overline{v_3}$ be the corresponding eigenvectors. It has been shown [**16, 25**] that for a bright tubular structure on a dark background

$$\lambda_1, \lambda_2 < 0 \quad and \quad \lambda_1| \approx |\lambda_2| \gg |\lambda_3| \tag{2.1}$$

To provide justification for this constraint, it is important to note that the eigenvalues λ_3 and λ_1 correspond to the minimal and maximal principal curvatures of the surface, respectively, and that for thin tubular structures, the curvature along the structure is much smaller than that in the cross-sectional direction. Furthermore, the eigenvector $\overline{v_3}$ associated with λ_3 will give the direction of the main axis of the tube, whereas $\overline{v_1}$ and $\overline{v_2}$ will give a basis in the plane orthogonal to the axis [**15, 25**]. It should be clear that such an analysis of the eigenvalues can also be performed for blob-like and plate-like structures as well.

After the eigenvalues have been computed, for each of the convolved images I_σ, a singe-scale response is computed, by applying a certain scalar function $F(\lambda_1, \lambda_2, \lambda_3)$ at each point of the image. This is typically a heuristic function that uses properties of the eigenvalues of the Hessian to try to capture how "vessel-like" the local structures are. It is constructed to give a high response only when the structure resembles a vessel. Several candidates for response functions have been proposed by different researchers [**26, 25, 16, 23**].

One of the key difficulties with multi-scale approaches is that one has to compare the results of a response function at different scales while the intensity and its derivatives are decreasing functions of scale. Hence, single response images have to be normalized. The choice of normalization procedures varies for different approaches and it is usually complex and non-intuitive [**26, 25**].

After the normalization step, a multi-scale response image at each point is computed as the maximum of the set of normalized single scale responses:

$$R_{multi}(\overline{x}) = max\{R_s(\overline{x}), s \in [s_{min}, s_{max}]\}.$$

In this multi-scale image, the vessels are brightened and non-tubular structures are darkened. This enhanced image can be visualized directly [16] or thresholded [40]. Related work includes methods [26, 25] which use the eigenvalues to estimate the centerlines of vessels. In addition, the scale at which the response is maximal at a particular voxel is stored. It corresponds to the radius of the vessel passing through this voxel. The full vasculature is then reconstructed using knowledge of centerlines and the associated radii.

2. Intensity Ridges

Another approach [3] proposes to find estimates of the centerlines of the vessels using the observation that they appear as intensity ridges in the image. The process presented in the paper tries to find 1D loci of the local maximas. These 1D loci are used as an approximation to the medial axes of the vessels. The width of the vessels is defined by the height of the ridges. The centerlines are detected by following the ridge, starting from an initial point (selected manually by a user). The method experiences problems at the branching points, where the assumption that an intensity ridge corresponds to a centerline of a vessel does not necessarily hold. The authors introduce certain "ridge termination" criteria to deal with this problem. While the ridge is generated using information only at a single scale, the width estimates are obtained using a multi-scale analysis.

3. A Statistical Approach

Wilson and Noble [49] propose a statistical approach to segmenting blood vessels. They introduce a mixture distribution for the data based on the physical properties of the blood to classify vessel voxels. The parameters of the distribution estimated using the expectation maximization (EM) algorithm define the thresholds used to segment the data. After the thresholding step, only a number of largest connected components are kept as a final segmentation.

From an analysis of the MRA data, the authors [49] suggest that three distributions can be used to represent it. Vessel voxels are assumed to arise from a uniform distribution, one Gaussian distribution represents tissue outside the head and a second represents brain tissue, eyes, skin, and bone. The mixture distribution is hence:

$$p(x) = w_0 \frac{1}{I} + \sum_{k=1}^{2} w_k \frac{1}{\sqrt{2\pi}\sigma_k} e^{-\frac{1}{2}\left(\frac{x-\mu_k}{\sigma_k}\right)^2}$$

where the weights w_k determine the proportion of each class, μ_k and σ_k are the mean and standard deviation of the Gaussian, and I is the total number of intensity levels. The EM algorithm is used to estimate all seven unknown parameters. A voxel i arises from a uniform distribution, i.e. it belongs to a vessel if $w_0 p(x_i|0) > w_k p(x_k|k)$ for $k = 1, 2$. This can be restated as:

$$. \; x_i > \mu_k + \sigma_k \sqrt{2ln\left(\frac{w_k}{\sigma_k} \frac{I}{w_0\sqrt{2\pi}}\right)}$$

This equation defines the high threshold. Slow blood flow at vessel walls and turbulent flow within aneurysms produce lower intensities in the MRA scan. Hence, some vessel voxels will not be detected by the high threshold. The authors [49] use the following heuristic to recover such voxels: if a voxel has sufficiently high intensity, and it is close to previously segmented vessel voxels, then it is assumed to belong to a vessel. The calculation of the low threshold involves computing the 3D distance transform [6] of the initial segmentation. Then for each distance value from 0 to 70, the same EM method is used to determine the threshold for each set of voxels of the original data at this distance. The low threshold is chosen as the minimum of these individual thresholds. This procedure for determining the low threshold is not fully justified since the three-class distribution of the MRA data is no longer appropriate for the voxels at some fixed distance away from the initially segmented bright vessels.

The authors report experimental results where it was found that over 92% of the vessel voxels segmented by the algorithm also appear in a manual segmentation. However, more than twice as many voxels are selected in the manual segmentation than by their approach.

4. Anisotropic Diffusion

We now review an approach to enhancing vessel-like structures. Anisotropic diffusion refers to the smoothing of an image to reduce noise, while preserving certain features of interest, such as edges. Krissian *et al.* [**24**] introduce an implementation of anisotropic diffusion which better preserves small tubular structures such as vessels. In 2D such smoothing can be implemented using the geometric heat equation [**22**]

$$\frac{\partial I}{\partial t} = div\left(\frac{\nabla I}{|\nabla I|}\right)|\nabla I|$$

where I is image intensity. This equation is equivalent to moving each level set of the image by its curvature [**17, 18**] which will have the desired effect of preserving elongated structures with small curvature while removing noise. In 3D, however, mean curvature motion tends to shrink the vessels since one of the principal curvatures (in the direction perpendicular to the vessel) will be very high. Instead, Krissian *et al.* [**24**] propose an explicit weighting of smoothing in the directions of the gradient, minimum and maximum curvatures.

To enhance a 3D image they use the following equation:

$$\frac{\partial I}{\partial t} = C_{\xi}(|\nabla I|)I_{\xi\xi} + C_{e_1}(|\nabla I|, |\kappa_1|)I_{e_1e_1} + C_{e_2}(|\nabla I|, |\kappa_2|)I_{e_2e_2}$$

where the vector ξ denotes the gradient direction, vectors e_1 and e_2 correspond to the maximal and minimal curvature directions, respectively, κ_1 and κ_2 are the values of the maximal and minimal curvatures respectively, and the coefficients C_ξ, C_{e_1}, C_{e_2} determine the amount of diffusion in each of these directions. The basic strategy is to maximize diffusion in the direction of a vessel (minimal principal curvature direction) and minimize diffusion in the direction perpendicular to it (maximal principal curvature direction). Such anisotropic filtering can be used as a preprocessing step for a vessel segmentation algorithm. It will reduce blob-like noise while preserving tubular structures. It should be noted, however, that the vessels will be slightly perturbed and their exact location will change.

5. Region Growing Based on Simple Points

Dokládal *et al.* [**13**] propose a method for segmenting blood vessels of the liver based on the topological properties of the vessel system. The algorithm relies on the hypothesis that the vessels of the liver form an object that is simply connected, contains no holes and no cavities, and that this object itself forms a cavity with respect to its complement. At the heart of this method is the notion of a simple or removable point. A simple point is defined as a point of an object whose deletion does not change the topology of the object. Bertrand and Malandain [**4**] proposed a characterization of a simple point based on determining the number of connected components in its 3x3 neighborhood. The calculation of removable points has also been used in research for computing skeletons [**4, 7**].

There are two versions of the algorithm for reconstructing the liver vessel system: segmentation by reconstructing the object and segmentation by reconstructing the background. When segmenting by reconstruction, a single point is manually placed inside the object. The resulting object is reconstructed by iteratively adding simple points to ensure that topology is preserved. During the reconstruction procedure, the points are processed in the order of their intensities, an ordering that is essential for the algorithm. The reconstruction stops when no simple points of intensity higher than a certain predetermined threshold can be added.

The other variation is segmentation by reconstructing the background in essentially the same fashion. The vessels are then obtained as the complement of the reconstructed background. This variation has the advantage that an initial seed need not to be placed manually; it can simply be a point on the border of the image.

Provided that strong assumptions about the topology of the vessel system holds, this technique can be a useful segmentation tool. In practice, however, it most often fails due to noise, partial volume effects and pathologies in the vascular network.

6. Co-dimension Two Flows

The system recently developed by Lorigo *et al.* [**31, 33, 32**] uses a variation of the curve evolution framework to address the blood vessel segmentation problem. Vessels are modeled as 1D curves in 3D space (having co-dimension two) that can branch arbitrarily. An initial curve estimate is obtained by thresholding the image and is then evolved to the curves in the data (the vessels). The main idea is to regularize the geometric flow with a force that is not based on the mean curvature of tubular surfaces, but on the curvature of the underlying 1D curves. The rationale is that classical mean curvature based regularization tends to shrink vessels and annihilate thin ones, since one of the principal curvatures is high.

This work has a mathematical foundation in the recent level set theory developed for mean curvature flows in arbitrary co-dimension [**2**] and has the intuitive behavior that wiggly tubular structures are "straightened" out. Also, the approach has a variational interpretation as the gradient flow that minimizes a weighted curvature functional. This flow turns out to be [**31, 33, 32**] :

$$\psi_t = \lambda(\nabla\psi, \nabla^2\psi) + \frac{g'}{g}\nabla\psi.\mathbf{H}\frac{\nabla\mathbf{I}}{|\nabla\mathbf{I}|}$$

Here ψ is an embedding surface whose zero level set is the evolving 3D curve, λ is the smaller nonzero eigenvalue of a particular matrix [**2**], g is an image-dependent weighting factor, **I** is the intensity image and **H** is its Hessian. For numerical simulations the evolution of the curve is depicted by the evolution of an ϵ-level set. With a suitable choice of g this flow should should remain regular while being attracted to boundaries of vessels. However, it is unclear how seeds placed within vessels, would grow *along* them to recover thin elongated structures. To encourage the latter behavior, the second term of the equation is multiplied with a factor $\rho(\nabla\psi.\nabla\mathbf{I})$ where ρ is a constant, giving rise to

$$\psi_t = \lambda(\nabla\psi, \nabla^2\psi) + \rho(\nabla\psi.\nabla\mathbf{I})\frac{g'}{g}\nabla\psi.\mathbf{H}\frac{\nabla\mathbf{I}}{|\nabla\mathbf{I}|}$$

Thus the flow is biased so that normals to the ϵ-level set align themselves (locally) to the direction of image intensity gradients (the inner product of $\nabla\psi$ and $\nabla\mathbf{I}$ is then maximized). However, the introduction of the multiplicative term is a heuristic and the framework loses its pure variational interpretation.

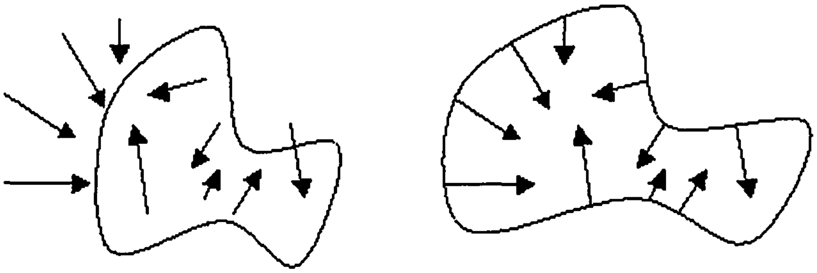
In this thesis we develop a vessel algorithm that is also based on geometric flows. Although we employ co-dimension one flows (we evolve curves in 2D and surfaces in 3D) and use different assumptions and intuitions, the work by Lorigo *et al.* [**31, 33, 32**] is the closest in spirit to our method. In the following chapter we shall present an overview of geometry-driven flows and their application to segmentation problems. We shall then introduce and develop the notion of a flux maximizing flow in 2D and 3D, which is the central contribution of this thesis.

CHAPTER 3

Flux Maximizing Flows

The curve and surface evolution framework has proven to be a powerful segmentation tool, especially in application to medical images [**8, 9, 34, 38, 45**] where objects of interest often have highly irregular shapes and contrast variations across boundaries can be low. Here the essential idea is to move a curve (in 2D) or surface (in 3D) under constraints from image-based forces so that it clings to object boundaries. Most of these curve and surface evolution schemes are numerically implemented using level set technology developed by Osher and Sethian [**37**]. Here the essential idea is to interpret the evolving curve or surface as the zero level set of a higher dimensional function, which is updated in time. The main advantage is that topological changes are handled with a fixed computational complexity. These methods will be discussed in greater detail in Chapter 4.

A theoretical advantage of curve and surface evolution schemes for shape segmentation is that several can be derived as gradient flows that minimize a particular weighted length or weighted area functional [**20, 46**]. Recent advances in the use of geometric flows include methods which take into account the statistics of the regions enclosed by the evolving curves [**38, 51**]. Further developments include multi-phase motions [**48, 10**] and the use of an external force field based on a diffused gradient of an edge map [**50**].



(a) A curve in a 2D vector field

(b) The inward flux through the curve is maximized

FIGURE 3.1. An illustration of the flux maximizing principle.

However, in practice these models often fail when applied to images of thin elongated structures such as blood vessels. In this chapter we shall derive geometric flows designed specifically for this purpose. The main intuition is that in angiography images, areas of flowing blood appear bright. This leads to the constraint that in the vicinity of blood vessel boundaries, the gradient vector field is locally orthogonal to them. Thus, a natural principal to use towards the recovery of these boundaries is to maximize the inward flux of the gradient vector field through an evolving curve (in 2D) or surface (in 3D). To illustrate, Figure 3.1(a) shows an initial closed curve placed in a dense 2D vector field. To maximize the inward flux of the vector field through the curve, the curve must evolve so that its inward normals are locally aligned with the vector field, Figure 3.1(b).

We begin this chapter with a brief review of related geometric flows in the literature with an emphasis on the length and area minimizing flows. We shall then derive the flux maximizing flows both in 2D and 3D.

1. Geometric Flows

Consider a boundary, either a curve in two dimensions or a surface in three dimensions, separating one region from another. Imagine that this curve/surface moves in a direction normal to itself (where the normal direction is oriented with

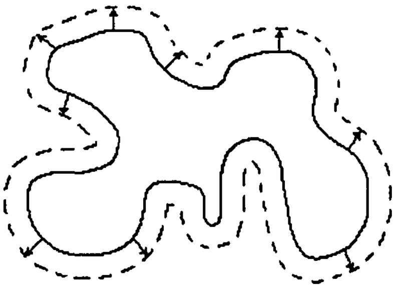


FIGURE 3.2. An illustration of a front propagating in time along its normals. The new position of the front is shown with a dashed line. The normals are shown only in some places to facilitate viewing.

respect to an inside and an outside) with a known speed function F (Figure 3.2). The goal is to track the motion as it evolves. For now we will consider motions of curves in 2D. However, the development will generalize to the case of moving surfaces.

Let γ be a simple, smooth, closed initial curve in R^2, and let $\gamma(t)$ be the one-parameter family of curves generated by moving γ along its normal vector field with speed F. Here, F is a scalar function. Let the position vector $\mathcal{C}(s,t) = (x(s,t), y(s,t))$ parameterize γ at time t , where $0 \leq s \leq S$, and assume periodic boundary conditions $\mathcal{C}(0,t) = \mathcal{C}(S,t)$. The curve evolution equation is then

$$\frac{\partial \mathcal{C}}{\partial t} = F\mathcal{N}, \tag{3.1}$$

where $\mathcal{N}$ is the unit inward normal to the curve. This is the basic equation that we will study and simulate.

2. Area and Length Minimizing Flows

Generally, the speed function F can be any scalar function. It can be a highly complex function simulating some physical phenomena or it can be as simple as a constant. We will now consider in some detail two interesting choices of F, that have been widely used in the literature. Both can be interpreted as gradient flows that minimize a particular geometric functional.

2.1. Curvature Flow. Consider the 2D case where $F = \kappa$, the Euclidean curvature of the curve. Then Eq. (3.1) becomes:

$$\frac{\partial \mathcal{C}}{\partial t} = \kappa \mathcal{N}. \tag{3.2}$$

This equation, known as the *geometric heat equation*, has been studied extensively in the literature [**17, 11, 14, 18**] . It has a number of properties which make it very useful in image processing. In particular, Eq. (3.2) is the Euclidean curve shortening flow, in the sense that under this evolution the Euclidean perimeter shrinks as quickly as possible. Since we will need a similar argument for the flux maximizing flow presented in this thesis, let us work out some details.

Define the length functional for $\mathcal{C} = \mathcal{C}(p,t)$, a smooth family of closed curves where t parameterizes the family and p the given curve:

$$L(t) = \int_0^1 \|\frac{\partial \mathcal{C}}{\partial p}\| \, dp.$$

Without loss of generality we assume that $0 \leq p \leq 1$. Differentiating with respect to t, and using integration by parts, one can show that

$$L'(t) = -\int_0^{L(t)} \langle \frac{\partial \mathcal{C}}{\partial t}, \kappa \mathcal{N} \rangle \, ds \; ,$$

where $ds = \|\frac{\partial \mathcal{C}}{\partial p}\| \, dp$ denotes the arc-length parameter. Thus, the direction in which $L(t)$ is decreasing most rapidly is obtained when $\frac{\partial \mathcal{C}}{\partial t} = \kappa \mathcal{N}$. Another interesting and fundamental property shown by Grayson, Gage and Hamilton [**17, 18**] is that simple closed curves converge to "round" points in the limit when evolving accordingly to Eq. (3.2), without developing singularities or topological changes , see Figure 3.3.

In 3D, when evolving surfaces with $F = H$, where H is the mean curvature of a surface, the flow can be shown to shrink surface area at the fastest possible rate. However, topological splits can occur, see Figure 3.4. Evans and Spruck and Chen, Giga and Goto have independently studied mean curvature flow of any hypersurface

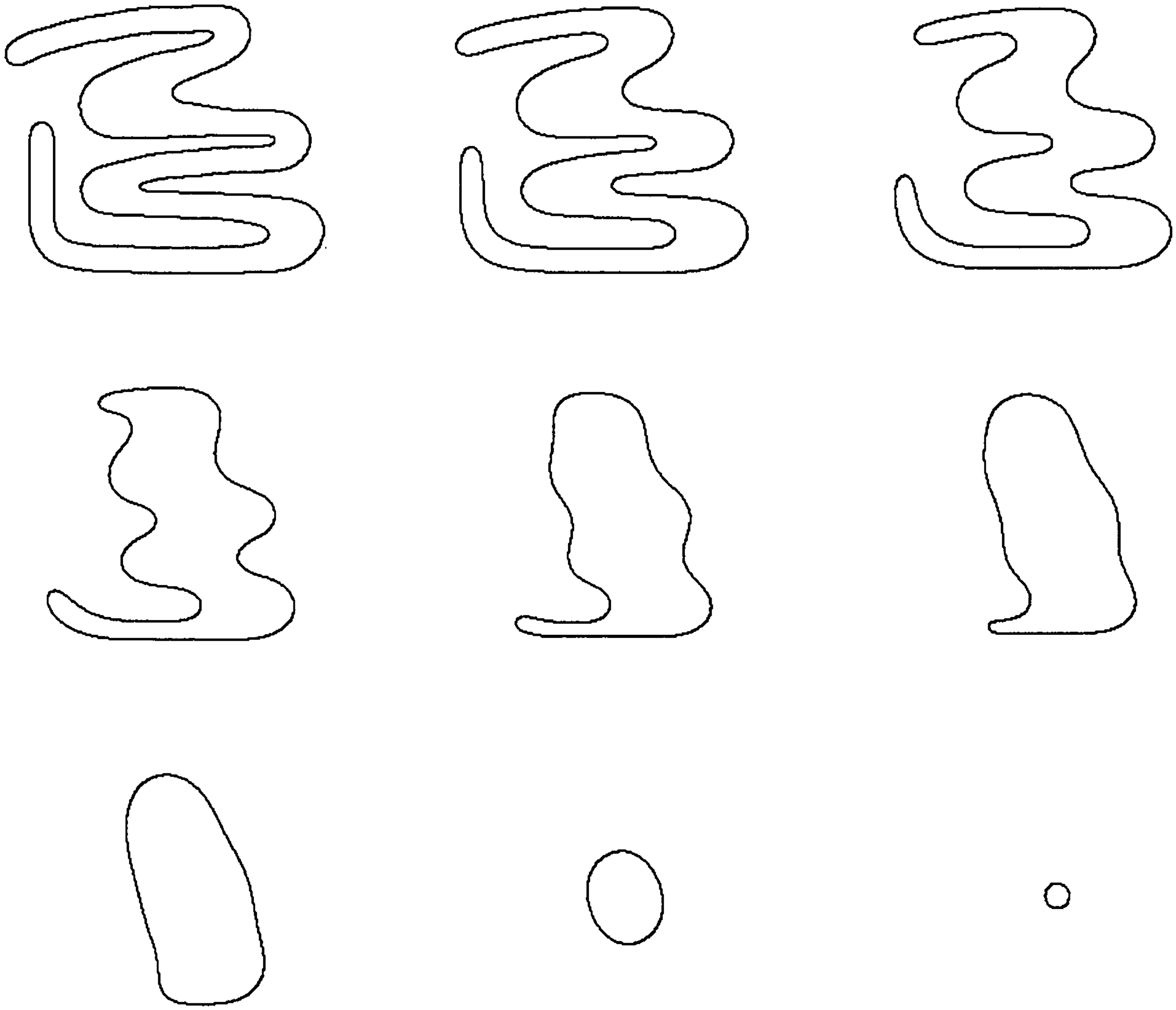


FIGURE 3.3. An illustration of a curve evolving accordingly to the equation $\frac{\partial \mathcal{C}}{\partial t} = \kappa \mathcal{N}$. A front propagating with speed equal to curvature must collapse smoothly to a round point in the limit [**17, 18**].

as a level set problem, and have proved the existence, uniqueness, and stability of solutions using viscosity theory [**11, 14**].

2.2. Constant Flow. Now consider the case where $F = 1$. This leads to the constant motion flow or Blum's "grassfire" transformation [**5**]

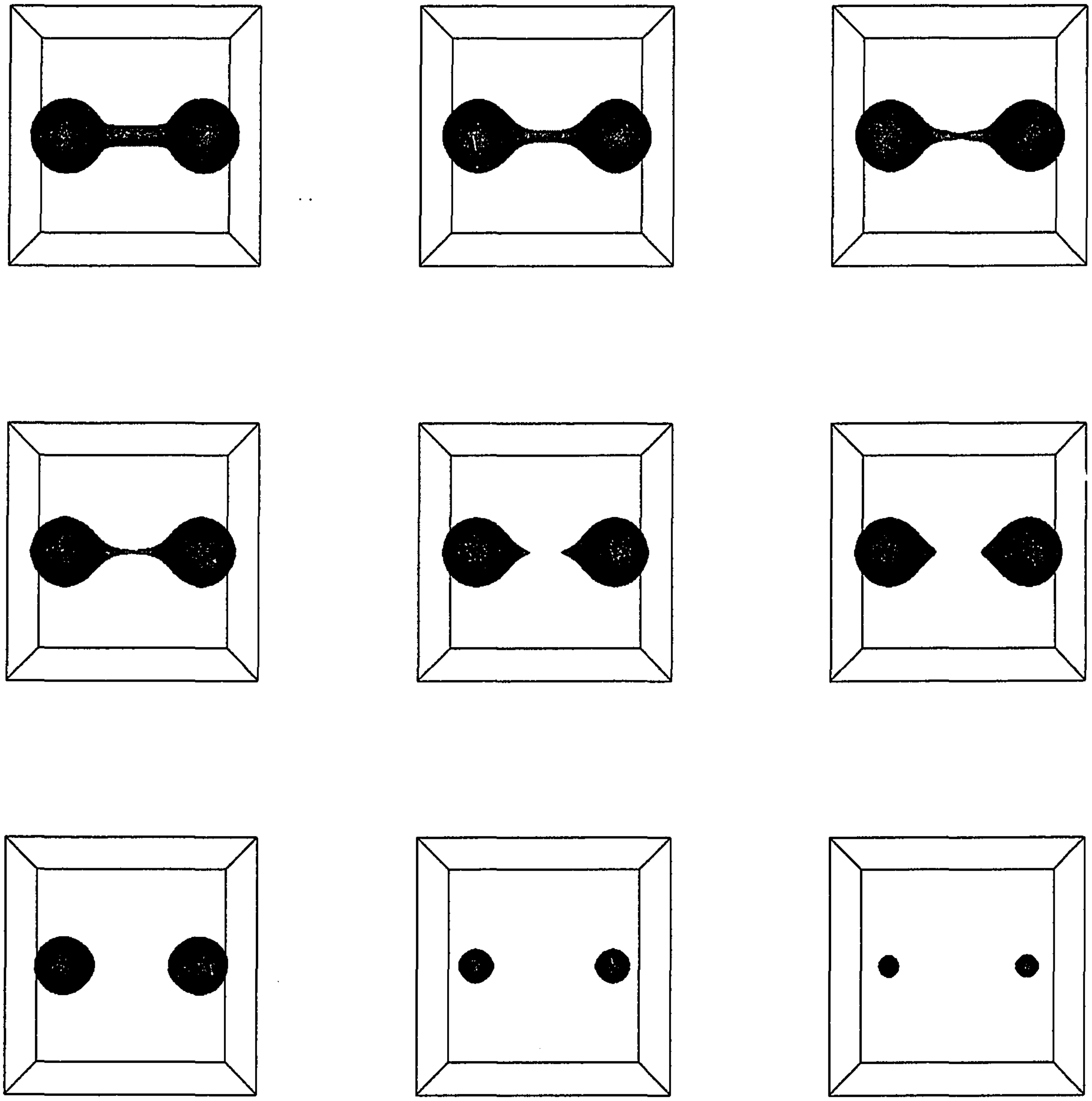


FIGURE 3.4. An illustration of a dumbbell evolving under its mean curvature in 3D. Because one of the principal curvatures around the neck is very high, the handle shrinks inward and disappears. This result shows that a simply connected surface propagating under its mean curvature in 3D can break into separate surfaces as it evolves [44].

$$\frac{\partial \mathcal{C}}{\partial t} = \mathcal{N}.$$

In analogy with the curvature flow that minimizes length, it can be shown that this evolution shrinks the area enclosed by the curve (in 2D) or the volume enclosed by the surface (in 3D) at the fastest possible rate. In contrast with the curvature flow which remains regular throughout the evolution, the constant motion flow can cause a smooth curve to evolve to a singular one and topological changes may occur, see Figure 3.5.

The constant motion and curvature motion evolutions were introduced by Osher and Sethian in the context of flame propagation in the early 80's [**42, 37**]. They were later introduced to the computer vision community in the context of visual shape analysis by Kimia, Tannenbaum and Zucker [**21, 21**]. Affine invariant versions of the curvature flow were also studied [**39**] and properties of scale spaces based on these flows were established [**1**]. In the following section we shall show how modifications of these flows can be adapted to the problem of shape segmentation.

3. Active Contours

The main idea behind using curve and surface evolution for segmentation is to introduce constraints derived from an image-based stopping force, so that they cling to features of interest in an intensity image. This type of motion can by simulated by the following equation, introduced independently by Casselles *et al.* [**8**] and Malladi *et al.* [**34**]

$$\frac{\partial \mathcal{C}}{\partial t} = \phi(\alpha + \beta \cdot \kappa)\mathcal{N}.$$

Here α and β are constants, κ is the Euclidean curvature, and the scalar function ϕ is constructed to have local minima near edges. The constant term α is needed to keep the curve moving in the desired direction and the curvature term $\beta \cdot \kappa$ regularizes

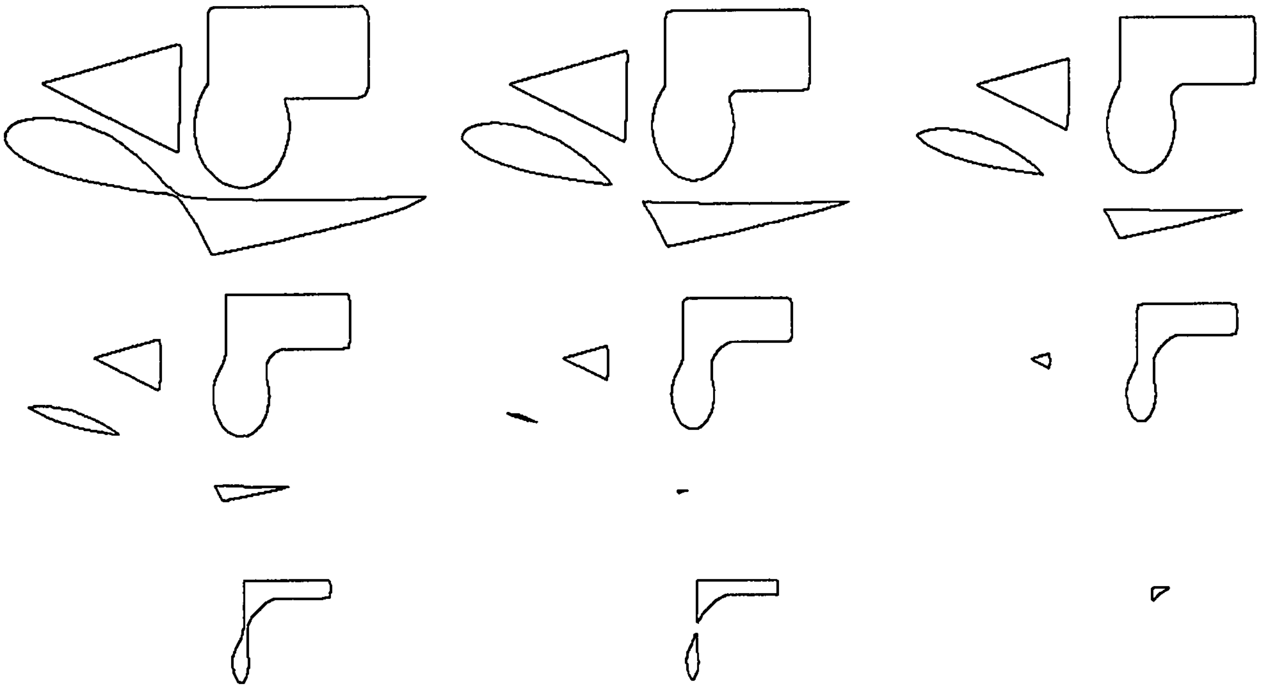


FIGURE 3.5. An illustration of a collection of curves evolving with a constant speed. Notice that sharp corners are developed. Also, one of the curves splits into two and then three pieces.

the evolving front. The multiplicative image-based stopping term ϕ, ensures that the curve halts near edges. In [8, 34] , for example, ϕ takes the form:

$$\phi = \frac{1}{1 + |\nabla G_\sigma \star I|^n}$$

where I is the grayscale image and G_σ is a Gaussian smoothing filter.

These active contour models have the significant advantage over classical snakes [19] that changes in topology due to the splitting and merging of multiple contours are handled in a natural way. Also, the original model can be refined by deriving weighted length and weighted area minimizing versions, where the essential idea is to define a conformal length or area metric and compute the corresponding gradient evolution equations [20, 46]. Figures 3.6 and 3.7 depict examples of shape segmentation in 2D and 3D using these flows.

It has been recognized that a practical weakness of most geometric flows with stopping terms based purely on local image gradients is that they may "leak" in the presence of weak or low contrast boundaries, are not suitable for segmenting textures and typically require the initial curve or curves to lie entirely inside or outside the regions to be segmented. Thus, a number of researchers have sought to derive flows which take into account the statistics of the regions enclosed by the evolving curves [38, 51]. Further developments include multi-phase motions, which allow triple points to be captured [48, 10], as well as the incorporation of an external force field based on a diffused gradient of an edge map [50]. However, most geometric flows are not able to capture elongated low contrast structures well, such as blood vessels viewed in 2D and 3D angiography images. At places where such structures are narrow edge gradients may be weak due to partial volume effects and it is also unclear how to robustly measure region statistics. Approaches to regularizing the flow in 3D by introducing a term proportional to mean curvature have the unfortunate effect of annihilating such structures. In this thesis we propose a new geometric flow to address these problems. Our approach is motivated by the simple observation that blood flows in the direction of vessels. Brightness in angiography images is proportional to the magnitude of the blood flow velocity. This leads to the constraint that in the vicinity of blood vessel boundaries, the gradient vector field of the image should be locally orthogonal to them. Thus, a natural principle to use towards the recovery of these boundaries is to maximize the inward flux of the gradient vector field through an evolving curve (in 2D) or surface (in 3D), see Figure 3.1. Both in 2D and 3D, this leads to a simple and elegant interpretation which is essentially parameter free.

4. Flux Maximizing Flows, 2D Case

Let $\mathcal{C} = \mathcal{C}(p, t)$ be a smooth family of closed curves evolving in the plane. Here t parameterizes the family and p the given curve. Without loss of generality we shall

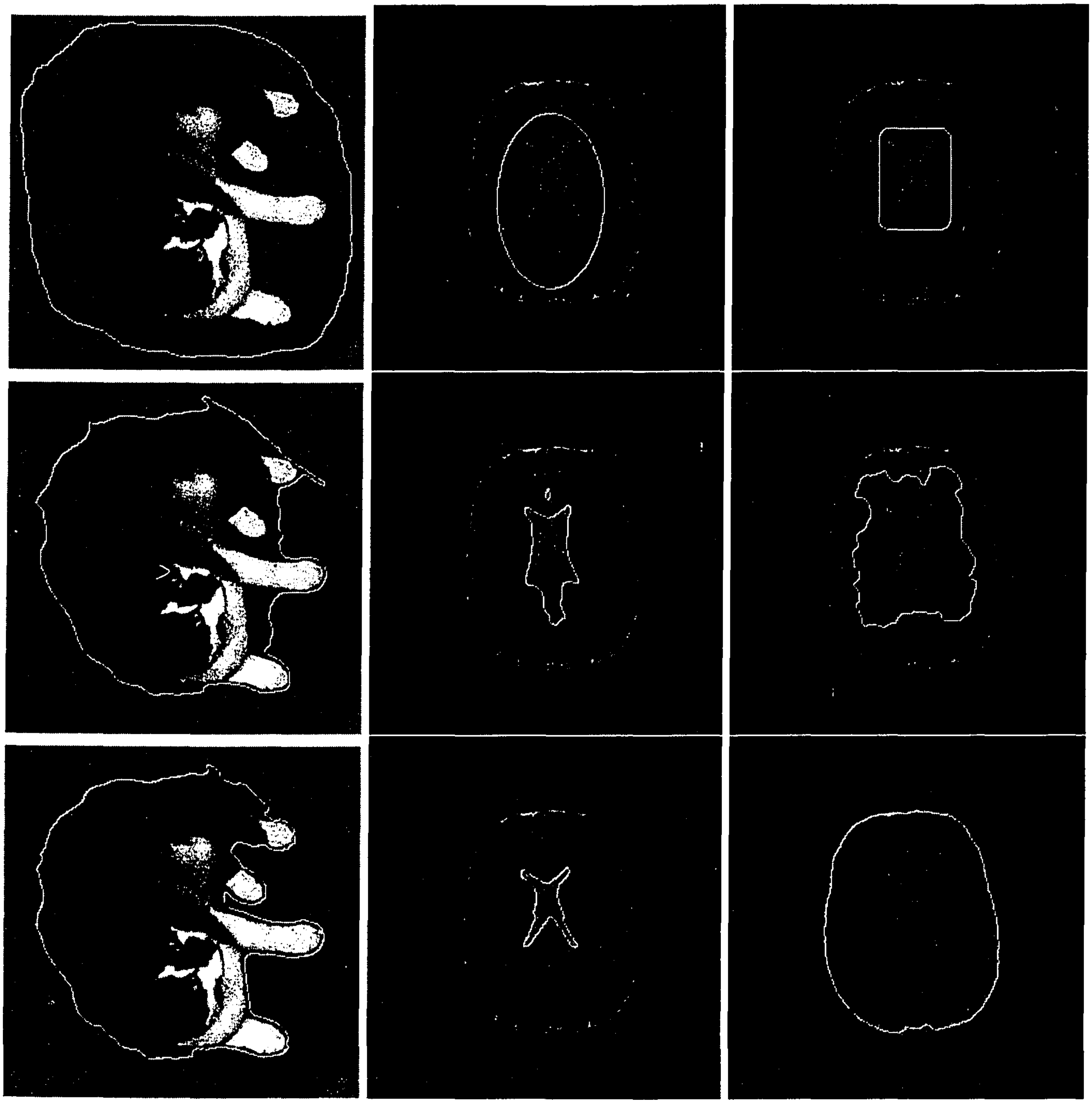


FIGURE 3.6. An illustration of shape segmentation using curve evolution based active contours. Different stages of the evolution are shown from top to bottom. COLUMN ONE: An inward flow for segmenting a black and white cat. COLUMN TWO: Segmentation of the brain ventricles on a 256x256 MRI image with inward flow. COLUMN THREE: Segmentation of the inner surface of the brain with an outward flow on the same MRI image.

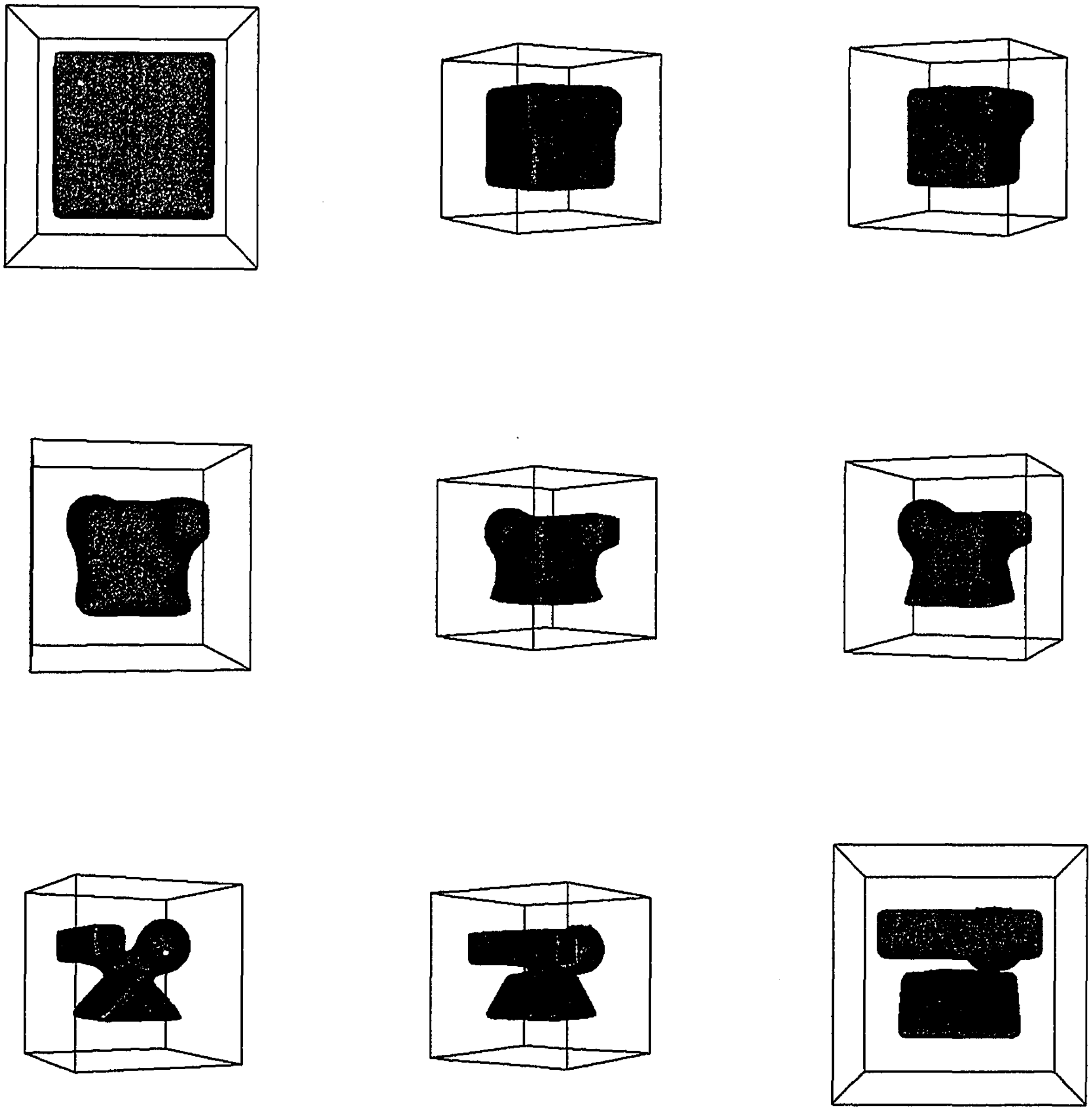


FIGURE 3.7. An illustration of shape segmentation using surface evolution based active contours on an artificial 3D scene composed of a sphere, a prism and a parallelepiped. The initial surface (top left corner) is a cube placed outside the objects. Different stages of the evolution are shown from left to right, top to bottom. Notice how three objects are captured using a single initial surface.

assume that $0 \leq p \leq 1$, i.e., that $\mathcal{C}(0,t) = \mathcal{C}(1,t)$. We shall also assume that the first derivatives exist and that $\mathcal{C}'(0,t) = \mathcal{C}'(1,t)$. The unit tangent $\mathcal{T}$ and the unit inward normal $\mathcal{N}$ to C are given by

$$\mathcal{T} = \frac{\begin{pmatrix} x_p \\ y_p \end{pmatrix}}{\|\mathcal{C}_p\|} = \begin{pmatrix} x_s \\ y_s \end{pmatrix}$$

$$\mathcal{N} = \frac{\begin{pmatrix} -y_p \\ x_p \end{pmatrix}}{\|\mathcal{C}_p\|} = \begin{pmatrix} -y_s \\ x_s \end{pmatrix}$$

where s is the arc-length parameterization of the curve. Now consider a vector field $\mathcal{V} = (V_1(x,y), V_2(x,y))$ defined for each point (x,y) in $\mathcal{R}^2$. The total inward flux of the vector field through the curve is given by the contour integral

$$Flux(t) = \int_0^1 \langle \mathcal{V}, \mathcal{N} \rangle \, \|\mathcal{C}_p\| \, dp = \int_0^{L(t)} \langle \mathcal{V}, \mathcal{N} \rangle \, ds \tag{3.3}$$

where $L(t)$ is the Euclidean length of the curve. The circulation of the vector field along the curve is defined in an analogous fashion as

$$Circ(t) = \int_0^1 \langle \mathcal{V}, \mathcal{T} \rangle \, \|\mathcal{C}_p\| \, dp = \int_0^{L(t)} \langle \mathcal{V}, \mathcal{T} \rangle \, ds.$$

The main theoretical result in 2D of this thesis is given by the following theorem:

THEOREM 1. *The direction in which the inward flux of the vector field $\mathcal{V}$ through the curve C is increasing most rapidly is given by $\frac{\partial \mathcal{C}}{\partial t} = div(\mathcal{V})\mathcal{N}$.*

In other words, the gradient flow which maximizes the rate of increase of the total inward flux is obtained by moving each point of the curve in the direction of the inward normal by an amount proportional to the divergence of the vector field. In order to tailor the flux maximizing flow to blood vessel segmentation, we shall consider the gradient $\nabla \mathbf{I}$ of the original intensity image $\mathbf{I}$ to be the vector field $\mathcal{V}$ whose inward flux through the evolving curve is maximized. The 2D flux calculation

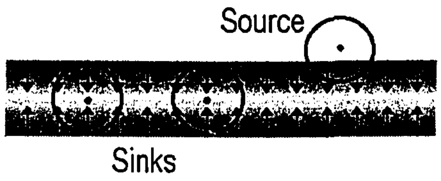


FIGURE 3.8. An illustration of the gradient vector field in the vicinity of a blood vessel. Assuming a uniform background intensity, at its centerline, at the scale of the vessel's width, the total outward flux, which is proportional to the divergence, is negative. Outside the vessel, at a smaller scale, the total outward flux is positive.

is illustrated in Figure 3.8. Hence, the flux maximizing flow in Eq (3.5) has the desirable effect that when seeds are placed within blood vessels, the sources outside boundaries prevent the flow from leaking.

Proof: Define the perpendicular to a vector $\mathcal{W} = (a, b)$ as $\mathcal{W}^\perp = (-b, a)$. The following properties hold:

$$\begin{aligned} \langle \mathcal{U}, \mathcal{W}^\perp \rangle &= - \langle \mathcal{U}^\perp, \mathcal{W} \rangle \\ \langle \mathcal{U}^\perp, \mathcal{W}^\perp \rangle &= \langle \mathcal{U}, \mathcal{W} \rangle . \end{aligned} \tag{3.4}$$

We now compute the first variation of the flux functional with respect to t

$$Flux'(t) = \underbrace{\int_0^1 \langle \mathcal{V}_t, \mathcal{N} \rangle \, \|\mathcal{C}_p\| \, dp}_{I_1} + \underbrace{\int_0^1 \langle \mathcal{V}, (\mathcal{N} \, \|\mathcal{C}_p\|)_t \rangle \, dp}_{I_2} .$$

Switching to parametrization by s for I_1 and using

$$\mathcal{V}_t = \left(\frac{\partial V_1}{\partial t}, \frac{\partial V_2}{\partial t} \right) = \left(\langle \nabla V_1, \mathcal{C}_t \rangle , \langle \nabla V_2, \mathcal{C}_t \rangle \right)$$

we have

$$I_1 = \int_0^{L(t)} \langle \mathcal{C}_t, x_s \nabla V_2 - y_s \nabla V_1 \rangle \, ds.$$

With $\mathcal{N} = (-y_p, x_p) / \, \|\mathcal{C}_p\|$, I_2 works out to be

$$\int_0^1 \left\langle \mathcal{V}, \begin{pmatrix} -y_{pt} \\ x_{pt} \end{pmatrix} \right\rangle dp.$$

Now, using integration by parts

$$I_2 = \underbrace{\left\langle \mathcal{V}, \begin{pmatrix} -y_t \\ x_t \end{pmatrix} \right\rangle \Bigg]_0^1}_{\text{equals 0}} - \int_0^1 \left\langle \begin{pmatrix} -y_t \\ x_t \end{pmatrix}, \mathcal{V}_p \right\rangle dp$$

Using the properties of scalar products in Eq. (3.4) and the fact that

$$\mathcal{V}_p = \left(\frac{\partial V_1}{\partial p}, \frac{\partial V_2}{\partial p} \right) = \left(\langle \nabla V_1, \mathcal{C}_p \rangle, \langle \nabla V_2, \mathcal{C}_p \rangle \right)$$

we can rewrite I_2 as follows

$$I_2 = \int_0^1 \left\langle \begin{pmatrix} x_t \\ y_t \end{pmatrix}, \mathcal{V}_p{}^{\perp} \right\rangle dp$$

$$= \int_0^1 \left\langle \mathcal{C}_t, \begin{pmatrix} -\langle \nabla V_2, \mathcal{C}_p \rangle \\ \langle \nabla V_1, \mathcal{C}_p \rangle \end{pmatrix} \right\rangle dp.$$

Switching to arc-length parameterization

$$I_2 = \int_0^{L(t)} \left\langle \mathcal{C}_t, \begin{pmatrix} -\langle \nabla V_2, \mathcal{T} \rangle \\ \langle \nabla V_1, \mathcal{T} \rangle \end{pmatrix} \right\rangle ds.$$

Combining I_1 and I_2, the first variation of the flux is

$$\int_0^{L(t)} \left\langle \mathcal{C}_t, x_s \nabla V_2 - y_s \nabla V_1 + \begin{pmatrix} -\langle \nabla V_2, \mathcal{T} \rangle \\ \langle \nabla V_1, \mathcal{T} \rangle \end{pmatrix} \right\rangle ds.$$

Thus, for the flux to increase as fast as possible, the two vectors should be made parallel:

$$\mathcal{C}_t = x_s \nabla V_2 - y_s \nabla V_1 + \begin{pmatrix} -\langle \nabla V_2, \mathcal{T} \rangle \\ \langle \nabla V_1, \mathcal{T} \rangle \end{pmatrix}.$$

Decomposing the above three vectors in the Frenet frame $\{\mathcal{T}, \mathcal{N}\}$, dropping the tangential terms (which affect only the parameterization of the curve) and making use of the properties of scalar products

$$\mathcal{C}_t = \{x_s\langle \nabla V_2, \mathcal{N}\rangle - y_s\langle \nabla V_1, \mathcal{N}\rangle$$
$$+\left\langle \begin{pmatrix} -\langle \nabla {V_2}^{\perp}, \mathcal{N}\rangle \\ \langle \nabla {V_1}^{\perp}, \mathcal{N}\rangle \end{pmatrix}, \mathcal{N}\right\rangle \}\mathcal{N}$$

Expanding all terms in the above equation

$$\begin{aligned} \mathcal{C}_t = (x_s(-V_{2x}.y_s + V_{2y}.x_s) - y_s(-V_{1x}.y_s + V_{1y}.x_s) + \\ \left\langle \begin{pmatrix} -V_{2y}.y_s - V_{2x}.x_s \\ V_{1y}.y_s + V_{1x}.x_s \end{pmatrix}, \mathcal{N}\right\rangle)\mathcal{N} \\ = (-V_{2x}.x_s.y_s + V_{2y}.{x_s}^2 + V_{1x}.{y_s}^2 - V_{1y}.x_s.y_s \\ +V_{2y}.{y_s}^2 + V_{2x}.x_s.y_s + V_{1y}.x_s.y_s + V_{1x}.{x_s}^2)\mathcal{N} \\ = (V_{1x}({x_s}^2 + {y_s}^2) + V_{2y}({x_s}^2 + {y_s}^2))\mathcal{N} \\ = (V_{1x} + V_{2y})\mathcal{N} = div(\mathcal{V})\mathcal{N} \,\square \qquad (3.5) \end{aligned}$$

As a corollary to Theorem 1, we have

COROLLARY 1. *The direction in which the circulation of of the vector field $\mathcal{V}$ along the curve $\mathcal{C}$ is increasing most rapidly is given by $\frac{\partial \mathcal{C}}{\partial t} = div(\mathcal{V}^{\perp})\mathcal{N}$.*

Proof: Using the properties of scalar products in Eq. (3.4)

$$\begin{aligned} Circ(t) &= \int_0^{L(t)} \langle \mathcal{V}, \mathcal{T}\rangle\, ds \\ &= \int_0^{L(t)} \langle \mathcal{V}^{\perp}, \mathcal{T}^{\perp}\rangle\, ds \\ &= \int_0^{L(t)} \langle \mathcal{V}^{\perp}, \mathcal{N}\rangle\, ds \end{aligned}$$

Hence the circulation of the vector field $\mathcal{V}$ along the curve is just the inward flux of the vector field $\mathcal{V}^{\perp}$ through it and the result follows from Theorem 1.

5. Flux maximizing flows, 3D Case

Let $\mathcal{S} : [0,1] \times [0,1] \to \mathcal{R}^3$ denote a compact embedded surface with (local) coordinates (u, v). Let $\mathcal{N}$ be the inward unit normal. We set

$$\mathcal{S}_u := \frac{\partial \mathcal{S}}{\partial u}, \quad \mathcal{S}_v := \frac{\partial \mathcal{S}}{\partial v}.$$

Then the infinitesimal area on $\mathcal{S}$ is given by

$$\begin{aligned} d\mathcal{S} &= (\|\mathcal{S}_u\|^2 \|\mathcal{S}_v\|^2 - \langle \mathcal{S}_u, \mathcal{S}_v \rangle^2)^{1/2} dudv \\ &= \|\mathcal{S}_u \wedge \mathcal{S}_v\| dudv. \end{aligned}$$

Let $\mathcal{V} = (V_1(x,y,z), V_2(x,y,z), V_3(x,y,z))$ be a vector field defined for each point (x, y, z) in $\mathcal{R}^3$. The total inward flux of the vector field through the surface is defined by the surface integral

$$Flux(t) = \int_0^1 \int_0^1 \langle \mathcal{V}, \mathcal{N} \rangle \, d\mathcal{S}. \tag{3.6}$$

We shall now show that our result in 2D extends to 3D as well.

Theorem 2. *The direction in which the inward flux of the vector field $\mathcal{V}$ through the surface $\mathcal{S}$ is increasing most rapidly is given by $\frac{\partial \mathcal{S}}{\partial t} = div(\mathcal{V})\mathcal{N}$.*

It should not come as a surprise that the flux maximizing flow has the same form in 3D as in 2D, since the flux formulation is unchanged. However, the calculation is more subtle. We also note that in fluid mechanics the circulation in 3D is defined along a chosen closed 3D curve on the surface, but this is not a surface integral. Hence, Corollary 1 does not generalize to 3D.

Proof: The essential idea is to calculate the first variation of the flux functional with respect to t:

$$Flux'(t) = \underbrace{\int_0^1 \int_0^1 \langle \mathcal{V}_t, \mathcal{N} \rangle \, \|\mathcal{S}_u \wedge \mathcal{S}_v\| dudv}_{I_1} + \underbrace{\int_0^1 \int_0^1 \langle \mathcal{V}, (\mathcal{N} \|\mathcal{S}_u \wedge \mathcal{S}_v\|)_t \rangle \, \|dudv}_{I_2}.$$

With $\mathcal{S} = (x(u,v,t), y(u,v,t), z(u,v,t))$, the unit normal vector is given by the normalized cross product of two vectors in the tangent plane:

$$\begin{aligned}\mathcal{N} &= \frac{\mathcal{S}_u \wedge \mathcal{S}_v}{||\mathcal{S}_u \wedge \mathcal{S}_v||} = \frac{(N_1, N_2, N_3)}{||\mathcal{S}_u \wedge \mathcal{S}_v||} \\ &= \frac{(y_u z_v - y_v z_u), (x_v z_u - x_u z_v), (x_u y_v - x_v y_u)}{||(y_u z_v - y_v z_u), (x_v z_u - x_u z_v), (x_u y_v - x_v y_u)||}.\end{aligned}$$

I_1 is then given by

$$\int_0^1 \int_0^1 \langle \mathcal{S}_t, (N_1 \nabla V_1 + N_2 \nabla V_2 + N_3 \nabla V_3) \rangle \, dudv,$$

where the integrand has the desired form of an inner product of $\mathcal{S}_t$ with another vector. For I_2, it would seem reasonable to express the normal in terms of the partial derivatives $x_u, x_v, y_u, y_v, z_u, z_v$ and then proceed. However, this does not lead to an obvious solution. In fact, it turns out to be advantageous to instead express the normal $\mathcal{N}$ as $\frac{\mathcal{S}_u \wedge \mathcal{S}_v}{||\mathcal{S}_u \wedge \mathcal{S}_v||}$. In this case, the integral can be rewritten as

$$\int_0^1 \int_0^1 \langle \mathcal{V}, (\mathcal{S}_u \wedge \mathcal{S}_v)_t \rangle \, dudv.$$

The trick now is to exploit the fact that for any vectors A, B and C, the following properties of inner products and cross products hold:

$$\begin{aligned} A \wedge B &= -B \wedge A \\ \langle A, (B \wedge C) \rangle &= \langle (A \wedge B), C \rangle \\ (A \wedge B)_t &= (A_t \wedge B) + (A \wedge B_t). \end{aligned}$$

Hence, I_2 can be written as

$$\begin{aligned} I_2 &= \int_0^1 \int_0^1 \langle \mathcal{V}, (\mathcal{S}_{ut} \wedge \mathcal{S}_v + \mathcal{S}_u \wedge \mathcal{S}_{vt}) \rangle \, dudv \\ &= \int_0^1 \int_0^1 \langle \mathcal{V}, (\mathcal{S}_{ut} \wedge \mathcal{S}_v) \rangle \, dudv + \int_0^1 \int_0^1 \langle \mathcal{V}, (\mathcal{S}_u \wedge \mathcal{S}_{vt}) \rangle \, dudv \\ &= \int_0^1 \int_0^1 - \langle \mathcal{V}, (\mathcal{S}_v \wedge \mathcal{S}_{ut}) \rangle \, dudv + \int_0^1 \int_0^1 \langle \mathcal{V}, (\mathcal{S}_u \wedge \mathcal{S}_{vt}) \rangle \, dudv \\ &= \int_0^1 \underbrace{\left[\int_0^1 - \langle (\mathcal{V} \wedge \mathcal{S}_v), \mathcal{S}_{ut} \rangle \, du \right]}_{I_3} dv + \int_0^1 \underbrace{\left[\int_0^1 \langle (\mathcal{V} \wedge \mathcal{S}_u), \mathcal{S}_{vt} \rangle \, dv \right]}_{I_4} du \end{aligned} \tag{3.7}$$

Using integration by parts, I_3 works out to be

$$\underbrace{- \langle (\mathcal{V} \wedge \mathcal{S}_v), \mathcal{S}_t \rangle]_0^1}_{\text{equals 0}} + \int_0^1 \langle \mathcal{S}_t, (\mathcal{V} \wedge \mathcal{S}_v)_u \rangle du$$

Similarly, using integration by parts, I_4 works out to be

$$\underbrace{\langle (\mathcal{V} \wedge \mathcal{S}_u), \mathcal{S}_t \rangle]_0^1}_{\text{equals 0}} - \int_0^1 \langle \mathcal{S}_t, (\mathcal{V} \wedge \mathcal{S}_u)_v \rangle dv$$

Combining I_3 and I_4, I_2 works out to be

$$\int_0^1 \int_0^1 \langle \mathcal{S}_t, (\mathcal{V} \wedge \mathcal{S}_v)_u - (\mathcal{V} \wedge \mathcal{S}_u)_v \rangle \, dudv.$$

It can now be seen that the integrand in I_2 has the desired form of the inner product of $\mathcal{S}_t$ with another vector. Hence, combining I_1 and I_2, the first variation of the flux is

$$\int_0^1 \int_0^1 \langle \mathcal{S}_t, N_1 \nabla V_1 + N_2 \nabla V_2 + N_3 \nabla V_3 + (\mathcal{V} \wedge \mathcal{S}_v)_u - (\mathcal{V} \wedge \mathcal{S}_u)_v \rangle \, dudv.$$

Thus, for the inward flux to increase as fast as possible, the two vectors should be made parallel:

$$\mathcal{S}_t = N_1 \nabla V_1 + N_2 \nabla V_2 + N_3 \nabla V_3 + (\mathcal{V} \wedge \mathcal{S}_v)_u - (\mathcal{V} \wedge \mathcal{S}_u)_v. \tag{3.8}$$

The above equation in fact is a legitimate form of the 3D flux maximizing gradient flow, but it looks rather complicated. It can be further simplified by noting that the components of the flow in the tangential plane to the surface $\mathcal{S}$ affect only the parameterization of the surface, but not its evolved shape. Hence, they can be dropped. The normal component of the flow can be calculated by taking the inner product of the right hand side of Eq. (3.8) with the unit normal vector

$$\mathcal{N} = \frac{(y_u z_v - y_v z_u), (x_v z_u - x_u z_v), (x_u y_v - x_v y_u)}{\|(y_u z_v - y_v z_u), (x_v z_u - x_u z_v), (x_u y_v - x_v y_u)\|}$$

to give

$$\begin{aligned} \mathcal{S}_t = \; & \langle (y_u z_v - y_v z_u)\nabla V_1 + (x_v z_u - x_u z_v)\nabla V_2 + (x_u y_v - x_v y_u)\nabla V_3 + \\ & (\mathcal{V} \wedge \mathcal{S}_v)_u - (\mathcal{V} \wedge \mathcal{S}_u)_v, \\ & \tfrac{(y_u z_v - y_v z_u),(x_v z_u - x_u z_v),(x_u y_v - x_v y_u)}{\|(y_u z_v - y_v z_u),(x_v z_u - x_u z_v),(x_u y_v - x_v y_u)\|} \rangle \mathcal{N} \end{aligned}$$

It is now a straightforward task to expand the terms in this expression:

$$\begin{aligned} \mathcal{S}_t = \; & \tfrac{(y_u z_v - y_v z_u)}{|||\mathcal{S}_u \wedge \mathcal{S}_v|||^2} (V_{1x}(y_u z_v - y_v z_u) + V_{1y}(x_v z_u - x_u z_v) + V_{1z}(x_u y_v - x_v y_u)) \\ + \; & \tfrac{(x_v z_u - x_u z_v)}{|||\mathcal{S}_u \wedge \mathcal{S}_v|||^2} (V_{2x}(y_u z_v - y_v z_u) + V_{2y}(x_v z_u - x_u z_v) + V_{2z}(x_u y_v - x_v y_u)) \\ + \; & \tfrac{(x_u y_v - x_v y_u)}{|||\mathcal{S}_u \wedge \mathcal{S}_v|||^2} (V_{3x}(y_u z_v - y_v z_u) + V_{3y}(x_v z_u - x_u z_v) + V_{3z}(x_u y_v - x_v y_u)) \\ + \; & \tfrac{1}{|||\mathcal{S}_u \wedge \mathcal{S}_v|||^2} \langle (\mathcal{V}_u \wedge \mathcal{S}_v), (y_u z_v - y_v z_u, x_v z_u - x_u z_v, x_u y_v - x_v y_u) \rangle \\ - \; & \tfrac{1}{|||\mathcal{S}_u \wedge \mathcal{S}_v|||^2} \langle (\mathcal{V}_v \wedge \mathcal{S}_u), (y_u z_v - y_v z_u, x_v z_u - x_u z_v, x_u y_v - x_v y_u) \rangle . \end{aligned}$$

With

$$\begin{aligned} \mathcal{V}_u \wedge \mathcal{S}_v = \; & (z_v(V_{2x}x_u + V_{2y}y_u + V_{2z}z_u) - y_v(V_{3x}x_u + V_{3y}y_u + V_{3z}z_u), \\ & x_v(V_{3x}x_u + V_{3y}y_u + V_{3z}z_u) - z_v(V_{1x}x_u + V_{1y}y_u + V_{1z}z_u), \\ & y_v(V_{1x}x_u + V_{1y}y_u + V_{1z}z_u) - x_v(V_{2x}x_u + V_{2y}y_u + V_{2z}z_u)). \end{aligned}$$

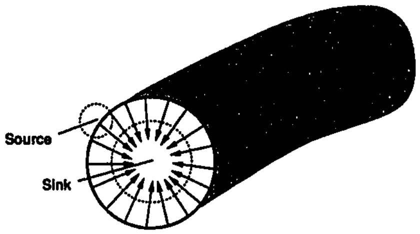


FIGURE 3.9. An illustration of the gradient vector field of an angiography image along a cross-section of a 3D blood vessel. Assuming a uniform background intensity, at the scale of the vessel's width the total inward flux is positive (a sink). Outside the vessel, at a smaller scale, the total inward flux is negative (a source).

and

$$\begin{aligned}-\mathcal{V}_v \wedge \mathcal{S}_u = \; & (-z_u(V_{2x}x_v + V_{2y}y_v + V_{2z}z_v) + y_u(V_{3x}x_v + V_{3y}y_v + V_{3z}z_v), \\ & -x_u(V_{3x}x_v + V_{3y}y_v + V_{3z}z_v) + z_u(V_{1x}x_v + V_{1y}y_v + V_{1z}z_v), \\ & -y_u(V_{1x}x_v + V_{1y}y_v + V_{1z}z_v) + x_u(V_{2x}x_v + V_{2y}y_v + V_{2z}z_v))\end{aligned}$$

the terms can be grouped and simplified. The curious result is that most cancel, leaving the following simple and elegant form for the 3D flux maximizing flow:

$$\mathcal{S}_t = (V_{1x} + V_{2y} + V_{3z})\mathcal{N} = div(\mathcal{V})\mathcal{N} \; \Box \tag{3.9}$$

In order to use the above result for blood vessel segmentation, in analogy to the 2D case we shall consider the gradient $\nabla\mathbf{I}$ of the original intensity image $\mathbf{I}$ to be the vector field $\mathcal{V}$ whose inward flux through the evolving surface is to be maximized. The 3D flux calculation is illustrated in Figure 3.9. Once again we have the desirable effect that when seeds are placed within blood vessels, the sources outside boundaries prevent the flow in Eq (3.9) from leaking.

CHAPTER 4

Numerical Implementation

The numerical implementation of flux maximizing flows is based on the use of level set methods developed by Osher and Sethian [**42, 37, 43, 35**]. The main idea behind level set methodology is to embed the propagating curve or surface as the zero level set of a higher dimensional function. By evolving this function and finding its zero-crossings, one obtains the evolution of the original curve or surface. We discuss level set methods is some detail and also describe some optimization techniques that are used to speed up the simulations.

A second important consideration is the computation of the velocity term based on the divergence of the gradient vector field. We shall exploit consequences of the divergence theorem to introduce an appropriate local scale for this calculation, while also making the computation numerically more robust.

Finally, we briefly review 3D visualization techniques used to present our results.

1. Level Set Methods

This discussion draws closely from [**43**], to which the reader is referred for further discussion. Another recent review on the use of level set methods is [**35**].

Let Γ be a front evolving accordingly to the equation

$$\frac{\partial \Gamma}{\partial t} = F\mathcal{N}, \tag{4.1}$$

where F is the speed of a point on the front and $\mathcal{N}$ is the unit inward normal at that point. The level set method was devised by Osher and Sethian in [**37**] as a simple and versatile method for computing and analyzing the motion of the interface Γ in two or three dimensions, see Figure 3.2. The explicit, *Lagrangian* way to evolve the front is to compute the normals analytically and recompute the positions of every point on the interface for the next time step. This formulation faces significant difficulties when the front is about to change topology or develop singularities. Interpolating the gaps between neighboring particles as they move apart in subsequent steps is another challenge. To address these issues, Osher and Sethian [**37**] developed an *Eulerian* formulation by introducing an implicit representation of the interface. The idea is to define a smooth (Lipschitz continuous) function $\Psi(\mathbf{x}, t)$ that represents the interface $\Gamma(t)$ as the set of points $\mathbf{x}$ where $\Psi(\mathbf{x}, t) = 0$. In practice, $\Psi(\mathbf{x}, t = 0)$ is typically a signed Euclidean distance function from $\mathbf{x}$ to $\Gamma(t = 0)$, i.e.

$$\Psi(\mathbf{x}, t = 0) = \pm d \tag{4.2}$$

and the plus(minus) sign is chosen if the point is outside(inside) the initial front $\Gamma(t = 0)$, see Figure 4.1. The goal now is to derive an equation for the evolving function $\Psi(\mathbf{x}, t)$ that embeds the motion of $\Gamma(t)$ as the zero level set, i.e, $\Gamma(t)$ is given by $\Psi(\mathbf{x}, t) = 0$ for all t. Differentiating Eq. (4.2) with respect to t:

$$\Psi_t + \nabla\Psi(\mathbf{x}(t), t) \cdot \mathbf{x}_t = 0.$$

The evolution equation we want to simulate is given by $\frac{\partial \Gamma}{\partial t} = F\mathcal{N}$. Since for any level set the unit inward normal $\mathcal{N} = -\frac{\nabla\Psi}{|\nabla\Psi|}$, the front Ψ evolves according to

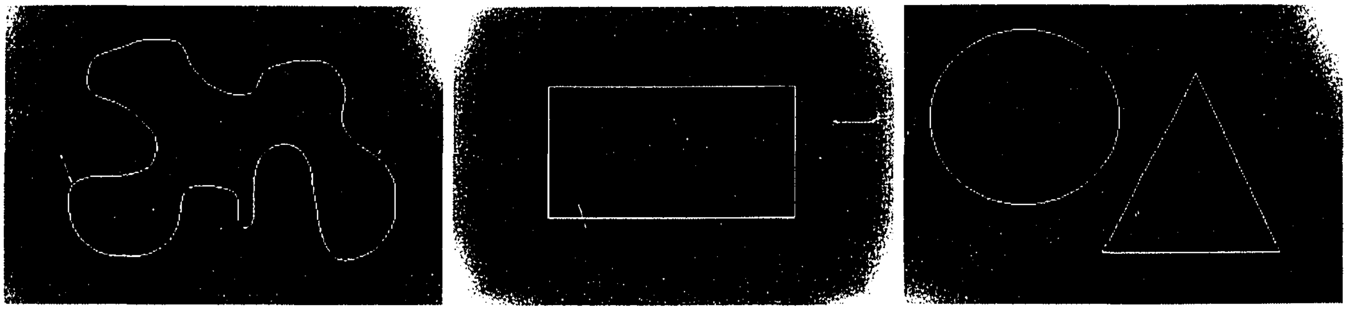


FIGURE 4.1. A Euclidean distance function as Ψ. Each point is assigned its Euclidean distance to the nearest point on the curve (shown in white). Brighter values indicate larger distances. In practice, points inside the curve are given a negative sign and those outside are given a positive sign.

$$\Psi_t = F|\nabla\Psi|. \tag{4.3}$$

Hence, if we want to simulate the motion of the interface Γ according to Eq. (4.1), we simply pick a function Ψ satisfying $\Psi(\mathbf{x}, t = 0) = \Gamma(0)$ and update it according to Eq. (4.3). Γ is then obtained as the zero level set of $\Psi(t)$. The evolving function $\Psi(\mathbf{x}, t)$ always remains a function as long as F is smooth. However, the zero level set, and hence the propagating interface $\Gamma(t)$, may change topology and form sharp corners as the function Ψ evolves. Another advantage of this implicit representation is that intrinsic geometric properties of the front are easily determined from the level set function Ψ. For example, at any point of the front, the unit outward normal vector is given by

$$\mathcal{N} = \frac{\nabla\Psi}{|\nabla\Psi|}$$

and the mean curvature of each level set at every point is the divergence of the normal:

$$\kappa = div\left(\frac{\nabla\Psi}{|\nabla\Psi|}\right).$$

For numerical simulations Eq. (4.3) is discretized. For example, on a 2D rectangular grid, at each grid point (i, j) an update rule for Ψ is:

$$\Psi_{ij}^{n+1} = \Psi_{ij}^{n} + \Delta t \cdot F|\nabla_{ij}\Psi_{ij}^{n}|.$$

This is a first-order update rule in time, where n is the iteration number and Δt is the time step. If greater accuracy is required, higher-order time integration schemes may be used. Special care must be taken to approximate $\nabla_{ij}\Psi_{ij}^{n}$ since F is a hyperbolic speed function for the flux maximizing flow. Specifically, when spatial derivatives are computed, information must propagate in the same direction as the evolving front. Numerical methods which enforce this constraint are known as *upwind* schemes [**28**]. Furthermore, if greater spatial accuracy is required, higher-order versions of these techniques may be used [**36**].

2. Optimization

The first step in implementing a level set evolution is the construction of the level function Ψ. The most popular choice for this function has been a signed Euclidean distance to the front. Each point is assigned its Euclidean distance to the closest point on the front, with interior and exterior points being given different signs. An exact distance function is a $O(mn)$ computation, where m is the number of points on the front and n is the total number of points in an input array. In 3D this can add a heavy computational burden. In practice an approximate Euclidean distance function is often used [**6**], where a mask is swept through the entire array a fixed number of times. Hence, the complexity is reduced to $O(n)$.

By constructing an embedding hypersurface Ψ, a level set method adds an extra dimension of computation to a problem. The update equation Eq. (4.3) must be applied at each grid point, in contrast to the evolution equation of the front Eq. (4.1), which is defined only for points at the interface. Furthermore, the notion of the velocity term F has to be smoothly extended to all level sets, not just the zero level set for which F is naturally defined. One way to do this is to assign to each grid point the value of F from the closest point on the zero level set. This however must

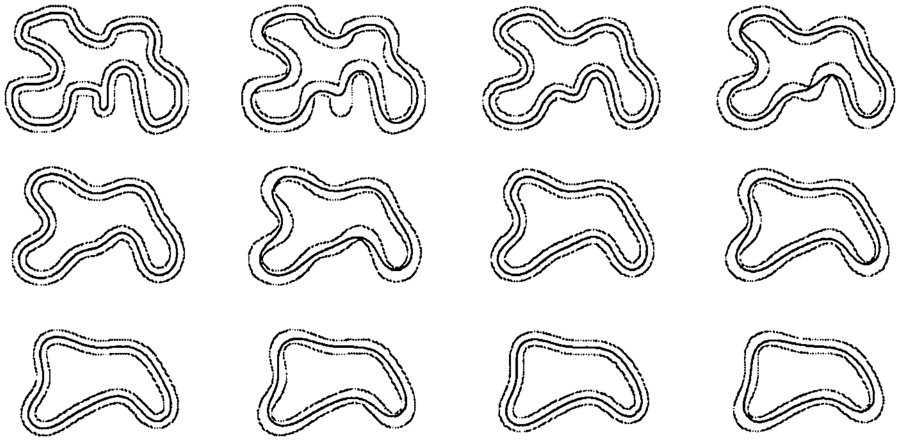


FIGURE 4.2. An illustration of the narrow band technique. All computations are performed in a small neighborhood of the front, as opposed to over the entire grid. When the front hits the boundary of the narrow band, the distance function is recomputed and a new band is placed around the front. Successive iterations are shown from left to right, top to bottom. The front is shown in black and the boundaries of the band are in gray.

be done for all points in the domain and hence the operation is computationally very expensive.

To combat these issues it was proposed [12] to perform all calculations only in a small neighborhood of the zero level set. This method became known as the *narrow band approach.* In the course of the evolution when the front approaches the boundary of the band, the calculation is stopped, Ψ is recomputed and a new band is built around the zero level set, as illustrated in Figure 4.2. In this case, the time for each iteration of the front reduces to $O(k)$ where k is the number of points in the band. The computation speed-up in using the narrow band approach depends entirely on the number of points on the front and the width of the band. Clearly if the size of the band is comparable to the size of the domain and/or if we have to perform re-initialization very often, we may end up with a loss of speed in comparison to updating the full grid. In practice, however, the narrow band implementation leads to a speed-up factor of 5 to 10.

3. Divergence and Flux

In order to tailor the flux maximizing flow to blood vessel segmentation, we consider the gradient $\nabla\mathbf{I}$ of the original intensity image $\mathbf{I}$ to be the vector field $\mathcal{V}$ whose inward flux through the evolving curve (or surface) is maximized. An important consideration in the implementation of Eq. (3.5) and Eq. (3.9) is that since the divergence of the vector field needs to be calculated, implicitly second derivatives of $\mathbf{I}$ are being used. In fact with $\mathcal{V} = \nabla\mathbf{I}$, Eq. (3.5) becomes

$$\frac{\partial\mathcal{C}}{\partial t} = div(\mathcal{V})\mathcal{N} = div(\nabla\mathbf{I})\mathcal{N} = \triangle\mathbf{I}\mathcal{N}.$$

Similar calculations could be done for Eq. (3.9). In other words, each point on the front is moved in the inward normal direction by an amount given by the Laplacian of $\mathbf{I}$. Note, however, that second derivatives $\mathbf{I}$ are not defined at places where the gradient vector field $\nabla\mathbf{I}$ becomes singular, which are precisely the locations of interest.

In order to address this issue and also incorporate a notion of scale that corresponds to the local width of a vessel we exploit a consequence of the divergence theorem. The divergence at a point is defined as the net outward flux per unit area, as the area about the point shrinks to zero. Via the divergence theorem,

$$\int_a \mathrm{div}(\mathcal{V})\mathrm{da} \equiv \int_{\mathrm{L}} < \mathcal{V}, \mathcal{N} > \mathrm{dl}. \tag{4.4}$$

Here Δa is the area, L is its bounding contour and $\mathcal{N}$ is the outward normal at each point on the contour. The formulation extends to 3D by replacing the contour integral with a surface integral.

For our numerical implementations we use this flux formulation along the boundaries of circles (in 2D) or spheres (in 3D) of varying radii, corresponding to a range of blood vessel widths. Thus, we integrate the divergence over an appropriate local neighborhood. The chosen flux value at a particular location is the maximum (magnitude) flux over the range of radii. In contrast to other multi-scale approaches where combining information across scales is non-trivial [**26**] normalization across scales is straightforward in our case. One simply has to divide by the number of entries in the discrete sum that approximates Eq. (4.4). This computation is illustrated in Figures 3.8 and 3.9.

4. Visualization

Displaying 3D information such as input images or level sets on 2D paper leaves a certain choice of options on which volume or surface rendering technique to use. We visualize 3D MRA and CRA input images of blood vessels using a maximum intensity projection (MIP) technique, Figure 1.2(a). In the MIP 3D data is projected onto a 2D plane by choosing the maximal value along the projection rays. Because intensity in angiography images is proportional to the speed of the blood, voxels containing blood vessels have higher values and consequently they are more likely to be seen on the MIP.

As an output of our segmentation algorithm we have a matrix in which zero crossings (the zero level set) corresponds to blood vessels. We use the freely available *Visualization Toolkit (VTK)* [**41**] that uses the marching cubes technique [**29**] to triangulate voxels corresponding to the zero crossings into isosurfaces. Before rendering, the surfaces are slightly smoothed to enhance their visualization, see Figure 1.2(b).

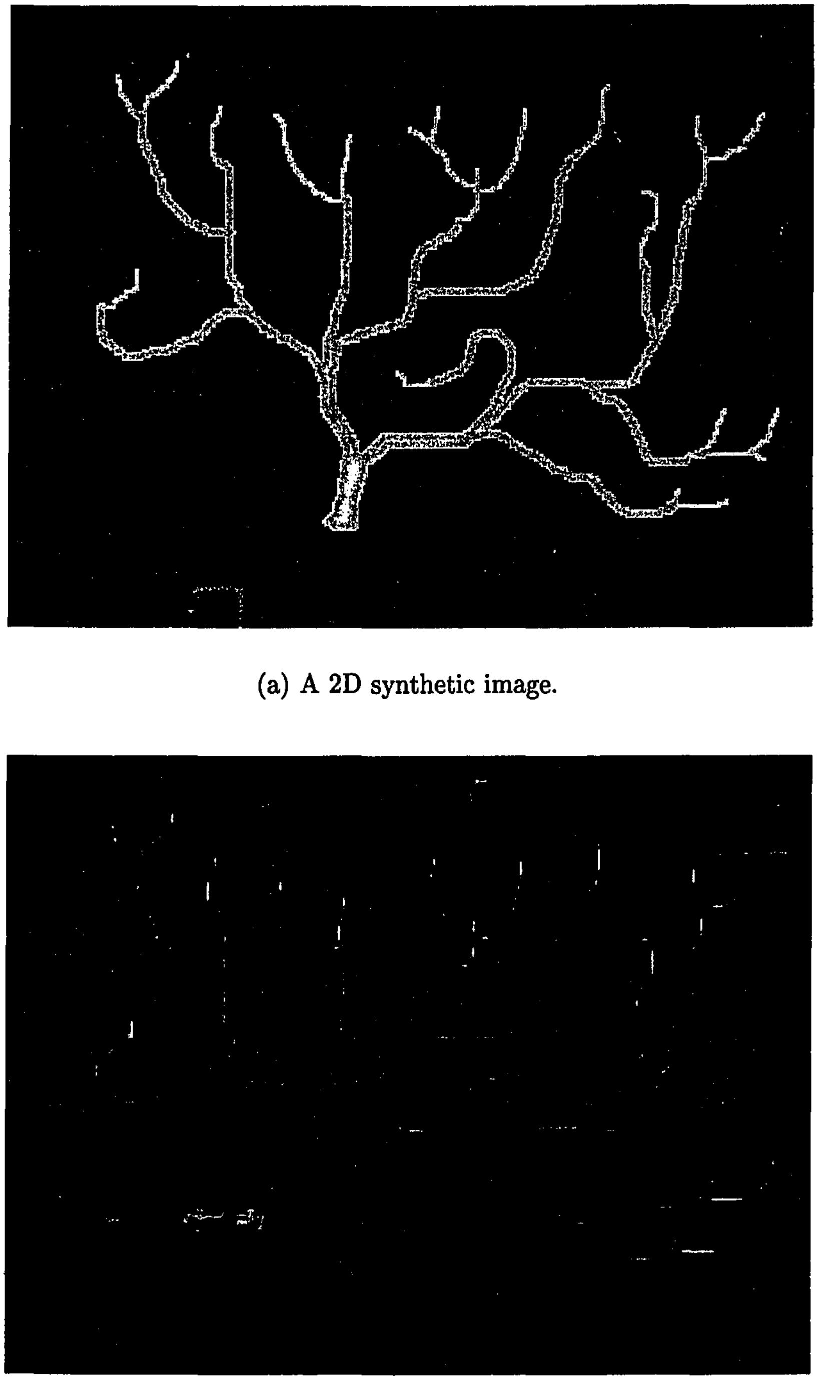
CHAPTER 5

Results

In this chapter we present examples that illustrate the application of flux maximizing flows to the segmentation of the vasculature. We demonstrate the advantages of our method on regions of a 2D angiography retinal image, on a 3D MRA image of the human head, as well as on several cropped regions of a 3D CRA image of the head. Our segmentation results are compared against results obtained by thresholding. Whereas this is only a crude form of segmentation it is nevertheless a first step used to initialize several vessel segmentation techniques. In order to better observe the properties of the flow and to illustrate the multi-scale flux calculation, we begin with a simulation on a synthetic 2D tree-like image.

1. A Synthetic Image

We start by illustrating the flux maximizing flow on a synthetic 2D image, as shown in Figure 5.1(a). This image was constructed as follows. First, a binary image was obtained by drawing an object resembling a 2D vascular tree. Next, the distance function was computed and displayed in the interior of this object. Hence, the centerlines of the branches appear brightest, and then intensity gradually decreases towards the boundaries. The idea is to mimic a real MRA image where intensity is proportional to the speed of the blood and blood flows faster in the center of a vessel.



(a) A 2D synthetic image.

(b) Its multi-scale divergence.

FIGURE 5.1. A synthetic image imitating a 2D vascular tree a), and the multi-scale outward flux (divergence) of its gradient vector field b).

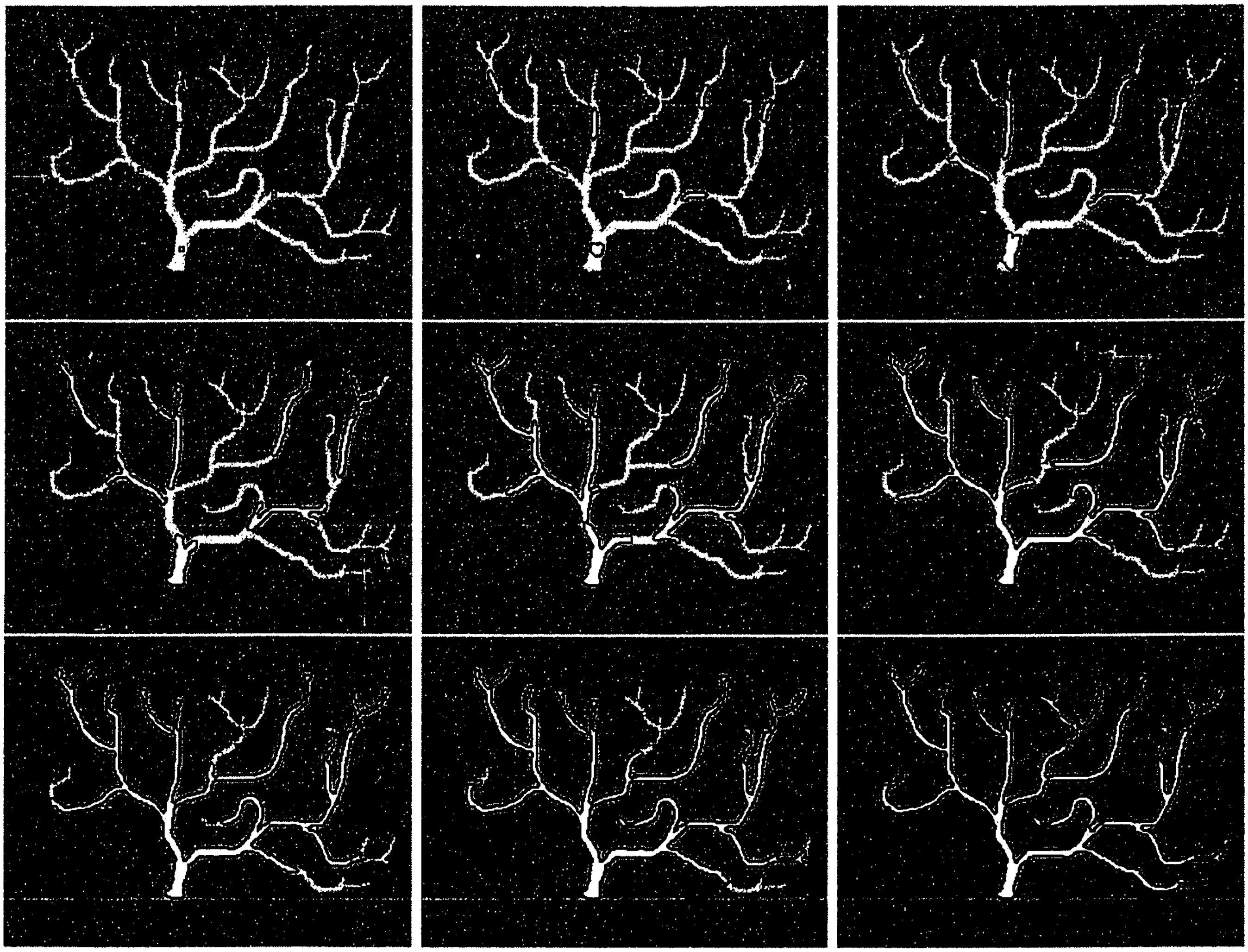
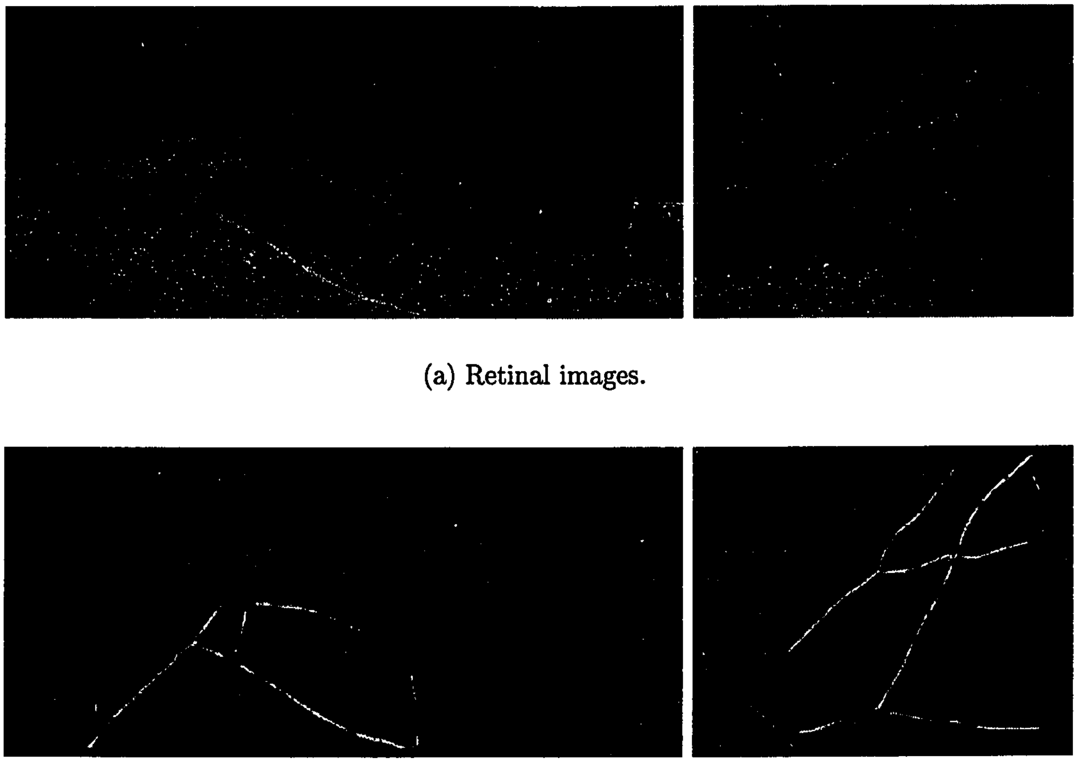


FIGURE 5.2. An illustration of the evolution of a few isolated seeds flowing according to Eq. (3.5) on the synthetic tree-like image in Figure 5.1(a). Consecutive evolution steps are shown from left to right, top to bottom.

Although it is trivial to segment this synthetic image, it serves as a clean illustration of the properties of the flux maximizing flow.

The first step in simulating the flow is the computation of the image-based velocity field that will drive the evolution. For flux maximizing flows the speed is defined by the divergence of the auxiliary vector field. For blood vessel segmentation we choose the gradient vector field of the original intensity image to be this auxiliary field. For our simulations we compute the flux along the boundaries of circles in 2D and spheres in 3D of varying radii that cover the range of expected vessel widths. The flux is then normalized by dividing it by the length of the circle in 2D or surface area of the



(a) Retinal images.

(b) Their multi-scale divergence.

FIGURE 5.3. Portions of a retinal angiography image a) and their corresponding multi-scale outward flux (divergence) b). The scale corresponds to possible vessel widths and ranges uniformly from 1 to 8 pixels.

sphere in 3D. The maximal (in magnitude) normalized flux value at each point is then assigned to this point. The multi-scale divergence computed in such a way over a range of radii from 1 to 6 pixels is shown in Figure 5.1(b). After the velocity field has been computed we can proceed with simulating the actual flow. Several seeds are placed manually inside the tree to initialize the flow, as illustrated in Figure 5.2. As expected, the curves evolve in the direction of branches thereby maximizing the flux of the gradient vector field through them. The algorithm has no restrictions at branch points and hence all branches are captured perfectly.

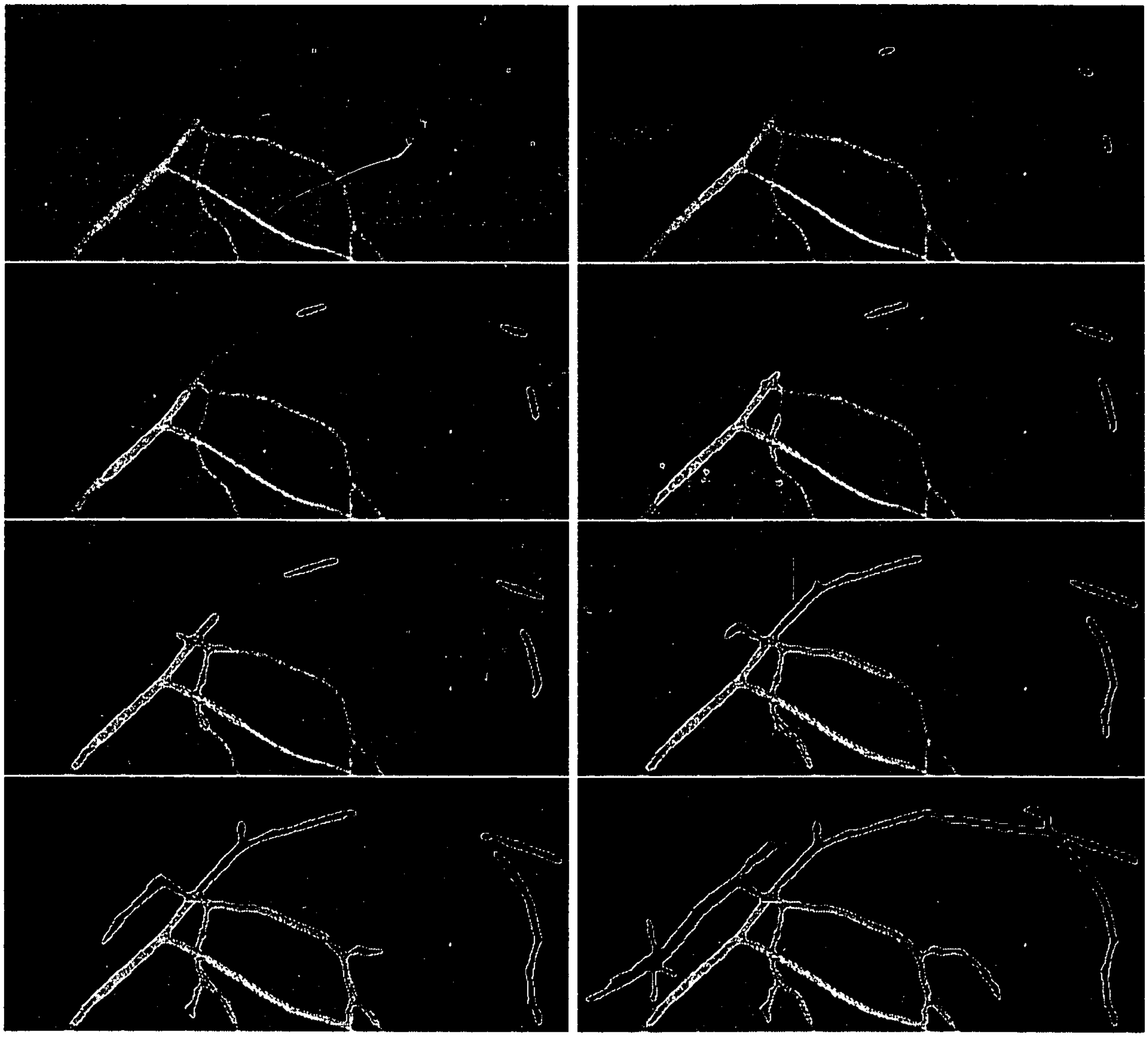
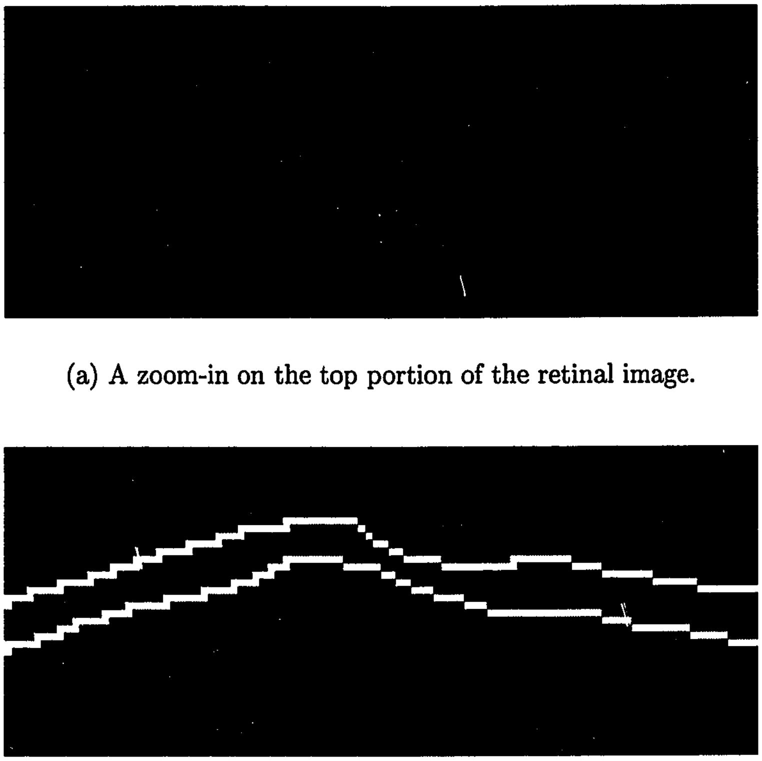


FIGURE 5.4. An illustration of the evolution of a few isolated seeds flowing according to Eq. (3.5). Notice that, as expected, they follow the direction of shading in the image to maximize the inward flux of the gradient vector field through their boundaries.

2. 2D Retinal Images

We now demonstrate the performance of our algorithm in 2D on portions of an angiography image of the retina. The inputs are greyscale images containing blood vessels with 256 intensity levels, as shown in Figure 5.3(a). The multi-scale outward flux of the gradient vector field of these images is illustrated in Figure 5.3(b). Here 8 scales uniformly distributed from 1 to 8 have been used.



(a) A zoom-in on the top portion of the retinal image.

(b) The segmentation results for this portion.

FIGURE 5.5. An illustration of the segmentation of a low contrast elongated structure. We zoom in on the top portion of the retinal image in Figure 5.4 and display the segmentation results obtained by the flux maximizing flow.

Figures 5.4 and 5.6 depict the flux maximizing flow generated by evolving several isolated seeds (Figure 5.4) or a single seed (Figure 5.6), placed inside the vessels. Although the seeds can be placed automatically in places of high inward flux (as we will show in 3D examples), these seeds have been placed manually strictly for the purpose of enhancing the demonstration of the effectiveness of the algorithm. Notice how curves are elongated in the direction of vessels to reconstruct thin or low contrast structures. The ability to recover low contrast structures is best demonstrated on the vessel located on the top portion of the image from Figure 5.4. A zoomed-in version of this vessel is shown in Figure 5.5(a), with the segmentation shown in Figure 5.5(b). Most other flows, particularly ones with a constant inflation term, would leak through such boundaries. The introduction of a curvature-based regularization term may

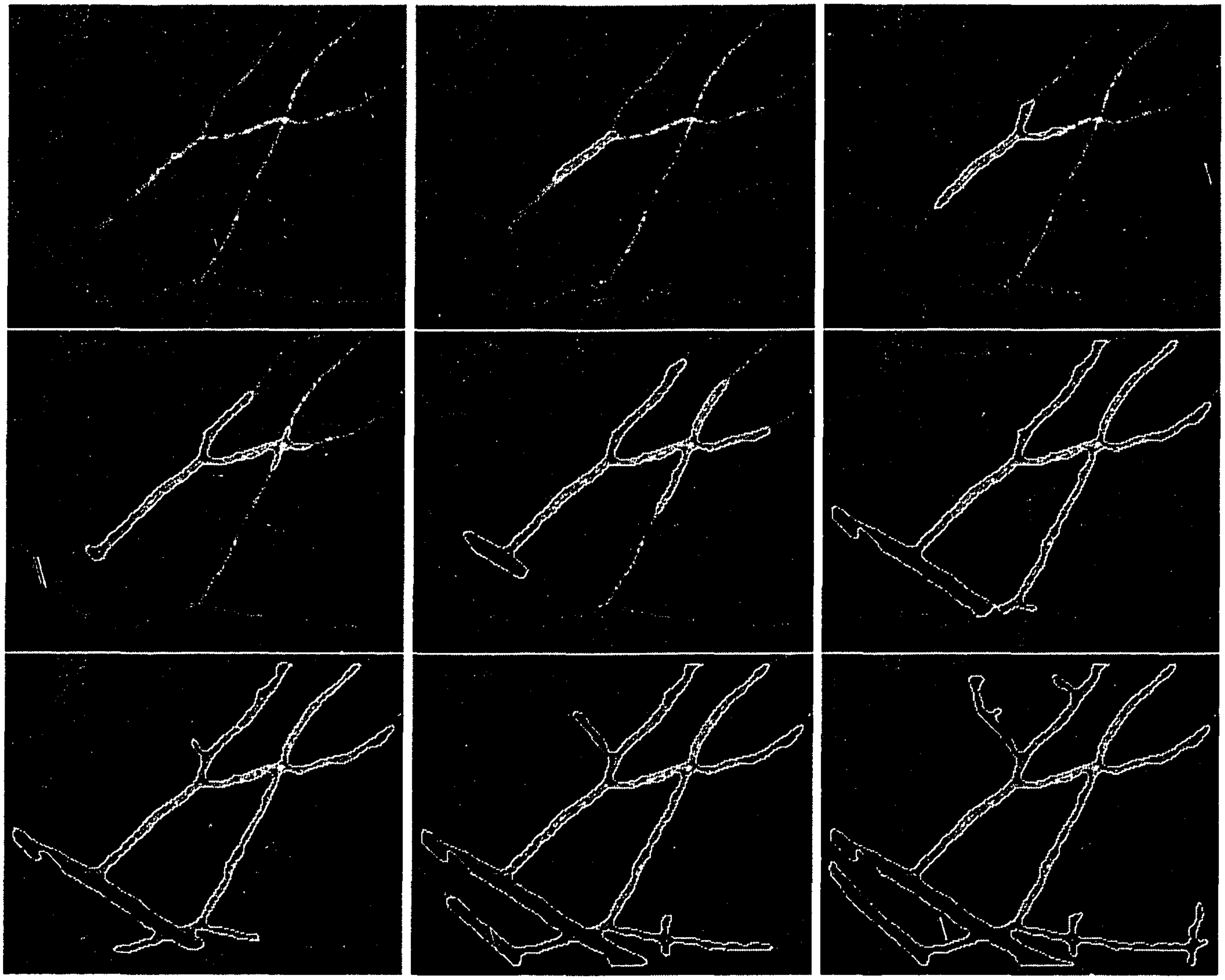
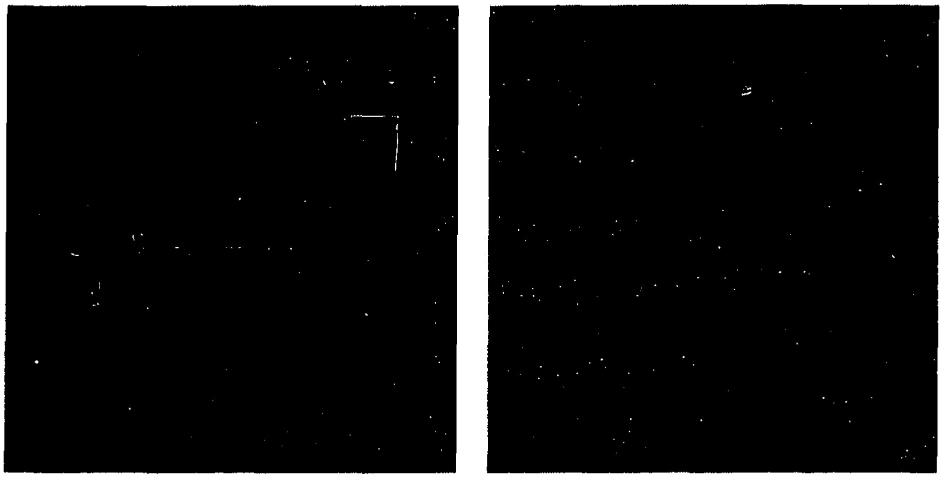


FIGURE 5.6. An illustration of a curve evolving according to Eq. (3.5). Notice that vessels of different width are well captured.

prevent leaking to an extent, but the flow would then be halted at narrow regions as well, since the curvature would dominate and would push the evolving curve back. Notice also that vessels of different radii are well captured due to the multi-scale computation of the flux.

3. A 3D Magnetic Resonance Angiography Image

The proposed blood vessel segmentation algorithm has been tested on 3D Magnetic Resonance Angiography (MRA) data. Figure 5.7 shows Maximum Intensity



(a) Maximum intensity projection (first view).

(b) Maximum intensity projection (second view).

FIGURE 5.7. Maximum intensity projections of a 60x60x60 portion of a 3D MRA image of blood vessels. An MIP of the full 256x256x100 dataset is illustrate in Figure 5.9.

Projections (MIPs) of a portion of an MRA image of blood vessels in the head, seen from two different viewing directions. This portion has dimensions 60x60x60 and has been cropped from the full MRA data of 256x256x100 voxels, whose MIP is shown in Figure 5.9. Figures 5.8 (for the cropped data) and 5.10 (for the full data) illustrate the evolution of a few isolated 3D spheres flowing according to the 3D flux maximizing equation flow (Eq. 3.9). These spheres were placed automatically by an initialization program which sampled the data and placed seeds uniformly in regions of high flux, which is similar to the idea of using "bubbles" [47]. Notice how the spheres elongate in the direction of blood vessels, which is once again the evolution we expect since it maximizes the rate of increase of inward flux through them. It is interesting to note that the main thick vessels are reconstructed first, since they have higher inward flux, followed by the reconstruction of thinner low contrast vessels. Such a hierarchical reconstruction can later be exploited for the classification of vessels by size.

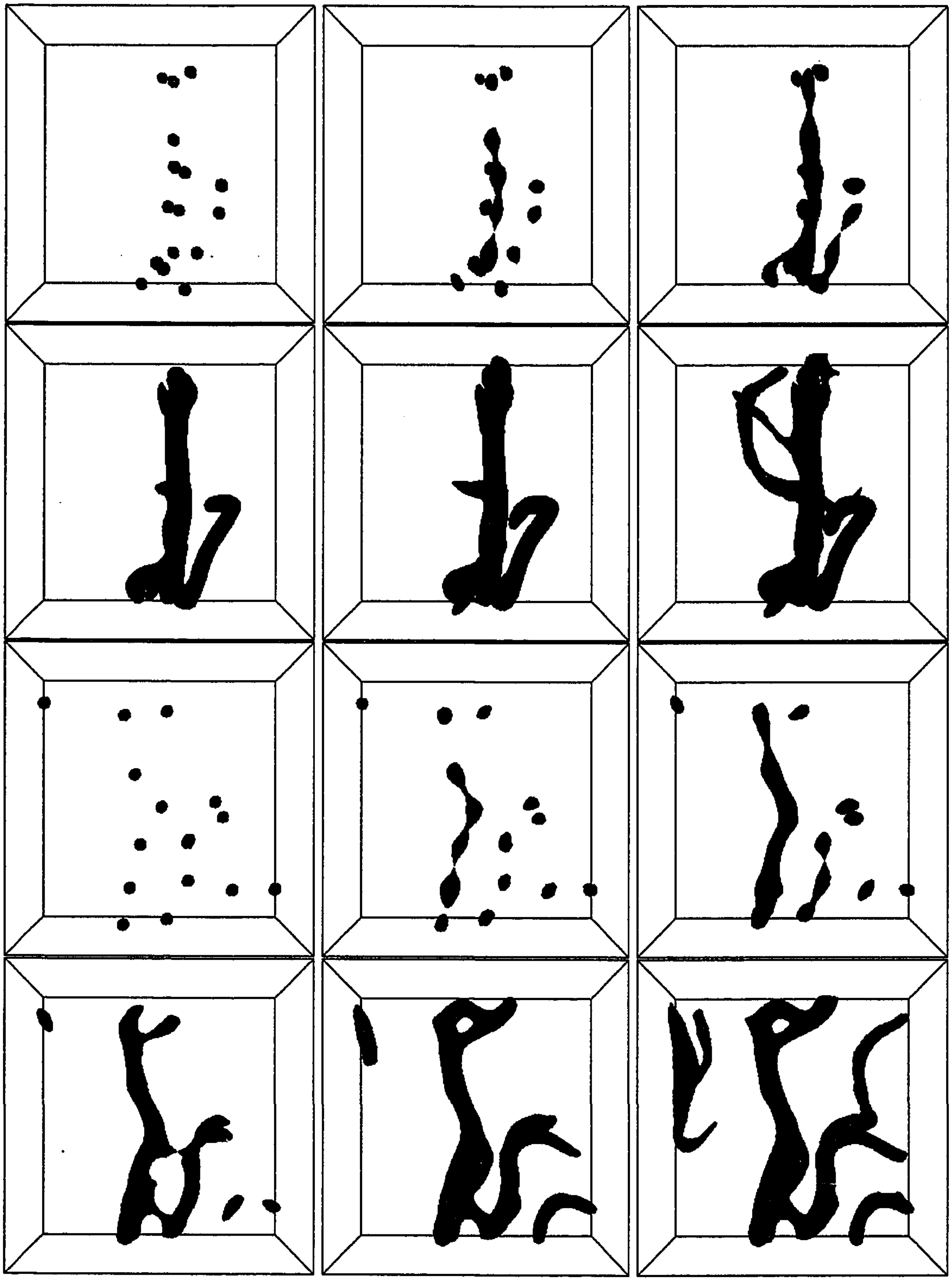


FIGURE 5.8. An illustration of the flux maximizing flow for a portion of a 3D MRA image of blood vessels in the head. Two distinct views of the same evolution are shown (top two rows and bottom two rows). The full reconstruction is shown in Figure 5.10.

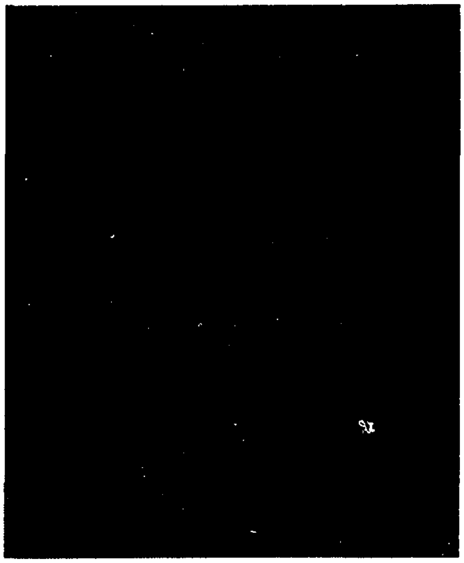


FIGURE 5.9. A maximum intensity projection of the 256x256x100 3D MRA image.

In Figure 5.11 we compare the segmentation results obtained with the flux maximizing flow with simple thresholding. High (conservative) thresholding, shown in Figure 5.11(a), does not detect thin or low contrast vessels. As we decrease the threshold, allowing more structures to be detected, most low contrast thin vessels are captured, but also many voxels corresponding to noise are incorrectly labeled as vessels (Figure 5.11(b)). Also, notice the effect of merging neighboring vessels as the threshold is lowered. In general it is unclear how to recover from poor results due to thresholding, despite this being a common step in many algorithms [**31**, **33**, **32**, **49**, **40**].

4. A 3D Computed Rotational Angiography Image

Computed Rotational Angiography (CRA) volumetric images are produced by acquiring projection x-ray images from many angles around the patient, followed by reconstruction using the Computed Tomography (CT) algorithm. For blood vessels to be seen, x-ray contrast material is injected directly into the vessel system before scanning.

The proposed algorithm based on flux maximizing flows has been tested for the segmentation of blood vessels in a 360 × 420 × 310 CRA image of the head. We

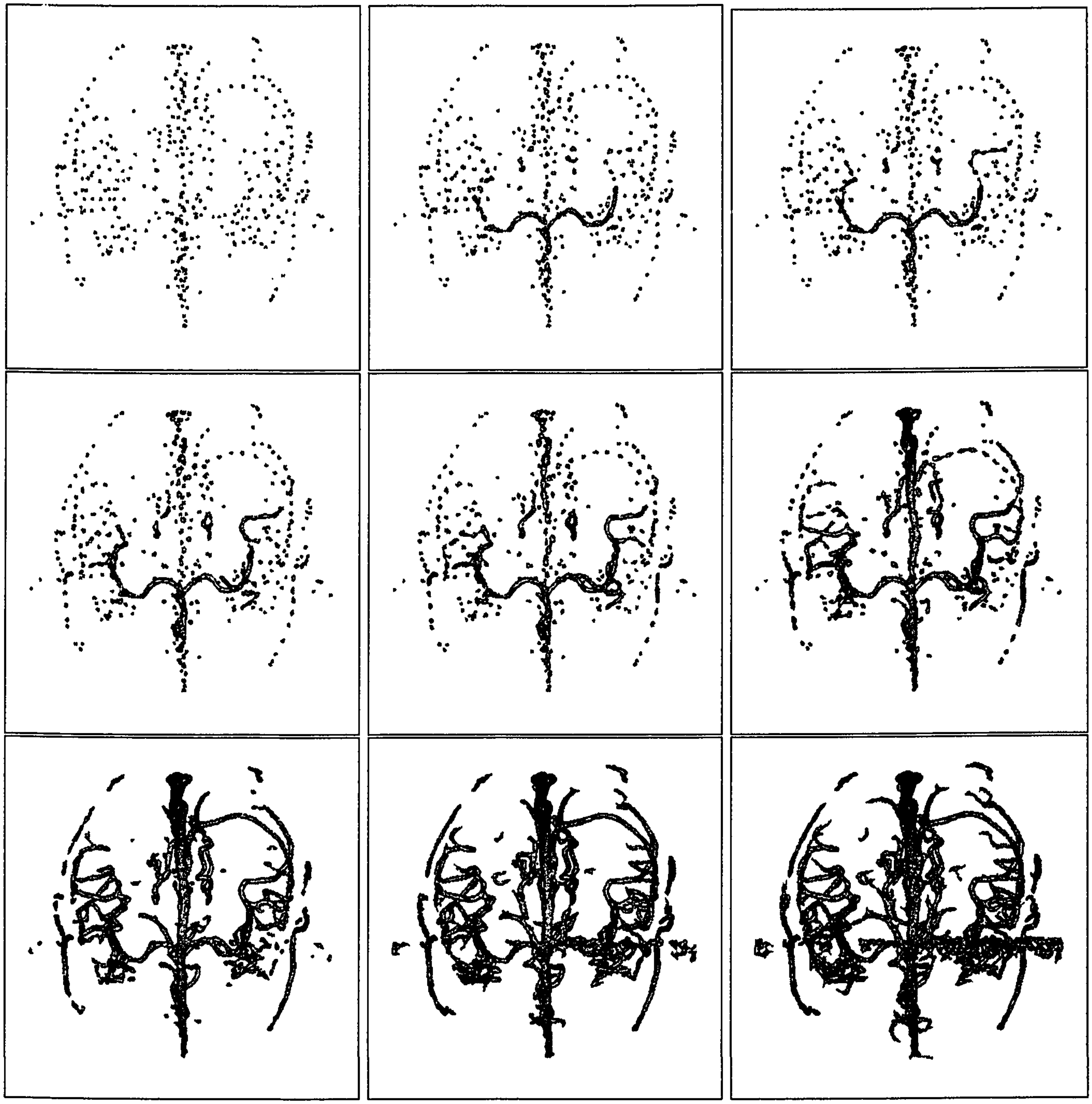
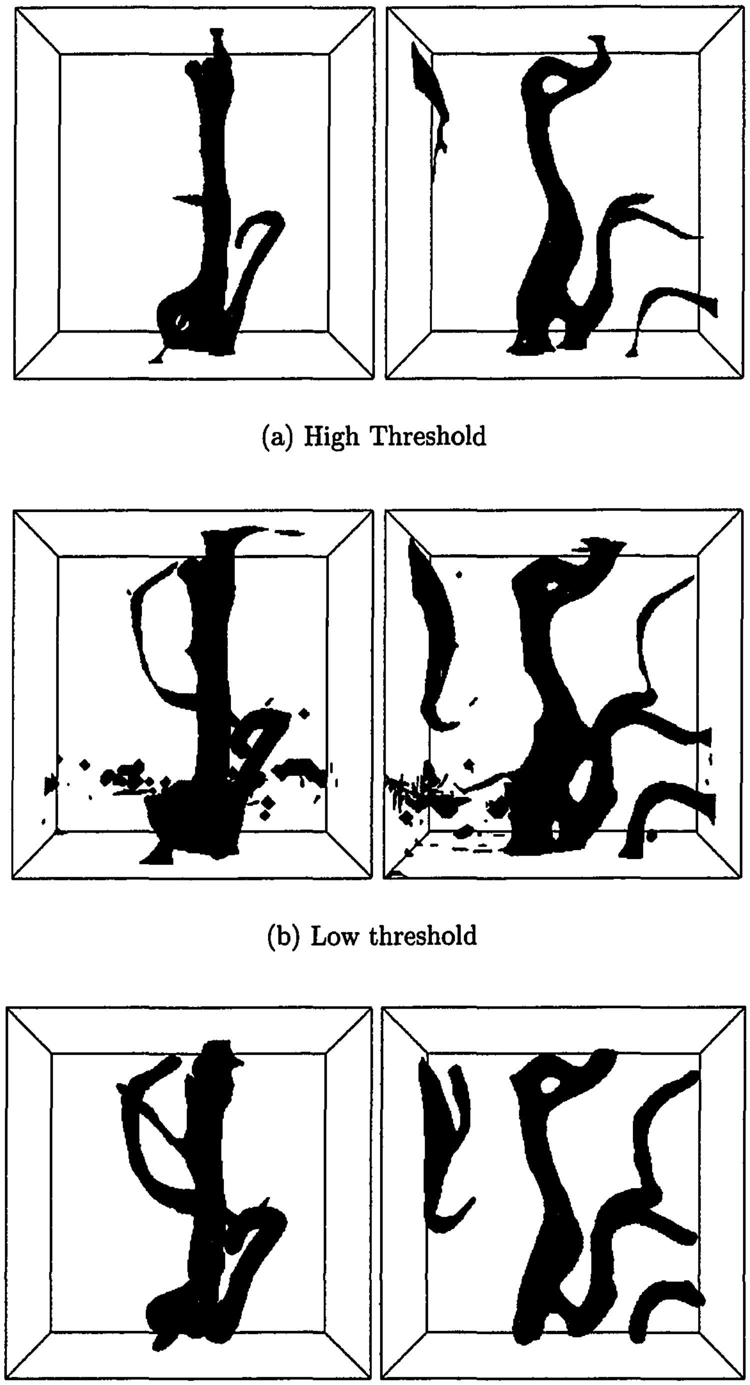


FIGURE 5.10. An illustration of the flux maximizing flow for the full 3D MRA image, of which a portion was shown in Figure 5.8. Images from left to right, top to bottom depict the successive evolution steps of a few isolated blobs. The main vessels, which have higher inward flux, are the first to be reconstructed.

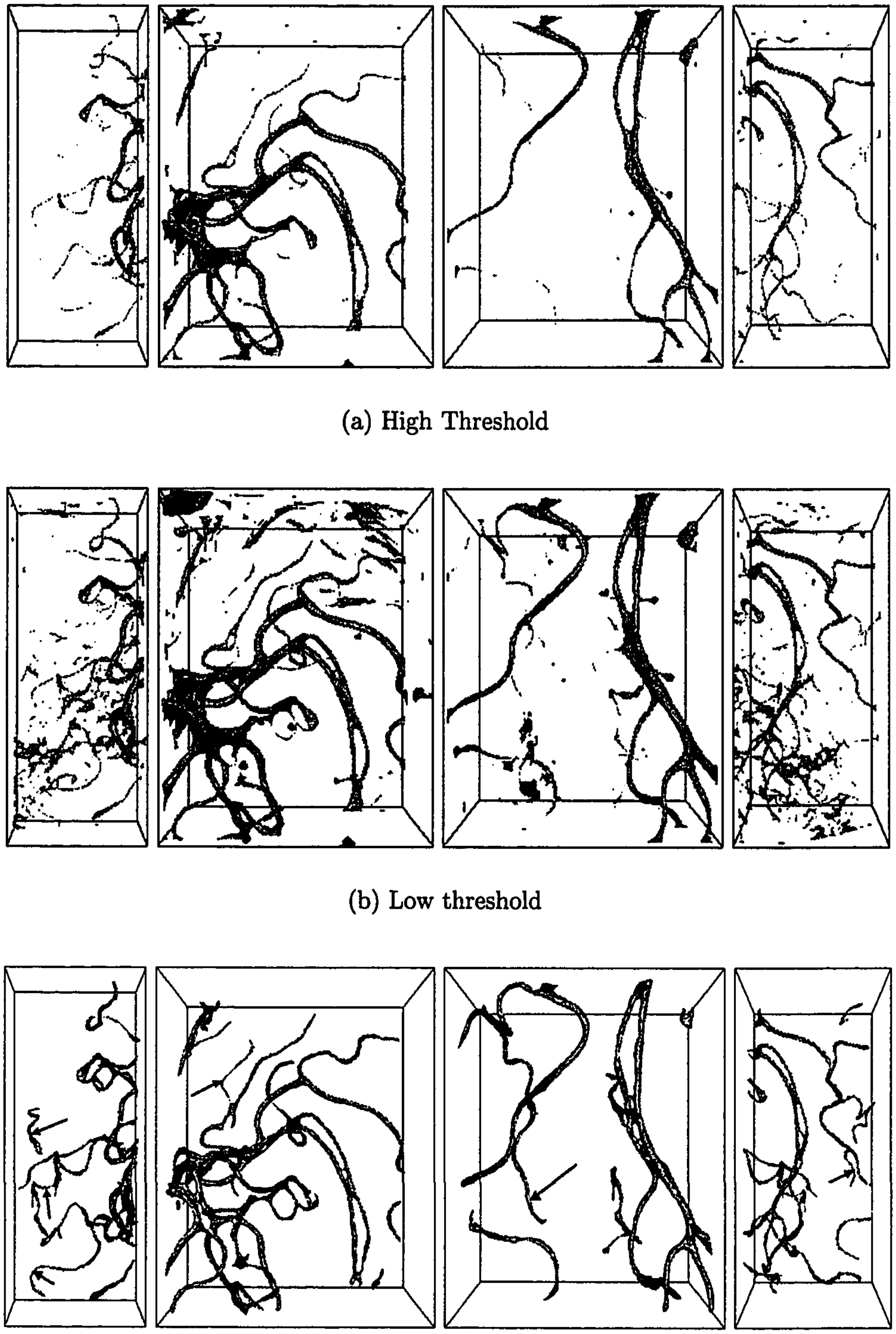


(a) High Threshold

(b) Low threshold

(c) Flux maximizing flows

FIGURE 5.11. A comparison of segmentation results obtained with flux maximizing flows c) with simple thresholding a) and b) on a portion of an MRA image shown from two different viewing directions. A conservative (high) threshold a) does not detect low contrast vessels, whereas a lower threshold b) incorrectly labels noise as vessels.



(a) High Threshold

(b) Low threshold

(c) Flux maximizing flows

FIGURE 5.12. A comparison of segmentation results obtained with flux maximizing flows c) with simple thresholding a) and b), on different regions of a CRA image. Arrows in c) point to some of the thin low intensity vessels that are not seen even in the low threshold image b), that already contains several noise artifacts.

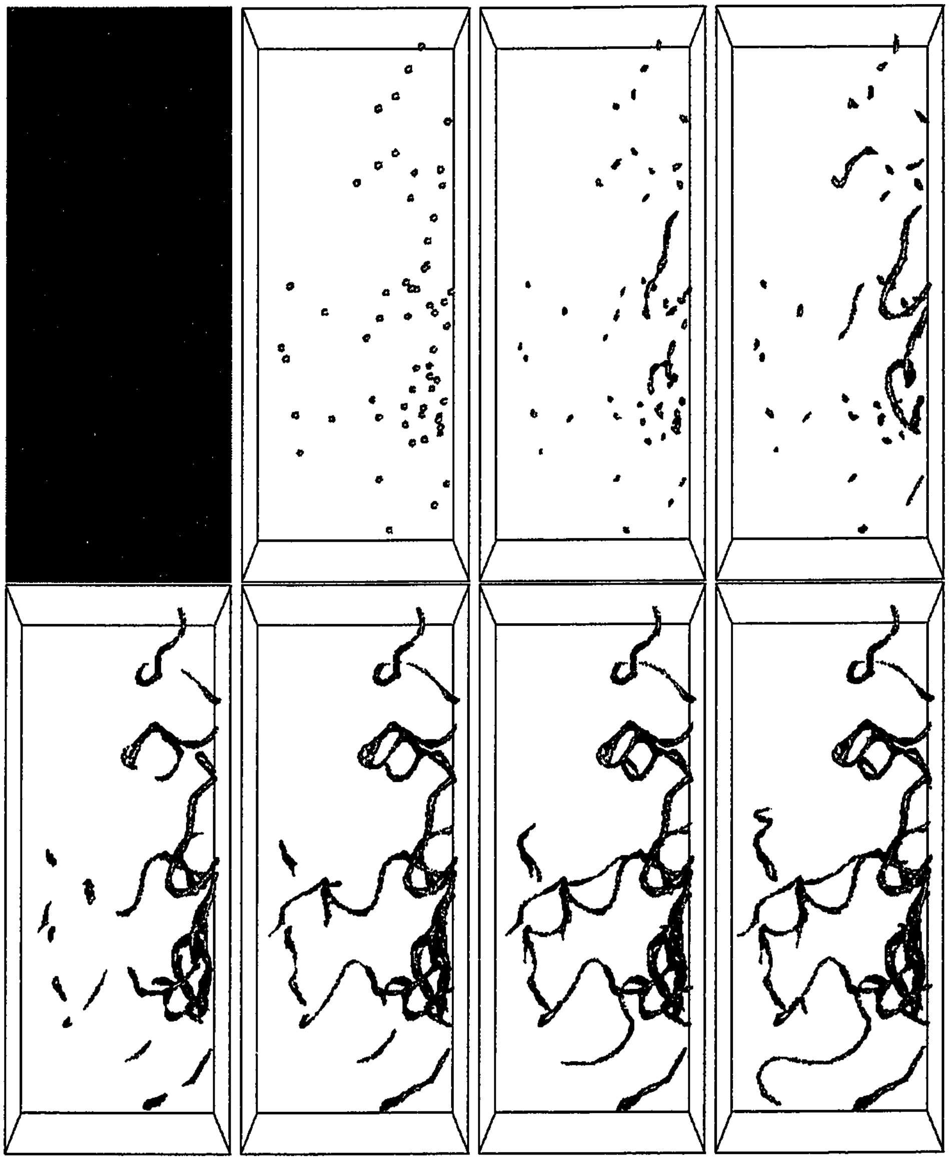


FIGURE 5.13. An illustration of the flux maximizing flow for a portion of a 360 × 420 × 310 3D CRA image of blood vessels in the head. A maximum-intensity projection of the region being viewed is shown on the top left. The other images depict the evolution of a few isolated spheres. Notice how the evolution follows the direction of blood flow to reconstruct the blood vessel boundaries.

illustrate the effectiveness of the method on different regions of this CRA image. These regions were selected to contain many vessels of different sizes and intensities in order to be able to better evaluate the performance of the proposed flow. Segmentations obtained by evolving several 3D spheres (placed automatically in areas

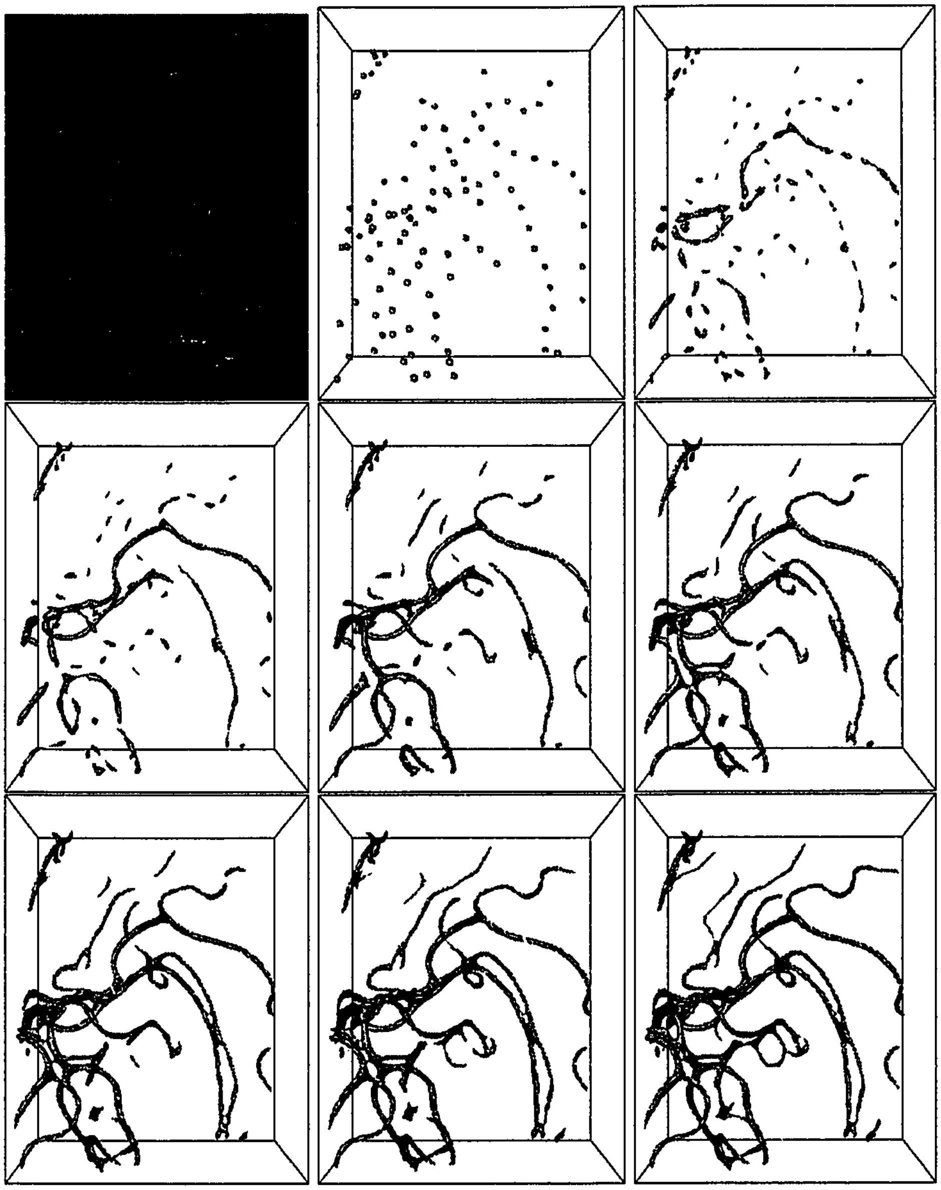


FIGURE 5.14. An illustration of the flux maximizing flow for a portion of a 360 × 420 × 310 3D CRA image of blood vessels in the head. A maximum-intensity projection of the region being viewed is shown on the top left. The other images depict the evolution of a few isolated spheres. Notice how the evolution follows the direction of blood flow to reconstruct the blood vessel boundaries.

of high inward flux) according to the 3D flux maximizing (Eq. 3.9) are shown in Figures 5.13, 5.14, 5.15 and 5.16. For each region, a maximum intensity projection is shown on the top left corner of the corresponding figure. The dimensions of the data are 79 × 161 × 66 voxels for the region in Figure 5.13, 103 × 122 × 97 voxels

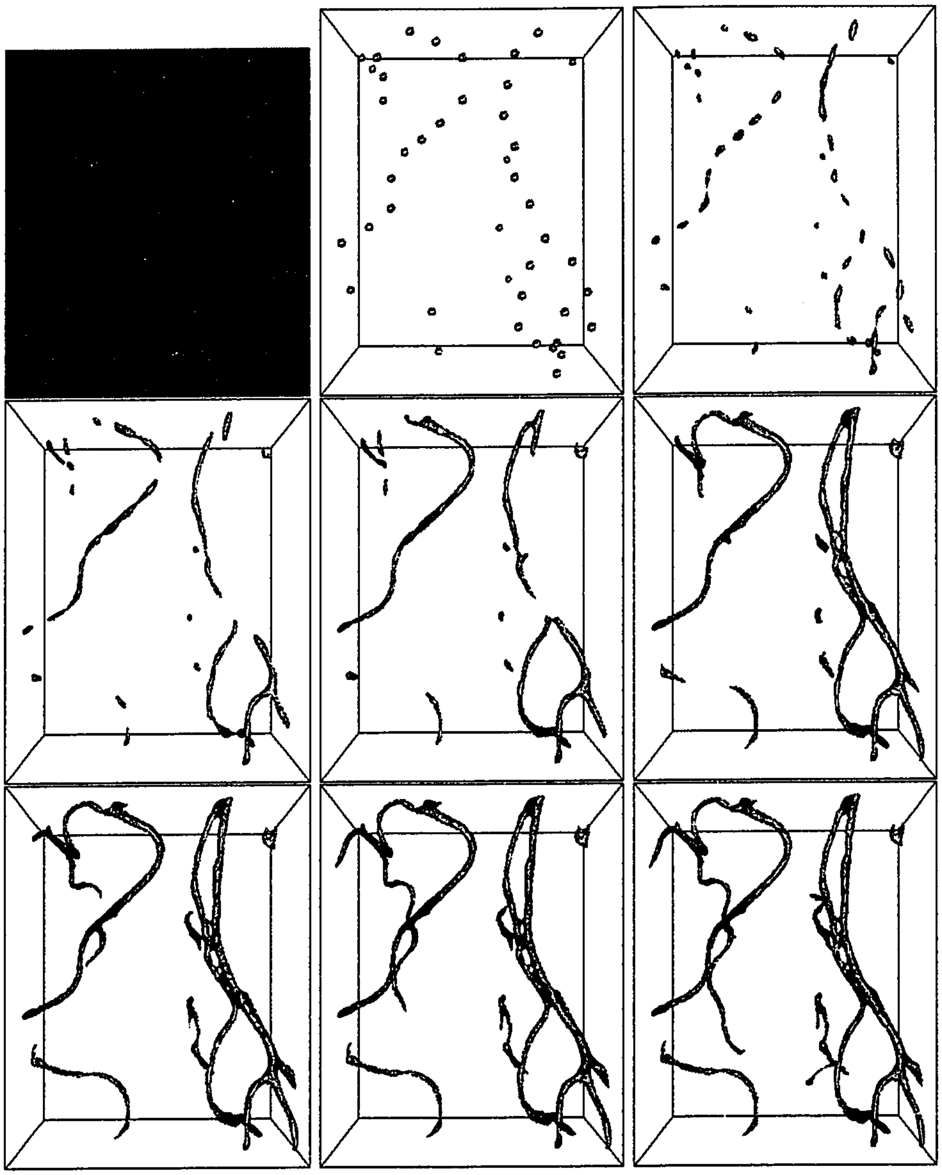


FIGURE 5.15. An illustration of the flux maximizing flow for a portion of a 360 × 420 × 310 3D CRA image of blood vessels in the head. A maximum-intensity projection of the region being viewed is shown on the top left. The other images depict the evolution of a few isolated spheres. Notice how the evolution follows the direction of blood flow to reconstruct the blood vessel boundaries.

for Figure 5.14, 103 × 98 × 79 voxels for Figure 5.15 and 161 × 173 × 79 voxels for Figure 5.16.

Figure 5.12 compares segmentation results obtained by our method with segmentation by thresholding. Notice that as with the MRA image (Figure 5.11(b)) a high

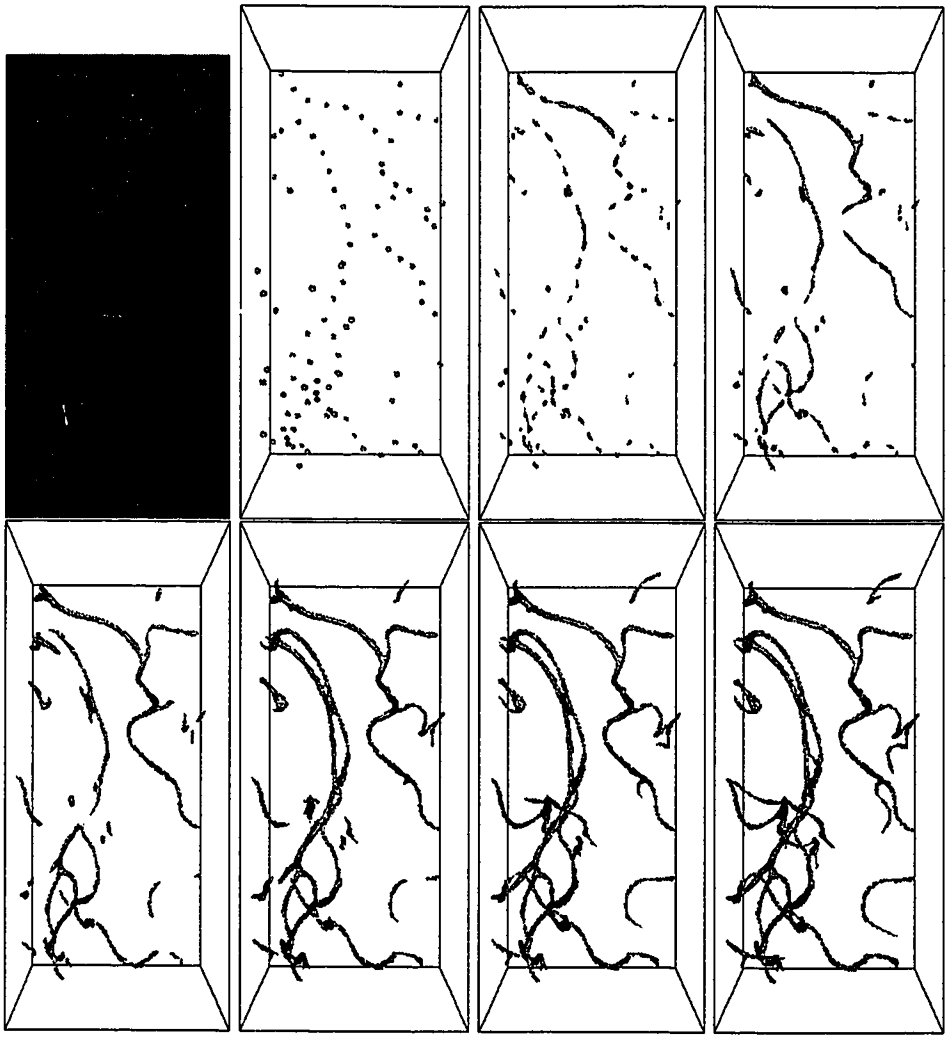


FIGURE 5.16. An illustration of the flux maximizing flow for a portion of a $360 \times 420 \times 310$ 3D CRA image of blood vessels in the head. A maximum-intensity projection of the region being viewed is shown on the top left. The other images depict the evolution of a few isolated spheres. Notice how the evolution follows the direction of blood flow to reconstruct the blood vessel boundaries.

(conservative) threshold does not capture thin low contrast vessels as illustrated in Figure 5.12(a). Also, with a high threshold, many thin vessels appear disconnected or "broken" into several pieces. Attempts to lower the threshold value do not produce good quality results, as demonstrated in Figure 5.12(b). One might think that because the resolution of the CT angiography image is higher than that of the MRA image ($360 \times 420 \times 310$ for CRA compared to $256 \times 256 \times 100$ for MRA), segmentation by thresholding will give superior results for the CRA image. However, this is in general not true because voxels corresponding to vessels in a CRA image exhibit a larger range of values than those in an MRA image. This range also overlaps the intensity values of anatomical structures other than vessels. Hence, attempts to cover this range with a global threshold operation will result in the capture of many more voxels than those corresponding to blood vessels. In contrast, the final segmentation results obtained with our segmentation method are very encouraging (see Figure 5.12(c)). Vessels of variable sizes and intensities are well detected. Moreover, extremely thin vessels that are not seen even in a low threshold image are captured. Several such vessels are pointed to by the black arrows in Figure 5.12(c).

In the thresholded images, in some places of high vessel density (such as the mid-bottom left portion of the second image to the left in Figure 5.12(b)) we observe that several distinct vessels are merged. This unfortunate effect is greatly reduced in the images segmented using flux maximizing flows (see the same portion in the second image to the left of Figure 5.12(c)). Nevertheless, vessels may merge when segmented by our algorithm. For example, the two long vessels in the second image to the left in Figure 5.12(c) come together in the mid right section of the image. This is typically the cause of partial volume effects during the scan; the resolution of the original CRA data set is not high enough to separate vessels when they are very close to one another.

Notice also, that since the evolution has the intuitive behavior that it follows the direction of blood flow, the final segmentation gives vessels that appear well connected.

CHAPTER 6

Discussion and Conclusions

1. Overview

In this thesis we have studied the problem of automatic segmentation of blood vessels in 2D and 3D angiography images. A solution to this problem will find applications in a number of clinical procedures including the diagnosis of several diseases and surgical planning and navigation during surgery. Several systems have been proposed by researchers to address this problem. These include multi-scale methods that try to detect line-like structures by analyzing the eigenvectors and associated eigenvalues of the Hessian matrix constructed at each voxel of the image [**26, 25, 16, 23, 30, 40**], an algorithm that detects vessels based on the assumption that their centerlines appear as ridges in intensity images [**3**], a statistical method [**49**], an anisotropic diffusion approach which filters vessels by smoothing data only along vessels [**24**], a topology-based region growing technique [**13**], and a co-dimension two mean curvature flow based approach [**31, 33, 32**]. We choose to follow the framework of co-dimension one geometric flows to segment blood vessels. While many methods developed within this framework have been applied to the segmentation problem [**8, 9, 34, 38, 45**], most will fail to detect low contrast thin structures such as blood vessels. To address this problem we derived the gradient flow which maximizes the rate of increase of inward flux of an auxiliary vector field through a closed curve or surface. The calculation lead to a simple and elegant interpretation in 2D and 3D that was essentially parameter free. We demonstrated the potential of the derived *flux maximizing flows* on several 2D and 3D images of blood vessels. All the geometric flows simulated in this thesis were implemented using the level set framework [**37**, **43**, **35**].

2. Summary of the Algorithm

The proposed segmentation algorithm proceeds in several stages:

- Input a 2D or 3D intensity image **I**, such as an MRA or CT image containing blood vessels.
- Compute the gradient vector field $\mathcal{V} = \nabla \mathbf{I}$.
- Compute the multi-scale divergence image of $\mathcal{V}$. Here, for stable numerical computation we exploit the divergence theorem rather than use the standard definition of the divergence.
- Place initial circles (in 2D) or spheres (in 3D) in the places of high inward flux, i.e., places where the divergence has low negative values.
- Use the level set framework to evolve initial curves (surfaces) according to the equation that moves the curves (surfaces) in such a way so that the rate of increase of inward flux of the gradient vector field through these curves/surfaces is maximal. We have proved that this equation has the same form in 2D and 3D:

$$M_t = div(\mathcal{V})\mathcal{N} \tag{6.1}$$

  where M is the evolving closed curve (2D) or closed surface in (3D), and $\mathcal{V}$ is the gradient vector field.
- Visualize the segmentation results.

3. Future Work

There are several ways to further extend work on blood vessel segmentation using flux maximizing geometric flows. The main idea of flux maximizing flows is to maximize the rate of increase of flux of an auxiliary vector field through a curve or surface. In order to apply this idea to blood vessel segmentation we have used the gradient vector field as this auxiliary field, which works well in practice. One other candidate for the vector field that could be a construction based on the eigenvectors of the Hessian matrix. Several researchers [**25, 16, 23, 30, 40**] have noted that for a tubular structure, the eigenvector corresponding to the smallest eigenvalue points in the direction of the tube, while the other two lie in the plane perpendicular to it. This and other (2.1) properties of the eigenvectors and eigenvalues of the Hessian have been used to construct heuristics for estimating the "probability" of each voxel to belong to a vessel. For flux maximizing flows one can simply construct a vector field of eigenvectors that are perpendicular to the main axes of a vessel. In this way, the flow that maximizes the rate of increase of inward flux of this field should follow the local direction of a vessel.

We chose not to introduce a regularization term in the variational formulation. Whether this can be incorporated in the derivation remains to be investigated. The flow could also be smoothed after it has converged using the anisotropic diffusion approach of [**24**], or the geometric heat equation or its affine invariant version in 2D. It would also be interesting to see whether the regularization technique based on the co-dimension two flow in 3D [**31, 33, 32**] could be incorporated in the flow.

More work could be done to validate the approach against ground truth or other blood vessel segmentation results [**32, 25, 49, 3**]. One difficulty with validation is that it is unclear what to consider as the ground truth because the actual blood vessels are ultimately hidden and cannot be measured directly. One way to avoid this problem is to create an artificial binary vascular tree and simulate the corresponding MRA data using an MRI simulator [**27**]. In this fashion, the original binary data could be considered as the ground truth.

Finally, the multi-scale divergence computation could easily be incorporated in the morphological reconstruction of vessels from [**13**]. Instead of ordering "removable“ voxels by their intensity one could order them by their divergence values. This will potentially lead to a more complete reconstruction of the vascular system. A similar ordering idea was used in [7] for creating medial surfaces.

4. Conclusion

To conclude, we have presented the novel idea of maximizing the flux of a vector field through a closed curve or surface and have derived the corresponding gradient flows from first principles.

We have also proposed a real-life application for these flows, the segmentation of blood vessels. We have investigated the performance of our segmentation algorithm on several 2D and 3D angiography images. The results demonstrate the potential of our algorithm for recovering vessels of various widths and sizes, including thin low contrast structures.

REFERENCES

[1] L. Alvarez, F. Guichard, P. L. Lions, and J. M. Morel, *Axioms and fundamental equations of image processing*, Archive of Rational Mechanics and Analysis **123** (1993).

[2] Luigi Ambrosio and Halile Mete Soner, *Level set approach to mean curvature flow in arbitrary codimension*, Journal of Differential Geometry **43** (1996), 693–737.

[3] A. Aylward, S.M. Pizer, and E. Bullittand D. Eberly, *Intensity ridge and width for 3d object segmentation and description*, IEEE Proc. Workshop Mathematical Models Biomedical Image Analysis, 1996, pp. 131–138.

[4] G. Bertrand and G. Malandain, *A new characterization of three-dimensional simple points*, Pattern Recognition Letters (1994), 1003–1011.

[5] H. Blum, *Biological shape and visual science*, Journal of Theoretical Biology **38** (1973), 205–287.

[6] G. Borgefors, *Distance transformations in arbitrary dimensions*, CVGIP, 1984, pp. 321–345.

[7] S. Bouix and K. Siddiqi, *Divergence-based medial surfaces*, European Conference on Computer Vision, June 2000, pp. 119–124.

[8] V. Caselles, F. Catte, T. Coll, and F. Dibos, *A geometric model for active contours in image processing*, Numerische Mathematik **66** (1993), 1–31.

[9] Vicent Caselles, Ron Kimmel, and Guillermo Sapiro, *Geodesic active contours*, Fifth International Conference on Computer Vision, IEEE Computer Society Press, 1995, pp. 694–699.

[10] Tony Chan and Luminita Vese, *An efficient variational multiphase motion for the mumford-shah segmentation model*, Asilomar Conference on Signals and Systems, October 2000.

[11] Y.G. Chen, Y. Giga, and S. Goto, *Uniqueness and existence of viscosity solutions of generalized mean curvature flow equations*, Journal of Differential Geometry **33** (1991), 749–786.

[12] D.L. Chopp, *Computing minimal surfaces via level set curvature flow*, Journal of Computational Physics **106** (1993), 77–91.

[13] P. Dokládal, C. Lohou, L. Perroton, and G. Bertrand, *Liver blood vessel extraction by 3-d topological approach*, Medical Image Conference and Computer Assisted Interventions (MICCAI), Springer-Verlag, 1999, pp. 98–105.

[14] L.C. Evans and J. Spruck, *Motion of level sets by mean curvature*, Journal of Differential Geometry **33** (1991), 635–681.

[15] Tracy L. Faber and Ernest M. Stokely, *Orientation of 3-d structures in medical images*, IEEE transactions on Pattern Recognition and Machine Intelligence, vol. 10, 1988, pp. 626–633.

[16] A. Frangi, W. Niessen, K. L. Vincken, and M. A. Viergever, *Multiscale vessel enhancement filtering*, MICCAI'98, 1998, pp. 130–137.

[17] M. Gage and R.S. Hamilton, *The heat equation shrinking convex plane curves*, Journal of Differential Geometry **23** (1986), 69–96.

[18] M. Grayson, *The heat equation shrinks embedded plane curves to round points*, Journal of Differential Geometry **26** (1987), 285–314.

[19] M. Kass, A. Witkin, and D. Terzopoulos, *Snakes: Active contour models*, International Journal of Computer Vision **1** (1987), 321–331.

[20] S. Kichenassamy, A. Kumar, P. Olver, A. Tannenbaum, and A. Yezzi, *Conformal curvature flows: From phase transitions to active vision*, Archives of Rational Mechanics and Analysis **134** (1996), 275–301.

[21] B. B. Kimia, A. Tannenbaum, and S. W. Zucker, *Toward a computational theory of shape: An overview*, Lecture Notes in Computer Science **427** (1990), 402–407.

[22] Benjamin B. Kimia and Kaleem Siddiqi, *Geometric heat equation and nonlinear diffusion of shapes and images*, Computer Vision and Image Understanding **64** (1996), no. 3, 305–322.

[23] Th. M. Koller, G. Gerig, G. Székely, and D. Dettwiler, *Multiscale detection of curvilinear structures in 2-d and 3-d image data*, ICCV'95, 1995, pp. 864–869.

[24] K. Krissian, G. Malandain, and N. Ayache, *Directional anisotropic diffusion applied to segmentation of vessels in 3d images*, International Conference On Scale Space Theories in Computer Vision, 1997, pp. 345–348.

[25] ———, *Model based detection of tubular structures in 3d images*, Tech. Report 3736, INRIA, Sophia Antinopolis, France, 1999.

[26] K. Krissian, G. Malandain, N. Ayache, R. Vaillant, and Y. Trousset, *Model-based multiscale detection of 3d vessels*, CVPR'98, 1998, pp. 722–727.

[27] R. K.-S. Kwan, A.C. Evans, and G.B. Pike, *An extensible mri simulator for post-processing evaluation*, Visualization in Biomedical Computing, Lecture notes in Computer Science, vol. 1131, Springer-Verlag, 1996, pp. 135–140.

[28] R. J. LeVeque, *Numerical methods for conservation laws*, Birkhäuser, Boston, 1992.

[29] W.E. Lorensen and H.E. Cline, *Marching cubes: a high resolution 3d surface construction algorithm*, SIGGRAPH '87, vol. 21, 1987, pp. 163–170.

[30] C. Lorenz, I.-C. Carlesen, T.M. Buzug, C. Fassnacht, and J. Weese, *A multi-scale line filter with automatic scale selection based on the hessian matrix for*

medical image segmentation, International Conference on Scale-Space, 1997, pp. 152–163.

[31] Liana M. Lorigo, *Curve evolution for medical image segmentation*, Ph.D. thesis, Massachusetts Institute of Technology, 2000.

[32] Liana M. Lorigo, Olivier Faugeras, Eric L. Grimson, Renaud Keriven, Ron Kikinis, Arya Nabavi, and Carl-Fredrik Westin, *Codimension-two geodesic active contours for the segmentation of tubular structures*, Conference on Computer Vision and Pattern Recognition, vol. 1, 2000, pp. 444–451.

[33] Liana M. Lorigo, Olivier Faugeras, Eric L. Grimson, Renaud Keriven, Ron Kikinis, and Carl-Fredrik Westin, *Codimension-two geodesic active contours for mra segmentation*, International Conference on Information Processing in Medical Imaging, 1999, pp. 126–139.

[34] Ravikanth Malladi, James A. Sethian, and Baba C. Vemuri, *Shape modeling with front propagation: A level set approach*, IEEE Transactions on Pattern Analysis and Machine Intelligence **17** (1995), no. 2, 158–175.

[35] S. Osher and R.P. Fedkiw, *Level set methods*, Tech. report, Computational and applied mathematics reports, University of California at Los Angeles, February 2000.

[36] S. Osher and C. Shu, *High-order nonoscillatory schemes for hamilton-jacobi equations*, Journal of Computational Physics **28** (1991), 907–922.

[37] S. J. Osher and J. A. Sethian, *Fronts propagating with curvature dependent speed: Algorithms based on hamilton-jacobi formulations*, Journal of Computational Physics **79** (1988), 12–49.

[38] N. Paragios and R. Deriche, *Geodesic active regions for supervised texture segmentation*, International Conference on Computer Vision, September 1999, pp. 926–932.

[39] G. Sapiro and A. Tannenbaum, *On affine plane curve evolution*, Journal of Functional Analysis **119** (1994), 79–120.

[40] Y. Sato, S. Nakajima, N. Shiraga, H. Atsumi, S. Yoshida, T. Koller, G. Gerig, and R. Kikinis, *Three-dimensional multi-scale line filter for segmentation and visualization of curvilinear structures in medical images*, Medical Image Analysis, 1998, pp. 143–168.

[41] W. Schroeder, K. Martin, and B. Lorensen, *The visualization toolkit: an object oriented approach to 3d graphics*, Prentice Hall PTR, Upper Saddle River, New Jersey, 1998.

[42] J. A. Sethian, *An analysis of flame propagation*, Ph.D. thesis, University of California, 1982.

[43] J. A. Sethian, *Level set methods*, Cambridge University Press, 1996.

[44] J.A. Sethian, *Numerical algorithms for propagating interfaces: Hamilton-jacobi equations and conservation laws*, Journal of Differential Geometry **31** (1990), 131–161.

[45] J. Shah, *Recovery of shapes by evolution of zero-crossings*, Tech. report, Dept. of Mathematics, Northeastern University, Boston, MA, 1995.

[46] Kaleem Siddiqi, Yves Bérubé Lauzière, Allen Tannenbaum, and Steven W. Zucker, *Area and length minimizing flows for shape segmentation*, IEEE Transactions on Image Processing **7** (1998), no. 3, 433–443.

[47] H. Tek and B. Kimia, *Deformable bubbles in the reaction-diffusion space*, Fifth International Conference on Computer Vision, IEEE Computer Society Press, 1995.

[48] Andy Tsai, Anthony Yezzi, , and Alan Wilsky, *A curve evolution approach to smoothing and segmentation using the mumford-shah functional*, Conference on Computer Vision and Pattern Recognition, June 2000, pp. 119–124.

[49] D. L. Wilson and A. Noble, *Segmentation of cerebral vessels and aneurysms from mr angiography data*, IPMI'97, 1997, pp. 423–428.

[50] Chenyang Xu and Jerry Prince, *Snakes, shapes and gradient vector flow*, IEEE Transactions on Image Processing **7** (1998), no. 3, 359–369.

[51] Anthony Yezzi, Andy Tsai, and Alan Willsky, *A statistical approach to snakes for bimodal and trimodal imagery*, International Conference on Computer Vision, September 1999, pp. 898–903.

[52] Alexander Vasilevskiy and Kaleem Siddiqi, *Flux maximizing geometric flows*, International Conference on Computer Vision, July 2001, pp. 149–154.

[53] Kaleem Siddiqi and Alexander Vasilevskiy, *3D flux maximizing geometric flows*, EMMCVPR, September 2001.

Document Log:

Manuscript Version 1 — 17 September 2001

Typeset by $\mathcal{A}_\mathcal{M}\mathcal{S}$-LaTeX — 17 September 2001

Alexander Vasilevskiy

Centre for Intelligent Machines, McGill University, 3480 University St., Montréal (Québec) H3A 2A7, Canada, *Tel.* : (514) 933-5795

E-mail address: `sasha@@cim.mcgill.ca`

Typeset by $\mathcal{A}_\mathcal{M}\mathcal{S}$-LaTeX