Detecting Open Surfaces in Three Dimensions

by

Biswajit Bose

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2009

Author ....
Department of Electrical Engineering and Computer Science
February 25, 2009

Certified by.....
W. Eric L. Grimson
Professor
Thesis Supervisor

Certified by........
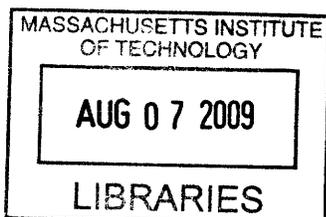John W. Fisher III
Principal Research Scientist
Thesis Supervisor

Accepted by ...
Terry P. Orlando
Chair, Department Committee on Graduate Students

Detecting Open Surfaces in Three Dimensions

by

Biswajit Bose

Submitted to the Department of Electrical Engineering and Computer Science
on February 25, 2009, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Electrical Engineering and Computer Science

Abstract

We present a novel level-set method for representing and detecting *open surfaces* embedded in three-dimensional image volumes. Open surfaces are two-dimensional manifolds with a one-dimensional boundary lying within a three-dimensional volume. Distinct portions of a closed surface can be modeled as open surfaces, as can very thin volumes with negligible thickness.

To detect open surfaces, we propose an interface likelihood model that captures the image appearance along a profile normal to the open surface. This allows statistical modeling of more complex surface-appearance characteristics than just voxel intensities or gradients. Appearance models of the surface are used in the level-set framework in two ways: firstly, to evolve the open surface in the normal direction for the purpose of detecting the location and shape of the surface, and secondly, to evolve the boundary of the open surface in a direction tangential to the surface in order to delineate the extent of the surface. We show that our models are well suited to detecting structures of interest in three-dimensional medical and geological images, and demonstrate their utility on challenging structural magnetic resonance (MR) datasets and seismic-reflection volumes.

Thesis Supervisor: W. Eric L. Grimson
Title: Professor

Thesis Supervisor: John W. Fisher III
Title: Principal Research Scientist

Acknowledgments

For their incredible support and encouragement, I would like to thank my thesis supervisors, thesis readers, office-mates over the years (especially Gerald Dalley and Xiaogang Wang), members of the CSAIL vision and graphics neighborhood, friends and well-wishers around the world, and, last but not least, my wonderful parents.

Kinh Tieu and Michael Richard Siracusa deserve a special mention for making sure my many hours in lab were not spent solely in research pursuits, for that would have been rather dull. Similarly, my roommates over the years in Edgerton House made the inevitable nights and weekends of working from home more tolerable.

Funding support from multiple sources, including the EECS Department at MIT, the Shell Oil Company and DARPA, made it possible to produce this thesis. And finally, thanks to the MIT Libraries, and to Google Scholar, without which this thesis would have taken quite a bit longer.

THIS PAGE INTENTIONALLY LEFT BLANK

Contents

THIS PAGE INTENTIONALLY LEFT BLANK

List of Figures

15

List of Tables

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 1

Introduction

We investigate the problem of detecting an open surface of finite extent in a three-dimensional image volume. An open surface of finite extent is defined as a two-dimensional manifold with a one-dimensional boundary embedded within a three-dimensional space such that the boundary of the manifold lies completely within a bounded volume. Building on methods used for related tasks such as image segmentation and multi-view 3D surface reconstruction, we propose an algorithm for solving this problem, and analyze its performance. Our solution leads us to develop an appearance model well-suited to representing open surfaces. We provide examples of useful applications of our method in medical imaging and seismic imaging.

1.1   Motivation

Understanding volumetric data, such as magnetic resonance (MR) images in biomedical studies, or seismic reflection data in geophysical problems, often requires detection and modeling of physical surfaces that have been imaged in a three-dimensional (3D) volume. Many of these surfaces are *open* surfaces, which means, intuitively speaking, that they are more similar in shape to a sheet of paper floating in space than to the surface of a sphere (which we classify as a *closed* surface). Unlike closed surfaces, open surfaces have a boundary, much like the edges of a sheet of paper. Detecting an open surface requires locating the surface within the volume and de-

termining the shape and extent of the surface, including the extent of the boundary of the open surface. More formally, we define an open surface as a co-dimension-1 manifold with boundary embedded in three-dimensional Euclidean space. A closed surface is a similar manifold, except it does not have a boundary. The open-surface detection problem has not been studied extensively in the published literature.

We present two examples of recent applications to illustrate how detecting open surfaces is useful. The first example concerns a medical imaging problem: detection of the primary visual cortex in the human brain. The visual cortex (Figure 1-2, left) is a region on the cortical surface of the brain, and is hence well modeled as an open surface. Identifying this region automatically from radiological images could significantly speed up the process of diagnosing medical anomalies in patients. The second example comes from the oil and gas industry. A vertical slice through a 3D seismic reflection volume of the earth's subsurface (Figure 1-2, top) shows the layered structure of sedimentary rock. Geophysicists search for suitable geometries in the alternating sand and shale rock layers to identify potential petroleum reservoirs. Due to the nature of the depositional process for these sedimentary rocks, each rock layer (or *horizon*) has finite extent, and can be modeled as an open surface in 3D. In the cross-sectional view seen in the figure, the open surface of the horizon appears as an open curve.

1.2   Objective

We propose to develop a framework for modeling open surfaces in 3D and an algorithm for detecting these surfaces starting from a suitable initialization. The basic input to our system will consist of a 3D intensity volume and an initial open surface. The output of the system will be a triangle mesh that marks the detected open surface. In addition, our model requires a training phase, during which the system will be provided with a few representative intensity volumes with the target open surfaces labeled[1] so that appearance models for surface points can be learned.

---

[1] In our experiments, one volume containing labeled surfaces has been sufficient to obtain training data. More generally, the precise number of volumes needed to learn a model of the surface will
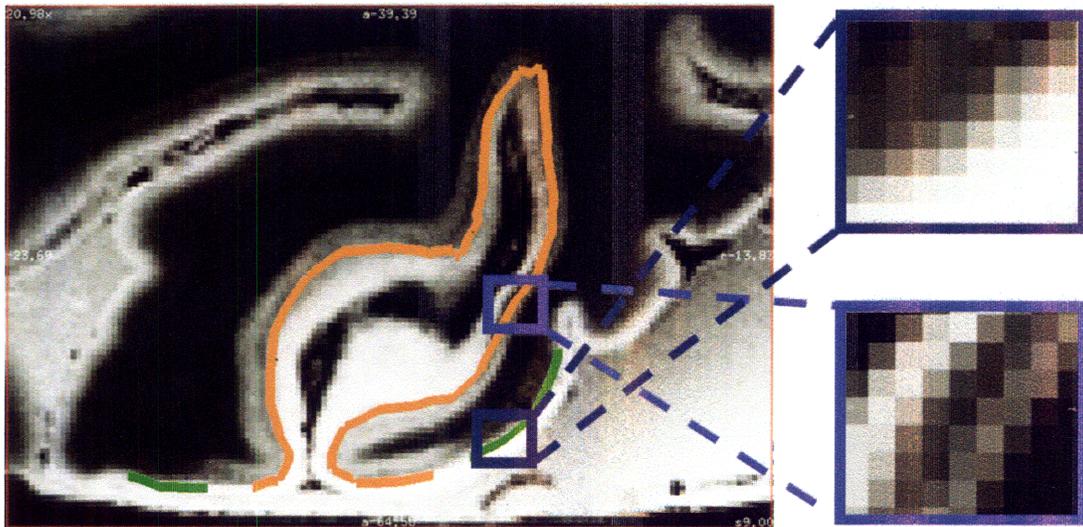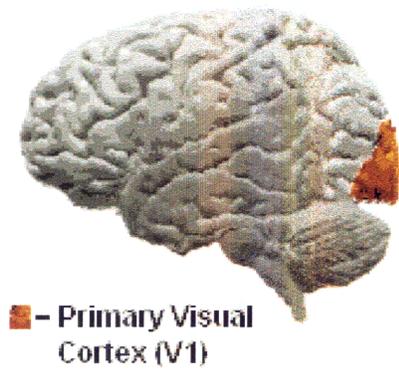
Figure 1-1: **Top:** human brain imaged from the side, with the primary visual cortex marked in orange. **Bottom** A slice through a magnetic resonance image of the brain, showing the visual cortex (marked in orange) and the cortical region around it (marked in green).

bin slice #178
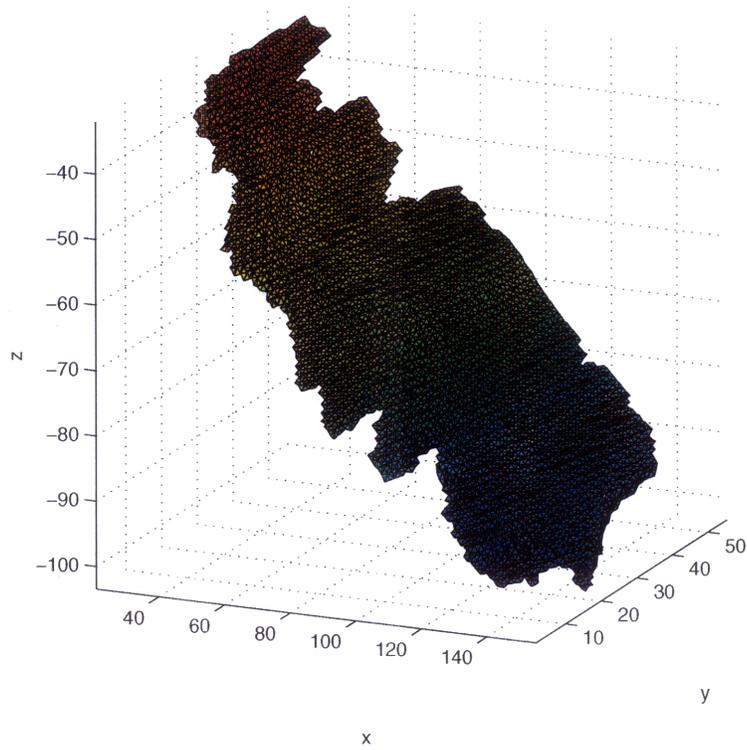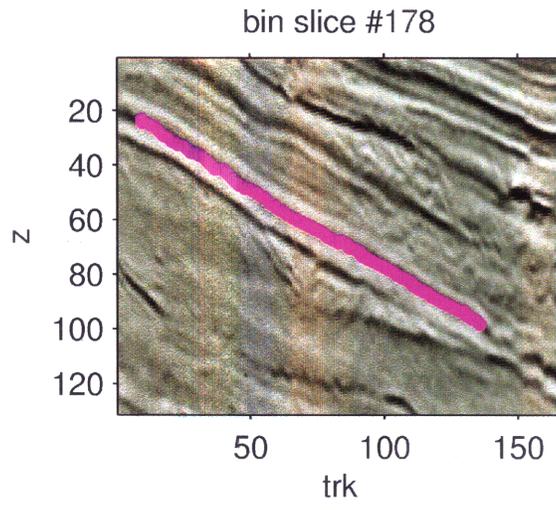
Figure 1-2: **Top:** 2D slice through a seismic data volume from the Gulf of Mexico. A detected horizon surface, seen as an open curve in this cross section, is marked in pink. **Bottom:** A 3D view of the detected horizon surface.

We note here that there are two common uses of the term **detection** in computer vision: one concerns determining whether or not an object is present in the image, while the other concerns localizing the object in the image (given that it is present). Our work uses the term detection in the latter sense. We could have used the term *localization*, but we note that open surface detection includes both localization of the open surface in space as well as determination of the shape and extent of the deformable surface.

1.3   Our idea in a nutshell

Our method for detecting a target open surface involves deforming (or *evolving*) an initial open surface under suitable data-driven and smoothness forces, till it matches the target surface. We *match* the evolving surface to the target in a maximum *a posteriori* (MAP) sense within a statistical framework[2]. The surfaces evolving and target surfaces are matched in location, shape *and* extent. The forces driving the evolution are designed to move the initial surface in a manner that brings it closer to the target, and to gradually disappear as the target is approached. Such a surface evolution framework is commonly used for *closed* surfaces [39, 51], but has only recently been studied in restricted cases for open surfaces [37, 61].

There are three major challenges concerning the open-surface detection problem:

1. **Choosing a suitable representation for an open surface.** Closed surfaces are easily and elegantly represented using level-sets, but this representation does not directly apply to open surfaces. We propose a construction using two level-set embedding functions to represent open surfaces.

2. **Designing a surface evolution algorithm that works for this representation.** Standard level-set-based surface evolution methods were designed for evolving a closed surface. We provide an algorithm for determining the location, shape and extent of an open surface through level-set evolutions of the

---

depend on the variability of surface appearance from one volume to another.

[2]As with many similar works, our solution is a local, not global, maximum to the MAP problem.

two embedding functions used in our representation.

3. **Designing appropriate forces to be used for open-surface evolution.**
Appearance-based forces used for evolving closed surfaces are typically derived from the intensity of voxels on the surface or from the voxels in the region *enclosed* by the closed surface. Both types of forces have limitations: the former type may not be distinctive enough and is susceptible to noise, while the latter is not applicable to open surfaces (which do not enclose a region). We propose features derived from intensity-profile vectors normal to the surface, to help capture the appearance of the open surface and its local surroundings.

We discuss these challenges further and compare and contrast our approach with previous work in Chapter 2.

1.4 Key contributions

As the closed-surface detection problem becomes a mature research topic, there is increasing interest in open-surface detection. At the same time, higher resolution imaging techniques are making more open-surfaces of interest directly observable in 3D image volumes. To place our work in context, we state here the three principal ways in which we advance the state of the art in open-surface detection:

1. **Integrated location, shape and extent detection.** We provide an open-surface evolution algorithm that starts from an initial surface and evolves it in the normal direction (to detect location and shape) and the tangential direction (to detect extent) relative to the surface. Previous methods mostly focus on one or the other direction of evolution exclusively.

2. **Surface appearance modeling.** Our profile-based appearance model allows for principled statistical modeling of surface appearance within a band surrounding the surface, not just on the surface itself. This is an extension of previously used local appearance models.

3. **Medical and seismic imaging applications.** We consider detection applications involving three types of open surfaces: Brodmann area detection in MR data, fault detection in seismic data and horizon detection in seismic data. Of these, the first two have not been previously attempted directly in 3D imagery.

1.5   Organization of the thesis

We begin by reviewing previous work in Chapter 2, and relating it to our problem. This provides the necessary technical background for our work. Chapter 3 presents our representation for open surfaces, as well as our algorithm for detecting open surfaces by evolving an initial surface. The next chapter discusses the forces used in our surface evolution, and introduces our appearance model for surfaces. Chapter 5 contains experimental results for our method on data from two sources—medical and seismic images. It also describes performance analyses of the algorithm, obtained using synthetic data. The last chapter summarizes the contributions of this thesis.

Appendix A provides an overview of methods for converting between implicit and explicit representations of surfaces, as this is a core component of our work.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 2

Background

We present here a discussion of previous research ideas that form the basis of our work. Significant among these are methods for evolving curves and surfaces, techniques for modeling surface appearance, and work on applications in medical and seismic imaging. On each of these topics, we also compare and contrast our approach with the methods presented.

Much work has been done on detecting interfaces[1] in image data. Typical interfaces of interest include curves in 2D images and surfaces in 3D images. For the sake of concreteness, we shall refer only to surfaces in what follows, unless otherwise stated. Many of the ideas discussed here apply to interfaces of arbitrary dimension.

Within the class of surfaces, an important distinction needs to be made between what we call *open* surfaces and *closed* surfaces. An *open* surface is a surface that is topologically equivalent to a disk[2]. In contrast, a *closed* surface is a surface that is topologically equivalent to a sphere. In practice, this means that these two kinds of surfaces are represented somewhat differently. See Figure 2-1 for a 2D illustration of the different types of interfaces.

We restrict our discussion here to detection of non-rigid or *deformable* surfaces. Much early work in computer vision was done on detecting surfaces or outlines of rigid

---

[1]By *interface*, we mean a manifold of co-dimension 1 relative to the space in which it is embedded, such as a set of points on a line, or a curve in 2D.

[2]To be precise, an open surface in this context is known as a *surface with boundary* in mathematical topology.
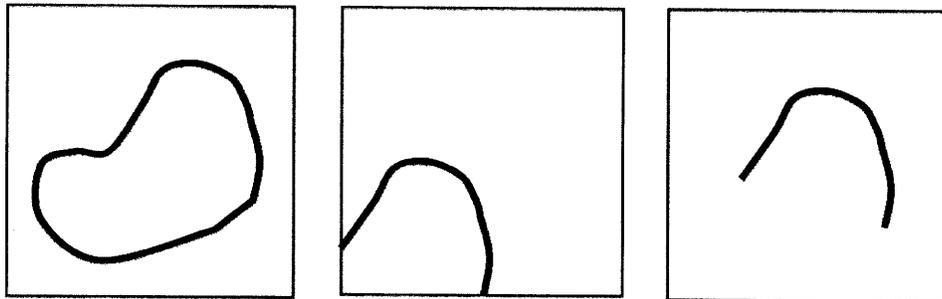
Figure 2-1: Three types of curves: closed curves, open curves whose ends lie on the boundary of the domain, and open curves whose ends lie completely within the domain. The first two types can be represented similarly, since standard level-set methods are applicable to both. We develop a level-set representation for the third type shown. Though this is a 2D example, similar properties hold for surfaces in 3D.

objects, using geometric properties and template matching [33, 35]. Representing and detecting deformable surfaces is significantly more challenging, as the variability in shape of the target surface is typically not easy to model. The most common approach is to start with an initial surface shape and *evolve* it till it closely matches the target surface. Our method falls within this class of methods.

In the following, we first discuss representation and detection of closed surfaces, which have been studied in much greater detail, and then present work on open surfaces.

2.1 Evolving Closed Surfaces

Closed surfaces can be represented and evolved in either an Eulerian (implicit) or Lagrangian (explicit) framework. We briefly discuss these two representations here.

In computer vision and computer graphics, explicit representations such as point sets [23] and polygon meshes [31, 39] were studied extensively first. Polygon meshes can be deformed/evolved using a front-tracking algorithm, to detect the shape of interest. For instance, snakes/active contours [19, 39] are widely used in the context of detecting object boundaries and other nonrigid surfaces.

Later, researchers turned their attention to implicit methods, using level-sets [51] for representing and evolving closed surfaces [11, 48]. The basic idea behind level-

set methods is to represent the surface implicitly as the zero level-set of a higher-dimensional embedding function. Surface deformation is then achieved implicitly by embedding-function deformation on a fixed grid (by solving a time-dependent PDE for the implicit surface). Implicit representations have been previously used for static interfaces in mathematical morphology [58]; however we are particularly interested in dynamic interfaces (*i.e.*, interfaces that are actively evolved). More details of the level-set representation are given in Section 2.2.

Eulerian evolution methods, unlike Lagrangian methods, do not require parametrization of the surface, and can more easily handle sharp corners and cusps that arise when evolving a surface. Further, these methods allow for seamless changes in surface topology as the surface evolves, and the methods generalize easily to higher dimensions [50, 59].

Using either representation, the problem of detecting a surface in an image volume can be posed as one of evolving the surface under the influence of internal and external forces [11, 70]. Internal forces represent prior knowledge about the likely shapes of the surface. They range from simple penalties on the magnitude of the first or second derivatives of the surface to more complex forces derived from trained models of surface shape. External forces account for image data. These forces tend to move the surface in a direction that minimizes mismatch between appearance of the current surface and appearance of the target surface. The target-surface appearance model may simply be a high intensity-gradient (in case of surfaces defined by edges) or could be a statistical model of appearance learnt from training data.

We note here that there is a large body of work dealing with detecting objects of interest in images, that indirectly also detect the boundary of the object [21, 28]. This boundary is typically a closed curve/surface. We do not elaborate on these methods, as they make use of the appearance of the object's interior in detecting the boundary. In contrast, our work focuses on interfaces for which the appearance away from the interface does not help significantly in detecting them.

2.2 Level-set Methods for Evolving Closed Interfaces

We present here an overview of level-set methods for implicitly evolving an interface. To illustrate the methods, we discuss evolution of curves in 2D. However, everything carries over directly to surfaces in 3D, and to even higher dimensions.

2.2.1 Mathematical Background

A level-set technique represents an evolving curve $C(t)$ that evolves with time t as the zero level set[3] of a real-valued embedding function $\Phi(\mathbf{x}, t)$, where $\mathbf{x} \in \mathbb{R}^2$ and $t \in [0, \infty)$. Thus, by construction,

$$C(t) = \{\mathbf{x_0}(t) \in \mathbb{R}^2 : \Phi(\mathbf{x_0}, t) = 0\}. \tag{2.1}$$

The time-evolution of the curve is typically defined by a force field (or *velocity function*) $\mathcal{F}$, a vector function indicating the rate of change of points on the evolving curve:

$$\mathcal{F} = -\frac{\partial C}{\partial t}. \tag{2.2}$$

To achieve this evolution in the implicit framework, the embedding function Φ is made to vary (or *evolve*) with time t in a way that $C(t)$ moves according to the specified flow of the curve. Figure 2-2 shows an example of how evolution of the embedding function leads to evolution of the implicit curve.

Evolution of the curve in $\mathbb{R}^2$ under a force field $\mathcal{F}$ can be related to a partial differential equation (PDE) for Φ as follows:

$$\mathcal{F} = -\frac{\partial C}{\partial t} = -\frac{\partial \mathbf{x_0}}{\partial t}. \tag{2.3}$$

Differentiating the constraint equation $\Phi(\mathbf{x_0}, t) = 0$ for the zero level set, and

---

[3]or zero *isocontour*.
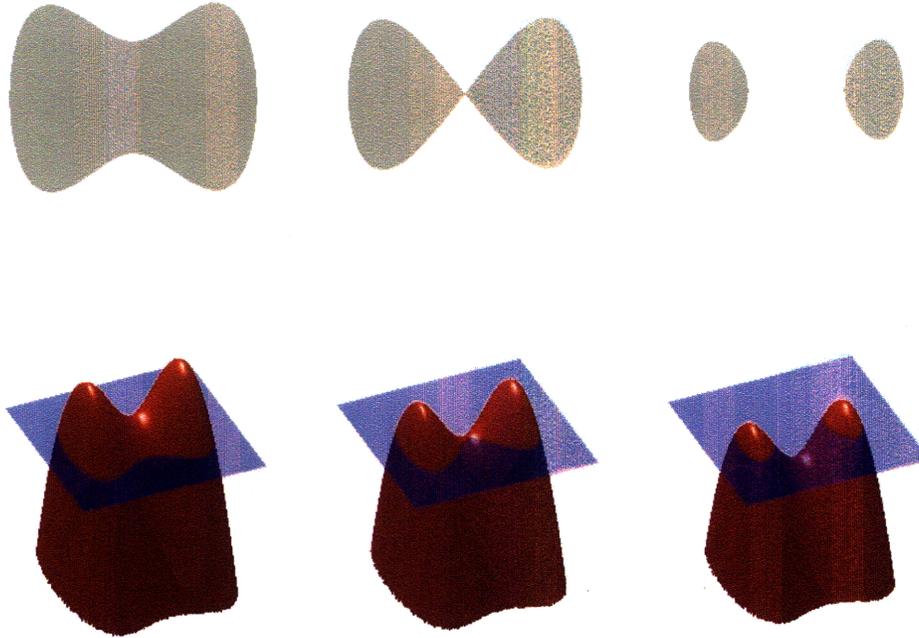
Figure 2-2: An illustration of the level-set method. The embedding function is shown in red at three different times from left to right. The plane marking the zero level for this function is in blue. The corresponding zero level set is the boundary of the gray region shown above the embedding function. As the function varies with time, the implicit curve (its zero level set) evolves in the plane. In this case, the function varies by moving down (*i.e.*, decreasing in value) with time, and the corresponding curve changes topology from a single-segment curve to two segments.

applying the chain rule, we get

$$\nabla\Phi(\mathbf{x_0}, t) \cdot \frac{\partial \mathbf{x_0}}{\partial t} + \Phi_t(\mathbf{x_0}, t) = 0. \tag{2.4}$$

Substituting into this from the previous equation, we get the level-set PDE:

$$\Phi_t(\mathbf{x_0}, t) = \mathcal{F} \cdot \nabla\Phi(\mathbf{x_0}, t). \tag{2.5}$$

2.2.2 Interpretation of Force Field

The surface evolution problem can be posed as statistical estimation problem [41] or a variational energy minimization problem [73]. We establish the connection between the two types of problems here.

In a statistical framework, we are trying to estimate a surface S given a set of surface measurements $\mathcal{Y}$ such that we maximize the posterior probability $p(S|\mathcal{Y})$ of the surface. The posterior probability can be expressed as the product of likelihood and prior, divided by the evidence (a normalization constant):

$$p(S|\mathcal{Y}) = \frac{p(\mathcal{Y}|S)p(S)}{p(\mathcal{Y})}. \tag{2.6}$$

Since we wish to maximize this expression with respect to S, we can take the logarithm, drop the constant that does not depend on S, and *minimize the negative* of the resulting *energy* $E(S)$:

$$
\begin{aligned}
E(S) &= -(\log p(S|\mathcal{Y}) + \log p(\mathcal{Y})) & (2.7)\\
&= -\log p(\mathcal{Y}|S) - \log p(S) & (2.8)\\
&= E_e(S) + E_i(S). & (2.9)
\end{aligned}
$$

In the energy-minimization framework, $E(S)$ consists of an internal regularization energy term $E_i(S)$ corresponding to the statistical likelihood and an external data-fidelity term $E_e(S)$ corresponding to the statistical prior (after taking the negative logarithms). The internal force field F_{Si} and external force field F_{Se} are represented respectively by the gradients/first-variations of the terms in the energy functional[4]. The force field $\mathcal{F}$ in Equation 2.5 is the sum of these internal and external forces.

In this way, the force field can be derived from either a statistical estimation or an energy minimization problem.

---

[4]Not all surface evolutions correspond to energy minimizations resulting in two forces [70].

2.2.3 Numerical Implementation

The level-set PDE is an initial value problem where the embedding function is the dependent variable. It is solved numerically on a fixed grid in $\mathbb{R}^2$, using a given initial curve $C(0)$ as the initial condition. Consequently, the initial curve must be the zero level-set of a suitably chosen initial embedding function. A common choice of embedding function is the *signed-distance function* $D_{C(0)}(\mathbf{x})$ of the initial curve $C(0)$: this is a function whose magnitude at any point $\mathbf{x}$ in $\mathbb{R}^2$ is simply the distance between $\mathbf{x}$ and the point on the curve $C(0)$ that is closest to $\mathbf{x}$. The sign of the signed-distance function is $+1$ for points outside the curve and -1 for points inside it. Thus, for a smooth curve, the corresponding signed-distance function is also smooth at points on the curve[5]. Further, $\|\nabla D_{C(0)}\| = 1$ wherever $D_{C(0)}$ is differentiable.

Note that the velocity function $\mathcal{F}$ was originally defined only on the curve, but the Equation 2.5 is to be solved numerically at points in a 2D grid. Points on the (continuous) curve may not lie on the grid. To get around this, a velocity function needs to be defined everywhere in the grid. While there are many ways of doing this, one popular approach is the so-called *velocity extension* technique [1], wherein the velocity at a point $\mathbf{x}$ in the domain is set equal to the velocity at the point on the curve that is closest to $\mathbf{x}$.

Using the signed-distance function of a curve as the implicit function has a number of benefits. The signed-distance function can be computed quickly using a fast-marching method [59]. When used in combination with velocity extension, a signed-distance function tends to stay a signed-distance function as the curve evolves [73].

A fast marching algorithm can be used to do velocity extension, making this process computationally very efficient [59].

Finally, a suitable numerical scheme needs to be chosen for discretizing the level-set PDE. Time discretization of Equation 2.5 can be done using a simple forward Euler method, which involves first-order differences. Spatial discretization needs to be chosen more carefully, to make sure the method is consistent (*i.e.*, the approximation

---

[5]unlike the distance function, which is positive everywhere and hence not differentiable on the curve.

error vanishes as the discretization step goes to zero). In general, an *upwind* difference scheme can be used, to ensure approximate derivatives are updated using the most current information about the solution. In addition to consistency, the PDE solution needs to be stable. Stability can be enforced by the Courant-Friedrichs-Lewy (CFL) condition [25].

In principle, there is a solution to Equation 2.5 for all time t. However, in practice, the evolution is performed till the curve C—or equivalently, the embedding function Φ—changes by less than a fixed threshold between iterations, or till a certain number of iterations have been performed. An explicit, triangle-mesh representation of the evolved surface can be obtained from the embedding function by using the marching-cubes algorithm [46].

2.3   Evolving Open Surfaces

Open surfaces have received much less attention than closed surfaces in the published literature. While the Lagrangian representation for them is similar to that for closed surfaces, the Eulerian representation is not as straightforward. This is because a typical level-set of an embedding function represents a closed surface or an open surface that reaches the boundaries of the domain[6]. Nevertheless, it is worth considering Eulerian representations of open surfaces because of the advantages of using level-set methods mentioned above.

One representation that has been previously used for modeling an open surface consists of two intersecting closed surfaces [14, 37, 45, 60, 61, 62]. The open surface forms part of one closed surface—let's call this the primary surface—and its boundary is marked by the curve along which the primary surface intersects the other surface—let's call that the secondary surface (see Figure 2-3). Both closed surfaces—primary and secondary—can then be represented implicitly, using respective embedding functions and level-sets.

---

[6]There are exceptional cases where a single level-set can represent an open interface with a boundary in the interior of the domain [47].

34
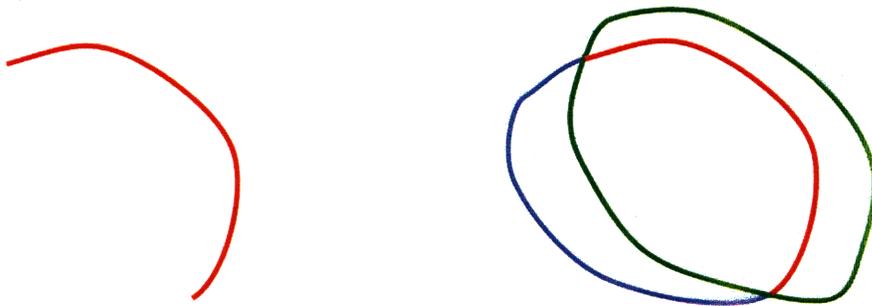
Figure 2-3: Representing an open surface with two closed surfaces, as seen in a slice through a 3D volume: (left) an open surface, seen as an open curve in a 2D slice, and (right) two closed surfaces, in blue and green, whose intersection is the red open surface.

For closed surfaces, each point on the surface evolves by moving in its local normal direction. Tangential direction motion is unnecessary, since that would only reparameterize the surface. However, tangential evolution *does* make sense for points on the boundary of an open surface, as this changes the extent of the surface. In fact, in the two-level-set representation, this can be achieved by normal evolution of the secondary surface while keeping the primary surface fixed.

The distinction between normal and tangential evolution for open surfaces is significant, as much previous work has focused solely on one or the other evolution. Normal evolution alone will move the open surface in 3D and change its shape, without explicitly altering its extent [60, 61]. This is useful when trying to locate an open surface of known extent. Tangential evolution alone will keep the location of the open surface fixed, while changing its extent. An example application of tangential evolution is detecting a region on a known 2D manifold that is embedded in 3D [62]. This is equivalent to the problem of detecting curves on surfaces in 3D [6, 14, 34].

Our work centers on evolving open surfaces in *both* normal *and* tangential directions. There has been some previous work in this area [37], but only in constrained settings with simple appearance models. We should mention that the related but different problem of evolving closed curves in 3D has also been studied [9]: in this

case, only the boundary of the open surface is of interest, while the rest of the surface is not modeled.

2.4 Appearance Models for Surfaces

Interface evolution is commonly driven by a combination of external and internal forces. The external forces depend on the data, while the internal forces reflect prior knowledge. For curves or surfaces embedded in image volumes, different representations of image appearance are used to derive the external force driving the curve/surface evolution[7].

A common choice is to represent the surface appearance using image gradients, wherein each point on the surface is made to evolve in the direction that maximizes the underlying gradient [24, 39, 40, 48]. The magnitude of the external force decreases as the gradient increases. Such a representation is good for detecting interfaces that show up as intensity edges in the image, which presumably are the locations of highest gradient. However, even in these situations, if the initial interface is far away from the target, local gradients may not move it in the correct direction and other forces need to be added to resolve the problem [19, 70]. Gradient-based representations are also typical of direct detection methods such as Canny edge detection [10] that scan the entire image in search of the interface instead of evolving an initial interface. However, unlike interface evolution methods, direct methods produce a disconnected set of point detections that need to be stitched together, instead of a continuous interface.

Alternatively, for closed surfaces, the appearance of the volume enclosed by the surface can be used to drive the evolution [13, 37, 41, 71, 74]. This is known as region-based evolution, and is used for segmentation of images into multiple disjoint regions. The advantages of using regions over using edges alone are that information is gathered from more image points, making the evolution less sensitive to initialization, and

---

[7]This is to be contrasted with curve/surface evolution in a volume containing sparse data, such as point measurements [61], where *appearance* information is not available.

image gradients are typically not used as features, thus decreasing sensitivity to noise. Further, suitable evolutions can be set up so that two regions of interest need not be separated by an intensity edge, but can still be detected based on different appearance statistics—such as two regions with same mean intensity but different intensity variance [41, 71], or regions with different texture features [52]. One disadvantage of region-based methods is that they do not work if the regions enclosed are inhomogeneous in appearance. Also, region-based methods are not directly applicable to open surfaces, since there is no *region* enclosed by an open surface.

In our work, we introduce a new statistical appearance model where each voxel on the surface is represented by a *vector* of voxel intensities extracted from a normal[8] profile centered at that voxel. This combines the benefits of locality needed for representing open surfaces with the added robustness to appearance provided by region-based methods. Some previous work has tried to combine the benefits of edge-based and region-based methods in other ways, such as using both local and global force terms to drive surface evolution [12, 52] or localizing the computation of the region-based term to points near the surface [42, 44]. We discuss our appearance model further in Chapter 4.

2.5 Surface detection in medical and seismic imaging

We now turn our attention to applications of open-surface detection.

Open surfaces appear in 3D image volumes of layered structures, such as medical and seismic images. In each case, the open surface being modeled is actually a volume (*i.e.*, has some small thickness), but it is more convenient to use a surface representation because of its ability to capture deformations easily. Open curves appear as the visible portions of closed curves in images, where part of the closed curve—an object boundary, for instance—has been occluded. Inter-object occlusion or self-occlusion occurs when a 3D object is viewed in a 2D image, producing open

---

[8]normal to the surface, that is.

curves. Here, we focus on open surfaces in 3D.

2.5.1 Surfaces in medical imaging

One goal of medical imaging is the extraction of information about anatomical structures from images. For example, the human cerebral cortex, heart and prostate have been widely studied in this regard [22, 64, 65]. The cortex is the outermost layer of the brain, and, due to its minimal thickness relative to its extent, is often modeled (to a first approximation) by a 2D manifold in 3D[9] [26]. Consequently, regions of the cortex can be treated as open surfaces.

Of particular importance is the parcellation of the cortex into the so-called Brodmann areas, which are structurally and functionally distinct cortical regions [8]. Brodmann areas are characterized by their distinct cellular structure and organization, as first observed under the microscope in stained histological sections by Brodmann [8] in 1909. Examples include the somatosensory cortex, the motor cortex and the primary and secondary visual cortices. Some Brodmann areas, such as the primary visual cortex, show characteristic lamination patterns due to the concentration of myelin fibres.

Most work on detecting Brodmann areas has been done in 2D, either in slices of the brain obtained via histological analysis of post-mortem specimens [3, 4, 57] or in 2D slices through 3D MR data [69]. These detections are used to build a probabilistic atlas of the cortical regions in a standard coordinate space, which can be used to map the regions to a new brain. However, the accuracy of this mapping is limited by the accuracy of transforming to the standard space, as well as by the inter-subject variability in location of the Brodmann areas relative to brain landmarks derived from the cortical folding pattern (sulci and gyri). The popularity of histology is partly because the necessary resolution for analyzing Brodmann areas in MR images was not available till very recently [4, 5]. Detection of Brodmann areas *in-vivo* from MR data is much preferred to histology, as it would lead to more precise localization of Brodmann areas in living patients, which could then be used in conjunction with

---

[9]though there are methods that directly model the volumetric geometry of the cortex [72].

Anticlinal Trap

Fault Trap

(a)   (b)
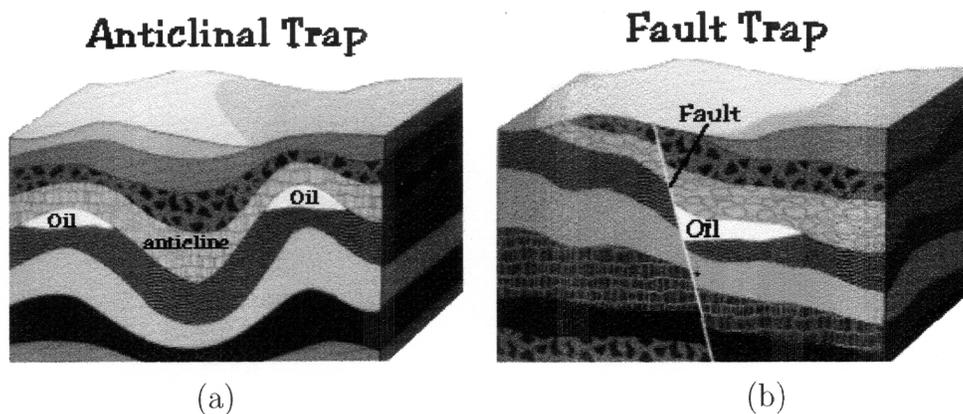
Figure 2-4: Two types of hydrocarbon traps: (a) an anticlinal trap produced by folding of horizons, and (b) a fault trap, produced by rock movement at a geologic fault.

neurological studies such as functional MRI for studying brain function [17]. Our method allows for direct 3D detection of Brodmann areas in high-resolution MR volumes, and is demonstrated for the primary visual cortex (Brodmann area 17) in Chapter 5.

2.5.2 Surfaces in seismic imaging

Sedimentary rocks, displaying a characteristic layered structure, are extensively analyzed in the oil and gas industry [20] as they are likely to contain reservoirs of petroleum. Typical sedimentary rocks contain alternating layers of sandstone and shale, created by repeated sediment deposition near a coastline over millions of years. 3D seismic imaging, based on acoustic reflections from rock interfaces at which rock properties change, captures this layered structure well. Each sedimentary layer, or *horizon* [27], presents itself as an open surface in the 3D seismic volume, bordered by faults or other geologic structures.

Interpreting seismic volumes from petroleum-bearing basins has traditionally been done manually by exploration geologists and geophysicists, either using paper and pencil or using a mouse and a computer screen. Recently, as the amount of data obtained through seismic surveys has grown rapidly, the emphasis in the oil exploration business has shifted to automating the interpretation of seismic volumes. In particu-
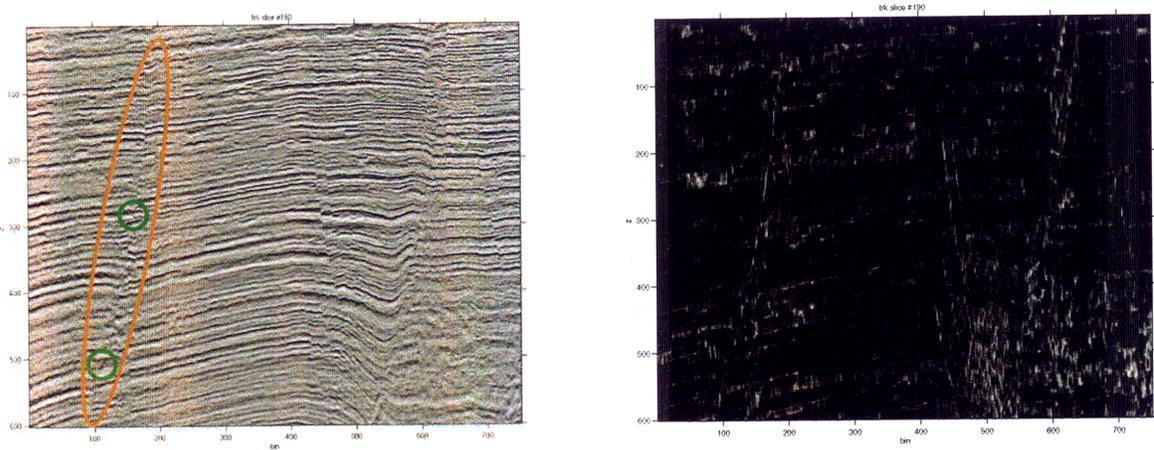
Figure 2-5: A geological fault (circled in orange) seen in a slice through seismic data (left), and the corresponding dissemblance volume computed from the seismic data (right). The green circles highlight regions of the fault where there is not much local evidence of the fault.

lar, automated highlighting of appearance features (or *attributes*) is used by geologists and geophysicists as an aid to manual interpretation of horizons [53]. Such automation is important for speeding up the process of analyzing large seismic volumes that cover many miles of the earth's subsurface.

The geologists and geophysicists are essentially searching for a horizon where all the conditions necessary for accumulation of a substantial quantity of petroleum seem to be present. This involves a source rock (the original source of organic matter that is converted to hydrocarbons) a reservoir rock where the hydrocarbons accumulate (typically a porous sandstone layer) and a trap that prevents hydrocarbons from escaping (often an impermeable shale cap, or a geologic fault). Figure 2-4 shows two illustrative examples of hydrocarbon traps.

The oil company uses the recommendations of the geoscientists in deciding where to drill exploration wells for oil and gas. Most exploration wells are *dry holes*, meaning that they do not contain significant amounts of oil or gas. Once a well hits a reservoir, further seismic measurements are made in the area to determine the best set of locations for production wells to be drilled. A single well can cost millions of dollars to drill, so it is crucial to extract as much information as possible about the probability of hitting an oil reservoir from all data (including seismic data) available beforehand.

A related problem is that of detecting geologic faults, which are discontinuities in the layered structure produced by relative movement of rocks on either side of the fault. These fault discontinuities have finite extent, and can hence also be modeled as open surfaces. Of particular interest are dip-slip faults, in which there is a vertical displacement between the rocks on either side of the fault surface, resulting in a visible discontinuity in the horizons. In contrast, strike-slip faults, arising from lateral displacement of rocks, may not provide significant evidence in seismic data.

Figure 2-5 shows a slice through a seismic volume, and an attribute volume called dissemblance that is computed from it to highlight dip-slip faults. Note that there is a fault in the region marked in orange, as seen from the vertical misalignment of the horizon layers on either side of it. Fault detection is not easy, though, and much prior knowledge is used by geoscientists to detect them. For instance, in the two regions circled in green, there is little evidence for the fault. This is because the horizon intensities on either side of the fault happen to match up, or because the horizons are not very prominent. As a result, there are gaps between the bright edges marking the fault in the dissemblance volume. Automated fault detection is a relatively recent but very important problem [18, 32, 36], because of the role faults play in creating traps for oil reservoirs.

2.6 Summary

As we have seen, there has been much work on detecting closed surfaces. Some work has been done on open surfaces, but typically for either normal evolution or tangential evolution alone. We present an integrated method for evolving open surfaces (Chapter 3), along with a statistical likelihood model of appearance from which external forces driving the evolution are derived (Chapter 4). Medical imaging and seismic imaging provide many examples of open-surface detection problems, only a few of which have been previously tackled. We present three applications of our integrated open-surface detection method to these two domains (Chapter 5).

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 3

Open-surface Representation and Evolution

We present here our level-set representation for open surfaces, and our algorithm for evolving an initial open surface to detect the target surface.

To help refresh the reader's memory about level-set methods, we provided a primer on level-set evolution in Section 2.2. We will assume here that the reader is familiar with level-set representation and evolution for closed surfaces. More details can be found in a text on level-set methods, such as those by Osher and Fedkiw [50] and Sethian [59].

3.1   Representing an Open Surface Implicitly

We represent an open surface implicitly using two level-set embedding functions (and hence two closed surfaces). This is similar to previous implicit representations [37, 61], but we add an additional constraint requiring orthogonality of the two surfaces where they intersect, as this gives us some nice properties. We choose an implicit representation for the reasons mentioned in Section 2.1.

We assume that an initial open surface $\mathcal{S}_o$, with boundary $\mathcal{C}_o$, is provided to us in some form. One such example that we consider in some detail is an open surface given as a triangle-mesh that does not self-intersect. This initial surface could be

(a)   (b)   (c)
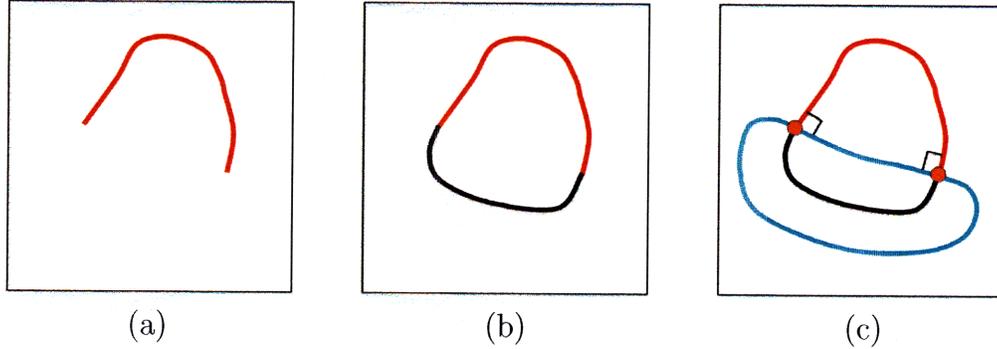
Figure 3-1: Relationship between primary and secondary surfaces. Note that this is a 2D example, so open surfaces in 3D are replaced by open curves in 2D, and closed curves in 3D are replaced by pairs of points in 2D. (a) shows the open surface we wish to detect in red. (b) shows the arbitrary closure of this open surface in black. The closed surface (red and black together) form the primary surface $\mathcal{S}_p$. (c) shows the secondary surface $\mathcal{S}_s$ in cyan. This implicitly defines the curve C_o, shown using orange dots, which forms the boundary of the open surface on the primary surface.

a small patch of the surface that was manually identified by an expert, or it could be the result of an automatic feature detection method that was run on the image volume. To convert an explicit representation of the open surface to an implicit one, we first *close* the open mesh by adding extra vertices and edges. We provide two algorithms for closing an open surface in Section 3.2[1]. We call the resulting closed surface the *primary* surface, $\mathcal{S}_p$. Only a part $\mathcal{S}_o \subset \mathcal{S}_p$ of the closed surface is *relevant*, *i.e.*, represents the open surface. C_o is the curve that forms the boundary between the relevant part $\mathcal{S}_o$ and the remaining, non-relevant, part of $\mathcal{S}_p$. To implicitly represent C_o, we use another closed surface, called the *secondary* surface, $\mathcal{S}_s$. This surface has the property that $C_o = \mathcal{S}_p \cap \mathcal{S}_s$. Further, by construction, we require that surface normals to $\mathcal{S}_p$ and $\mathcal{S}_s$ are orthogonal where the two surfaces intersect. We present our algorithm for constructing the secondary-surface from the primary surface—a process we call label extension—in Section 3.3. The orthogonality property is significant as it leads to a simplified surface evolution algorithm, which we present in Section 3.4. Figure 3-1 illustrates the relationships among the surfaces we construct[2].

---

[1]We provide the details of our automatic mesh-closing algorithm for the sake of completeness. In practice, this is a detail, and there are other ways of finding an initial closed surface (including interactive initialization)

[2]Note that this figure uses a 2D example for illustration.

3.2 Closing an Open Surface to Construct a Primary Surface

As an input to our surface detection process, we are given a triangle mesh representing an open surface. Our task is to close it somehow, so that the original open surface lies on the resulting closed primary surface, which in turn can be represented and evolved implicitly using level-set methods. The closed mesh should thus have no boundary edges (and thus be topologically equivalent to the surface of a sphere), or should extend to the edges of the domain.

Closing an arbitrary open mesh is non-trivial. We present here a few different methods we have developed for this task, and discuss their pros and cons.

3.2.1 Explicit Construction of Closed Surface

We calculate oriented[3] normals at each vertex of the open mesh, and move by an amount ϵ along each normal to create a new vertex. New edges are added to connect the new vertices among themselves in a manner paralleling the connection of the old vertices. Vertices on the original boundary are then connected pairwise to vertices on the new boundary, using a pair of triangles. This produces a closed mesh. The above steps are illustrated in Figure 3-2 for a simple, planar mesh.

In order for the closed mesh to be useful for our purposes, we require it to be non-self-intersecting. This depends on ϵ and the precise choice of vertex normals[4]. While it is possible to guarantee this property by construction, the resulting method is quite complex. Instead, equivalent results can be obtained in practice, by keeping ϵ small (relative to the typical gap between vertices) and repeatedly sampling vertex normals till a non-self-intersecting mesh is obtained. In our experiments, we started with each vertex normal calculated as the average of adjacent face normals, and typically required 5-10 samples to obtain a suitable closed mesh.

---

[3]By oriented, we mean that all the normals must point outwards on one side of the surface.

[4]Note that while normals for mesh faces are uniquely defined, there is no clear consensus on how to define normals for mesh vertices.
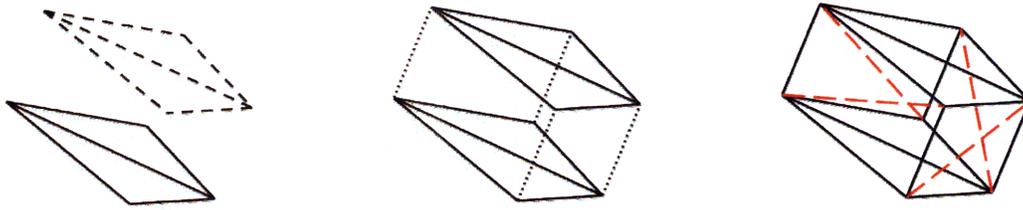
Figure 3-2: An illustration of the steps in the explicit construction of a closed surface from an open one. (Left) The vertices of the open triangle mesh (shown in solid lines) are moved in their local normal direction, and connected by edges (shown in dashed lines) in a manner paralleling the structure of the open mesh. (Center) Each vertex on the boundary of the open mesh is connected to its corresponding vertex by an edge (shown in dotted lines). (Right) Additional edges (shown in red dashed lines) are added along the boundary, forming a closed triangle mesh.
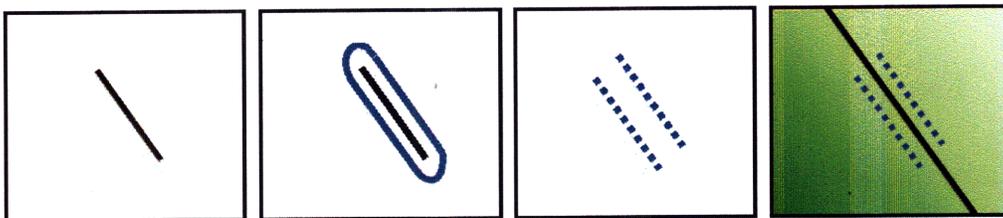
Figure 3-3: An illustration of the steps in the implicit construction of a closed curve from an open one. (Left) The original open curve. (Center left) The d-th level set of the distance function for the open surface. (Center right) Two open surfaces approximately equidistant from the original surface, obtained after removing points within $d + 1$ of the boundary of the original surface. (Right) Solution to Laplace's equation, shown in green as the implicit function that represents the desired closed surface.

3.2.2   Implicit Construction of Closed Surface

One shortcoming of the explicit construction described above is that the primary surface so obtained will have very large curvature near the boundaries of the open surface from which it is derived. This sharp change of orientation can be hard to smooth out during normal evolution. Hence, we explored an alternative, implicit closure construction that avoids introducing the unnecessary curvature. The steps in the construction are also illustrated by a 2D example in Figure 3-3.

To begin with, let us assume we have available *two* additional open surfaces, one

on either side of the initial open surface, and both equidistant from the initial surface and of similar extent to it. We describe later in this section how these two surfaces may be obtained from our single initial surface. We rasterize the given meshes for these two additional open surfaces within the volume, labeling the voxels of one mesh as +1 and those from the other mesh as −1. Then, we solve Laplace's equation within the volume, using the labeled voxels as constraints (*i.e.*, boundary conditions). The solution is a smooth interpolation of the labels within the volume. The zero level set of the solution is used as our initial closed surface.

Now, to obtain the two additional surfaces from a single initial surface, we use the following method:

1. Run the closest point transform within the volume to obtain the distance function from the initial surface

2. Consider the closed surface S_d defined by the d-th level-set of this distance function. This is the set of all points at distance d from the initial surface.

3. Split S_d into two parts by removing all points that are within distance $d + 1$ of the *boundary* of the initial open surface. Effectively, this removes from S_d those points whose closest point on the initial surface lay on the boundary of that surface.

4. Use these two disjoint parts as the two additional open surfaces in the above procedure.

3.3 Label Extension to Construct a Secondary Surface

The secondary surface is (re)initialized at the start of the evolution and after every few iterations of normal or tangential evolution. This is done implicitly by (re)initializing the secondary level-set embedding function. This helps ensure that the secondary

surface is orthogonal to the primary surface, and that the secondary function remains a signed-distance function (as typically used in level-set methods).

First, points on the primary surface, $\mathcal{S}_p$, are labeled as relevant or non-relevant using either the initial open-surface mesh or the sign of the current secondary embedding function, Ψ_s. Then, we calculate a closest-point transform in the volume. To do this, a fast-marching method is used to fill the volume with a binary label field $\mathcal{L}$ that indicates at each voxel whether the closest point to that voxel on $\mathcal{S}_p$ is labeled relevant or non-relevant. To get the new secondary embedding function, Ψ_s, we solve the eikonal PDE $|\nabla\Psi_s| = 1$ with a fast marching method (Section 11.4.1 of [59]), using the zero level-set of $\mathcal{L}$ as the initial condition.

As a consequence of the closest-point transform being run on labels defined on the primary surface, the secondary zero level-set, $\mathcal{S}_s$, has the desired orthogonality property. This can be seen as follows. Consider a point $\mathbf{x_1}$ on the secondary surface $\mathcal{S}_s$ that is a small distance δ away from the primary surface. Since $\mathbf{x_1}$ lies on $\mathcal{S}_s$, it belongs to the zero level-set of the label field $\mathcal{L}$. By construction, points in $\mathcal{L}$ are equidistant from the relevant and non-relevant regions. Let us say $\mathbf{x_a}$ and $\mathbf{x_b}$ are relevant and non-relevant points respectively at a small distance ϵ to $\mathbf{x_1}$, where $\epsilon > \delta$. Thus, $\mathbf{x_1}$, $\mathbf{x_a}$ and $\mathbf{x_b}$ form an isosceles triangle. However, the mid-point, $\mathbf{x_2}$, of the line joining $\mathbf{x_a}$ and $\mathbf{x_b}$ is also equidistant from them. For small enough ϵ and a smooth primary surface, $\mathbf{x_2}$ lies on the secondary surface. Using the properties of an isosceles triangle, the line from the apex to the mid-point of the base is orthogonal to the base. Thus, the primary surface and secondary surface are orthogonal where they intersect.

We use the term *label extension* to describe the above construction, due to its similarity to velocity extension [1]. Both processes require computation of a closest-point transform. In fact, label extension is simply velocity extension with binary velocities. Like velocity extension, label extension is speeded up in practice by running it only within a narrow band.

3.4 Level-set-based Evolution of an Open Surface

The core of the open-surface evolution process consists of alternating two level-set evolution tasks:

1. **Normal evolution**: the primary surface is evolved using a level-set method, while the secondary surface is held fixed.

2. **Tangential evolution**: the secondary surface is evolved, while the primary surface is fixed.

Each task involves multiple iterations of level-set evolution of the corresponding closed surface. The two tasks could also be alternated after a single iteration of each evolution, but this would be computationally expensive because the secondary surface needs to be reconstructed when switching tasks. This reconstruction step is required to restore the orthogonality of the primary and secondary surfaces. The orthogonality property, in turn, means that small level-set perturbations to either the primary or secondary surface preserves the properties of the other.

We now discuss these two evolution tasks in detail. Figure 3-4 gives a graphical summary of our alternating surface evolution process.

3.4.1 Normal evolution

The goal of our *normal evolution* task is to determine the location and shape—but not the extent—of the target surface in 3D. Within a statistical framework, this is a maximum *a posteriori* (MAP) estimation problem, where we are trying to estimate a closed surface that contains the target open surface. As we show in Chapter 4, surface evolution forces can be obtained from the gradient descent procedure for this MAP optimization problem. Our normal evolution task has been developed within this framework. Note that we are assuming we have statistical models—likelihoods and priors—for both the relevant and non-relevant regions of the primary surface; the corresponding likelihood models will be discussed in more detail in Section 4.1.
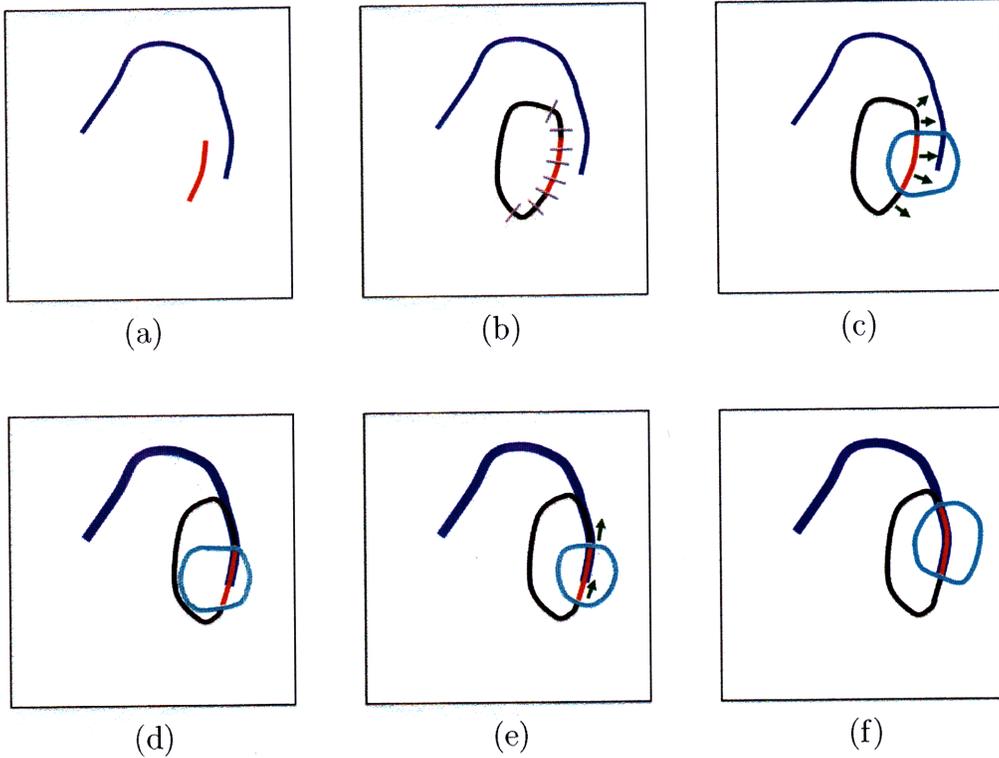
Figure 3-4: An illustration of one round of the alternating steps of normal and tangential evolution. (a) Initial open surface (in red) and target ground-truth open surface (in blue). (b) The initial open surface is closed, forming a primary surface, S_p, that consists of a relevant (red) and a non-relevant (black) region. Profile vectors (discussed in Chapter 4) at points on the primary surface are shown in gray. (c) Normal evolution will cause the primary surface to move in its normal direction. The secondary surface, S_s, shown in light-blue, is orthogonal to the primary surface. (d) As a result of normal evolution, the open surface moves closer to the target location. However, its extent remains unchanged, since the secondary surface has not evolved. (e) The secondary surface is re-initialized by label extension. Next, tangential evolution will cause the secondary surface to move in *its* normal direction. (f) As a result of tangential evolution, the extent of the relevant region is better aligned with the ground truth.

At the start of a normal evolution task, we are given a closed primary surface S_p, and a secondary surface S_s that has been constructed using label extension (Section 3.3) to be orthogonal to S_p where the two intersect. The surface S_p is evolved over time in the local normal direction using standard level-set techniques [50]. We represent S_p as the zero level-set of a 3D function Ψ_p, which we call the primary level-set embedding function. Following [50], evolution of S_p involves solving the nonlinear

PDE:

$$(\Psi_p)_t = \mathcal{F}_p \cdot \nabla \Psi_p, \tag{3.1}$$

where $\mathcal{F}_p(x)$ is a force field that specifies the velocity with which point x on the primary surface moves during surface evolution.

During normal evolution, we mostly ignore the fact that only part of the primary surface is actually relevant. However, there is a subtlety: during normal evolution, we modify the force field near the boundary between the relevant and non-relevant regions. We discuss this modification in Section 3.5. Apart from this, we do not distinguish between points in relevant and non-relevant regions during normal evolution. Since the non-relevant region results from an arbitrary closure of the open surface, there is a chance that it will affect the evolution by intersecting the relevant region (and hence causing topology change). Such intersection is unlikely when using the implicit construction of the primary surface (Section 3.2.2), since in that case the non-relevant region moves far away from the relevant region. Even if the explicit construction is used, such intersection is unlikely if the surface profile appearance (as discussed in Section 4.1) is not symmetric about the center, as this would prevent both the relevant and non-relevant regions being attracted to the same point from opposite sides.

The force field[5] we use for normal evolution is a sum of two terms:

$$\mathcal{F}_p = \mathcal{F}_{pl} + \mathcal{F}_{ps}, \tag{3.2}$$

where $\mathcal{F}_{pl}$ is an external data-fidelity (or *likelihood*, in case of statistical models) force term, and $\mathcal{F}_{ps}$ is an internal regularization (or *smoothness prior*, in case of statistical models) force term. Note that only the force in the direction normal to the surface matters, since $\nabla \Psi_p$ is oriented along the normal to the zero level-set. Any force in the tangential direction would only lead to reparameterization of the (closed) surface. This is fine, since tangential evolution will be handled separately later.

We consider the likelihood term in more detail in Chapter 4, where we discuss

---

[5]The force field is also known as the velocity function.

the relative advantages of different likelihood models and propose our own likelihood model. The smoothness term comes from a statistical smoothness prior $p(S_p)$, which is an exponential distribution that favors primary surfaces with smaller surface area:

$$p(S_p) = \exp(- \iint\limits_{S_p} da).$$

(3.3)

The smoothness term in our force field is then the first variation of negative logarithm of the prior, as discussed in Section 2.2.2:

$$\mathcal{F}_{ps} = -b_p \kappa_p \mathcal{N}_p,$$

(3.4)

where $b_p > 0$ is the weight factor, κ_p is the mean curvature of the surface at this point and $\mathcal{N}_p$ is the outward normal. Note that $\mathcal{F}_{ps}$ is the 3D analog of the commonly-used 2D curve-shortening flow [15]. We use two different values of the weight b_p for different parts of the surface; we elaborate on this in Section 3.5, after outlining the tangential evolution process. Both the curvature and the local normal can be calculated directly from the level-set function Ψ_p as:

$$\mathcal{N}_p = \frac{\nabla \Psi_p}{|\nabla \Psi_p|},$$

(3.5)

$$\kappa_p = \nabla \cdot \mathcal{N}_p$$

(3.6)

The effect of the smoothness term is to make the surface more rounded by smoothing out any ripples.

In principle, only the location of the zero level-set of Ψ_p matters; the variation of Ψ_p away from the zero level-set should not affect the answer. However, in practice, the choice of the embedding function affects the accuracy of solution [73] and the speed of computation. We follow the standard approach of setting Ψ_p to be the signed-distance function for the closed surface and of using velocity extension of the force field from the zero-level set into the volume [1] to perform the level-set evolution, as discussed in Section 2.2.3. Further, we adopt a narrow-band approach for speeding
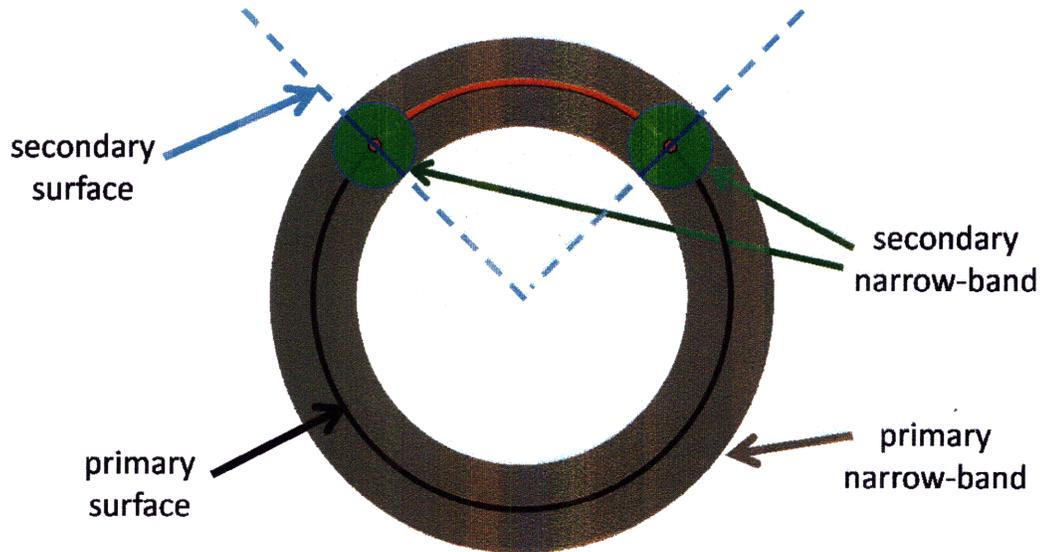
Figure 3-5: Narrow-bands used in level-set computation. Shown here is a 2D example involving an open curve. The primary curve is the black circle here in 2D. The open curve is a region on the top of the circle, marked in red. The intersection of this boundary with the primary curve is the pair of points (marked by small red circles). In theory, the corresponding secondary surface constructed by label extension would be a wedge, marked by the dashed blue lines. In practice, normal evolution is performed in the narrow band (marked in blue in this slice) around the primary curve, while tangential evolution and label extension are performed in the narrow band (marked in green in this image) around the boundary of the open curve.

up our evolution [59].

Figure 3-5 shows a 2D example of a circular primary *curve* drawn in black, with the corresponding narrow band used for normal evolution marked in gray. The other elements of the figure concern tangential evolution, and are explained in the next section.

Before a normal evolution task begins, the secondary surface has been designed to be orthogonal to the primary where they intersect. During the normal evolution task, points on the primary surface move in their local normal direction. Thus, to a first approximation, the labeling of points on the primary surface (relevant/non-relevant) is unchanged by normal evolution. In practice, we choose the number of iterations of normal evolution so that the deviation from orthogonality at the end of the task is small.

3.4.2   Tangential evolution

The goal of our *tangential evolution* task is to determine the extent of the target surface in 3D, assuming its location and shape have been estimated by normal evolution. The estimation of location and shape only need to be accurate near the current boundaries of the evolving open surface.

Estimating the extent of an open surface is a segmentation problem on the primary surface, where we are trying to partition the closed surface into its relevant and non-relevant regions. A segmentation problem can be posed as one of supervised classification, assuming suitable training samples are available, where the objective is to determine the partition boundary that minimizes probability of error. Boundary evolution forces can be obtained from the corresponding gradient-descent solution for the segmentation boundary.

Our tangential evolution task fits within this framework. Tangential evolution continues our open-surface detection by evolving the secondary surface, $\mathcal{S}_s$. The intersection of the primary surface $\mathcal{S}_p$ and the secondary surface $\mathcal{S}_s$ is a closed curve that marks the boundary of the relevant region on $\mathcal{S}_p$. We do tangential evolution by keeping $\mathcal{S}_p$ fixed and evolving $\mathcal{S}_s$. This is also done using a level-set method.

We represent $\mathcal{S}_s$ as the zero level-set of a 3D secondary level-set embedding function, Ψ_s. The embedding function Ψ_s is negative in the relevant region of the primary surface $\mathcal{S}_p$, and positive in the non-relevant region. As with normal evolution, evolution of the secondary surface involves solving a nonlinear PDE:

$$(\Psi_s)_t = \mathcal{F}_s \cdot \nabla \Psi_s, \tag{3.7}$$

where $\mathcal{F}_s(x)$ is a force field specifying the evolution velocity of point x on the secondary surface.

We reconstruct $\mathcal{S}_s$ at the beginning of a tangential evolution task so that it is orthogonal to $\mathcal{S}_p$ where the two surfaces intersect (see Section 3.3 for details). The orthogonality property ensures that normal evolution of the primary function Ψ_p will not change the boundary of the relevant region (to a first-order approximation). At

the same time, *normal* evolution of the secondary function Ψ_s will induce *tangential* evolution of the boundary of $\mathcal{S}_o$ along $\mathcal{S}_p$. Therefore, we hold one surface fixed while performing level-set evolution of the other. This is in contrast to previous open-surface evolution methods [7, 61], where both level-sets are evolved together, and no separate tangential evolution is performed. Separating the two evolutions allows us to more easily control the interaction between them, as discussed in Section 3.5.

Similar to normal evolution, the force field $\mathcal{F}_s$ for tangential evolution also consists of a sum of terms, a likelihood term, $\mathcal{F}_{sl}$, and a smoothness term, $\mathcal{F}_{ss}$:

$$\mathcal{F}_s = \mathcal{F}_{sl} + \mathcal{F}_{ss}. \tag{3.8}$$

Both these terms are derived from a statistical model. The likelihood term is derived from a region-based model of *curve evolution on the primary surface*: we discuss in more detail in the next chapter. The smoothness term for the secondary surface comes from a statistical smoothness prior $p(C_o)$ on the space curve C_o. The curve C_o is the boundary of the open surface, and is defined by the intersection of the primary and secondary surfaces. The prior on this curve is an exponential distribution that favors curves with smaller curve length:

$$p(C_o) = \exp(-\oint_{C_o} ds). \tag{3.9}$$

As discussed by Ambrosio and Soner [2], the negative logarithm of this prior is minimized (or equivalently, the prior is maximized) on a surface S containing the curve by the evolution:

$$(C_o)_t = \kappa_o \mathcal{N}, \tag{3.10}$$

where κ_o is the curvature of the curve and $\mathcal{N}$ the normal to the curve within the tangent space of S.

Now, we impose a statistical smoothness prior $p(S_s)$ on the secondary surface,

which is an exponential distribution that favors surfaces with smaller surface area:

$$p(S_s) = \exp\left(- \iint\limits_{S_s} da\right).$$

(3.11)

The smoothness term in our force field is then the first variation of negative logarithm of the prior, as discussed in Section 2.2.2:

$$\mathcal{F}_{ss} = -b_s \kappa_s \mathcal{N}_s,$$

(3.12)

where $b_s > 0$ is a weight factor, κ_s is the mean curvature of the secondary surface at this point and $\mathcal{N}_s$ is the outward normal. Since the secondary surface is, by construction, orthogonal to the primary surface where the two intersect, we have (with reference to Equation 3.10):

$$\mathcal{N}_s = \mathcal{N},$$

(3.13)

$$\kappa_s = \kappa_o.$$

(3.14)

Comparing Equations 3.12 and 3.10, we see that choosing this smoothness prior on the secondary surface effectively implies a curve length prior on the boundary curve C_o.

In practice, tangential evolution is implemented using the same standard techniques as normal evolution, including velocity extension and narrow-banding. Note that secondary velocities are only defined on the space curve C_o lying on both S_p and S_s, so velocity extension starts from this curve and fills the volume. Further, we achieve a significant speed-up by only performing tangential evolution in a narrow band around this space curve, $i.e.$, where the narrow band around the primary surface intersects the band around the secondary surface. The narrow band used for tangential evolution is marked in green in the 2D slice shown in Figure 3-5. As seen from the figure, this narrow band is much smaller than the entire domain.

After every few iterations, as well as at the end of the tangential evolution task,

the secondary surface is re-initialized via label extension (Section 3.3), to guarantee continued orthogonality to the primary surface. This is done in a narrow-band around the primary surface, as shown in Figure 3-5, in order to limit the computational burden. The reconstructed portion of the secondary is shown in solid blue, while the (hypothetical) location of the rest of it is in dashed blue.

3.5 Interaction between Normal and Tangential Evolution

So far from our description, it might seem that normal evolution and tangential evolution are independent of each other. However, there is a subtlety that makes them dependent. In general, we want a strong smoothness term on the primary surface, to prevent it from becoming too jagged. The standard smoothness term is a curve length penalty, which tends to shrink the primary surface. Near the boundary of the open surface, though, we may wish to expand the primary surface during normal evolution instead of shrinking it, so that the relevant region can grow if necessary during subsequent tangential evolution.

To mitigate this problem, we use a much weaker smoothness term for normal evolution in the vicinity of the boundary C_o. In other words, the weight b_p in Equation 3.4 is small near C_o, and larger elsewhere. As C_o changes due to tangential evolution, the force field for normal evolution changes accordingly. We show in Chapter 5 that this change has the desired effect on primary surface evolution.

Another concern involving the two level-set evolutions is the possibility of spurious intersections between the primary and secondary surfaces away from the open-surface boundary. Previous works have considered this explicitly, since the forces driving the evolution are well understood only near the relevant region of the open surface, and could lead to arbitrary behavior for points on the primary and secondary surfaces far away from it. However, in our case, we can avoid spurious intersections, simply by using a narrow-band method for tangential evolution and re-initializing our secondary

surface through regular label extension. The reasoning behind this is as follows. Spurious intersections can occur only when a portion of the secondary surface that is not initially touching the primary surface moves in a manner that brings it to touch the primary surface. Since we are using velocity extension, the motion of a secondary surface point not touching the primary surface is identical to the motion of the corresponding point on the intersection of the zero-level sets from which the velocity was copied. Further, the narrow-band approach stipulates that the maximum motion of any point on the surface before surface reinitialization is equal to the width of the narrow band. Putting these all together, we can avoid spurious intersections of the secondary with the primary surface if no two points of the primary surface lie within a box the size of the narrow band such that any path connecting these points on the primary surface must pass outside the box. In practice, we found that a small enough narrow-band could always be found after the first round of normal evolution (as this smooths the initial primary surface).

Chapter 4

Likelihood Forces for Evolving Open Surfaces

In the previous chapter, we presented our algorithm for evolving open surfaces without specifying the form of the data-fidelity terms in the force field. We now turn our attention to a specific choice of data terms for normal and tangential evolution that can be used for many applications involving open-surface detection.

Following the widespread adoption of probabilistic methods for detection problems, we use a statistical model of appearance of our open surface [41, 52, 74]. The appearance model makes use of a probability density, $p(\mathbf{y}|\theta)$, based on a measurement $\mathbf{y}$ of the underlying appearance, such as voxel intensity or a texture measure, and model parameters θ. As an example, a simple model could be a one-dimensional Gaussian, with the measurement $\mathbf{y}$ being the brightness of a surface voxel and the parameters θ being the mean and variance of the Gaussian.

In the following, we propose a more complex statistical model for open-surface detection (Section 4.1). We use our model to obtain the data-fidelity force terms—or *likelihood* force terms—for normal evolution (Section 4.2) and tangential evolution (Section 4.3). For normal evolution, the likelihood force, $\mathcal{F}_{pl}$, comes from a maximum-likelihood estimation problem for the primary surface, which implies moving the surface in the direction where the surface appearance maximizes the likelihood of the surface under our chosen likelihood model. For tangential evolution, the likelihood

force, $\mathcal{F}_{sl}$, results from a maximum-likelihood estimation problem for the boundary curve of the open surface, which we show is equivalent to a supervised segmentation problem involving binary classification with minimum probability of error. While both evolutions have statistical models underlying them, the difference lies in the fact that normal evolution involves locating a closed surface—the primary surface— in space while tangential evolution involves labeling a closed surface—the primary surface again—in a binary manner.

Given that the above likelihood forces are combined with their respective smoothness forces from Section 3.4 to determine the final force fields in our framework, our solution for each evolution task involves locally maximizing the posterior probability of a surface or a curve under our chosen statistical model.

4.1　Profile-based Statistical Likelihood Model

Two common types of data terms used for curve/surface-evolution are edge-based [11, 39] and region-based [13, 54, 71] terms. Edge-based terms come from statistical models that use only appearance measurements on the interface, whereas region-based terms are calculated from energies that also use the appearance within the region enclosed by the interface. Region-based terms are more robust to noise and incorrect initialisation, as discussed in Section 2.4. However, while edge-based terms can be defined for both open and closed surfaces, region-based terms are only defined for closed surfaces.

For our normal evolution task, we would like the benefits of region-based terms, but we are evolving an open surface. Note that even though normal evolution involves a closed primary surface, we cannot use a region-based evolution since the primary surface is chosen arbitrarily to pass though the open surface. Hence, appearance measurements taken from within the *region* enclosed by the primary surface do not provide useful information.

The appearance measurements we propose to use are *intensity profiles* along the normal direction to the surface, at points centered on the open surface (as illustrated

60

by the gray lines in Figure 3-4b). These profiles are vectors whose elements are voxel intensities. Such profile-based descriptors can also be thought of as a generalization of edge-based models to include the appearance of regions near the edge. A profile vector of length three, for example, would essentially capture edge appearance only. Longer profiles can represent some of the appearance information further away from the interface as well. Profile vectors are very suitable if there is a distinct layered appearance around the surface of interest, but significant variability in intensity far away from it. Use of the profile vector allows us to capture not just the distribution of intensities on/near the surface, but also the geometric profile structure of the surface appearance.

In all of our tasks, the statistical likelihood model is over normalized normal vectors centered on the primary surface. Each observed normal profile vector, $\mathbf{y_o}$, is normalized in intensity by first subtracting the element-wise mean from each element and then scaling the element intensities of the profile, to obtain a unit vector, $\mathbf{y}^1$. Normalization helps reduce the effects of variation in average intensity from one part of the data volume to another, such as may be produced by a varying bias-field in a magnetic resonance (MR) volume, or by varying reflectivity properties in a seismic volume. In certain cases, normalization removes much of the useful information about the profiles, and should be avoided. For instance, if the main difference between two sets of profile vectors is the average intensity, this distinguishing feature will be lost upon normalization.

We model each normalized profile vector $\mathbf{y}$ of length l as being generated independently from a latent low-dimensional Gaussian random vector α embedded in the high-dimensional space, plus an additive Gaussian noise vector $\mathbf{n}$:

$$\mathbf{y} = \Phi\alpha + \mathbf{n}, \tag{4.1}$$

where Φ is an $l \times k$ projection matrix of orthonormal *basis vectors* that transforms the k-dimensional intrinsic Gaussian to the l-dimensional vector space of the observed

---

[1]Note that this is different from simply scaling the profile to make it a unit vector.

profile. Thus, we have the Gaussian variables

$$\alpha \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}) \tag{4.2}$$

and

$$\mathbf{n} \sim \mathcal{N}(\mathbf{m}, \mathbf{R}), \tag{4.3}$$

which implies that

$$\mathbf{y} \sim \mathcal{N}(\mathbf{m}, \Phi \mathbf{Q} \Phi^T + \mathbf{R}). \tag{4.4}$$

Without loss of generality, we restrict $\mathbf{Q}$ to be diagonal [55]. Further, we restrict $\mathbf{R}$ to be a multiple of the identity matrix, so as to prevent the noise from trivially explaining all the structure in the data [55].

The choices of l and k are problem-dependent. We choose l large enough that we get few local maxima in location estimates during normal evolution, but small enough that we get good discriminability between the relevant and non-relevant region models during tangential evolution. We choose k so that the basis vectors explain 90% of the variance of the normalized profile appearance for a particular application. As shown in Chapter 5, there is typically a sharp drop-off in variance as we move through the basis vectors in order of variance, indicating that a low-dimensional likelihood model is suitable.

The parameter set, θ, of this model, M, are learned via probabilistic principal component analysis (PPCA) [55, 66] on a training set of profile vectors. The set θ of a PPCA model consists of the matrix of basis-vectors, Φ, the diagonal covariance matrix $\mathbf{Q}$ of the latent random vector α and the mean $\mathbf{m}$ and covariance $\mathbf{R}$ of the noise vector $\mathbf{n}$.

PPCA is similar to the hugely popular principal component analysis (PCA) [38] in identifying a low-dimensional basis that captures the essence of high-dimensional data. In fact, both are special cases of factor analysis. PPCA—also known as sensible PCA or SPCA—has the advantage over PCA of being based on a probabilistic model, allowing it to be used in statistical techniques [55].
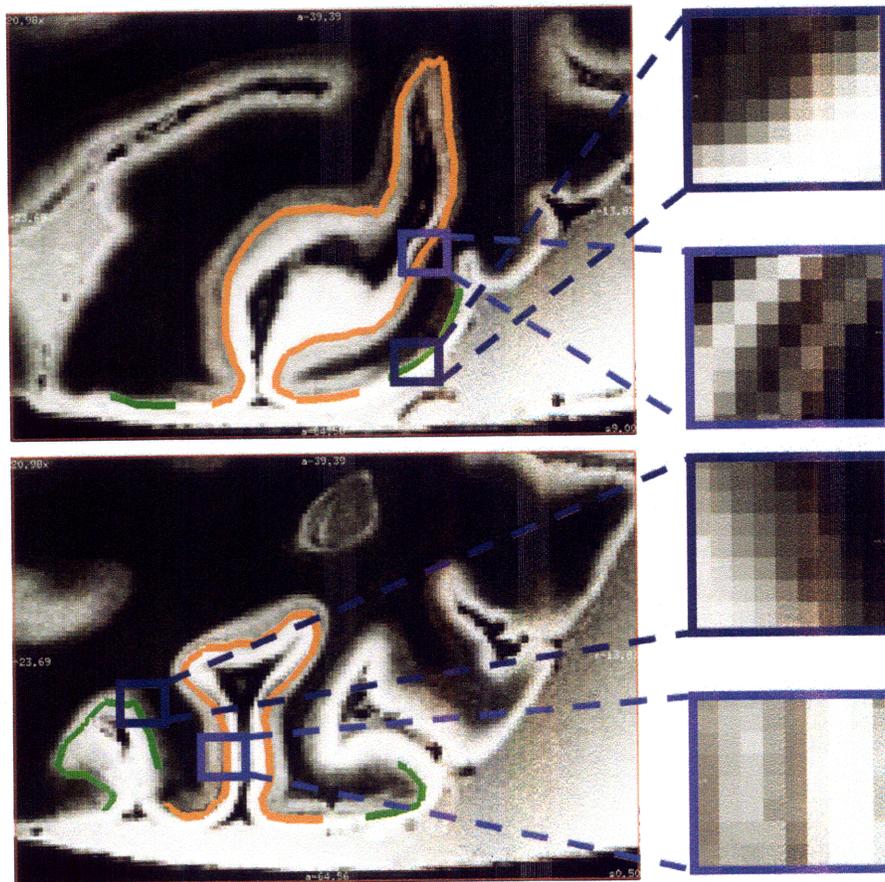
Figure 4-1: (Left) 2D slices through a 3D MR data volume, with an open surface of interest—called Brodmann area 17, and indicated by the *stria* of Gennari—marked in orange. The *non-stria* region just outside the boundary of the relevant open surface is marked in green. (Right) Zoomed-in parts of the stria and non-stria are shown: note the distinctive dark band of the *stria*, seen in the middle of any profile vector centered on the surface and oriented along the normal to the surface.
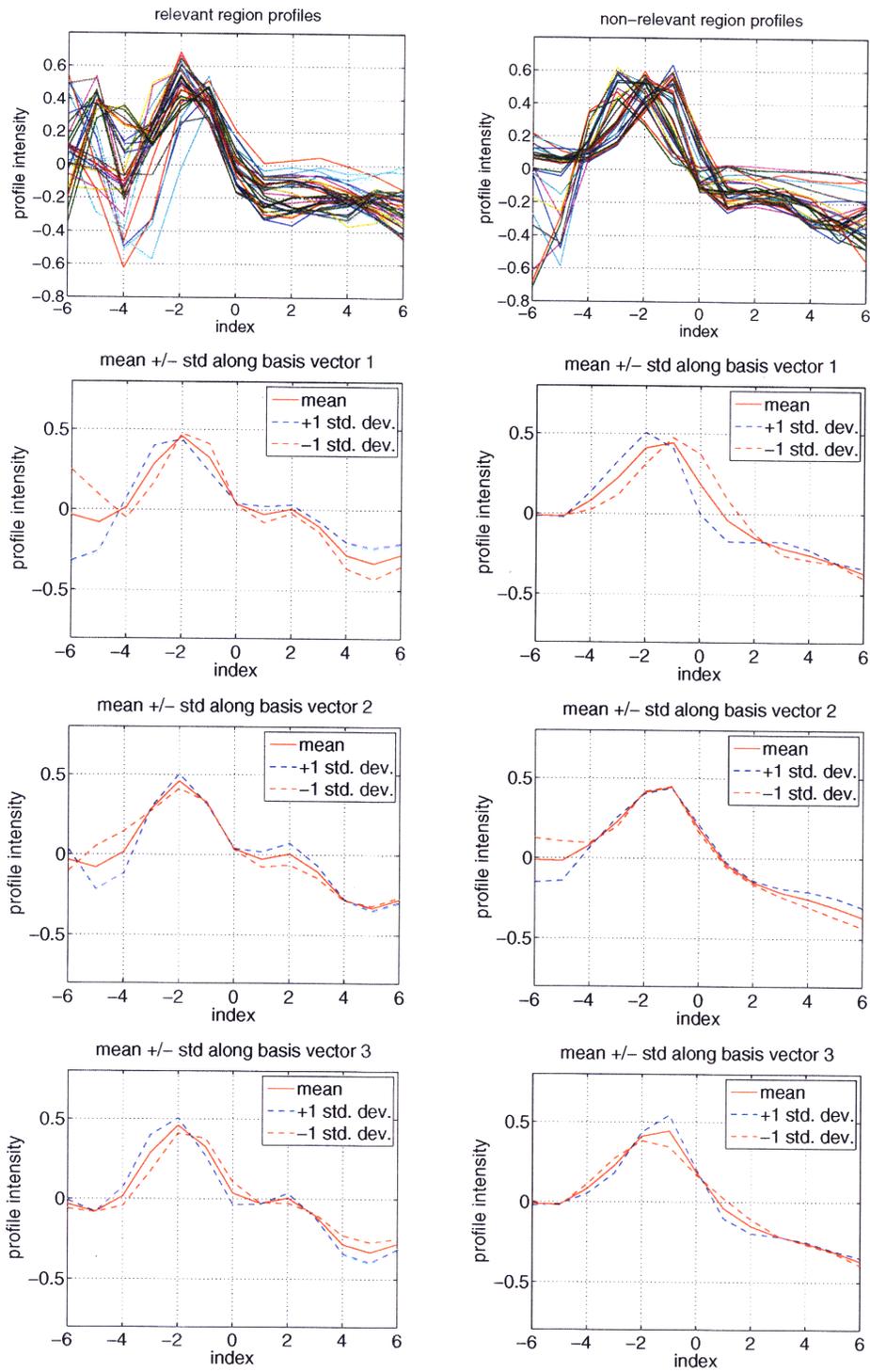
Figure 4-2: Sample normal profile vectors (top row) and PPCA model parameters (mean vector +/- standard-deviation × basis-vector, next 3 rows) for relevant (left) and non-relevant (right) regions of the primary surface in the MR data of Figure 4-1. Each profile vector is centered on the primary surface and is of length 13. The vectors are plotted as functions, with the element indices on the x-axis and the element values on the y-axis.

During the training stage, separate PPCA models, M_1 and M_2, are learnt for profiles from the *relevant* and *non-relevant* regions (as defined in Section 3.1) respectively. Force terms derived from these PPCA models are then used for normal and tangential evolution, as discussed in the next two sections.

Slices from an example data volume are shown in Figure 4-1, with the relevant region of a surface marked in orange. On the right side of the figure are zoomed-in portions of the surface of interest, taken from both the relevant and non-relevant regions. A sample set of normalized profile vectors from these relevant and non-relevant regions is seen in Figure 4-2 (left). The significant parameters of their corresponding PPCA likelihood models—mean and basis vectors—are shown in Figure 4-2 (right). Note that the typical profile obtained from either region varies from high intensity near one end to low intensity near the other. This characteristic overall shape of the profiles is used for localization of the surface during normal evolution. The relevant model M_1 shows an additional dip in intensity in the middle, due to the dark band formed along the open surface of interest. This dip/darkening can be seen both in the zoomed-in surface pictures (Figure 4-1) and estimated PPCA profile models (Figure 4-2). It is this statistical difference between the likelihood models that allows us to discriminate between the two regions during tangential evolution.

4.2 Forces Guiding Normal Evolution

We have two probability distributions, $p(\mathbf{y}|M_1)$ and $p(\mathbf{y}|M_2)$, over profile vectors, corresponding to the relevant and non-relevant regions. For normal evolution, we calculate a mixture distribution $p(\mathbf{y}|M_m)$ from these two, assuming equal prior probability of the two models:

$$M_m = \frac{1}{2}M_1 + \frac{1}{2}M_2. \tag{4.5}$$

Ideally, the prior probability would reflect the relative number of samples in the relevant and non-relevant regions, but in the absence of this knowledge for an arbitrary primary surface, one can work within a min-max framework to obtain the best possible worst-case performance irrespective of the priors [56]. This is equivalent to assuming

equal priors for the two regions.

Given the location x of a voxel on the current primary surface, with profile vector $\mathbf{y}$, we can calculate its likelihood $p(\mathbf{y}|M_m)$ under this mixture model. To estimate the entire surface, we wish to maximize the likelihood of the set of points on the surface taken together. Using an independence assumption across points, and taking the negative logarithm (as discussed in Section 2.2.2), we get the energy function:

$$E_{pl}(S_p) = -\int_{S_p} \log p(\mathbf{y}|M_m)da. \tag{4.6}$$

To estimate the maximum likelihood surface location locally, we should ascend the gradient of this log likelihood function. We calculate the first variation/gradient and use it as the likelihood force term $\mathcal{F}_{pl}$ for evolving the primary surface:

$$\mathcal{F}_{pl} = -\nabla \log p(\mathbf{y}|M_m). \tag{4.7}$$

4.3 Forces Guiding Tangential Evolution

In simple terms, the tangential evolution forces move the boundary of the open surface so as to include voxels whose appearance vector matches the relevant-region likelihood model better than the non-relevant region model, while excluding voxels which do not.

For tangential evolution, we use our two likelihood models in a *region-based evolution* framework [13, 74] to find a curve on a surface. We assume that there are two true underlying regions *on the primary surface*—the true relevant-region R^{+*} and the true non-relevant region R^{-*}—and that the voxels in these regions are independent and identically distributed samples from the likelihood models M_1 and M_2 respectively. Note that the term *region* here refers to a portion of a surface in 3D, instead of the interior area of a 2D curve or interior volume of a 3D surface (as in the standard *region-based* methods [13, 41]).

We now formulate an estimation problem for the boundary C_o of the open surface. We express the posterior probability of C_o given the set $\mathcal{Y}$ of all observed profiles $\mathbf{y}(\mathbf{x})$

from each point **x** on the primary surface as:

$$p(C_o|\mathcal{Y}) \propto p(\mathcal{Y}|C_o)p(C_o). \tag{4.8}$$

The prior $p(C_o)$ was discussed previously in Section 3.4.2 (Equation 3.9). Here we focus on the likelihood $p(\mathcal{Y}|C_o)$. Any given C_o divides the set of points on the primary surface S_p into a relevant region R^+ and a non-relevant region R^-, so that $S_p = R^+ \cup R^-$. We can write the likelihood by dividing the primary surface into the two regions and using the point-wise independence assumption as follows:

$$p(\mathcal{Y}|C_o) = \prod_{\mathbf{x} \in R^+} p(\mathbf{y}(\mathbf{x})|M_1) \prod_{\mathbf{x} \in R^-} p(\mathbf{y}(\mathbf{x})|M_2). \tag{4.9}$$

Taking the negative logarithm to convert to an energy function (as discussed in Section 2.2.2), we get:

$$-\log p(\mathcal{Y}|C_o) = -\int_{R^+} \log p(\mathbf{y}(\mathbf{x})|M_1) da - \int_{R^-} \log p(\mathbf{y}(\mathbf{x})|M_2) da. \tag{4.10}$$

We need to minimize this expression as part of maximizing the posterior probability of the boundary curve. To make the minimization simpler, we add a constant Z that does not affect the location of the optimum:

$$Z = \int_{R^+ \cup R^-} (\log p(\mathbf{y}(\mathbf{x})|M_1) + \log p(\mathbf{y}(\mathbf{x})|M_2)) da. \tag{4.11}$$

Thus, we finally get the likelihood energy $E_l(C_o)$ for the boundary curve:

$$E_l(C_o) = -\log p(\mathcal{Y}|C_o) + Z \tag{4.12}$$

$$= \int_{R^+} \log p(\mathbf{y}|M_2) da + \int_{R^-} \log p(\mathbf{y}|M_1) da. \tag{4.13}$$

Finally we move from optimizing over the curve C_o to optimizing over the entire

secondary surface S_s, using the same energy function. Hence:

$$E_{sl}(S_s) = \int_{R^+} \log p(\mathbf{y}|M_2)da + \int_{R^-} \log p(\mathbf{y}|M_1)da. \qquad (4.14)$$

We choose this form of energy function since it is both directly related to the optimal solution of the underlying classification problem and is easy to optimize. Using the Euler-Lagrange equations for this energy function, as described by Tsai [67], the force field that minimizes the likelihood energy term for tangential evolution is given by a log-posterior-ratio:

$$\mathcal{F}_{sl} = \log \frac{p(\mathbf{y}|M_2)}{p(\mathbf{y}|M_1)} \mathcal{N}_s. \qquad (4.15)$$

This force is only defined at points along the boundary curve C_o on the secondary surface $\mathcal{S}_s$. To evolve the entire secondary surface, we use velocity extension (as discussed in Section 2.2.3), whereby we set the force at each point in the remainder of $\mathcal{S}_s$ as equal to the force at the corresponding closest point on C_o.

4.4 Summary

Having defined the level-set forces $\mathcal{F}_l^p$ and $\mathcal{F}_l^s$, we can plug these into our open-surface evolution algorithm of Section 3.4. The precise form of the PPCA models used in computing these forces—*i.e.*, the size of the profile vectors and the number of basis vectors—will vary by application. We discuss some example applications in the next chapter.

Chapter 5

Open-surface Detection Experiments

In this chapter, we describe experiments where we applied our level-set-based open-surface detection method to three practical applications. One application is for medical imaging, while the other two address problems in 3D seismic imaging. For each case, a different, domain-specific, initialization is used, and the likelihood features are adapted to the data. Given the initial surface, and a suitable training set for the likelihood models, the surface detection algorithm used in each case is the same, as discussed in Chapter 3.

After discussing these applications, we present a performance evaluation of multiple aspects of our algorithm on simulated data. In all our experiments, we used the C++ open-source image processing toolkit called CImg [16] for level-set evolution. The remaining algorithms were in Matlab. Code for probabilistic PCA (PPCA) was taken from the Netlab Toolbox [49].
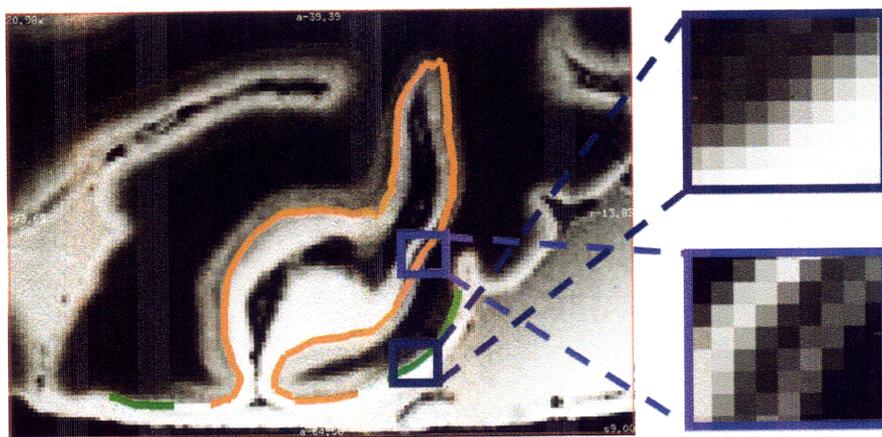
Figure 5-1: (Left) 2D slice through a 3D MR data volume, with the open surface of interest—called Brodmann area 17, and indicated by the *stria* of Gennari—marked in orange. The *non-stria* region just outside the boundary of the relevant open surface is marked in green. (Right) Zoomed-in parts of the stria and non-stria are shown: note the distinctive dark band of the *stria*, seen in the middle of any profile vector centered on the surface and oriented along the normal to the surface.

5.1 Application: Detecting Primary Visual Cortex in MR Data

We detect the primary visual cortex or Brodmann area 17 (BA 17) of the brain, by modeling it as an open surface. The Brodmann areas lie in the grey matter near the surface of the brain. BA 17 is marked by a bundle of myelinated fibers called the *stria of Gennari*, which appear as a dark band in a contrast-inverted magnetic resonance (MR) image.

We ran our surface detection experiments on 250μm isotropic ex-vivo MR brain data obtained using a multiecho FLASH pulse sequence at 7T [4]. At this resolution, the stria appears as a dark band 1-2 voxels thick (see Figure 5-1).

5.1.1 Initialization

For our experiments, we marked an initial open surface mesh $\mathcal{S}_o$ manually about half-way between the gray-white interface—the boundary between the grey matter and the white matter—and the pial surface—the membrane covering the outer brain

surface—with parts of the mesh lying in the relevant region (BA 17) and parts of it outside. The extent of our initial surface was chosen to be the region around a predicted open-surface boundary obtained from mapping a probabilistic atlas of the Brodmann area to this brain [30]. While this does not cover the entire stria region, it is situated at the place where there is most uncertainty about the extent of the Brodmann area (since, in the interior of the relevant region far away from the true boundary, the atlas-based mapping does not affect the extent). Note that such an initialization could potentially be obtained automatically in future, using techniques that localize the grey-white and pial surfaces lying on either surface of the stria (*e.g.*, [29]); however this has not been done at this resolution yet. To obtain the initial primary surface, the open surface was closed using the explicit construction algorithm described in Section 3.2, and the resulting primary surface was converted to an implicit representation following the method given in Appendix A.2.

The accuracy necessary for initialization is described in more detail in Section 5.4.

5.1.2 Training Data

A ground-truth cortical surface for this brain was obtained from a neuro-anatomist. This surface was provided as a triangle mesh consisting of over 3000 manually labelled vertices. The mesh consisted of two disconnected components: one component mesh was marked as stria (*i.e.*, relevant) and the other as non-stria (*i.e.*, non-relevant). Part of the ground-truth mesh was used for training the relevant and non-relevant region likelihood models, while a separate part was used for testing and evaluation.

For the training part of the mesh, 500 normal profile vectors of length $l = 13$ were obtained at the vertices, and PPCA models of dimensionality $k = 3$ for stria and non-stria learnt from them. The localization accuracy of the mesh vertices, as provided by the expert, was +/- 1 voxel (250μm). In comparison, the main discriminating feature between stria and non-stria profile vectors is a dark band only 1-2 voxels thick. Hence, the error in the ground truth would offset the profiles and seriously reduce our accuracy in differentiating stria from non-stria. Note that such misalignment is unavoidable, because of the volume of data needed for precisely marking the cortical
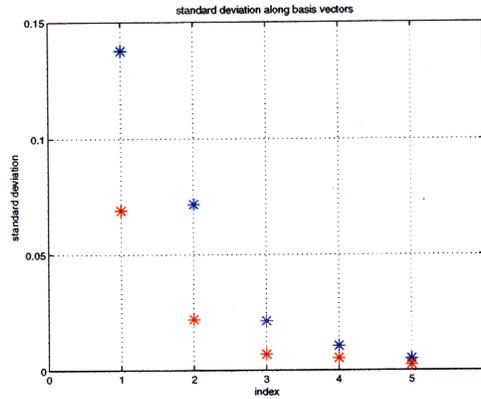
Figure 5-2: Standard deviations along basis vectors in PPCA model. The blue points were noted before realigning the training set of profile vectors, while the red points were noted after.

region, and the tedious nature of the manual labeling task.

To reduce the effect of this error, the training profiles were re-aligned with the learnt PPCA models, by repeatedly shifting (offsetting) each profile vector in the normal direction by a few voxels and calculating the likelihood under the PPCA model for each shift. Each profile vector was then replaced by the shifted version at which maximum likelihood occured, and the PPCA models were re-estimated. The resulting models had significantly lower variances along the low-dimensional basis vectors, as shown in Figure 5-2. This in turn implies that the new likelihoods were higher, since the entropy of the distribution varies with the product of the variances, and a lower entropy distribution has higher likelihoods for its most probable samples.

5.1.3   Open-surface Evolution

In our experiments, normal and tangential evolution were performed alternately in rounds. One round of normal evolution consisted of 30 level-set iterations, while one round of tangential evolution consisted of 50 level-set iterations. The number of iterations per round was chosen based on experiments on a validation set separate from the test set, to ensure that at least 50% of the surface voxels had evolved by at least one voxel before switching evolutions. A round might have had fewer iterations

than the above if the corresponding evolution had converged, in the sense that all voxel updates were less than a motion threshold of 0.02 voxels.

In our result, the RMS distance between the detected BA 17 boundary and ground-truth boundary[1] was about 1.75mm. This corresponded to an average classification rate of 80% for the stria and non-stria regions (*i.e.* 4 out of 5 voxels in these regions were correctly classified). On average, the total distance moved by a point on the BA boundary in both normal and tangential directions, between the initialization and the final result, was 15mm (60 voxels).

Figure 5-3 shows zoomed-in views of sample 2D slices through our 3D results to illustrate one round each of normal evolution and tangential evolution. The images correspond to MR data of size 10mm by 7mm. Only the primary surface, S_p, is shown in each slice (in blue and yellow). To avoid cluttering, the auxiliary surface is not shown, except that it implicitly passes through the boundaries of the visible relevant region.

We note a few salient points. As shown in row (d), if tangential evolution is performed first, without any normal evolution, it converges to a local minimum. In contrast, the result of performing both normal and tangential evolution (row (f)) is much more accurate. Note that even though the initial relevant region (row (c)) overlaps significantly with the ground truth non-stria region, the final relevant region after both normal and tangential evolution (row (f)) does not have any holes in it (*i.e.*, the topology is correct). This is mainly because of strong smoothing term used while evolving the auxiliary surface. The apparent discontinuities in the relevant region shown are due to visualization of a 3D surface in 2D slices. Further, note that after normal evolution (row (e)), parts of the primary surface (in the relevant region) actually move away from the ground truth surface. This is not necessarily an error, since the evolved surface in this region actually matches the typical stria profile better.

Figure 5-4 illustrates the advantage of alternating normal and tangential evolution. The original surface is shown in (a), followed by the result of one round of tangential

---

[1]taken to be the mid-point of the region where the ground-truth labeling was ambiguous.
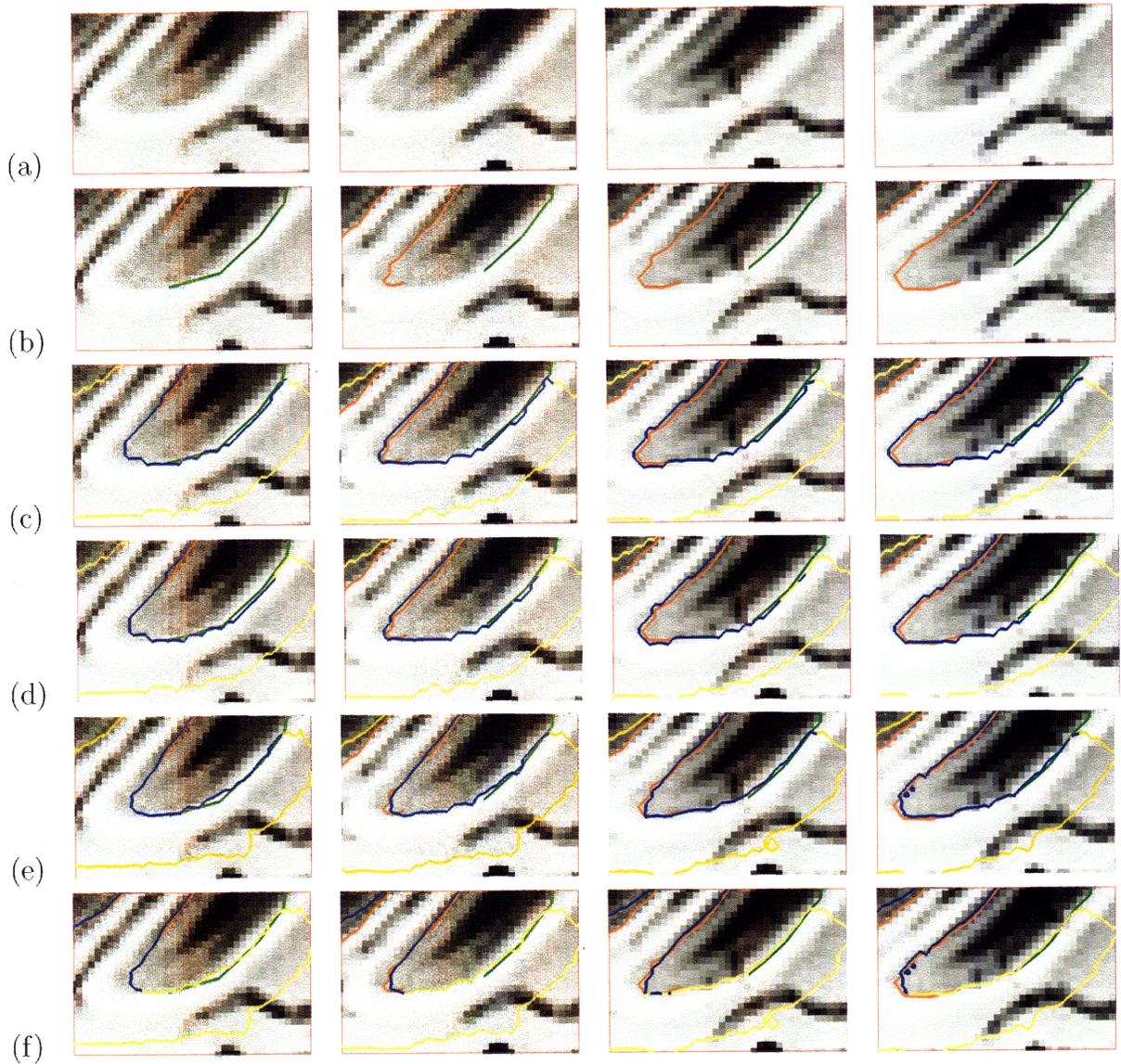
Figure 5-3: Multiple 2D slices through our (3D) results. The 4 columns are 4 different slices through the MR data. The first row simply shows the MR intensities. In row (b), ground truth labeling is shown in orange (striated Brodmann area 17) and green (non-striated region outside BA 17). Results of surface evolution are shown in subsequent rows, in blue (BA 17: relevant region) and yellow (non-relevant). The third row shows the initial surface before normal and tangential evolution. Row (d) shows the (rather poor) result of performing tangential evolution alone, without any normal evolution. Row (e) shows the results of normal evolution alone. Row (f) shows results after one round each of both normal and tangential evolution are performed. Gap between orange and green curves is due to a region where ground truth labeling was ambiguous, as determined by the expert labeller.

(a)	(b)	(c)	(d)
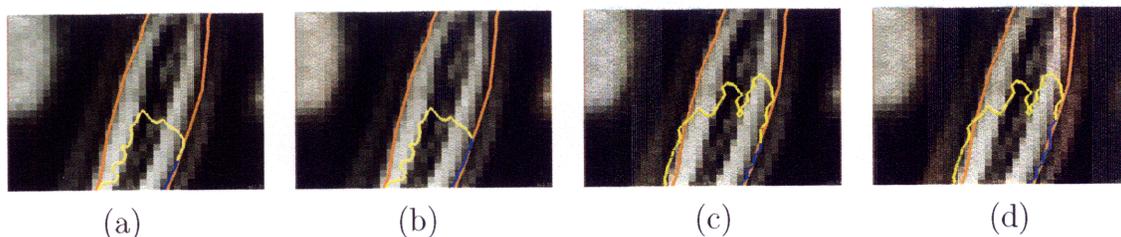
Figure 5-4: A 2D slice through our (3D) results, illustrating the need for alternating normal and tangential evolution. Shown here are (a) an initial configuration, (b) the results of one round of tangential evolution, (c) the next round of normal evolution and (d) the following round of tangential evolution. Color scheme used is the same as in Figure 5-3.

evolution in (b). Tangential evolution converges early in this case, because the original primary surfaces deviates from the cortical surface (due to its arbitrary closure). The next round of normal evolution fits the cortex better, as seen in (c). Finally, a subsequent round of tangential evolution further extends the relevant region, as shown in (d).

5.2 Application: Detecting a Horizon in 3D Seismic Data

Our next application consists of detecting horizon surfaces in 3D seismic-reflection volumes.

5.2.1 Initialization

We asked an expert geophysicist to identify 9 points lying on a portion of a horizon in a seismic volume obtained from the Gulf of Mexico. The expert was presented with 3 vertical slices through the volume, and asked to label 3 points in each slice that lay on or near[2] the horizon. These 9 points were used to create an initial open surface using thin-plate spline interpolation in MATLAB, as shown in Figure 5-5. A smaller

---

[2]*Near* here means within 5 voxels, at a resolution where the typical gap between horizons is about 14 voxels.
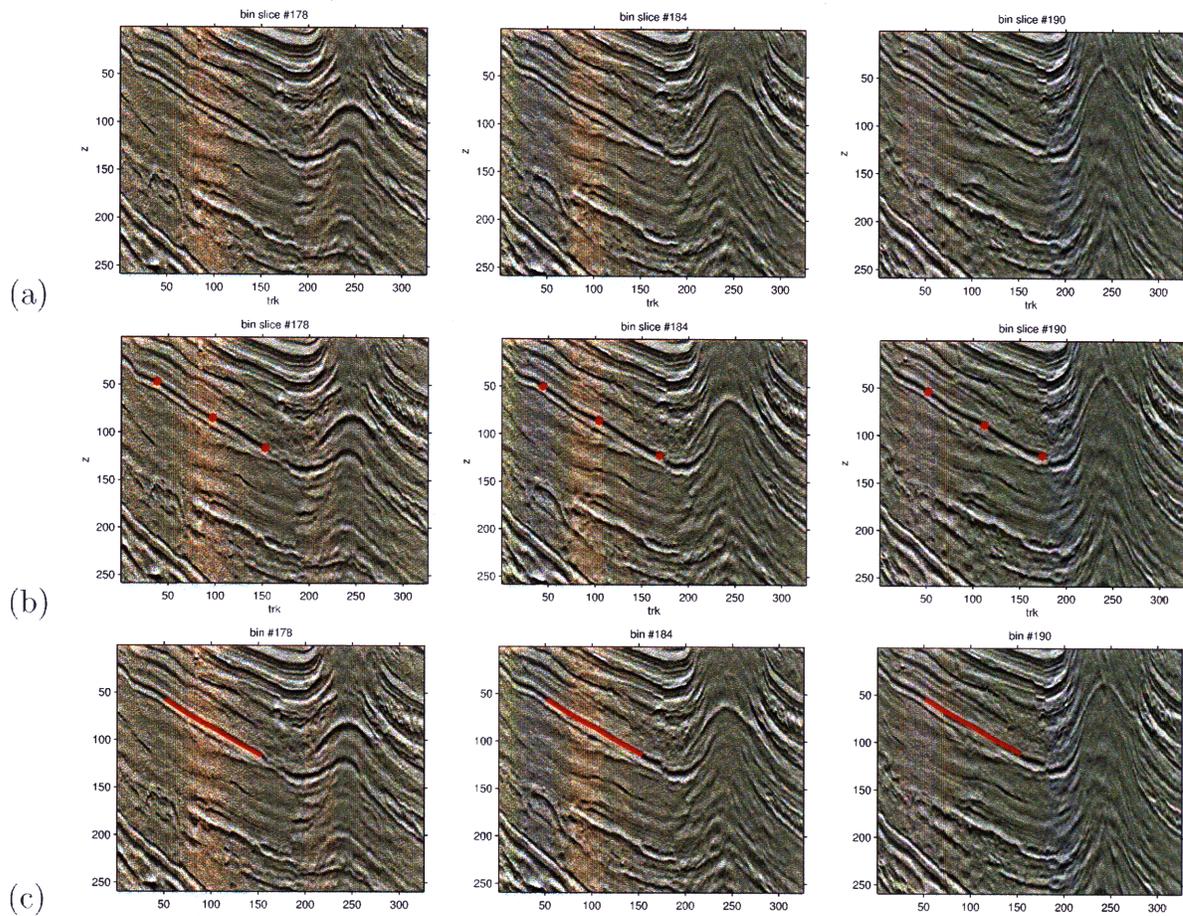
Figure 5-5: Multiple 2D slices showing the initialization used for horizon detection in a seismic volume: (a) three vertical slices through seismic volume (one in each column), (b) expert-labeled points on a horizon in these three slices, and (c) spline-interpolated surface fit to the expert-labeled points.

subvolume was then extracted from the seismic volume (to speed up processing), and the initial surface was closed within this subvolume using the implicit construction algorithm described in Section 3.2. This is illustrated in Figure 5-6(a).

5.2.2 Labeled Data

The seismic data volume consists of multiple horizons stacked one above the other. For training purposes, two such adjacent horizons were manually labeled by a geophysicist, and 500 profile vectors from these horizons were used for training the relevant-region likelihood model. The non-relevant region model was trained on a combination of 200 profiles from random locations in the volume and 200 profiles centered on points lying midway between the two labeled horizons.

In addition, eight horizons (not adjacent to the above two) were also manually labeled, to provide ground truth for testing our algorithm.

5.2.3 Open-surface Evolution

For each horizon surface, we ran our open-surface detection algorithm for 100 iterations of normal evolution, followed by 100 iterations of tangential evolution. As in Section 5.1.3, these numbers were chosen to allow at least 50% of the voxels on each surface to move before switching evolutions. Here, we report average statistics from these runs.

The mean distance between the open-surface boundaries of our result and the ground truth was 11 voxels, with a standard deviation of 4 voxels (where 1 voxel = 15 feet). This is to be compared with the mean distance between the boundaries of our initialization and the ground truth, which was 69 voxels. The improvement in localization was mostly due to tangential evolution, which led to an average cumulative motion of 57 voxels for points on the open-surface boundary. The cumulative motion for points on the primary surface due to normal evolution was, on average, 3 voxels. However, this normal motion was critical to the success of tangential evolution. We recalculated the statistics with tangential evolution alone, and found a mean inter-
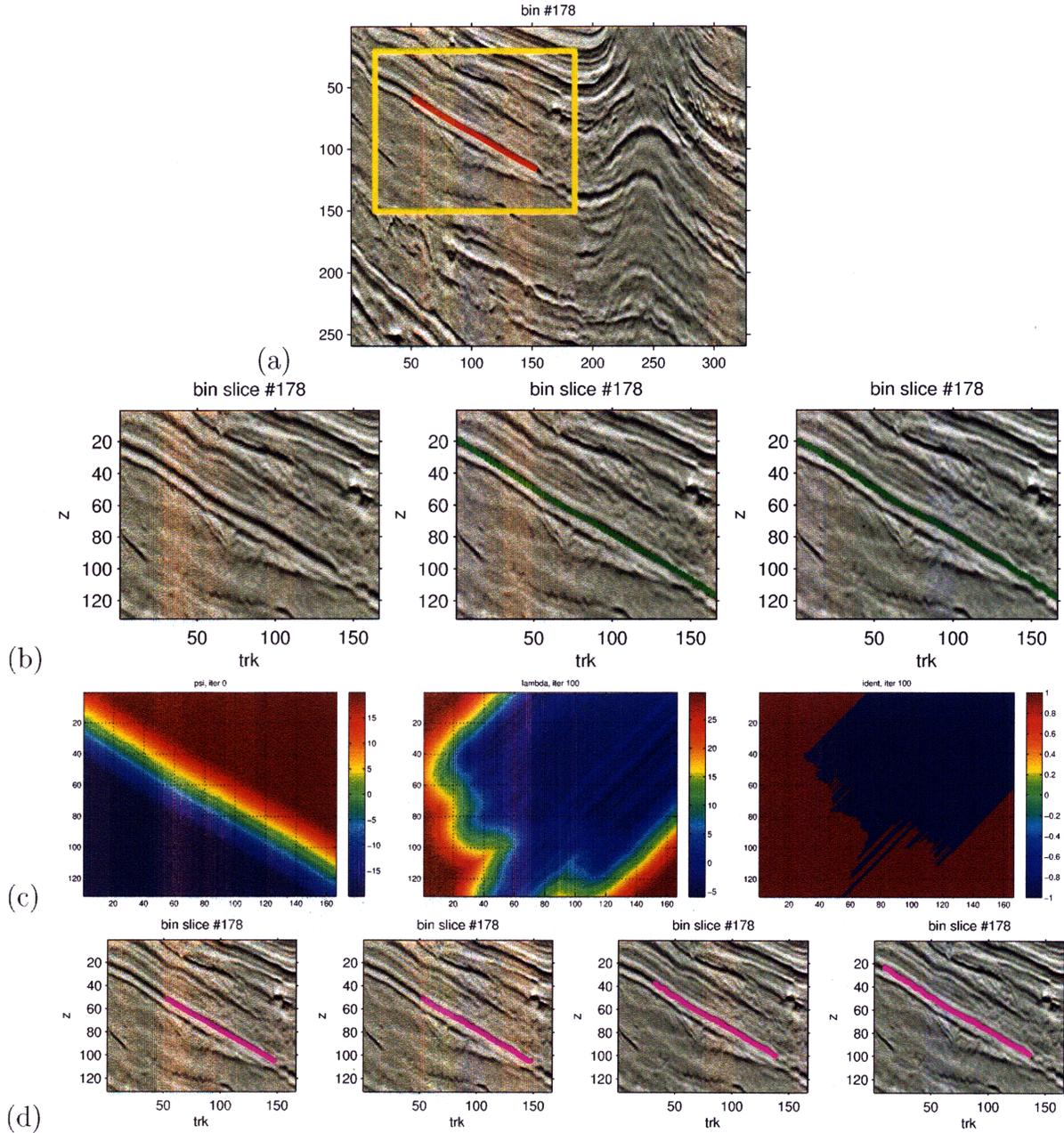
Figure 5-6: We show here a single vertical slice through the seismic volume, illustrating several stages of the surface evolution algorithm: (a) a sub-volume is specified around the initial surface, (b) from left to right, we see the sub-volume slice, the initial primary surface and the primary surface (after normal evolution), (c) from left to right, we see a slice through the primary implicit level-set function, the secondary implicit level-set function and a binarization of the secondary function, (d) the open-surface, from left to right, seen initially, after normal evolution, during tangential evolution, and after tangential evolution.

78

boundary distance of 32 voxels. This is because in the absence of normal evolution, the profiles used in tangential evolution are not well-aligned, *i.e.*, they are not all centered on the true horizon. Consequently, more of the relevant-region profiles get misclassified as non-relevant, and vice versa.

We note that discussion with an expert in geophysics indicates there is some uncertainty in the ground truth labeling provided to us, of the order of 1 voxel in location and 3 voxels in extent.

Figures 5-5 and 5-6 show slices from a sample run, at various stages of the algorithm.

5.3 Application: Detecting a Fault in Seismic Data

Our third application aims at detecting geologic faults in 3D seismic data. It is a little more ambitious than the previous two applications, since fault surfaces are known within the geophysical community to be notoriously difficult to detect [32].

5.3.1 Fault-surface Appearance Model

Evidence for faults shows up in seismic volumes as discontinuities in horizon layers. These discontinuities are easily identified by an observer, since to the human eye, the horizons seem to *pop out* [63] from the seismic volume after a few seconds of observation (see Figure 5-7(a) for example, where many such fault-lines can be seen in each 2D slice). However, for an automatic detection technique, this would require detecting horizons first, which is in itself non-trivial. Further, there is little evidence for faults between the sharply-defined horizons, or in regions where the displaced horizons happen to match in intensity with each other. Hence, even though prior knowledge indicates that major faults tend to be continuous and typically extend beyond 2-3 horizons, it is not easy to detect them using an automated image-processing technique.

Geophysicists often use attribute volumes calculated from a seismic volume to further *highlight* faults. We follow this idea, making use of a pre-computed attribute

79

volume called the *dissemblance* volume as the data for our open-surface detection task. For each voxel of the original volume, the dissemblance volume is the negative of the local correlation of vertical profile vectors centered at that voxel and at its 4 neighboring voxels in a horizontal plane. Horizons show high correlation laterally, and hence low dissemblance. The highest dissemblance is expected along faults, or at randomly occuring points within a heterogeneous rock structure (such as a salt body). Sample seismic volume slices, and the corresponding dissemblance slices, are shown in Figure 5-7(a,b).

To account for the lack of evidence for faults in the gaps between visible horizons, we changed the likelihood feature used for this task. Instead of considering only a single normal profile vector $\mathbf{y}(\mathbf{x})$ at each surface point $\mathbf{y}$, we consider a *set* of profile vectors computed at neighboring surface points in each direction within a radius r_f. We then find the n_f profiles $\mathbf{y}(\mathbf{x_1}), \mathbf{y}(\mathbf{x_2}), \ldots, \mathbf{y}(\mathbf{x_{n_f}})$ with highest likelihood within this set, and use the average profile $\bar{\mathbf{y}}$ as the vector representing this point:

$$\bar{\mathbf{y}} = \frac{1}{n_f} \sum_{i=1}^{n_f} \mathbf{y}(\mathbf{x_i}). \tag{5.1}$$

By selecting the best profiles from a set, we essentially make the appearance likelihood more *robust* to gaps by averaging within a window. Of course, there is a trade-off, since the larger the neighborhood radius r_f, the greater the robustness on the one hand, but the lower the accuracy of localization of the surface boundary and the longer the computation time on the other. In our implementation, we chose $r_f = 15$ and $n_f = 10$, after experimenting with two labeled fault surfaces and making sure tangential evolution did not stop too early as the initial open-surface grew in size. The results were quite sensitive to the value of r_f, as, for instance, $r_f = 10$ or less almost always failed to grow the fault surface adequately. We therefore recommend choosing r_f appropriately for each seismic volume being considered, though the same value can be typically be used for all faults in the volume.
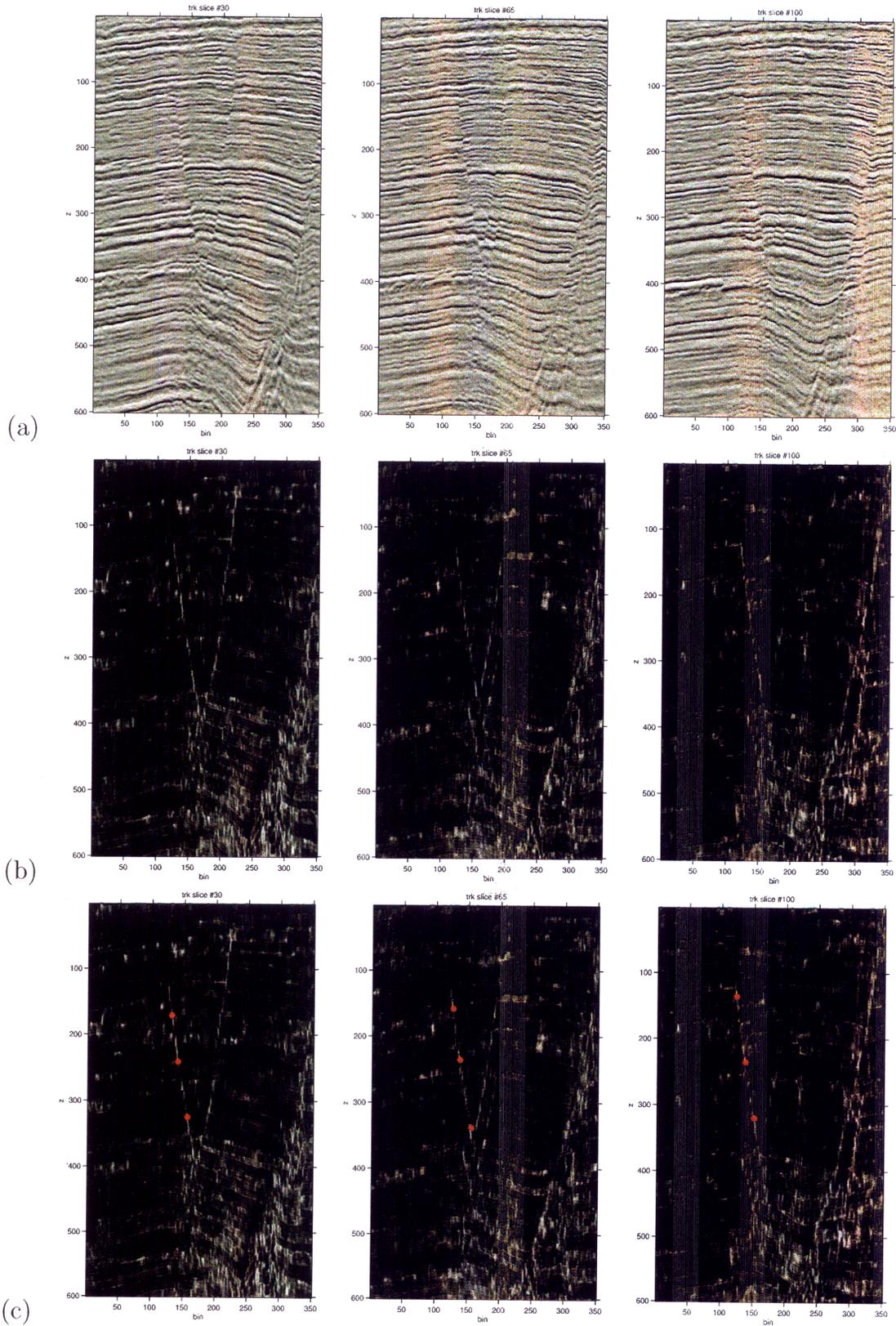
Figure 5-7: Multiple 2D slices showing the initialization used for fault detection in a seismic/dissemblance volume. Seen here are (a) vertical slices from the seismic volume, (b) corresponding slices from the dissemblance volume and (c) expert labeling of a single fault.

5.3.2 Initialization

As with the horizons, for each fault in our test set, an expert geophysicist identified 9 points lying on a portion of a the fault in a dissemblance volume. The expert was presented with three vertical slices through the dissemblance volume and the corresponding seismic volume, and asked to click on three points in each slice that lay on or near the fault (see Figure 5-7). These nine points were used to create a smooth initial open surface using thin-plate spline interpolation [68] in MATLAB with smoothing parameter value 0.5.

5.3.3 Labeled Data

We experimented with two seismic data volumes from the Gulf of Mexico. Each seismic volume consisted of multiple faults with different orientations and extents. A volume 1000 voxels deep, 300 samples long and 300 samples wide contained about 10 large faults (more than 200 voxels in height) and 50 small faults. For training purposes, 4 large faults in one volume were manually labeled by a geophysicist, and 500 normal profile vectors from these fault meshes, calculated using Equation 5.1, were used for training the relevant-region PPCA likelihood model. The non-relevant region model was trained on 300 profiles from random locations in the volume and 300 profiles centered on points obtained by extrapolating the labeled faults beyond their true extent. As with the MR data (Section 5.1.2), the fault models were re-estimated after aligning the profile samples with the initially-learned model.

In addition, 7 other large faults from the two volumes were manually labeled, to provide ground truth for testing, and two more faults were used for parameter selection.

5.3.4 Open-surface Evolution

For each fault in the test set, we ran our open-surface detection algorithm for 100 iterations of normal evolution, followed by 100 iterations of tangential evolution. The number of iterations was chosen in the same way as for horizons. The mean distance
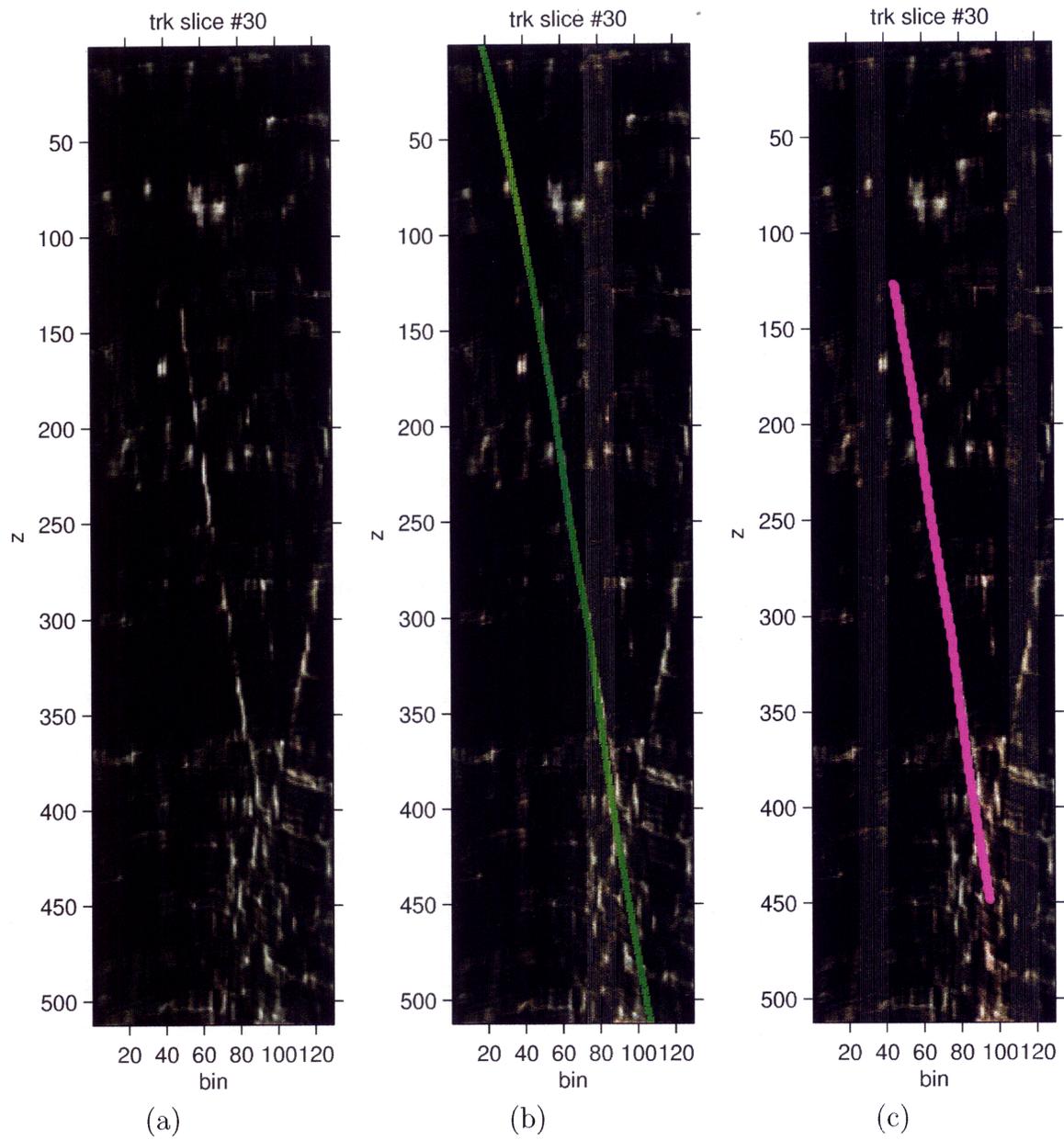
Figure 5-8: Dissemblance volume (a), primary surface (b), and final open-surface result (c) for fault detection, all shown in a 2D slice through the 3D volume.

between the open-surface boundaries of our result and the ground truth fault surface was 44 voxels, with a standard deviation of 20 voxels (where 1 voxel = 15 feet). This is to be compared with the mean distance between the boundaries of our initialization and the ground truth, which was 105 voxels. Also, to put this in context, the average linear extent of a fault was 550 voxels. As determined by a geophysical expert, there is some error even in the manual labeling provided to us as ground truth, of the order of 8-10 voxels.

As with horizon detection, the improvement in localization was mostly due to tangential evolution, but once again, the small amount of normal motion played a crucial role in the success of tangential evolution. With tangential evolution alone, the re-computed statistics indicated a mean inter-boundary distance of 60 voxels.

The accuracy of fault detection is considerably lower than that of horizon detection mostly because there is less appearance information characterizing faults. For instance, the transition from a horizon region to a non-horizon region is marked by a distinct change in profile vectors, while faults die away much more gradually. Another factor affecting the accuracy of fault detection is the neighborhood size we used for the sake of robustness in computing appearance vectors on the fault surface (as discussed at the beginning of this section). A third reason is that the signal-to-noise ratio of the seismic image gets steadily poorer as we go down in the volume, so the bottom of a fault is very hard to detect.

Nonetheless, our results are likely useful to the practicing geophysicist, since they can easily be augmented by some manual interaction. Our method saves the geophysicist's time, by allowing him to concentrate on the regions of the fault that most need his expertise to detect. Based on the typical resolution at which a geophysicist labels faults in a volume for initial inspection—one sample every 50 voxels or so in either direction along a fault surface—we estimate a 75% reduction in the number of samples that need to be manually marked.

Figures 5-7 and 5-8 show slices from a sample run, at various stages of the algorithm.

(a)   (b)
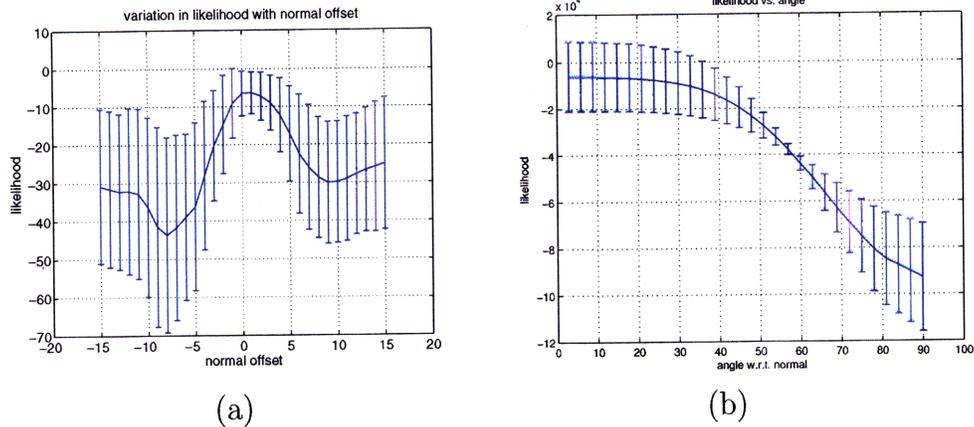
Figure 5-9: Plots showing variation of profile likelihood with (a) translation and (b) rotation. For these data, the plot on the left shows the learned appearance model localizes the target open surface, on average, from an initialization within $+/-$ 8 voxels in the normal direction, and rotated by up to 40 degrees.

5.4 Performance Analysis

In this section, we characterize the performance of our open-surface detection algorithm using simulated data and a variety of metrics.

5.4.1 Likelihood Variation with Translation

The success of normal evolution depends on how the likelihood-based force terms vary with translation in the normal direction. Since normal evolution essentially performs gradient ascent, it finds a local optimum in surface location. This implies that the maximum distance moved in the normal direction, on average, is the distance to the nearest local minimum in likelihood.

We simulated multiple sample profiles at different normal offsets from a PPCA model learnt for the *stria* in the medical-imaging application. In Figure 5-9(a), we show the average likelihood as a function of offset, for mis-aligned profiles that have been translated along the normal direction. There is a local minimum, on average, at a distance of 4 voxels from the true surface location. This gives us an indication of what the typical accuracy of the initialization must be in the normal direction for our method to work. An initial surface whose closest point is more than 4 voxels

(a)

(b)

(c)
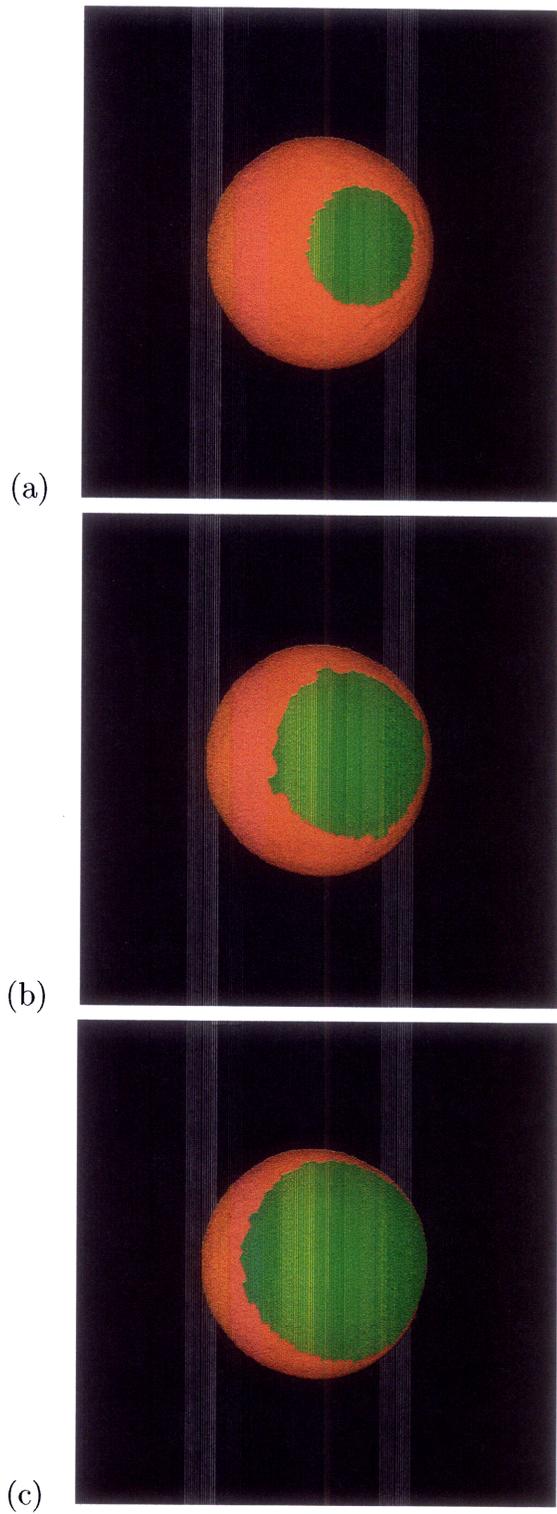
Figure 5-10: Stages in the evolution of an open-surface on a simulated sphere: (a) initialization, (b) after 25 iterations of tangential evolution, and (c) final state, after 75 iterations.

away from the target along the normal direction is likely not going to converge to target surface, due to the nature of the local optimization used. On the other hand, if a majority of surface points on the initial surface are within 4 voxels, as measured along the normal direction, convergence is very likely. The smoothness term plays an important role here, as it allows individual points to move past local minima in appearance if that is the direction in which neighboring points are moving.

For our applications, this issue has not been a major constraint. With the medical data, the outer surfaces of the cortical ribbon—the pial surface and the grey-white interface—are localizable by other methods, and lie 8-12 voxels apart. The stria of Gennari typically lies halfway between these two cortical surfaces. For horizon detection, automated trackers exist that scan the seismic volume to directly detect points on a horizon and then follow the horizon by correlation tracking to provide an initial surface. Fault detection, on the other hand, typically requires some manual supervision anyway, so the accuracy constraint described here is unlikely to be a limitation.

Another feature to note from the plot in Figure 5-9(a) is the width of the peak, as a sharper peak implies better localization. In this case, an offset of $+/-1$ voxel is possible, as the likelihood drop in this range may be overcome by the decrease in smoothness penalty.

5.4.2   Likelihood Variation with Rotation

Normal evolution not only translates the open surface, but also rotates it. It achieves this by varying the magnitude of the normal force as one moves along the surface. Hence, it is worth looking at the variation of likelihood-based force terms with rotation of the normal profile relative to the ground-truth surface.

We simulated multiple sample profiles at different angles relative to a ground-truth surface from a PPCA model learnt from seismic data. Figure 5-9(b) plots the variation of average likelihood with angular offset between 0 and 90 degrees. The likelihood falls off rapidly beyond 40 degrees, and is an order of magnitude lower at 90 degrees. This gives us a sense of how well-aligned the initial surface needs to be

for the evolution to be successful.

Given the constraint on translation of the surface in the normal direction, as described in the previous sub-section, we do not want the initial open surface to be significantly rotated relative to the ground truth surface. In the examples we presented, it is relatively easy to come up with an automatic initialization where the relative rotation of the open surface and the ground truth is small. In the medical imaging case, such an initialization could be obtained by registering the volume with another pre-labeled volume [30]. For the seismic data, a well-aligned initial open surface for horizons can be obtained by edge detection (in case of horizons) or by plane-fitting (in case of faults). However, large relative rotation is possible for the remaining, non-relevant part of the primary surface, especially if the explicit construction algorithm of Section 3.2 is used for closing the open surface. For this reason, we prefer to use the implicit construction for creating the primary surface.

5.4.3 Testing Tangential Evolution

We tested the accuracy of tangential evolution on two simulated datasets.

The first dataset consists of circular patch on a spherical primary surface. The initial patch corresponds to a cone centered at the center of the sphere with half-angle 30 degrees. The ground truth patch is for a similar cone with half-angle 60 degrees. Three stages in the process are shown in Figure 5-10: in this case, the single initial surface patch successfully grows outwards on the sphere to match the ground truth.

The second dataset is similar, but the primary surface is a box instead of a sphere, with the patch located on one side of it.

For both datasets, we tested two things:

- the sensitivity of the result to the difference in likelihood between the relevant and non-relevant regions, and

- the sensitivity of the result to the topology of the initial patch.

Sensitivity to likelihood difference. We varied the PPCA models of the two regions (relevant and non-relevant) and studied its effect on the accuracy of

the resulting open surface. The PPCA models for the profiles in the two regions corresponded to spherical Gaussians with identical covariance matrices. Starting from parameters for which the evolution occured successfully, two kinds of variation were employed: reducing the distance between the means, and increasing the variance. Both of these variations increase the similarity between the relevant and non-relevant region likelihood models. In fact, one type of variation can be converted to the other by shifting and scaling the data. Multiple iterations were run with samples from these PPCA models.

The evolution showed two different behaviors: it either completed successfully with the initial patch growing to match the true patch, or it failed completely with the initial patch shrinking and disappearing. As both types of variation were applied, the time taken for successful evolution increased, till the evolution started breaking down, after which the time to breakdown decreased. This is because the likelihood force term steadily decreased, till a point where it was dominated by the smoothness prior term. Since the smoothness prior favors shorter curve lengths for the boundary of the open surface, it shrinks the open-surface and makes it disappear.

The results were correct in over 95 percent of cases when the likelihood term was at least twice as large as the smoothness term. We used this as a benchmark in choosing the relative weights of likelihood and smoothness forces for the practical applications described in the previous sections of this chapter.

Sensitivity to topology. We varied the topology of the initialization by varying the secondary surface, with the spherical primary kept fixed and the two likelihood models sufficiently distinct. We considered 4 initial cases in which to test the behavior of the algorithm:

1. Single circular patch in middle of true relevant region: this case works as expected, as shown in Figure 5-10.

2. Two circular patches within true relevant region: the two patches grow and merge to give the same result as case 1.

3. Single circular patch completely outside true relevant region: the patch shrinks

till it disappears, leading to an incorrect result.

4. Two circular patches with one on either side of true relevant-region boundary: the interior patch grows to match the true open-surface, while the exterior patch shrinks and disappears.

In each case, the algorithm behaved as expected. The failure in case 3 is due to the widely-understood limitation of the local optimization framework used in most level-set methods.

Through the above performance tests, we have established the range of initial conditions for which our open-surface evolution algorithm is likely to work well in our chosen applications. Given the significant differences between our three applications, we also expect the results to be representative of other open-surface detection tasks.

Chapter 6

Conclusion

We have proposed a novel open-surface evolution algorithm that evolves a surface in both the normal and tangential directions in 3D. The evolution is done in a level-set framework, by alternately updating two implicit functions.

We have also proposed a principled statistical model for modelling structured appearance of surfaces along the normal direction. This model fits naturally within the level-set framework, and generalizes the traditional edge-based and region-based appearance models. For normal evolution, a mixture of relevant and non-relevant profiles is used for learning an appearance model, while for tangential evolution, a region-based likelihood term is used to solve the classification problem of relevant versus non-relevant region.

The method has been applied to a medical imaging problem, the automatic detection of the primary visual cortex (Brodmann area 17), and two seismic imaging problems, horizon detection and fault detection. It looks particularly promising for detection of the extent of these open surfaces. This opens up possibilities in these respective fields, including automatic analysis of cortical structure directly from high resolution MR data—possibly even acquired invivo—without requiring histological samples, and rapid assisted hydrocarbon-basin interpretation in the petroleum industry. We believe our formulation can also be applied for locating other cortical and sub-cortical regions.

The primary advantage of our proposed algorithm is that it provides an integrated

solution for detecting open surfaces in 3D intensity volumes. The use of profile vectors for the appearance likelihood makes the method applicable to a broad class of interface detection problems, including closed surface detection problems. For open surfaces, our process for constructing the secondary surface from the primary surface (using the orthogonality constraint) implies that, in contrast with previously published methods, the derivation of forces is significantly simplified and spurious intersections between the primary and secondary surfaces are automatically avoided for many surfaces of interest.

There are a number of possible directions in which the present research could be extended. These include incorporating a global shape prior on the open surface, improving the appearance model of the non-relevant region, modeling variation in appearance, geometry and scale (*i.e.*, length) of profile vectors and exploring alternative smoothing forces. We discuss a couple of these issues here.

Global statistical priors on shape have previously been proposed for closed surfaces within the level-set framework (see for example, [43]), and we believe these can be transferred to open surfaces. Such priors would be useful for avoiding many of the local minima currently present during our first normal evolution task. The priors are particularly meaningful for fault surfaces, which tend to be very smooth, and cortical surface regions, which have reasonably consistent geometric folding patterns.

The constant length of the profile vectors along the surface makes appearance modeling simple—PPCA can be used directly, for example—but may not correctly model the physical process behind the generation of the surface's layered appearance. In the medical example, for instance, cortical thickness varies as one moves along the folds, and the corresponding profile of intensity variation expands or contracts in scale. Further, the bending of surface suggests that curved profiles might be more appropriate than straight profiles. In other words, as we step away from the surface along the normal vector to get the profile values, the normal vector might change in direction continually. Finally, the appearance of the profile vectors may show structured variation from one part of the surface to another, which is not captured by the current assumption that the vectors are independently and identically distributed

under a PPCA model.

We believe we have explored a problem of growing practical importance, proposed a novel approach to solving the problem based on a principled theoretical framework, and demonstrated interesting results on two very different applications. We hope and expect to see much further work on the topic of detecting open surfaces in the years to come.

THIS PAGE INTENTIONALLY LEFT BLANK

Appendix A

Switching Between Implicit and Explicit Representations

We summarize here methods for converting an implicit surface representation to an explicit one and vice versa. Even though most of work involves manipulation of implicit representations, we may be provided an input that is represented explicitly, or may require an output in an explicit form. To handle these situations, the following two types of methods can be used.

A.1   Implicit to Explicit

For going from an implicit representation to an explicit one, we first discuss the case of a closed surface. The conversion for an open surface is a simple extension.

We are given an embedding function in $\mathbb{R}^3$, and we wish to obtain its zero level-set as a triangle mesh. The standard algorithm for solving this problem is called *marching cubes*, and was proposed by Lorensen and Cline [46] in the context of computer graphics.

A.2   Explicit to Implicit

We are given a closed surface as a triangle mesh, and we wish to obtain an embedding function whose zero level-set matches this mesh. While there are many possible embedding functions that can be constructed to match a mesh, a very common choice is to create a signed-distance function from the mesh, as discussed in Section 2.2.3. A simple algorithm for doing this consists of two tasks:

1. binarizing the volume, to mark the interior and exterior of the mesh, and,

2. solving an eikonal equation on the binarized volume to get the distance function from the triangle mesh.

Binarizing a volume. Having chosen a bounding volume around the triangle mesh, we run a ray-intersection algorithm along one of the axes of the volume. From one of the six surfaces of the bounding volume, parallel rays centered at the voxels on the surface are sent into the volume in a direction normal to the surface. For each ray, we record all points at which it intersects a triangle of the triangle mesh. After calculating all ray-triangle intersections, we step along each ray, marking voxels as -1 or $+1$ to indicate the interior or exterior of the mesh. Each ray starts with values of $+1$, and the sign of the value changes every time we cross a recorded intersection point.

Creating the distance function for the binarized volume. We first identify the points that lie closest to the surface, namely the points which have a binary value different from one of their neighbors. We set the magnitude of the signed-distance function at these points to be 0.5 voxels, and the sign to match the sign of the binary volume. Note that this is only accurate to $+/-0.5$ voxels. Next, we use these points as the initial condition for solving the eikonal equation in the entire volume, with the help of a fast-marching method.

This gives us the required embedding function for a closed surface mesh. For an open surface mesh, we have two choices:

1. use the explicit construction of Section 3.2.1 to obtain a closed surface mesh, and then use the above algorithm, or

2. use the implicit construction of Section 3.2.2 to directly obtain an implicit representation of a corresponding closed surface.

In either case, the implicit closed surface so obtained is the primary surface. We then need to construct the secondary surface using the method described in Section 3.3. These two closed surfaces together provide an implicit representation of the open surface.

THIS PAGE INTENTIONALLY LEFT BLANK

Bibliography

[1] D. Adalsteinsson and J. Sethian. The fast construction of extension velocities in level set methods. *Journal of Computational Physics*, 148:2–22, 1999.

[2] L. Ambrosio and H. Soner. Level set approach to mean curvature flow in arbitrary codimension. *Journal of Differential Geometry*, 43:693–737, 1996.

[3] J. Annese, A. Pitiot, I. Dinov, and A. Toga. A myelo-architectonic method for the structural classification of cortical areas. *Neuroimage*, 21(1):15–26, 2004.

[4] J. Augustinack, A. van der Kouwe, M. Blackwell, D. Salat, C. Wiggins, M. Frosch, G. Wiggins, A. Potthast, L. Wald, and B. Fischl. Detection of entorhinal layer II using 7 Tesla magnetic resonance imaging. *Annals of Neurology*, 57(4):489–494, 2005.

[5] E. Barbier, S. Marrett, A. Danek, A. Vortmeyer, P. van Gelderen, J. Duyn, P. Bandettini, J. Grafman, and A. Koretsky. Imaging cortical anatomy by high-resolution MR at 3.0T: Detection of the stripe of Gennari in visual area 17. *Magnetic Resonance in Medicine*, 48(4):735–738, 2002.

[6] A. Bartesaghi and G. Sapiro. A system for the generation of curves on 3d brain images. *Human Brain Mapping*, 14(1):1–15, 2001.

[7] M. Bertalmio, G. Sapiro, and G. Randall. Region tracking on level-sets methods. *IEEE Transactions on Medical Imaging*, 18(5):448–451, May 1999.

[8] K. Brodmann. Vergleichende Lokalisationslehre der Großhirnrinde in ihren Prinzipien dargestellt auf Grund des Zellenbaues, 1909.

[9] P. Burchard, L.-T. Cheng, B. Merriman, and S. Osher. Motion of curves in three spatial dimensions using a level set approach. *Journal of Computational Physics*, 170(2):720–741, 2001.

[10] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.

[11] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *International Journal of Computer Vision*, 22(1):61–79, 1997.

[12] A. Chakraborty, L. Staib, and J. Duncan. Deformable boundary finding in medical images by integrating gradient and region information. *IEEE Transactions on Medical Imaging*, 15(6):859–870, 1996.

[13] T. Chan and L. Vese. Active contours without edges. *IEEE Transactions on Image Processing*, 10(2):266–277, 2001.

[14] L-T. Cheng, P. Burchard, B. Merriman, and S. Osher. Motion of curves constrained on surfaces using a level-set approach. *Journal of Computational Physics*, 175(2):604–644, 2002.

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

[16] Cimg library. http://cimg.sourceforge.net/.

[17] S. Clare and H. Bridge. Methodological issues relating to in vivo cortical myelography using mri. *Human Brain Mapping*, 26(4):240–250, 2005.

[18] I. Cohen, N. Coult, and A. Vassiliou. Detection and extraction of fault surfaces in 3d seismic data. *Geophysics*, 71(4):21–27, 2006.

[19] L. Cohen. On active contour models and balloons. *Computer Vision, Graphics and Image Processing: Image Understanding*, 53:211–218, 1991.

[20] C. Conaway. *The Petroleum Industry: A Nontechnical Guide*. PennWell Publishing, 1999.

[21] T. Cootes, G. Edwards, and C. Taylor. Active appearance models. In *Proceedings of the International Conference on Information Processing in Medical Imaging*, 1998.

[22] T. Cootes, A. Hill, C. Taylor, and J. Haslam. The use of active shape models for locating structures in medical images. 1993.

[23] T. Cootes and C. Taylor. Active shape models - 'smart snakes'. In *Proceedings of the British Machine Vision Conference*, 1992.

[24] T. Cootes, C. Taylor, D. Cooper, and J. Graham. Active shape models—their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, 1995.

[25] R. Courant, K. Friedrichs, and H. Lewy. Uber die partiellen differenzengleichungen der mathematischen physik. *Mathematische Annalen*, 100(1):32–74, 1928.

[26] A. Dale, B. Fischl, and M. Sereno. Cortical surface-based analysis: I. segmentation and surface reconstruction. *Neuroimage*, 9(2):179–194, 1999.

[27] G. Dorn. Modern 3-d seismic interpretation. *The Leading Edge*, 17(9):1262–1273, 1998.

[28] P. Felzenszwalb. Representation and detection of deformable shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(2):208–220, 2005.

[29] B. Fischl and A. Dale. Measuring the thickness of the human cerebral cortex from magnetic resonance images. *Proceedings of the NAS*, 97(20):11050–11055, 2000.

[30] B. Fischl, N. Rajendran, E. Busa, J. Augustinack, O. Hinds, B. Yeo, H. Mohlberg, K. Amunts, and K. Zilles. Cortical folding patterns and predicting cytoarchitecture. *Cerebral Cortex*, 18(8):1973–1980, 2008.

[31] J. Foley, A. van Dam, S. Feiner, and J. Hughes. *Computer Graphics*. Addison-Wesley, second edition, 1995.

[32] D. Gibson, M. Spann, and J. Turner. Automatic fault detection for 3d seismic data. In *Digital Image Computing : Techniques and Applications (DICTA)*, 2003.

[33] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[34] M. Hofer and H. Pottmann. Energy-minimizing splines in manifolds. *ACM Transactions on Graphics*, 23(3):284–293, 2004.

[35] B. Horn. *Robot Vision*. MIT Press, 1986.

[36] W.-K. Jeong, R. Whitaker, and M. Dobin. Interactive 3d seismic fault detection on the graphics hardware. In *International Workshop on Volume Graphics*, 2006.

[37] H. Jin, A. Yezzi, and S. Soatto. Region-based segmentation on evolving surfaces. In *Proceedings of the International Conference on Information Processing in Medical Imaging*, 2004.

[38] I. Jolliffe. *Principal Component Analysis*. Springer-Verlag, 1986.

[39] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, January 1988.

[40] S. Kichenassamy, A. Kumar, P. Olver, A. Tannenbaum, and A. Yezzi. Gradient flows and geometric active contour models. In *Proceedings of the International Conference on Computer Vision*, 1995.

[41] J. Kim, J. Fisher, A. Yezzi, M. Cetin, and A. Willsky. A nonparametric statistical method for image segmentation using information theory and curve evolution. *IEEE Transactions on Image Processing*, 14(10):1486–1502, 2005.

[42] S. Lankton and A. Tannenbaum. Localizing region-based active contours. *IEEE Transactions on Image Processing*, 17(11):2029–2039, 2008.

[43] M. Leventon, E. Grimson, and O. Faugeras. Statistical shape influence in geodesic active contours. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2000.

[44] C. Li, C.-Y. Kao, J. Gore, and Z. Ding. Implicit active contours driven by local binary fitting energy. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[45] M. Li, C. Kambhamettu, and M. Stone. A level set approach for shape recovery of open contours. In *Proceedings of the Asian Conference on Computer Vision*, 2006.

[46] W. Lorensen and H. Cline. Marching cubes: A high resolution 3d surface construction algorithm. *Computer Graphics*, 21(4):163–169, 1987.

[47] L. Lorigo, O. Faugeras, E. Grimson, R. Keriven, R. Kikinis, A. Nabavi, and C.-F. Westin. Codimension-two geodesic active contours for the segmentation of tubular structures. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2000.

[48] R. Malladi, J. Sethian, and B. Vemuri. Shape modeling with front propagation: A level set approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(2):158–175, 1995.

[49] Netlab. http://www.ncrg.aston.ac.uk/netlab/index.php.

[50] S. Osher and R. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*. Springer-Verlag, 2003.

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

[52] N. Paragios and R. Deriche. Geodesic active regions and level set methods for supervised texture segmentation. *International Journal of Computer Vision*, 46(3):223–247, 2002.

[53] A. Roberts. Curvature attributes and their application to 3d interpreted horizons. *First Break*, 19(2):85–100, 2001.

[54] R. Ronfard. Region-based strategies for active contour models. *International Journal of Computer Vision*, 13(2):229–251, 1994.

[55] S. Roweis and Z. Ghahramani. A unifying review of linear gaussian models. *Neural Computation*, 11(2):305–345, 1999.

[56] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 2003.

[57] A. Schleicher, K. Amunts, S. Geyer, P. Morosan, and K. Zilles. Observer-independent method for microstructural parcellation of cerebral cortex: A quantitative approach to cytoarchitectonics. *Neuroimage*, 9(1):167–177, 1999.

[58] J. Serra. *Image Analysis and Mathematical Morphology, Volume 2: Theoretical advances*. Academic Press, 1988.

[59] J. Sethian. *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*. Cambridge University Press, 1999.

[60] P. Smereka. Spiral crystal growth. *Physica D: Nonlinear Phenomena*, 138(3).

[61] J. E. Solem and A. Heyden. Reconstructing open surfaces from image data. *International Journal of Computer Vision*, 69(3):267–275, 2006.

[62] A. Spira and R. Kimmel. Segmentation of images painted on parametric manifolds. In *European Signal Processing Conference (EUSIPCO)*, 2005.

[63] M. Stemmler, M. Usher, and E. Niebur. Lateral interactions in primary visual cortex: a model bridging physiology and psychophysics. *Science*, 269(5232):1877–1880, 1995.

[64] W. Sun, M. Cetin, R. Chan, and A. Willsky. Learning the dynamics and time-recursive boundary detection of deformable objects. *IEEE Transactions on Image Processing*, 17(11):2186–2200, 2008.

[65] P. Thompson and A. Toga. A surface-based technique for warping three-dimensional images of the brain. *IEEE Transactions on Medical Imaging*, 15(4):402–417, 1996.

[66] M. Tipping and C. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B*, 61(3):611–622, 1999.

[67] A. Tsai. *Curve Evolution and Estimation-Theoretic Techniques for Image Processing*. PhD thesis, Massachusetts Institute of Technology, 2000.

[68] G. Wahba. *Spline Models for Observational Data*. SIAM, 1990.

[69] N. Walters, S. Eickhoff, A. Schleicher, K. Zilles, K. Amunts, G. Egan, and J. Watson. Observer-independent analysis of high-resolution MR images of the human cerebral cortex: in vivo delineation of cortical areas. *Human Brain Mapping*, 28(1):1–8, 2007.

[70] C. Xu and J. Prince. Snakes, shapes and gradient vector flow. *IEEE Transactions on Image Processing*, 7(3):359–369, 1998.

[71] A. Yezzi, A. Tsai, and A. Willsky. A statistical approach to snakes for bimodal and trimodal imagery. In *Proceedings of the International Conference on Computer Vision*, 1999.

[72] X. Zeng, L. Staib, R. Schultz, and J. Duncan. Segmentation and measurement of the cortex from 3-d mr images using coupled-surfaces propagation. *IEEE Transactions on Medical Imaging*, 18(10):927–937, 1999.

[73] H. Zhao, T. Chan, B. Merriman, and S. Osher. A variational level set approach to multi-phase motion. *Journal of Computational Physics*, 127:179–195, 1996.

[74] S. Zhu and A. Yuille. Region competition: Unifying snakes, region growing, and bayes/mdl for multiband image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(9):884–900, 1996.