

UNIVERSITY OF CALIFORNIA  
Los Angeles

Mathematical Techniques  
in Object Matching and Computational Anatomy:  
a New Framework Based on the Level Set Method

A dissertation submitted in partial satisfaction of the  
requirements for the degree Doctor of Philosophy  
in Biomathematics

by

Wei-Hsun Liao

2003



The dissertation of Wei-Hsun Liao is approved.

---

Kenneth Lange

---

Tom Chou

---

Marvin Bergsneider

---

Luminita Vese

---

Sung-Cheng Huang, Committee chair

University of California, Los Angeles

2003

# Contents

<b>0</b>	<b>Preface – an Introduction</b>	<b>1</b>
<b>1</b>	<b>Partial Differential Equations, the Level Set Method, and Image Processing</b>	<b>4</b>
1.1	Introduction	5
1.2	Fundamental Formulations of the Level Set Method	5
1.3	The Level Set Dictionary	10
<b>2</b>	<b>Examples of Image Processing Techniques based on the Level Set Method</b>	<b>15</b>
2.1	Image Segmentation based on the Level Set Method	16
2.2	Image De-noising Techniques based on PDE's and the Level Set Method	18
2.3	A Modified Anisotropic Diffusion Applied to Dynamic PET FDG Brain Images	20
<b>3</b>	<b>Image Blending with Contrast Invariance</b>	<b>26</b>
3.1	Introduction	27
3.2	Description of the Model	28
3.3	Euler-Lagrange Equations and the Corresponding Gradient Descent Time Dependent Partial Differential Equations	33
3.4	Implementation	36
3.5	Results	39
<b>4</b>	<b>Computational Anatomy and the Semi-Lagrangian Level Set Method</b>	<b>49</b>
4.1	Introduction	50
4.2	Small Deformation Matching through Regularizer on the Displacement	52
4.3	Large Deformation through Diffeomorphisms by Infinite Dimensional Group Actions	53
4.4	The Semi-Lagrangian Level Set Method	61
<b>5</b>	<b>Two-Dimensional Object Matching using the Level Set Method</b>	<b>67</b>
5.1	Shape Matching	68
5.2	Open Curve Matching	70
5.3	Incorporating the Level Set Method to Diffeomorphisms Generated by	75

	Infinite Dimensional Group Actions	
5.4	The Modified Beg's Algorithm	76
5.5	Distance Function Re-Initialization	78
5.6	Results	79
<b>6</b>	<b>Length Shortening Flow of Open Curves in Two-Dimensional Space</b>	<b>83</b>
6.1	Basic Formulations for Length Shortening Flow of Curves in 2D	84
6.2	How to Move the End Points of an Open Curve Using Level Set Implicit Formulation	85
6.3	A Second Strategy for Mapping Open Curves in 2D (When the End Points Are Known)	86
<b>7</b>	<b>Level Set Based Landmark Matching</b>	<b>90</b>
7.1	Landmark Matching – First Approach	91
7.2	Landmark Matching – Second Approach	93
7.3	A Numerical Example of Level Set based Landmark Matching	95
<b>8</b>	<b>Equivalence of Objects under the Action of Translation, Scaling, and Rotation</b>	<b>97</b>
8.1	The Equivalence Classes of Objects	98
8.2	The Invariance of Objects Represented by Their Signed Distance Functions	99
8.3	Minimizing the Hausdorff Metric Between Objects	101
<b>9</b>	<b>Extensions to Three Dimensional Object Matching</b>	<b>104</b>
9.1	Level Set Dictionary for 3-Dimensional Curves	105
9.2	Length Shortening Flow of Curves in 3D	108
9.3	Open Curve Matching in 3D	111
<b>10</b>	<b>Numerical Experiments</b>	<b>118</b>
10.1	Shape Matching	119
10.2	Open Curve Matching	122
10.3	Landmark Matching	123
<b>11</b>	<b>Conclusion and Future Directions</b>	<b>161</b>
11.1	Conclusion	162
11.2	Future Directions	164
	<b>Bibliography</b>	<b>168</b>

# List of Figures

1.1	The Eikonal equation computed on a 128 by 128 grid in which the speed function is one with initial front at point $x=20$ and $y=60$ , and three impenetrable blocks. The result can be interpreted as the shortest distance to the point	9
2.1	A Dynamic Brain FDG PET Imaging before applying the (modified) anisotropic diffusion proposed by Perona and Malik.	24
2.2	The same dynamic brain FDG PET imaging as in figure 1 processed by the (modified) anisotropic diffusion proposed by Perona and Malik.	25
3.1	An image blending sequence from a circular shape in (a) to an ellipse. (b), (c), and (d) are the intermediate images along the sequence.	41
3.2	An image blending sequence from the initial image (a): the distance function to the point $x = 80.5$ , $y = 20.5$ , to the target, the distance function to the point $x = 30.5$ , $y = 70.5$ . (b), (c), and (d) are the intermediate images along the sequence. Notice that the image is actually transformed to the reverse of the target image.	42
3.3	An image blending sequence from the initial image to the target image (shown in the next figure Fig. 4). (b), (c), and (d) are the intermediate images along the sequence. Notice that the image is transformed such that one of the two objects in the image is reverse of the target image.	44
3.4	The target image in figure 3.	45
3.5	<b>Left panel:</b> the SPGR MRI brain image of a control subject. <b>Right panel:</b> the PET FDG brain image of the same plane from the same subject.	48
3.6	The hybrid image of the two brain images in figure 5	48
4.1	Front merging of two circles with constant speed implemented by the semi-Lagrangian level set method.	64
4.2	Front breaking into two pieces with constant speed implemented by the semi-Lagrangian level set method.	65
5.1	(a): Illustration of a general set-up for object matching with the level set based method. Objects in the template will be represented by level set	69

functions  $\phi$ 's. The corresponding objects in the study by level set functions  $\phi$ 's. (b): Illustration of four sub-regions divided by the level set functions, when the shapes in the template and study have overlap in space.

5.2	Illustration of how to represent an open curve using level set functions.	71
5.3	A matching problem of two arcs in the template (a) and the study (b).	80
5.4	The deformation field obtained by the proposed method for matching figure 3(a) to 3(b).	80
5.5	The Transformation process of the template (a) (one pair of oval shapes) to the study (f) (a pair of oval shapes without overlapping with the shapes in the template)	81
5.6	The final deformation field of one pair of oval shapes to another non-overlapping pair in figure 5.	82
7.1	The diffeomorphism given by the level set based landmark matching with distance re-initialization. The landmarks to be matched are (25, 30) to (30, 25), and (40, 35) to (35, 40) on a 64 by 64 image.	96
9.1	Illustration of how to represent a curve in 3D and calculate the length of the curve using the intersection of two level set functions.	108
10.1	(a): a brain MRI image right after traumatic brain injury. (b): MRI image of the same patient three months later. (c): warped image of (b) to match (a).	120
10.2	The diffeomorphism of moving the circle centered at the grid point (35, 35) with radius 5 to (45, 45) using the modified Bet's algorithm calculated on the unit square discretized to a 64 by 64 grid.	121
10.3	(a): the zero level sets of two level set functions arbitrarily initialized. (b): the warped zero level sets of the level set functions with the arc in the middle matched to the line segment joining grid points (30, 60) and (90, 60). (c): the warped zero level sets of the level set functions without enforcing matching of the endpoints.	124
10.4	(a): the deformation field of figure 3(b). (b): the deformation field of matching the line segment in (a) with oscillations added.	125
10.5	The same testing problem as in figure 3 and 4(a) is computed using the	127

modified Beg's algorithm. The deformation field obtained in this case is different from figure 5(a).

- 10.6 The diffeomorphism given by the level set based landmark matching with distance re-initialization. The landmarks to be matched are (16, 32) to (16, 27), (29, 35) to (29, 40), (29, 29) to (29, 34), (35, 35) to (35, 30), (35, 29) to (35, 24), and (50, 32) to (50, 37) on a 64 by 64 image. 128
- 10.7 Two random curves are drawn by hand on a 64 by 64 grid for illustrating landmark matching, curve matching, and the equivalence class of images under the action of translation, scaling, and rotation. 130
- 10.8 **Upper panel:** the positions of the two open curves after matching the end points of the curves. **Lower panel:** the corresponding deformation field. 131
- 10.9 **Upper panel:** the positions of the two open curves after matching the two curves. **Lower panel:** the corresponding deformation field. 132
- 10.10 The positions of the two open curves after taking into account the equivalence class of the curve in the study by factoring out actions of translation, scaling, and rotation. The Hausdorff metric between two curves (approximated by raising the power of the distance function to the 7-th power) is used as the cost function. The four parameters in this example are  $a = -2.442$ ,  $b = -6.978$ ,  $r = 0.7340$ , and  $\theta = -0.3865$ . 134
- 10.11 The positions of the two open curves after taking into account the equivalence class of the curve in the study by factoring out actions of translation, scaling, and rotation. The original cost function between two curves is used without raising the distance function to higher power than one. The four parameters in this example are  $a = -2.582$ ,  $b = -5.582$ ,  $r = 0.5314$ , and  $\theta = -0.4828$ . 135
- 10.12 **Upper panel:** the positions of the two open curves after matching the two curves using the target in figure 10 where translation, scaling and rotation are factored out. **Lower panel:** the corresponding deformation field. 136
- 10.13 The oval shape carried by the geodesic flow that links the two oval shapes in figure 2 at time 0.1 (a), time 0.2 (b), time 0.3(c), 0.4 time (d), time 0.5 (e), time 0.6 (f), time 0.7 (g), time 0.8(h), and time 0.9 (i). 142
- 10.14 The geodesic flow that links the two oval shapes in figure 2 at time 0.1 (a), time 0.2 (b), time 0.3(c), 0.4 time (d), time 0.5 (e), time 0.6 (f), time 0.7 (g), time 0.8(h), and time 0.9 (i). 151

- 10.15 The geodesic flow that links the two open curves in figure 7 without taking into account the equivalence class at time 0.1 (a), time 0.2 (b), time 0.3(c), 0.4 time (d), time 0.5 (e), time 0.6 (f), time 0.7 (g), time 0.8 (h), and time 0.9 (i). 160

## **List of Tables**

- 2.1 Comparison of % SD of Original and New Images at Various Times 23

## VITA

March 2, 1974	Born, Chia-Yi, Taiwan
1998	M.D. National Taiwan University Taipei, Taiwan
1998-1999	First year resident Department of Neurology National Taiwan University School Hospital
1999-2003	Department of Biomathematics University of California, Los Angeles Los Angeles, CA, USA

## PUBLICATIONS AND PRESENTATIONS

BrainPET 2001 (June, 2001), The Use of MM Algorithm for Regularization of Parametric Images in PET.

Brain Imaging Using PET (Academic Press), Use of MM Algorithm for Regularization of Parametric Images in PET, Pages: 107-114.

Medical Imaging Conference 2001(IEEE MIC), Optimal Design in PET Data Acquisition: a New Approach Using Simulated Annealing and Component-wise Metropolis Updating.

Society of Nuclear Medicine, annual meeting 2002: A Modified Anisotropic Diffusion for PET FDG Dynamic Images.

Neuroimage: A Variational Approach for Noise Removal of Parametric Images in Tracer Kinetic Modeling, July, 2002, VOL 16, No. 3, S 68-69.

IEEE TRANSACTIONS ON NUCLEAR SCIENCE: Optimal Design in PET Data Acquisition: a New Approach Using Simulated Annealing and Component-wise Metropolis Updating, October 2002, VOL 49. Issue 5, pp 2291-2296.

UCLA Computational Applied Mathematics Report 02-59 (November, 2002): From Landmark Matching to Shape and Open Curve Matching: a Level Set Approach.

UCLA Computational Applied Mathematics Report 03-02 (January, 2003): Computational Anatomy, Object Matching, and the Level Set Method.

UCLA Image Processing Seminar (January, 2003): Computational Anatomy, Object Matching, and the Level Set Method (invited talk).

UCLA Applied Mathematics Colloquium (April, 2003): The Semi-Lagrangian Level Set Method and its Application to Computational Anatomy (invited talk).

IEEE International Conference on Image Processing (ICIP) 03: A New Framework of Object Warping: a Semi-Lagrangian Level Set Approach (oral presentation).

International Workshop on Biomedical Image Registration (2003): Computational Anatomy, Object Matching, and the Level Set Method (oral presentation).

Lecture Notes in Computer Science: Computational Anatomy and Implicit Object representation: a Level Set approach (in print)

## ABSTRACT OF THE DISSERTATION

Mathematical Techniques  
in Object Matching and Computational Anatomy:  
a New Framework Based on the Level Set Method

by

Wei-Hsun Liao

Doctor of Philosophy in Biomathematics  
University of California, Los Angeles, 2003  
Professor Sung-Cheng Huang, Chair

During the past decade, lots of progress has been made in biomedical image processing with the help of modern computers. These technological advances allow the use of large-scale computational algorithms and techniques of which the implementation was almost impossible just few years ago.

This dissertation is about biomedical imaging processing. More precisely, it is about the incorporation of the level set method, computational anatomy and other related fields in biomedical imaging processing.

A general framework of level set based object matching technique is developed in this dissertation. In order to apply the level set method, it has to be re-formulated in the semi-Lagrangian reference formulation rather than the Eulerian reference in the original level set method. The level set based object matching techniques is then incorporated to the framework of diffeomorphisms generated by infinite dimensional group actions in

computational anatomy. With this formulation, a metric can be rigorously defined by quantifying the underlying grid deformation based on ideas borrowed from continuum mechanics. Moreover, a geodesic flow can be defined and computed that links one object to the other. A unifying approach for matching objects of different types including shapes, open curves, and landmarks in both two-dimensional and three-dimensional space will be presented.

Formulations for taking into account the equivalence of objects under the action of translation, scaling and rotation will also be discussed. Furthermore, we show that the method of level set based object matching is closely related to minimizing the Hausdorff metric between objects, a classic result in geometric measure theory. This provides a sound and rigorous foundation for the level set based object matching.

# Preface

## An Introduction

*Abstract-This dissertation is about the level set method and its application to biomedical image processing in the field of computational anatomy and some other closely related fields. In this section, we will briefly describe the scope of this dissertation, the problems we wish to solve, and the goals we wish to achieve.*

During the past decade we have seen some of the most amazing progress in computational techniques due to the rapid technological advances in modern computers. Due to this progress, People have started to simulate the phenomenon seen in real world using highly complicated computational algorithms and resulted in tremendous success. In biomedical sciences, we have seen the most promising future in the study of genetic codes (resulted from the large amount of data obtained through gene sequencing) and in the field of biomedical image processing due to the help of modern computers and computational techniques.

This dissertation is about biomedical imaging processing. More precisely, it is about the incorporation of the level set method to computational anatomy and other related fields in biomedical imaging processing.

The level set method is a powerful tool for interface tracking. The interface is being implicitly represented as the zero level set of the corresponding level set function of one higher dimension. With this implicit representation, tracking of the interface becomes updating the values of the level set function on the grid points. Because of the Eulerian reference nature, the main advantage of the level set method is that it treats topological changes naturally. The other advantage of the level set method is, due to the implicit representation, it provides an elegantly and efficient mathematical and computational technique of representing and evolving interfaces.

However, the main advantage of allowing topological changes of the level set method becomes the main drawback if one wishes to apply it to computational anatomy. In computational anatomy, anatomical structures are considered embedded in the underlying

image and thus is not Eulerian in nature. In order to incorporate the level set method into the framework of computational anatomy, the level set method has to be re-interpreted in a new manner, namely, in the semi-Lagrangian reference formulation.

A general framework of level set based object matching technique based on the semi-Lagrangian implementation is developed in this dissertation. This new framework is then tied to the framework of comparing images through diffeomorphisms generated by infinite dimensional group actions in computational anatomy. A unifying approach for matching objects of different types will be presented with generalization to 3D object comparison. Formulations for taking into account the equivalence of images under the action of translation, scaling and rotation will also be discussed along with a theoretical connection with the classic result of the Hausdorff metric (defined between two sets of points) in geometric measure theory. Numerical results on the matching of 2D objects including shapes, open curves, and landmarks will be presented.

We will start with reviewing the fundamental techniques of image processing using modern techniques borrowed from calculus of variations, partial differential equations, and the level set method in chapter 1, 2 and 3. In chapter 4, an overview of computational anatomy will be discussed, followed by a detailed discussion of level set based object matching. Numerical examples will be presented in chapter 10 while a summary and some future directions are outlined in the final chapter of this dissertation.

# Chapter 1

## Partial Differential Equations, the Level Set Method, and Image Processing

*Abstract-Traditionally, image processing in the engineering society involves tasks including, for example, image segmentation, edge detection and denoising with a wide variety of filters. Since the past decade, the use of partial differential equations, statistics, and wavelet theory in image processing is rapidly growing and expanding. In the first chapter, we will briefly review the use of partial differential equations-especially the level set method-in modern image processing.*

## 1. Introduction

The level set method, first proposed by Osher and Sethian in [1], allows an interface  $\Gamma$  (a curve in 2D or a surface in 3D) to be represented implicitly by the zero level set of a function  $\Phi$  (the level-set function) of one higher dimension

$$\Gamma = \{x \mid \Phi(x) = 0\} \quad (1)$$

With this representation, an image is divided into two regions by the level set function:  $\Phi > 0$  and  $\Phi < 0$ . In the original level set formulation, negative values were assigned to the area inside the interface  $\Gamma$ . In this dissertation, we will adopt a different approach from the original level set formulation by assigning positive values inside the interface instead.

## 2. Fundamental Formulations of the Level Set Method

In this section we will follow closely the discussion in [2].

### Formulation in the form of initial value problem

Let us now derive the fundamental formulation for the level set method. We will derive the formulation for evolving curve in 2D though 3D is treated similarly and results in the same formulation.

Let the interface be  $x(t)$ , a closed curve in 2D that is parameterized by a parameter  $t$  and recall the definition of the level set function

$$\Phi(x(t), t) = 0. \quad (2)$$

Now let us differentiate the above equation with respect to  $t$

$$\Phi_t + \nabla\Phi \cdot x'(t) = 0 \quad (3)$$

Assuming that the interface is moving according to a speed function  $F(x, t)$ , we have

$$\begin{aligned} F &= x'(t), \\ \Phi_t + \nabla\Phi \cdot F &= 0. \end{aligned} \quad (4)$$

If  $F$  has only the component pointing in the outward normal direction without the tangential component, then the above equation is the same as

$$\frac{\partial\Phi}{\partial t} + F|\nabla\Phi| = 0. \quad (5)$$

These two equations are the fundamental formulation of the level set method. It is an initial value problem with the initial value given by

$$\Phi_0 = \Phi(t = 0) \quad (6)$$

with the initial interface  $\Gamma$  being the zero level set of  $\Phi_0$ . One way (possibly the best way) to initialize  $\Phi_0$  is to use the signed distance function of  $\Gamma$  as described in the next section.

Equation 5 falls into a general category of the so-called Hamilton-Jacobi equations. Schemes borrowed from hyperbolic conservation law have been shown to successfully

solve these problems numerically. For a review on these schemes please refer to [3] and the references therein.

In this chapter, let us describe the simplest first order upwind scheme in 2D for a convex speed function  $F$

$$\Phi_{i,j}^{n+1} = \Phi_{i,j}^n - \Delta t [\max(F_{i,j}, 0) \nabla^+ + \min(F_{i,j}, 0) \nabla^-] \quad (7)$$

$$\nabla^+ = [\max(D_{i,j}^{-x} \Phi, 0)^2 + \min(D_{i,j}^{+x} \Phi, 0)^2 + \max(D_{i,j}^{-y} \Phi, 0)^2 + \min(D_{i,j}^{+y} \Phi, 0)^2]^{1/2} \quad (8)$$

$$\nabla^- = [\max(D_{i,j}^{+x} \Phi, 0)^2 + \min(D_{i,j}^{-x} \Phi, 0)^2 + \max(D_{i,j}^{+y} \Phi, 0)^2 + \min(D_{i,j}^{-y} \Phi, 0)^2]^{1/2} \quad (9)$$

A slightly different first order upwind scheme, as suggested in [4], is the building block for solving Eikonal equation (described in the next section)

$$\nabla^+ = [\max(D_{i,j}^{-x}, -D_{i,j}^{+x}, 0)^2 + \max(D_{i,j}^{-y}, -D_{i,j}^{+y}, 0)^2]^{1/2}, \quad (10)$$

$$\nabla^- = [\max(-D_{i,j}^{-x}, D_{i,j}^{+x}, 0)^2 + \max(-D_{i,j}^{-y}, D_{i,j}^{+y}, 0)^2]^{1/2}. \quad (11)$$

## **Formulation in the form of boundary value problem-the Eikonal Equation**

Level set method has also a boundary value problem formulation in the case when each point is crossed by the interface at most once. It can be considered as computing the “arrival time”  $T(x)$  of the interface at each point. The equation that the arrival time satisfies is the following Eikonal type boundary value problem

$$F|\nabla T| = 1 \quad (12)$$

The boundary value is given by

$$T(x) |_{\Gamma} = 0 \quad (13)$$

here  $\Gamma$  is the initial interface at time zero.

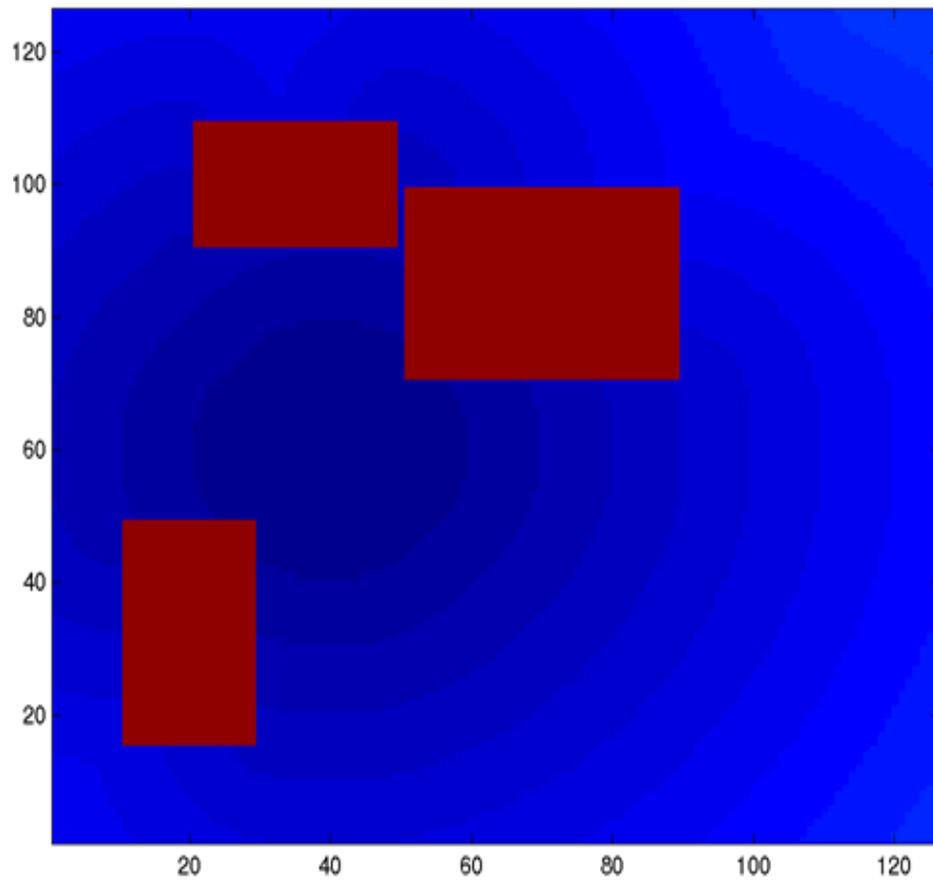
If we use the scheme in equation (10) for numerical discretization, solving the Eikonal equation becomes solving the following starting from the initial interface

$$[\max(D_{i,j}^{-x}T, -D_{i,j}^{+x}T, 0)^2 + \max(D_{i,j}^{-y}T, -D_{i,j}^{+y}T, 0)^2]^{1/2} = \frac{1}{F_{i,j}} \quad (14)$$

The above equation could be solved efficiently using the fast marching method [5-7], which involves a heap sorting structure in implementation. In [8], the fast marching method is generalized to deal with triangulations and generating geodesics on a manifold. Approaches for solving the equations of the Eikonal type other than the fast marching method are also being actively investigated. For example, a different technique-the “sweeping” technique-is proposed and studied in [9].

## **A numerical example**

When the speed function is one in the Eikonal equation, the arrival time reduces to the distance function at each grid point to the initial front. It can also be applied to optimal path planning by assigning different speeds in different regions and zero for impenetrable obstacles. Figure 1 shows the distance function to the point (40,60) with three impenetrable blocks.



**Fig. 1:** The Eikonal equation computed on a 128 by 128 grid in which the speed function is one with initial front at point  $x=40$  and  $y=60$  and three impenetrable blocks. The result can be interpreted as the shortest distance to the point.

### **The advantage of the level set method**

Tracking the front with the level set method has several advantages over the traditional method of putting particles on the front and tracking the particles. The main advantages are

1. **Topological changes in the evolving interface are easily handled**
2. **Direct generalization to interface evolution in higher dimensions**
3. **Efficient numerical schemes are available that guarantees correct weak solutions satisfying the entropy condition**

### **3. The Level Set Dictionary**

In this section, we will describe how common quantities in image processing can be calculated in the level set framework.

#### **The unit normal vector to the interface**

The unit normal vector to the interface is simply

$$n = \frac{\nabla\Phi}{|\nabla\Phi|} \quad (15)$$

In the case when the level set function takes positive values inside the interface, this normal vector points “inward”.

#### **The area enclosed by the interface**

The area (A) enclosed by the interface could be recovered by integrating one in the region where the corresponding level set function takes positive values

$$A = \int_{\Omega} H(\Phi(x, y)) dx dy \quad (16)$$

Here  $H$  is the Heaviside function as defined in the following way

$$H(x) = \begin{cases} 0, & x < 0 \\ 1, & x \geq 0 \end{cases} \quad (17)$$

### **The length of the interface**

The length of the interface could be obtained by the length of the zero level set of the corresponding level set function  $\Phi$

$$L = \int \delta(\Phi) |\nabla \Phi(x, y)| dx dy \quad (18)$$

Here  $\delta$  is the delta dirac function, the derivative of the Heaviside function in the sense of distributions.

### **The curvature of the front at any point**

The curvature  $k$  of the level curve of the level set function passed through each point is the divergence of the unit normal vector at that point

$$k = \operatorname{div}\left(\frac{\nabla \Phi}{|\nabla \Phi|}\right) . \quad (19)$$

In image processing, the curvature term occurs naturally in the Euler-Lagrange equation of various energy minimization problems. For example, motion under curvature arises in

length shortening of a curve (minimizing equation 18 with respect to the level set function  $\Phi$ )

$$\frac{\partial \Phi}{\partial t} = |\nabla \Phi| \operatorname{div} \left( \frac{\nabla \Phi}{|\nabla \Phi|} \right) \quad (20)$$

Due to the hyperbolic nature of equation 20, central differencing should be used for discretization instead of the upwind scheme introduced in previous sections. Moreover, motion under curvature plays an important role in many applications as it smoothes out the boundaries due to its non-linear heat equation property. It is shown in [10-12] that, all simple closed curves shrink to a point under motion of curvature regardless of their initial shapes.

## Numerical approximations for Heaviside and Delta function

Numerically, approximations are used to calculate Heaviside and Delta function. Different choices are available depending on the smoothness of the approximating functions. For example, the following approximations are used extensively

$$H_a(x) = \begin{cases} \frac{1}{2}(1 + x/a + \sin(\pi x/a)) & \text{if } |x| \leq a \\ 1 & \text{if } x > a \\ 0 & \text{otherwise} \end{cases} \quad (21)$$

$$\delta_a(x) = \begin{cases} \frac{1}{2a}(1 + \cos(\pi x/a)) & \text{if } |x| \leq a \\ 0 & \text{otherwise} \end{cases} \quad (22)$$

A second choice that allows more smoothness is the following  $C^\infty$  approximation of the Heaviside function

$$H_a = \frac{1}{2} \left( 1 + \frac{2}{\pi} \tanh\left(\frac{x}{a}\right) \right) \quad (23)$$

The approximation of the delta function in this case is then obtained by differentiating the equation (23)

$$\delta_a = \frac{a}{\pi(a^2 + x^2)} \quad (24)$$

## Extension of the Speed Function

A subtle point of the level set formulation is that the velocity is only available on the interface, or the zero level set of the level set function. However, we need to provide speed function for level sets other than the zero level set in order to evolve the whole level set function. There is no “correct” way of extending the speed function; yet it would be desirable if the extended speed function preserves the signed distance property of the level set function. As discussed in [2, 13], this is equivalent to requiring that the extended speed function satisfies the following relation

$$\nabla F^{extended} \cdot \nabla \Phi = 0 \quad (25)$$

An alternative is to “re-initialize” the level set function to be the signed distance function of its zero level set by the following iteration as first proposed in [14, 15].

$$\begin{aligned}\varphi_\tau &= \text{sign}(\Phi(t))(1 - |\nabla \varphi|) \\ \varphi(0, \bullet) &= \Phi(t, \bullet)\end{aligned}\tag{26}$$

In image processing, often times we only concern about the zero level set. In these cases, it might be helpful to look at the alternative form of the level set formulation

$$\frac{\partial \Phi}{\partial t} + F \delta(\Phi) = 0\tag{27}$$

The above equation only changes the level set function values nearby the interface, and thus is similar to the idea of narrow banding as studied extensively in [16].

## Chapter 2

# Examples of Image Processing Techniques based on the Level Set Method

*Abstract-In this chapter, we will briefly describe some of the latest and most important image processing techniques based on the level set method. For image segmentation, the geodesic active contours and active contours without edges will be discussed. For image denoising, the anisotropic diffusion proposed by Perona and Malik and the method based on the total variation will also be presented. Furthermore, we derive a modified anisotropic diffusion as proposed by Perona and Malik and apply it to dynamic PET FDG brain imaging. We examine the effectiveness of removing noise especially in the first few minutes of the dynamic scan in which the images are very noisy.*

# 1. Image Segmentation based on the Level Set Method

## Active contours without edges by Chan and Vese

In [17, 18], Chan and Vese proposed the following variational model for segmentation of a bimodal image  $u$

$$\min_{c_1, c_2, \Phi} E = \lambda_1 \int_{\Phi > 0} (u - c_1)^2 + \lambda_2 \int_{\Phi < 0} (u - c_2)^2 + \mu \times \text{length}(\Phi = 0) \quad (1)$$

The idea is to segment the image into two regions: inside and outside of the zero level set, with each region represented by a constant intensity value ( $c_1$  and  $c_2$ ). A penalty (tuned by the weight  $\mu$ ) on the length of the zero level set is added to ensure proper segmentation.

This formulation leads to the following gradient descent PDE

$$\frac{\partial \Phi}{\partial t} = \delta(\Phi) \left[ \mu \text{div} \left( \frac{\nabla \Phi}{|\nabla \Phi|} \right) - \lambda_1 (u - c_1)^2 + \lambda_2 (u - c_2)^2 \right] \quad (2)$$

This approach can be easily extended to images that can be segmented into  $2^n$  regions by adding more level set functions and using the intersections of the level set functions to represent different regions.

## Geodesic active contour

Geodesic Active Contour model was first proposed in [19, 20]. In this model, they try to find a curve  $C$  that detects objects in an image  $u$  by minimizing “weighted” length of the curve

$$\min E(C) = \int g(|\nabla u(C(s))|) ds. \quad (3)$$

Here  $s$  is the arc length and  $g$  is often called an “edge detector function”, which is a positive and decreasing function of its argument such that it stops the curve evolution while edges are detected. The level set formulation of this problem by imbedding  $C$  as the zero level set of  $\Phi$  is shown to be

$$\frac{\partial \Phi}{\partial t} = |\nabla \Phi| \operatorname{div} \left( g(u) \frac{\nabla \Phi}{|\nabla \Phi|} \right) \quad (4)$$

Notice that the speed function resembles the curvature term introduced in equation 19 of chapter 1. It is common to expand the divergence term in equation (4) and obtain the following alternative form

$$\frac{\partial \Phi}{\partial t} = |\nabla \Phi| g(u) \operatorname{div} \left( \frac{\nabla \Phi}{|\nabla \Phi|} \right) + \langle \nabla g, \nabla \Phi \rangle \quad (5)$$

It is also noticed that by modifying equation 5 in the following way, the detection of non-convex objects could be made easier

$$\frac{\partial \Phi}{\partial t} = |\nabla \Phi| g(u) \left[ \operatorname{div} \left( \frac{\nabla \Phi}{|\nabla \Phi|} \right) + \alpha \right] + \langle \nabla g, \nabla \Phi \rangle \quad (6)$$

The discretization of the above equation includes a mixture of hyperbolic and parabolic terms, and thus requires separate treatment of those terms. Please refer to [19-21] for more detailed discussion.

## **2. Image De-noising Techniques based on PDE's and the Level Set Method**

### **The anisotropic diffusion by Perona and Malik**

Anisotropic diffusion is a nonlinear smoothing technique, as opposed to the isotropic diffusion, which is exactly the heat equation applied to an image. In [22], Perona and Malik proposed one of the first anisotropic diffusion algorithm for de-noising an image  $u$  by evolving  $u$  according to the following PDE

$$\frac{\partial u}{\partial t} = \text{div}(c(|\nabla u|)\nabla u) \quad (7)$$

Here  $c$  is the diffusion coefficient. This PDE can be obtained by minimizing the following cost function

$$\min E(u) = \int_{\Omega} f(|\nabla u|)d\Omega \quad (8)$$

The diffusion coefficient  $c$  can be written as a function of  $f$ :

$$c(|\nabla u|) = \frac{f'(|\nabla u|)}{|\nabla u|} \quad (9)$$

### **Image denoising cast in the form of energy minimization problem**

Let  $R$  be a linear operator and  $u_0$  be a noisy image that we wish to recover. The recovered image  $u$  can be written in the form

$$u_0 = Ru + \eta \quad (10)$$

Here  $\eta$  is random noise with mean 0. A class of approaches for recovering  $u$  is looking at the following minimization problem by adding a penalty on the size of the gradient of the image

$$\min_u \int \frac{1}{2} (u - Ru_0)^2 dx + \alpha \int P(|\nabla u|) dx \quad (11)$$

Here  $\alpha$  is the weight of the penalty, and  $P$  is the penalty function. One of the earliest attempts to recover images with this variational formulation is in [23] where the penalty function is simply chosen to be the identity and the corresponding technique is called the total variation noise removal. The most interesting thing of this approach is that it results in an Euler-Lagrange equation that could be interpreted as motions under curvature for all level curves in the images.

### 3. A Modified Anisotropic Diffusion Applied to Dynamic PET FDG Brain Images

As in the original anisotropic diffusion proposed by Perona and Malik [22], the formulation could be written in this form:

$$\frac{\partial u}{\partial t} = \text{div}(c(|\nabla u|)\nabla u) = \text{div}\left(\frac{f'(|\nabla u|)}{|\nabla u|}\nabla u\right) \quad (12)$$

This only involves one image. In PET dynamic images, there are a series of images. For example, in FDG PET dynamic images, there are about a series of 20 time frames and each time frame is a 3D image of a brain acquired during that time period. The 3D image of a brain at any time frame has about 64 planes and each plane is an image of size 128 by 128 or 256 by 256). Thus, in order to smooth PET dynamic images, the modified anisotropic diffusion should have the following features:

**It is able to smooth while preserving edges**

**It has a variable smoothing property. We want it to smooth more in certain time frames where the images are noisy and less in the other frames.**

This is closely related to anisotropic diffusion in vector-valued image, yet we notice that there are some inherent differences. Firstly, different frames in the PET images do not carry the same amount of information. For instance, images obtained in the first few minutes in FDG PET contain mostly noise while the images obtained in the last few are

always much clearer. As a result, in PET dynamic images we hope to achieve an inhomogeneous smoothing across frames.

The modified anisotropic smoothing model we will use comes from a heuristic extension of the original Perona-Malik model by replacing the gradient with the average gradient

$$\frac{\partial u_i}{\partial t} = \text{div}(c(|\overline{\nabla u}|)\nabla u_i) \quad (13)$$

Here the subscript  $i$  refers to the  $i$ -th time frame, and  $\overline{\nabla u}$  is a properly defined average gradient of the image. This formulation has a variational form when the average gradient is defined in the following way

$$\overline{\nabla u} \equiv \begin{pmatrix} w_1 \frac{\partial u_1}{\partial x_1} & \cdots & w_m \frac{\partial u_m}{\partial x_1} \\ \vdots & \vdots & \vdots \\ w_1 \frac{\partial u_1}{\partial x_n} & \cdots & w_m \frac{\partial u_m}{\partial x_n} \end{pmatrix} \quad (14)$$

Here  $w$ 's are the weights. In equation (14), we simply use the Frobenius norm for the average gradient. Under this choice the PDE equation (13) could be viewed as the gradient decent of the following variational problem

$$\min E(u_1, \dots, u_m) = \int_{\Omega} f(|\overline{\nabla u}|) d\Omega \quad (15)$$

This formulation leads to a modified anisotropic diffusion except for that now the weight appears in front of the divergence operator. We then notice that in order to get smoother images in the first few images, we have to smooth the images in different time scales.

Since the  $t$  in the proposed anisotropic diffusion is an artificial time, we then can absorb the weight into  $t$  and use different scales for different frames in implementation.

In generating the results, we further replace the gradient defined above with simple weighted average as we could capture more of the edge information by simple weighted average. The reasoning is based on the observation that by using simple weighted average noise tends to cancel out when calculating the average gradient.

The PDE can be implemented similarly as in the original scheme proposed by Perona and Malik

$$(u_{i,j}^{n+1} - u_{i,j}^n) / \Delta t = c_N \nabla_N u_{i,j}^n + c_S \nabla_S u_{i,j}^n + c_E \nabla_E u_{i,j}^n + c_W \nabla_W u_{i,j}^n + c_U \nabla_U u_{i,j}^n + c_D \nabla_D u_{i,j}^n \quad (16)$$

Here N, S, E, W, U, D stand for north, south, east, west, up, and down respectively and

$$\begin{aligned} C_N &= f(|\nabla_N u_{i,j}^n|) \\ \nabla_N u_{i,j}^n &= u_{i-1,j}^n - u_{i,j}^n \end{aligned} \quad (17)$$

Similarly, we define the coefficients along the other five directions. As already mentioned in Perona and Malik's original paper, this discretization is not exact for the original equation yet our experimentation shows that it performs well and compares to other less crude schemes. Another point is that this scheme also preserves the sum of image intensities of all pixels within each time frame and satisfies the maximum principle.

We applied this algorithm to PET images of injured brains (n=8). After the images were processed, regions of interest (ROIs) ranging from 9 to 13 pixels were drawn in the hemisphere contralateral to the lesion. Both the mean and the percent standard deviation

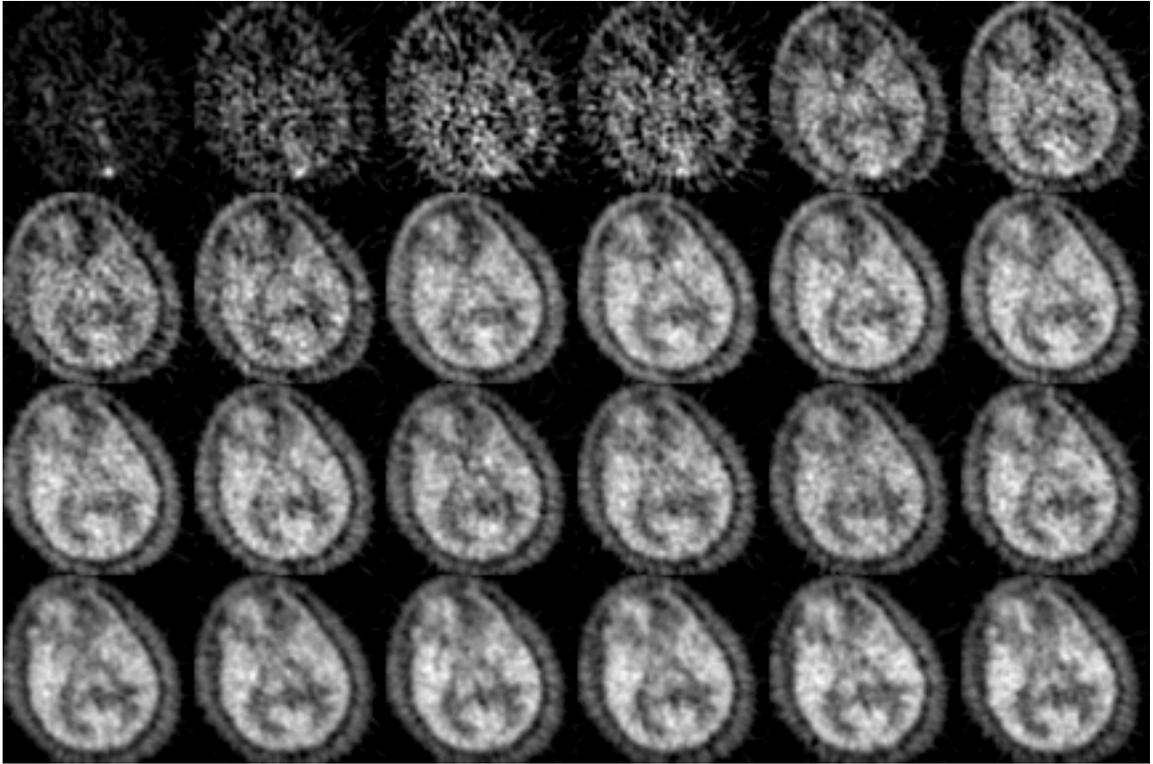
(percent SD) were computed within the ROIs for all time frames, before and after application of the algorithm. Percent SD of pixel values within ROI was calculated. Table 1 shows that the mean value within the ROIs remained the same while the Percent SD decreased significantly. These findings were consistent in all eight studies. The algorithm produced images that yielded smoother time activity curves and lower Percent SD at all time frames.

**Table 1 Comparison of % SD of Original and New Images at Various Times**

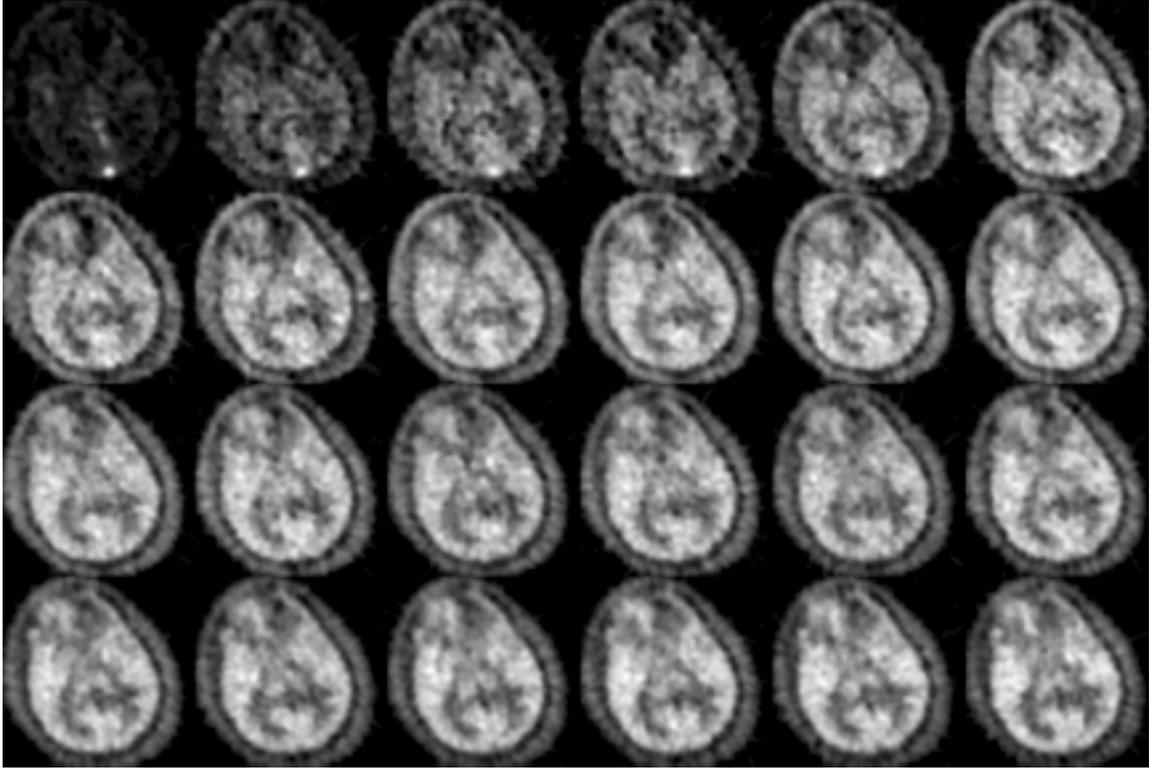
	<b>45 sec</b>	<b>7 min</b>	<b>37.5 min</b>
<b>Original image</b>	<b>17.6 ± 11.5</b>	<b>7.3 ± 4.0</b>	<b>5.2 ± 1.2</b>
<b>Smoothed image</b>	<b>4.7 ± 2.9</b>	<b>3.2 ± 1.9</b>	<b>4.0 ± 1.4</b>

In figure 1 and 2, we show the results in a particular plane of a control subject. We compare the images of different time frames of that specific plane before and after applying the modified anisotropic diffusion. It is clear from the figures that the first few frames now have much clearer pictures than before.

The original image format is in CTI version 6 and all images are from subjects in the UCLA Brain Trauma Project. All results are computed in 3D and implemented using C++ on Sun Solaris Unix system.



**Fig. 1: A dynamic brain FDG PET imaging before applying the (modified) anisotropic diffusion proposed by Perona and Malik.**



**Fig. 2: The same dynamic brain FDG PET imaging as in figure 1 processed by the (modified) anisotropic diffusion proposed by Perona and Malik.**

## Chapter 3

# Image Blending with Contrast Invariance

*Abstract—This chapter presents a new method for image blending. The proposed model is based on matching the gradient fields of two images. We first define equivalent classes of images in which images are considered equivalent under similarity group actions. The advantage of the approach is that images are identified by their relative contrasts and thus is scale free. By modifying the group actions, we examine different models under different choices of equivalent classes. The corresponding Euler-Lagrange equations associated to these models are derived. Fast and efficient solvers employing the Additive Operator Splitting (AOS) scheme will be reviewed. The proposed models are tested on several test images as well as real brain MRI and PET images from normal control subjects.*

## 1. Introduction

The mathematical formulation and theories of objects, shapes and images have been studied extensively for a long time in computer vision, statistics, geology, biomedical science and image processing literature. In this chapter, we examine the issue of image blending and propose a new method for achieving image blending different from simple linear interpolation, or cross dissolving.

The technique of linear interpolation (cross dissolving) for image blending between two images  $f$  and  $g$  on a spatial domain  $\Omega$  in  $R^2$  could be viewed as minimizing a common choice of cost function-the sum of squared difference-between two images

$$Cost(f, g) = \int_{\Omega} (f - g)^2 dx. \quad (1)$$

Minimizing this energy with respect to  $f$  simply gives us a linear interpolation between images  $f$  and  $g$ .

Although this cost function is the most direct and commonly used one, it is definitely not the best and the only choice. First, it is not scale invariant. If we rescale one of the images then we obtain a different cost function. Moreover, two images that only differ in scaling are considered different under this cost function. In this chapter, we will propose different cost functions from equation (1) that take into account the equivalence of images.

## 2. Description of the Model

### A natural way of defining “equivalent” images

We want to define differences between images in a way that the difference is defined in terms of equivalent classes of images. Difference between two images will only be nonzero when they do not belong to the same class. But how do we define equivalence? Human perception of an image depends mostly on the contrast of the image. Areas with high contrast are perceived as boundaries of objects while areas with more homogeneous intensities are often viewed as inside an object. So our goal is to construct cost functions that only take into account the relative change in the intensity values of images.

Let us first look at the following definition of equivalence. We consider two images  $I_1$  and  $I_2$  to be equivalent if and only if there exist real numbers  $k_1 > 0$  and  $k_2$  such that the following holds

$$I_1(x) = k_1 I_2(x) + k_2, \quad \forall x \in \Omega. \quad (2)$$

So the similarity should include translation ( $k_2$ ) and rescaling ( $k_1$ ) of the intensities. We could argue then that we would generalize the definition in equation 2 by relaxing our previous restrictions on  $k_1$ . By allowing negativity of  $k_1$ , we arrive at a second definition of equivalence.

Now we need to endow these new definitions with a proper cost function such that it is always non-negative and is zero only between images in the same class. Let two images  $f$

and  $g$  be in  $\Omega$ , and we consider the following inequality that is based on Cauchy-Schwarz inequality

$$\sqrt{\int |\nabla f|^2 dx} \int |\nabla g|^2 dx - \int (\nabla f \cdot \nabla g) dx \geq 0 \quad (3)$$

It is easy to see that the equality holds only when these two images are equivalent in the sense of equation (2). The reason of looking at this inequality is that it does not depend on the scale of the image, which is exactly what we wish for. In order to apply this equality to our space, we need to restrict our images to the Sobolev space  $H^1(\Omega)$ . From now on, all images are assumed to be in the space  $H^1(\Omega)$ . Moreover, constant images are excluded and the reason will become evident later.

### **A scale-free cost function between two images**

We notice that the inequality (3) is closely related to the following minimization problem and a metric can be defined in the sense of equation 2 (allowing negative  $k_I$ ).

**Given two images  $f$  and  $g$ , find  $c$  such that the following energy  $E$  is minimized**

$$\min_c E(c) = \int |\nabla f - c \nabla g|^2 dx \quad (4)$$

By expanding the integral it is easy to show that the energy is minimized when

$$c = \frac{\int (\nabla f \cdot \nabla g) dx}{\int |\nabla g|^2 dx} \quad (5)$$

and the minimum is just:

$$\int |\nabla f|^2 dx - \frac{[\int (\nabla f \cdot \nabla g) dx]^2}{\int |\nabla g|^2 dx} = \{\int |\nabla f|^2 dx \int |\nabla g|^2 dx - [\int (\nabla f \cdot \nabla g) dx]^2\} / \int |\nabla g|^2 dx. \quad (6)$$

Before moving on, we need the following lemma that is again just a restatement of Cauchy-Schwarz inequality (or Minkowski inequality):

**Lemma:**

$$(\int |\nabla f|^2 dx)^{1/2} + (\int |\nabla g|^2 dx)^{1/2} \geq (\int |\nabla f + \nabla g|^2 dx)^{1/2}. \quad (7)$$

We now are ready to propose a cost function to quantify the distance  $dist(f, g)$  between two images  $f$  and  $g$  by normalizing the energy defined above (equation 4) such that the difference does not depend on the scale of  $f$

$$dist(f, g) \triangleq \min_c \frac{(\int |\nabla f - c \nabla g|^2 dx)^{1/2}}{(\int |\nabla f|^2 dx)^{1/2}} = (1 - \frac{[\int (\nabla f \cdot \nabla g) dx]^2}{\int |\nabla f|^2 dx \int |\nabla g|^2 dx})^{1/2} \quad (8)$$

Furthermore, we can show that this distance function is a metric in the sense of equation 2 allowing negativity of  $k_I$ .

**Proof**

Notice that the distance is symmetric with respect to  $f$  and  $g$ . Moreover, it is always non-negative and is zero if and only if  $f$  and  $g$  are equivalent image classes. Thus we only need to show the triangular inequality to finish our proof. That is, we have to prove the following inequality for each pair of  $f$  and  $g$ :

$$\text{dist}(f, g) + \text{dist}(g, h) \geq \text{dist}(f, h) \quad (9)$$

Let

$$c_1 = \min_c \int |\nabla f - c \nabla g|^2 dx, \quad (10)$$

$$c_2 = \min_c \int |\nabla g - c \nabla h|^2 dx, \quad (11)$$

$$c_3 = \min_c \int |\nabla f - c \nabla h|^2 dx. \quad (12)$$

and  $k$  be such that

$$\nabla k = \nabla g - c_2 \nabla h. \quad (13)$$

We have

$$\left( \int |\nabla f - c_3 \nabla h|^2 dx \right)^{1/2} = \min_c \left( \int |\nabla f - c \nabla h|^2 dx \right)^{1/2} = \min_c \left( \int \left| \nabla f - \frac{c}{c_2} \nabla g + \frac{c}{c_2} \nabla k \right|^2 dx \right)^{1/2} \quad (14)$$

Now take  $c = c_2 c_1$  and recall that  $c_1 = \frac{\int (\nabla f \cdot \nabla g) dx}{\int |\nabla g|^2 dx}$ . We obtain

$$\begin{aligned} & \left( \int |\nabla f - c_3 \nabla h|^2 dx \right)^{1/2} \leq \left( \int |\nabla f - c_1 \nabla g + c_1 \nabla k|^2 dx \right)^{1/2} \leq \left( \int |\nabla f - c_1 \nabla g|^2 dx \right)^{1/2} + c_1 \left( \int |\nabla k|^2 dx \right)^{1/2} \\ & = \left( \int |\nabla f - c_1 \nabla g|^2 dx \right)^{1/2} + \frac{\int \nabla f \cdot \nabla g dx}{\left( \int |\nabla f|^2 dx \int |\nabla g|^2 dx \right)^{1/2}} \frac{\left( \int |\nabla f|^2 dx \right)^{1/2}}{\left( \int |\nabla g|^2 dx \right)^{1/2}} \left( \int |\nabla g - c_2 \nabla h|^2 dx \right)^{1/2} \\ & \leq \left( \int |\nabla f - c_1 \nabla g|^2 dx \right)^{1/2} + \frac{\left( \int |\nabla f|^2 dx \right)^{1/2}}{\left( \int |\nabla g|^2 dx \right)^{1/2}} \left( \int |\nabla g - c_2 \nabla h|^2 dx \right)^{1/2}. \end{aligned} \quad (15)$$

Now divide both sides by  $(\int |\nabla f|^2 dx)^{1/2}$  and we get the desired inequality.

### From the metric to our models

Now we could state the following models corresponding to the distance function defined above. We are going to propose three different models. The difference between the first two models is whether we allow the negativity of  $k_l$  in equation (2). The third of them will be a further generalization of the first two. Each one of them has slightly different properties and leads to its own partial differential equation. These equations are very interesting in their own rights and are closely related to the heat equation.

For two images  $f$  and  $g$  in  $\Omega$ , we look at the following minimization problems to match the gradient field of  $f$  to the gradient field of  $g$ :

#### Model A

$$\min_f E(f) = \sqrt{\int |\nabla f|^2 dx \int |\nabla g|^2 dx} - \int (\nabla f \cdot \nabla g) dx. \quad (16)$$

It is easy to see that this model transforms  $f$  to another image equivalent to  $g$  in the sense of equation (2).

#### Model B

$$\min_f E(f) = \int |\nabla f|^2 dx \int |\nabla g|^2 dx - (\int \nabla f \cdot \nabla g dx)^2. \quad (17)$$

This model is different from model A in that we allow negative values of  $k_l$  in equation (2) in the definition of equivalent images.

### Model C

We propose a third model that is even more nonlinear by applying the absolute function to the integrand of the last term on the right hand side of equation (17). It also turns out to be the most interesting of the three models. By equivalent class in this space we mean that the following cost function between two images is 0:

$$\min_f E(f) = \int |\nabla f|^2 dx \int |\nabla g|^2 dx - \left( \int |\nabla f \cdot \nabla g| dx \right)^2 . \quad (18)$$

It is easy to see that images that are equivalent in the sense of equation (16) and (17) are also equivalent in equation 18 but not vice versa.

## 3. Euler-Lagrange Equations and the Corresponding Gradient Descent Time Dependent Partial Differential Equations

The Euler-Lagrange equations associated with these models are presented in the following

### Model A:

$$-\sqrt{\int |\nabla g|^2 dx / \int |\nabla f|^2 dx} \Delta f + \Delta g = 0 \quad (19)$$

### Model B:

$$-(\int |\nabla g|^2 dx) \Delta f + (\int \nabla f \cdot \nabla g dx) \Delta g = 0 \quad (20)$$

**Model C:**

$$-(\int |\nabla g|^2 dx) \Delta f + (\int |\nabla f \cdot \nabla g| dx) \operatorname{div}(\operatorname{sgn}(\nabla f \cdot \nabla g) \nabla g) = 0 \quad (21)$$

A common way of solving these Euler-Lagrange equations is to introduce an artificial time  $t$  and let the initial image  $f_0$  be  $f$  at  $t=0$  and to employ the gradient descent that corresponds to the following time dependent partial differential equations to compute the steady state solutions.

**Model A:**

$$\frac{\partial f}{\partial t} = \sqrt{\int |\nabla g|^2 dx / \int |\nabla f|^2 dx} \Delta f - \Delta g . \quad (22)$$

**Model B:**

$$\frac{\partial f}{\partial t} = (\int |\nabla g|^2 dx) \Delta f - (\int \nabla f \cdot \nabla g dx) \Delta g . \quad (23)$$

**Model C:**

$$\frac{\partial f}{\partial t} = (\int |\nabla g|^2 dx) \Delta f - (\int |\nabla f \cdot \nabla g| dx) \operatorname{div}(\operatorname{sgn}(\nabla f \cdot \nabla g) \nabla g) . \quad (24)$$

The models we propose so far have a major disadvantage. Since they are not properly normalized, images are collapsed to constant images when time goes to infinity. In order to avoid this problem, we could normalize the energy so that the energy is always between 0 and 1 (this is exactly why we exclude constant images). In other words, we could also look at the following modified versions of our models.

### Model A'

$$\min_f E(f) = 1 - \frac{\int \nabla f \cdot \nabla g \, dx}{\left( \int |\nabla f|^2 \, dx \int |\nabla g|^2 \, dx \right)^{1/2}}. \quad (25)$$

The gradient descent PDE is:

$$\frac{\partial f}{\partial t} = \left( \frac{\int \nabla f \cdot \nabla g \, dx}{\int |\nabla f|^2 \, dx} \Delta f - \Delta g \right) \bigg/ \sqrt{\int |\nabla f|^2 \, dx \int |\nabla g|^2 \, dx}. \quad (26)$$

### Model B'

$$\min_f E(f) = 1 - \frac{\left( \int \nabla f \cdot \nabla g \, dx \right)^2}{\int |\nabla f|^2 \, dx \int |\nabla g|^2 \, dx}. \quad (27)$$

The gradient descent PDE is:

$$\frac{\partial f}{\partial t} = \left\{ \frac{\left( \int \nabla f \cdot \nabla g \, dx \right)^2}{\int |\nabla f|^2 \, dx} \Delta f - \left( \int \nabla f \cdot \nabla g \, dx \right) \Delta g \right\} \bigg/ \int |\nabla f|^2 \, dx \int |\nabla g|^2 \, dx. \quad (28)$$

### Model C'

$$\min_f E(f) = 1 - \frac{\int |\nabla f \cdot \nabla g| \, dx}{\left( \int |\nabla f|^2 \, dx \int |\nabla g|^2 \, dx \right)^{1/2}} \quad (29)$$

The gradient descent PDE is:

$$\frac{\partial f}{\partial t} = \left[ \frac{\int |\nabla f \cdot \nabla g| \, dx}{\int |\nabla f|^2 \, dx} \Delta f - \operatorname{div}(\operatorname{sgn}(\nabla f \cdot \nabla g) \nabla g) \right] \bigg/ \sqrt{\int |\nabla f|^2 \, dx \int |\nabla g|^2 \, dx}. \quad (30)$$

## 4. Implementation

In this section, the Adaptive Operator Splitting (AOS) scheme—a semi-implicit scheme—will be discussed. The fully explicit scheme spends a short time in each iteration, but it is only conditionally stable and takes many iterations to reach the steady state solution. On the other hand, the semi-implicit scheme is unconditionally stable and takes fewer iterations to reach the steady state solution. The same scheme is used in later chapters for solving image registration problems with the Horn and Schunck regularization.

The fully explicit scheme can be easily constructed by using central difference scheme for terms on the right hand side, which are the spatial derivatives of the equation, and forward Euler discretization on the time derivative. Let us write down the scheme for implementing model C'. Since the denominator on the right hand side is always positive, by ignoring the denominator we still have a gradient descent time evolution PDE. That means the PDE we solve looks like

$$\frac{\partial f}{\partial t} = \frac{\int |\nabla f \cdot \nabla g| dx}{\int |\nabla f|^2 dx} \Delta f - \text{div}(\text{sgn}(\nabla f \cdot \nabla g) \nabla g). \quad (31)$$

First let us recall the standard notations for finite differences. Let  $h$  be the spatial step size, and  $(x_i, y_j) = (ih, jh)$  be the grid points ( $0 \leq i, j \leq m-1$ ). Let  $f_{i,j}^n = f(n \Delta t, x_i, y_j)$  be an approximation of  $f(t, x, y)$ . The usual finite difference operators are then

$$\Delta_0^x f = (f_{i+1,j} - f_{i-1,j})/2h, \Delta_+^x f = (f_{i+1,j} - f_{i,j})/h, \Delta_-^x f = (f_{i,j} - f_{i-1,j})/h. \quad (32)$$

Similar operators are defined for approximating the derivatives with respect to  $y$ . The Laplacian of  $f$  at point  $(x_i, y_j)$  then could be approximated by

$$\Delta_{i,j} f = \frac{\Delta_+^x f - \Delta_-^x f}{h} + \frac{\Delta_+^y f - \Delta_-^y f}{h}. \quad (33)$$

Using these notations, we calculate the following coefficients at time step  $n$  by imposing the Neumann boundary conditions

$$\int |\nabla f^n \cdot \nabla g| dx = \sum_{i,j} |\Delta_0^x f_{i,j}^n \cdot \Delta_0^x g_{i,j} + \Delta_0^y f_{i,j}^n \cdot \Delta_0^y g_{i,j}|, \quad (34)$$

$$\int |\nabla f^n|^2 dx = \sum_{i,j} |\Delta_0^x f_{i,j}^n \cdot \Delta_0^x f_{i,j}^n + \Delta_0^y f_{i,j}^n \cdot \Delta_0^y f_{i,j}^n|. \quad (35)$$

Then the fully explicit scheme computes the solution  $f_{i,j}^{n+1}$  at next time level by

$$\Delta_+^t f_{i,j}^n = \frac{\int |\nabla f^n \cdot \nabla g| dx}{\int |\nabla f^n|^2 dx} \Delta_{i,j} f^n - \text{sgn}(\Delta_0^x f_{i,j}^n \cdot \Delta_0^x g_{i,j} + \Delta_0^y f_{i,j}^n \cdot \Delta_0^y g_{i,j}) \Delta_{i,j} g. \quad (36)$$

Notice there is a natural CFL type restriction on the step size in the form that  $\frac{\Delta t}{h^2}$  should

be bounded by a constant which depends on the dimensionality of the problem.

### Semi-implicit scheme

The AOS scheme was used in image processing by Weickert for efficiently solving non-linear diffusion filtering problem [19-21, 24-29]. The main idea is to replace the

Laplacian operator on the right hand side at the current time step with the next step that results in an implicit scheme. To be more precise, let the vector  $\vec{f}^n = (f_{0,0}^n, f_{0,1}^n, \dots, f_{m-1,m-1}^n)$  be a lexicographical ordering of the values at the grid points at time step  $n$  and  $A_l$  be the usual 3-point finite difference approximation operator of the second order derivative along the  $l$ -th space coordinate. We denote the finite

difference approximation of  $\frac{\int |\nabla f^n \cdot \nabla g| dx}{\int |\nabla f^n|^2 dx}$  described above by  $\alpha^n$  and let us also define

the vector  $\vec{F}^n$  by

$$\vec{F}^n = (F_{0,0}^n, F_{0,1}^n, \dots, F_{m-1,m-1}^n) \quad (37)$$

where

$$F_{i,j}^n = \text{sgn}(\Delta_0^x f_{i,j}^n \cdot \Delta_0^x g_{i,j} + \Delta_0^y f_{i,j}^n \cdot \Delta_0^y g_{i,j}) \Delta_{i,j} g . \quad (38)$$

The discretization now reads

$$\frac{\vec{f}^{n+1} - \vec{f}^n}{\Delta t} = \alpha^n \sum_{l=1}^d A_l \vec{f}^{n+1} - \vec{F}^n , \quad (39)$$

$$\text{or } \vec{f}^{n+1} = \left( I - \Delta t \cdot \alpha^n \sum_{l=1}^d A_l \right)^{-1} (\vec{f}^n - \Delta t \cdot \vec{F}^n) . \quad (40)$$

The trick of AOS is to replace the above problem by solving the following instead:

$$\vec{f}^{n+1} = \frac{1}{d} \sum_{l=1}^d \left( I - d \cdot \Delta t \cdot \alpha^n A_l \right)^{-1} (\vec{f}^n - \Delta t \cdot \vec{F}^n) . \quad (41)$$

This smart splitting has the same local truncation error as the original semi-implicit scheme and is of order one in time and order two in space. Moreover it is unconditionally stable. By splitting the operator into a coordinate-by-coordinate fashion we now only need to invert a tri-diagonal matrix along each coordinate and this allows an  $O(m^2)$  implementation by the Thomas algorithm. From our experience a speed-up of at least a magnitude of 10 compared to the fully explicit scheme could be achieved by using the AOS scheme presented here.

It should be noted that the AOS scheme could be applied to solve any inhomogeneous heat equation with a source term, and thus is used in later chapters to solve problems that involves the Horn and Schunck type regularization when computing deformation field in image warping.

## **5. Results**

### **Test of the basic models**

We will present our results on Model A, Model B and Model C'. For results presented in this section all calculations were performed on a 128 by 128 grid and the spatial step size  $h$  is 1.

#### **Model A**

We test this model by blending a circle to an ellipse. The test image pair is generated in the following way:

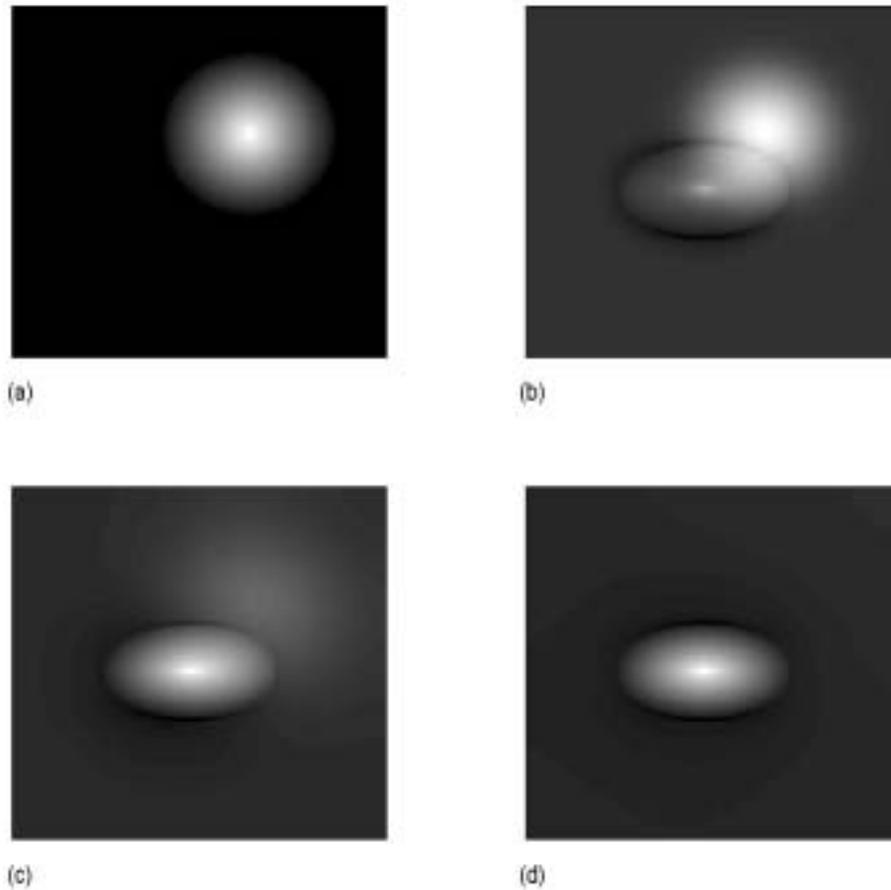
Target image  $g$ :

$$\begin{aligned} g(x, y) &= -\sqrt{3(x-60.5)^2 + (y-60.5)^2} + 30 && \text{when } \sqrt{3(x-60.5)^2 + (y-60.5)^2} \leq 30 \\ g(x, y) &= 0 && \text{else.} \end{aligned}$$

The initial image  $f_0$ :

$$\begin{aligned} f_0(x, y) &= -\sqrt{(x-80.5)^2 + (y-80.5)^2} + 30 && \text{when } \sqrt{(x-80.5)^2 + (y-80.5)^2} \leq 30 \\ f_0(x, y) &= 0 && \text{else.} \end{aligned}$$

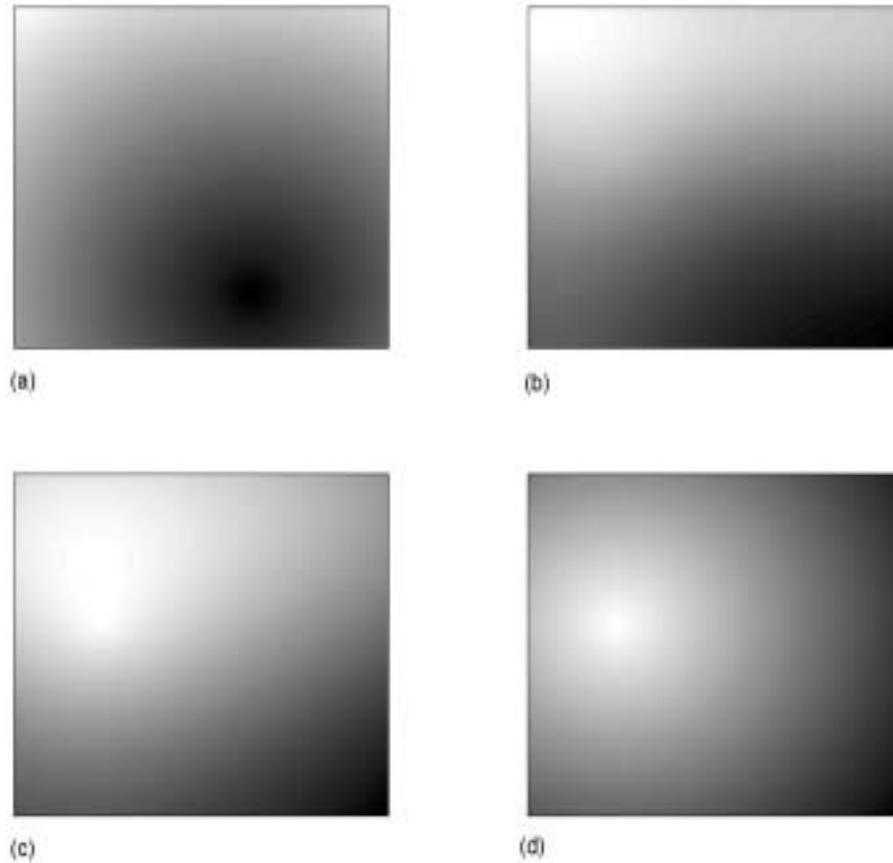
The image  $f$  is shown in Fig. 1(a). And the intermediate images during blending process are shown in Fig. 1(b)-(d). We notice that along the blending process the values outside the circles also change with time, which does not happen in simple interpolation technique. This phenomenon generates a flow-like motion that gives novel visual effects. These effects might be closely related to the resemblance of the governing PDE to the heat equation.



**Fig. 1: An image blending sequence from a circular shape in (a) to an ellipse. (b), (c), and (d) are the intermediate images along the sequence.**

## **Model B**

For this model we chose to use simple distance functions for both target and initial image. The target image  $g$  is the distance function to the point  $(30.5, 70.5)$  while the initial image  $f$  at  $t=0$  is another distance function to the point  $(80.5, 20.5)$ . The results are shown in Fig. 2. The initial image  $f$  is Fig. 2(a). The intermediate images are Figs. 2(b)-(d). Notice that the final image Fig. 2(d) is actually the inverse image of  $g$  instead of  $g$  itself.



**Fig. 2: An image blending sequence from the initial image (a): the distance function to the point  $x = 80.5, y = 20.5$  to the target, the distance function to the point  $x = 30.5, y = 70.5$ . (b), (c), and (d) are the intermediate images along the sequence. Notice that the image is actually transformed to the reverse of the target image.**

### **Model C'**

For this more complicated model we turn to the following pair of images that has two distinct objects in each of the image. The template image  $g$  is generated in the following way:

$$\begin{aligned}
g(x, y) &= -\sqrt{(x-34.5)^2 + (y-74.5)^2} + 30 && \text{when } \sqrt{(x-34.5)^2 + (y-74.5)^2} \leq 30 \\
g(x, y) &= -\sqrt{(x-80.5)^2 + (y-30.5)^2} + 20 && \text{when } \sqrt{(x-80.5)^2 + (y-30.5)^2} \leq 20 \\
g(x, y) &= 0 && \text{else}
\end{aligned}$$

We define the initial image  $f_0$  by:

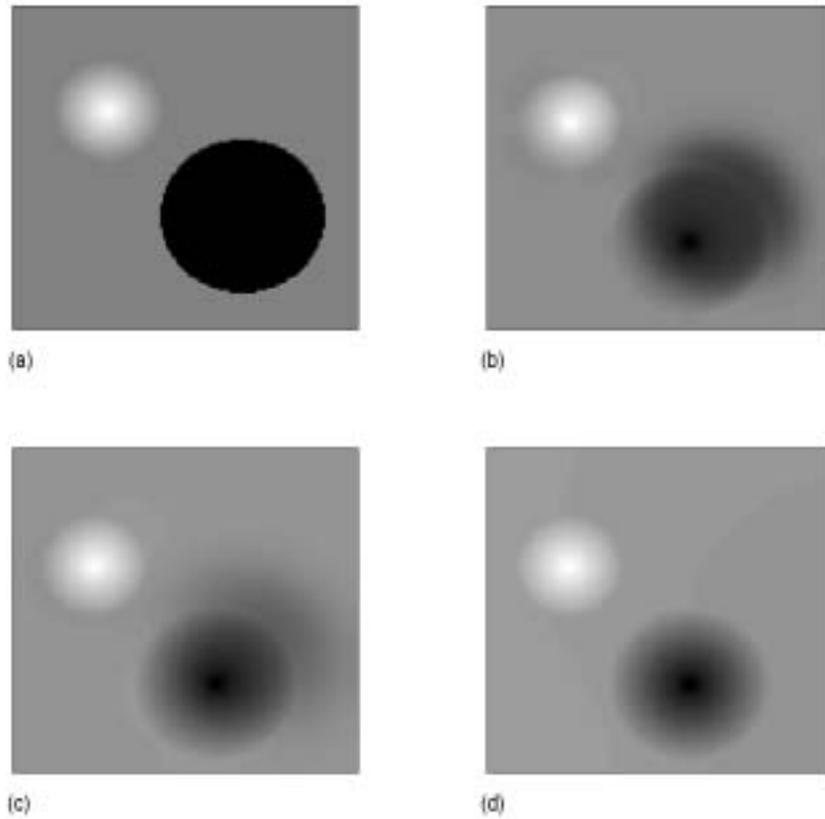
$$\begin{aligned}
f_0(x, y) &= 0 && \text{when } \sqrt{(x-44.5)^2 + (y-84.5)^2} \leq 30 \\
f_0(x, y) &= -\sqrt{(x-85.5)^2 + (y-35.5)^2} + 40 && \text{when } \sqrt{(x-85.5)^2 + (y-35.5)^2} \leq 20 \\
f_0(x, y) &= 20 && \text{else.}
\end{aligned}$$

These test images are such that there are two corresponding objects in the images and we would like to see if the initial image could be transformed into an image with the same two objects but with different orientation combination from the template image. The initial image  $f$  is shown in Fig. 3(a). The final result is shown in Fig. 3(d). We again plot the intermediate images in Fig. 3(b) and (c). In order to compare the result to the target image  $g$ , we also plot  $g$  in Fig. 4.

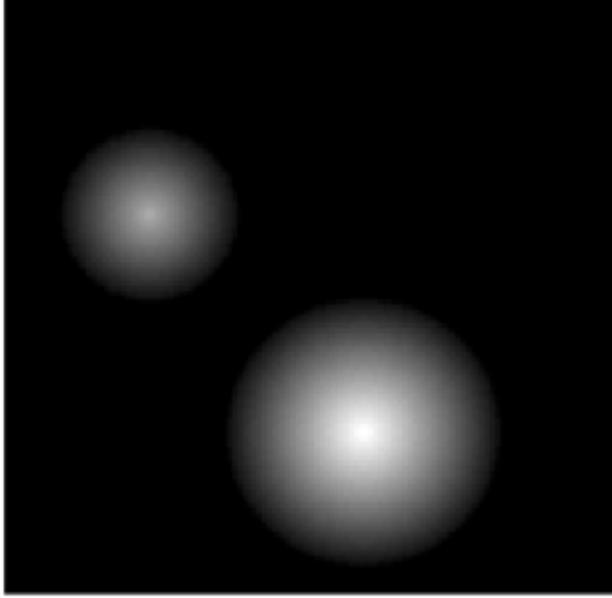
By comparing the final result Fig. 3(d) to the template image Fig. 4, we notice that one of the objects has been flipped along the  $x$ - $y$  plane while the relative contrast of these two objects remains the same. This is exactly what we expected to see. Interestingly enough, if we use a slightly different initial image to  $f$ , we could get different final images with various orientation combinations of the two objects (results not shown here).

This experiment suggested that this model depends heavily on the initial image and thus it is hard to predict what final image will be by simply looking at the initial image. At first sight this seems to limit the possible application of this model. Yet as shown in the

next section, model C' actually provides very promising results when we test it on real brain images.



**Fig. 3: An image blending sequence from the initial image (a) to the target image (shown in the next figure Fig. 4). (b), (c), and (d) are the intermediate images along the sequence. Notice that the initial image is transformed such that one of the two objects in the image is reverse of the target image (shown in Fig. 4).**



**Fig. 4: The target image in figure 3.**

### **Extension of the basic models and results from biomedical images**

Simple modification of model C' allows us to combine features of two images and to generate a hybrid or fusion image. We consider denoising an image  $f_0$  by solving  $f$  in the following problem:

$$\min E(f) = \frac{1}{2} \int (f - f_0)^2 dx + c \left[ 1 - \frac{\int |\nabla f \cdot \nabla g| dx}{\left( \int |\nabla f|^2 dx \int |\nabla g|^2 dx \right)^{1/2}} \right]. \quad (42)$$

The PDE that governs this minimization problem is:

$$\frac{\partial f}{\partial t} = (f_0 - f) + \left[ \frac{\int |\nabla f \cdot \nabla g| dx}{\int |\nabla f|^2 dx} \Delta f - \operatorname{div}(\operatorname{sgn}(\nabla f \cdot \nabla g) \nabla g) \right] \bigg/ \sqrt{\int |\nabla f|^2 dx \int |\nabla g|^2 dx} \quad (43)$$

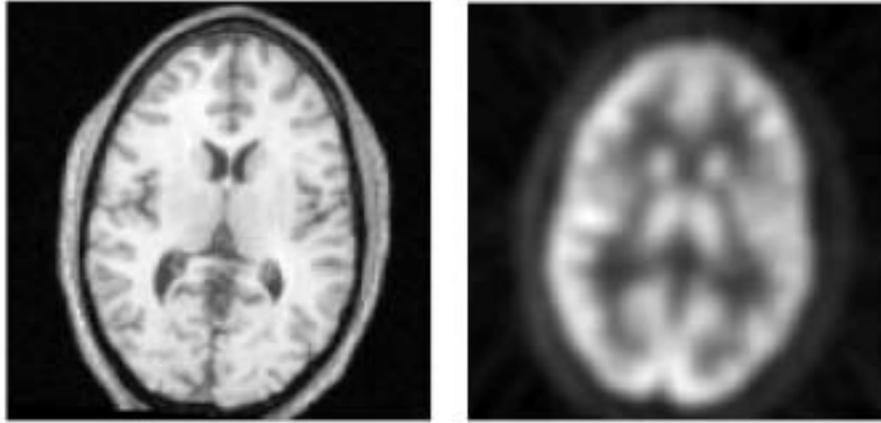
The above variational problem could be viewed as a “prior-based” image denoising. The idea is that  $f_0$  is a corrupted image that we would like to recover. Furthermore, we know that  $g$  contains the same information but we do not know exactly how the intensity values relate in these two images. By applying the above model we seek to find a recovered image that is close to  $g$  while constraining the result to be not far away from the initial image  $f_0$ .

In order to test this modified model, we turn to biomedical imaging for test images. In biomedical imaging, images of different modalities are often obtained for the same subject. This is especially true in brain imaging as different images could provide different information on brain structures and functions. Common imaging techniques available to brain imaging include Positron Emission Tomography (PET), Computed Tomography (CT) and Magnetic Resonance Imaging (MRI). We chose to test our model on MRI and PET images.

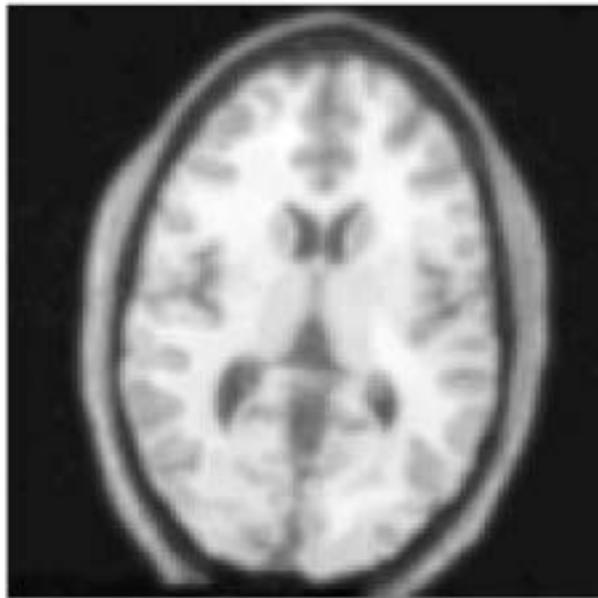
6-Fluoro-Deoxy-Glucose (FDG) PET imaging and MRI imaging of the brain were obtained from a normal subject in the Brain Trauma Project at UCLA. The available sequences in MRI images include SPGR, T2, and T2-star. All MRI images were first co-registered to PET using mutual information [30, 31] as the similarity measure. The mutual information between MRI and PET images were optimized using Powell’s multi-dimensional search algorithm.

To investigate the issue, we chose SPGR MRI images as the test images since SPGR images had the best spatial resolution and image quality. After MRI-PET co-registration was performed, plane 27 of SPGR images (left panel of figure 5) and PET images (right

panel of figure 5) were extracted. We then used MRI image as the initial image and the corresponding PET image as the template. The fusion image was calculated by minimizing the energy proposed in equation 42. The weight  $c$  of the regularizer is determined empirically. The final  $c$  used in this test was  $2 \times 10^6$ . The corresponding PDE was solved at  $t = 2$  to get the final fusion image. The result is shown in figure 6. Note that the overall contrast of the fusion images is very much like that of the initial SPGR image while the shape of the gray and white matters resemble that of the corresponding PET image.



**Fig. 5: Left panel: the SPGR MRI brain image of a control subject. Right Panel: The PET FDG brain image of the same plane from the same subject.**



**Fig. 6: The hybrid image of the two brain images in figure 5.**

## Chapter 4

# Computational Anatomy and the Semi-Lagrangian Level Set Method

*Abstract. In this chapter, we review the latest results of computational anatomy. Most recently, object warping in the computational framework is being formulated using diffeomorphisms generated through infinite dimensional group actions. We will give an overview of this approach. Secondly, we will briefly describe the semi-Lagrangian level set method. The semi-Lagrangian implementation of the level set method provides a theoretical link between the level set method and computational anatomy and is the foundation on which we will build our level set based object matching.*

## 1. Introduction

Computational anatomy [32, 33] is an emerging new discipline that deals with analyzing and making sense of the large collection and database of brain imaging. A fundamental problem in computational anatomy is image warping, or dynamically mapping one brain dataset to another through diffeomorphic transformation. In this paper, we will focus on the matching of anatomically important objects, which could be volumes, shapes, surfaces, curves, and points.

Object warping is a challenging problem not only in computational anatomy but also in computer vision, pattern recognition as well as many other scientific fields. In the past decade, several strategies of non-rigid warping algorithms have been proposed that could be divided into two groups: landmark based and dense matching.

Landmark matching involves first identifying user-defined landmarks that need to be matched. By interpolating the discrete matching of the landmarks, one tries to obtain a dense diffeomorphism for the whole image. Dense matching starts by forming a cost function that is minimized when the objects are matched. In order to ensure smooth matching, a regularizing term on the deformation field is added.

In this chapter, we will use the terms template and study to denote the images to be matched. Let us denote the template image as  $T(x)$  and the study image as  $S(x)$  which are

images on the spatial domain  $\Omega \subset R^n$ . The problem of image warping is to find a displacement field  $u(x)$  at each point  $x$  such that a properly defined distance measure, which will be denoted by  $D(T,S,u)$ , between the deformed template and the study is minimized. The displacement field is a vector field such that given any displacement field  $u$  the deformed template is given by  $T(x-u)$ . The term displacement is used because it can be viewed as how a point in the template is moved away from its original location. The most common way to define the measure between the deformed template and the study image is based on the  $L^2$  norm

$$D(T, S, u) = \frac{1}{2} \int_{\Omega} |T(x-u) - S(x)|^2 dx. \quad (1)$$

Gradient descent of the corresponding Euler-Lagrange equation is often used to minimize this distance measure

$$\frac{\partial u(x, t)}{\partial t} = f(x, u(x, t)), f(x, u) = [T(x-u) - S(x)] \cdot \nabla T|_{x-u}. \quad (2)$$

The function  $f(x, u)$  (up to a sign), which is often called the force field or the body force, describes the derivative of the distance measure with respect to the displacement field  $u$ .

Several models for regularizing the deformation field have been proposed. We will give an overview by looking at those with the most theoretical interests

## 2. Small Deformation Matching through Regularizer on the Displacement

### Hyper-elastic Matching

In hyper-elastic matching [34, 35], the authors tried to draw analogy between image warping and deforming elastic plates. Under the assumption of linear elasticity, which holds for relatively small displacement field only, we arrive at the following equation that should hold at equilibrium

$$\mu\Delta u(x,t) + (\mu + \lambda)\nabla(\nabla\cdot u(x,t)) = f(x, u(x,t)). \quad (3)$$

Here  $\mu$  and  $\lambda$  are the *Lame* constants. Due to this linear elasticity assumption, large-magnitude displacements are severely penalized and thus hyper-elastic model is not suitable for problems in which large and highly nonlinear deformation is needed.

### The Horn and Schunck Functional

Another big category of regularizers, originally in the optical flow problem, was based on the regularizer first proposed by Horn and Schunck in [36, 37]

$$R(u) = \frac{1}{2} \sum_{j=1}^d \|\nabla u_j\|_{L^2}^2 \quad (4)$$

This penalty term is well known to smooth isotropically across the discontinuities. Thus, it is not suitable to regularize optical flow since discontinuities in the velocity field often

exist on the boundary of moving objects. The same regularizing term for image warping was proposed in [38] and the term fast diffusion registration was used.

Several variants of (4) have also been proposed to account for anisotropy. One of the earliest effort to account for anisotropy based on the Horn and Schunck functional is due to Nagel [39, 40] by smoothing less across image boundaries

$$R(u) = \nabla u_i D(\nabla I) \nabla u_i^T \quad (5)$$

Here  $D(\nabla I)$  is a regularized projection matrix perpendicular to the gradient of the image  $I$ . For detailed discussion on other variants of this regularizer, please refer to the review paper [41])

It should be noted that none of these ensures diffeomorphism although they are easier to compute for generating small deformation.

### **3. Large Deformation through Diffeomorphisms by Infinite Dimensional Group Actions**

#### **Formulations based on continuum mechanics**

As mentioned in the previous section, working on the displacement field itself will not guarantee diffeomorphism. In the past decade, many researchers have tried to establish rigorous theories based on continuum mechanics that ensure diffeomorphic transformation by working on the forward and inverse mapping directly (see [32, 33, 42-56] and the references therein).

In summary, Let  $g^{-1}(x)=x-u$  which models the deformation field, and  $G$  be the group which is formed by all the diffeomorphisms that map  $\Omega \in X$  to itself. We now define the orbit (or mathematical anatomy) of an anatomical image  $I$  under the left action of group  $G$

$$G \cdot I = \{I \circ g : g \in G\} \quad (6)$$

We need to define a distance between two configurations (denoted by  $\rho$ ) on the orbit by quantifying the diffeomorphism needed to go from one configuration to the other. Moreover, we require this distance to be left-invariant with respect to the group action. Given the fact that these two configurations could be viewed as being generated from a reference element which could be taken to be any element on the orbit, the left invariance simply states that this distance is independent of the choice of this reference element. In mathematical terms, we require that

$$\rho(I, I') = d(id, g)|_{I=g \cdot I'} = d(g' \cdot id, g' \cdot g)|_{g' \cdot I = g' \cdot g \cdot I'} \quad (7)$$

Here  $g$  is the diffeomorphism that carries  $I$  to  $I'$ , and the second equality states the left-invariance property of this distance.

Using formulations borrowed from continuum mechanics, we can construct the distance by associating a path that links two elements in a mathematical anatomy. A path  $g$  in  $G$  is defined as a diffeomorphic flow in  $[0,1]$

$$g(x, t) = g_t(x), g_t \in G \text{ for } \forall t \in [0, 1] \quad (8)$$

A path  $g$  is linked to its velocity field  $v_t$  by the following equation

$$\frac{\partial}{\partial t} g_t(x) = v_t(g_t(x)) \quad (9)$$

Once the forward path  $g_t$  is defined, the inverse path is uniquely determined by

$$g_t^{-1}(g_t(x)) = x \quad (10)$$

Let us first define  $D$ , the Jacobian operator, acted on a scalar function  $f$  is the row vector where the  $i$ -th element is the partial derivative of  $f$  with respect to the  $i$ -th spatial derivative

$$(Df)_i = \frac{\partial f}{\partial x_i} \quad (11)$$

If  $f = (f_1, f_2, \dots)$  is a multi-valued function, then the Jacobian operator acted on  $f$  is a matrix with the  $(i, j)$ -th element defined as

$$(Df)_{i,j} = \frac{\partial f_i}{\partial x_j} \quad (12)$$

By differentiating the above equation with respect to  $t$ , we get

$$\begin{aligned} 0 &= \frac{\partial g^{-1}(g(x,t),t)}{\partial t} + \sum_i \frac{\partial g^{-1}(g(x,t),t)}{\partial x_i} \frac{\partial g_i(x,t)}{\partial t} \\ &= \frac{\partial g^{-1}(g(x,t),t)}{\partial t} + Dg^{-1}(g(x,t),t)v(g(x,t),t). \end{aligned} \quad (13)$$

Thus we derive the relationship between the inverse map and the velocity

$$\frac{\partial}{\partial t} g_t^{-1}(x) = -Dg_t^{-1}(x)v_t(x). \quad (14)$$

As originally proposed by [53, 56], given a path  $g(x, t)$  and its associated velocity  $v(x, t)$ , we define the energy of the path by

$$\int_{t=0}^1 \|v_t\|_L^2 dt = \int_{t=0}^1 \int_{\Omega} \langle Lv_t(x), Lv_t(x) \rangle dx dt \quad (15)$$

Here  $L$  is a differential operator acting on the velocity field.

Following the discussion in [43], an intuitive way to understand these formulations is to start with a discretized path  $g$  in  $t \in [0, 1]$  (starting from the identity initially) associated with the incremental displacement field  $u_k$  in the form

$$g_k = (id + u_k) \circ (id + u_{k-1}) \circ \dots \circ (id + u_1) \quad (16)$$

$$u_k = v_k \Delta t \quad (17)$$

Here  $v_k$  is the average speed in  $k$ -th time interval

We also associate the path with the following energy

$$E(g) = \sum_i \|v_i\|_L^2 \quad (18)$$

From equation (16), we get

$$g_k - g_{k-1} = u_k \circ g_{k-1} \quad (19)$$

Now let  $\Delta t$  goes to zero, equation (19) recovers equation (9), and the discretized energy (18) becomes the continuous version in equation (15).

## Building metric on diffeomorphisms

Furthermore, let us define the momentum  $p$  by

$$p(x, t) = p_t(x) = L^\dagger L v_t(x). \quad (20)$$

Here  $L^\dagger$  is the adjoint operator of  $L$ . It can be shown that diffeomorphisms could be ensured under mild restrictions on the operator  $L$  (see [57]). We then have the following theorem which could be found in [42, 58].

### Theorem

Let  $g_0$  and  $g_1$  be elements in  $G$ , the function  $d$  defined as

$$d(g_0, g_1) = \left( \inf_{\frac{\partial}{\partial t} g_t^{-1} = -Dg_t^{-1} v_t; g_{t=0} = g_0, g_{t=1} = g_1} \int_{t=0}^1 \|v_t(x)\|_L^2 dt \right)^{\frac{1}{2}} \quad (21)$$

is a left-invariant metric on  $G$ . Moreover, the geodesic satisfies the Euler-Lagrange equation

$$\frac{dp_t}{dt} + (v_t \cdot \nabla) p_t + \text{div}(v_t) p_t + \sum_i \nabla((v_t)_i)(p_t)_i = 0. \quad (22)$$

The left-invariance of this metric comes from the fact that any path that link  $g_0$  and  $g_1$  also links  $g \cdot g_0$  and  $g \cdot g_1$  for any  $g$  in  $G$ . For a proof and more detailed discussion, please refer to [42] and the references therein.

The above equation is sometimes called the Template Matching Equation, and could be viewed as a generalized Euler equation or momentum equation of the Navier-Stokes equations. In the (incompressible) Euler equation for a perfect fluid, the kinetic energy is

$$\int_0^T \|v_t\|^2 dt = \int_0^T \int_{\Omega} \langle v_t, v_t \rangle dx dt, \quad (23)$$

or equivalently, we could assume the differential operator  $L$  is chosen to be the identity and the momentum is proportional to the velocity itself. Under this fluid dynamics set up, and consider an incompressible flow with divergence-free velocity field (and thus volume-preserving)

$$\operatorname{div}(v_t) = 0 \quad (24)$$

The relation in the theorem reduces to the ordinary Euler equation encountered in the Navier-Stokes equation

$$\frac{\partial v_t}{\partial t} + (v_t \cdot \nabla)v_t = \nabla \text{pressure} \quad (25)$$

To summarize, we introduce the differential operator  $L$  and generalize the Euler equation in such a way that we relax the volume-preserving constraint while diffeomorphism could still be guaranteed by imposing minor restrictions on the operator  $L$ . For a complete treatment in this direction, please refer to the classic text in mechanics [59] and also [60].

## Sufficient conditions for diffeomorphism

In [57], the author describes the conditions on the differential operator  $L$  that ensures a diffeomorphism on the flow. To summarize, in 3D the proper space in which the velocity field  $v_t$  should lie to guarantee diffeomorphic flow is the product space  $[W_0^{3,2}(\Omega)]^3$ , where the Sobolev space  $W_0^{3,2}(\Omega)$  is the closure of  $C_0^{3,2}(\Omega)$  under the norm

$$\|f\|_{W_0^{3,2}(\Omega)} = \left( \int_{\Omega} \sum_{|a| \leq 3} |D^a f|^2 dx \right)^{1/2} \quad (26)$$

Thus, diffeomorphisms could be guaranteed if the  $L$ -norm of the velocity field is finite and dominates the above norm component-wise.

## Building metric on mathematical anatomy

With the theorem (21), we now can construct a metric satisfying symmetry and triangular inequality between two elements  $I_1$  and  $I_2$  on mathematical anatomies

$$\begin{aligned} \rho(I_1, I_2) &= d(id, g) |_{I_1=I_2} \\ &= \left( \inf_{\frac{\partial}{\partial t} g_t^{-1} = -Dg_t^{-1}v_t, I_1=I_2(g_{t=1}), g_{t=0}=id} \int_{t=0}^{t=1} \|v_t(x)\|_L^2 dt \right)^{1/2} \end{aligned} \quad (27)$$

## Choices of the differential operator $L$

Different choices of  $L$  have been proposed. For example, the viscous fluid matching proposed by Christensen [61-63] used the Navier-Stokes formulation, which can be

viewed as the linear elastic operator applied to the velocity field. The advantage of this method is that it allows large-magnitude deformations since stress constraining the deformation relaxes over time. The partial differential equation that describes the deformation under this model could be written in the following form

$$\mu\Delta v(x,t) + (\mu + \lambda)\nabla(\nabla \cdot v(x,t)) = f(x, u(x,t)). \quad (28)$$

Here  $v$  is the velocity field and is related to the displacement field  $u$  in a Eulerian framework.

Another closely related choice that is widely used by the Miller's school [32, 42, 52] is

$$L = (-a\Delta + bI)^m, \quad m \in \mathbb{N}. \quad (29)$$

This category of the operators is self-adjoint, and acts separately on each dimension of the velocity field. Furthermore, the inverse of the operator  $L^\dagger L$  could be viewed as a low-pass filter, and thus is a smoothing operator. In this dissertation, we follow the idea of Miller's school and numerical examples are also carried out using this choice of operator. Numerically, Fourier transform techniques will be employed to inverse this operator, and will be described in more details in the next chapter.

## **Gradient descent, PDE, and smoothing filter**

The concept of applying regularization could be closely tied to smoothing filter applied to the force field as discussed in [43, 48].

## 4. The semi-Lagrangian Level Set Method

The original level set method for moving an interface is a Eulerian framework. However, this Eulerian reference limits the application of the level set method in computational anatomy and related fields where the underlying grid deformation is the main interest. In order to overcome this difficulty, the semi-Lagrangian level set method as proposed in [64-66] should be employed rather than the original level set method.

The main concept in the semi-Lagrangian approach is backtracking the front instead of forward tracking. In other words, for each point on the front at time step  $n$ , we look at where this point was at time step  $n-1$ . Because of this backtracking nature, the semi-Lagrangian implementation shares the same advantage as the original level set method, namely, allowing topological changes.

The following is an intuitive derivation of the fundamental evolution equation in the semi-Lagrangian formulation. For any level set function  $\phi(x)$ , we calculate  $\phi^{n+1}(x)$  at time step  $n+1$  by linking it to the level set function in the previous time step  $\phi^n(x)$  using the backward tracking displacement field  $u^n$  in the following way

$$\phi^{n+1}(x) = \phi^n(x - u^n) \quad (30)$$

For a concrete example, let us consider minimizing the area enclosed by the zero level set of the level set function

$$\min_u \int_{\Omega} H(\phi(x - u)) dx \quad (31)$$

The PDE for  $u$  reads

$$\frac{du}{dt} = \delta(\phi(x-u))\nabla\phi(x-u) \quad (32)$$

This relation implies that, with time discretization  $\Delta t$ , the displacement field  $u^n$  at time step  $n$  is obtained by the following equation

$$\frac{u^n}{\Delta t} = \delta(\phi^n)\nabla\phi^n \quad (33)$$

As in the original level set formulation, we can extend this equation to every point on  $\Omega$  other than the zero level set by the following intuitive way

$$\frac{u^n}{\Delta t} = 1 \cdot \frac{\nabla\phi^n}{|\nabla\phi^n|} \quad (34)$$

To explain why this is the right extension, we can re-derive the standard level set formulation for minimizing the area enclosed by the zero level set in which the level set function undergoes motion with constant speed -1. First, let us expand  $\phi^{n+1}(x)$  up to first order by using equation (34)

$$\begin{aligned} \phi^{n+1} &= \phi^n(x - u^n) \\ &\approx \phi^n(x) + \nabla\phi^n \cdot u^n \\ &= \phi^n(x) + \nabla\phi^n \cdot \frac{\nabla\phi^n}{|\nabla\phi^n|} \Delta t = \phi^n + |\nabla\phi^n| \Delta t \end{aligned} \quad (35)$$

Now let  $\Delta t$  goes to zero, equation (35) becomes the standard level set evolution equation with constant speed -1!

$$\frac{\partial\phi}{\partial t} - |\nabla\phi| = 0 \quad (36)$$

Using similar arguments, every standard level set formulation has now an equivalent semi-Lagrangian formulation. In fact, given any speed function  $F$ , the corresponding evolution equation in the semi-Lagrangian form is simply

$$\frac{du}{dt} + F \cdot \frac{\nabla \phi}{|\nabla \phi|} = 0 \quad (37)$$

If the speed function is available only on the zero level set, then the alternative evolution equation is

$$\frac{du}{dt} + \delta(\phi) F \cdot \nabla \phi = 0 \quad (38)$$

For solving this equation numerically, the standard technique for discretizing Hamilton-Jacobi equations could be used. For a concrete example, let us outline the first order upwind scheme for semi-Lagrangian level set method in the case of constant speed motion with speed minus one. In this case, we can use the following first order upwind scheme on the  $x$  derivative

$$\phi_x = \max(D_x^- \phi, -D_x^+ \phi, 0) \quad (39)$$

**1 Initialize  $\phi$ , pick time step  $\Delta t$  and set  $\Phi^0 = \phi$**

**2 At time step  $n$ , obtain the backward tracking displacement  $u$  by solving the following equation**

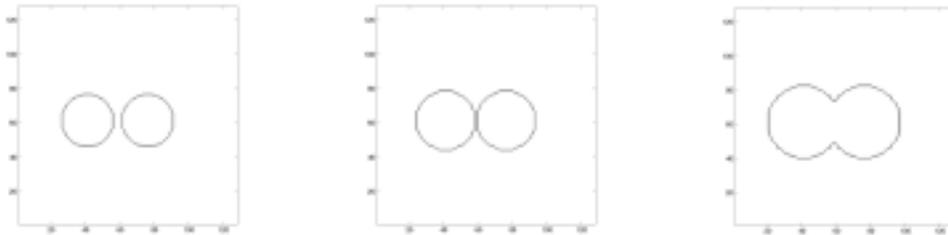
$$\frac{du}{dt} = \frac{\nabla \Phi^n}{|\nabla \Phi^n|} \quad (40)$$

**3 Solve  $\Phi^{n+1}$  by interpolation**

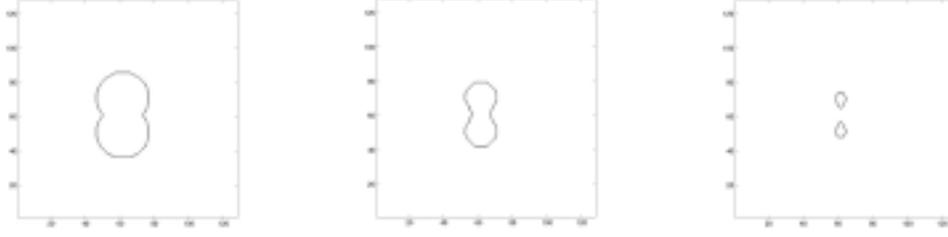
$$\Phi^{n+1}(x) = \Phi^n(x - u) \quad (41)$$

**Go back to main loop**

Motions under other speed functions could also be implemented in the semi-Lagrangian framework. In the following, two numerical examples showing front merging and breaking with constant speed 1 or  $-1$  are presented.



**Fig. 1: Front merging of two circles with constant speed implemented by the semi-Lagrangian level set method.**



**Fig. 2: Front breaking into two pieces with constant speed implemented by the semi-Lagrangian level set method.**

### **The modified semi-Lagrangian implementation**

In order to properly track the underlying grid deformation, the semi-Lagrangian implementation should be further modified by tracking the displacement field all the way to the initial data. In other words, instead of using the interpolated image at each step to calculate the next step, we store and increment the displacement field that allows us to track backward, at each time step, the displacement field with respect to the initial image. Thus, in computational anatomy, the semi-Lagrangian formulation we should adopt is slightly different from equation (37) and (38) in that we now have dependence of the displacement of  $u$  on the right hand side of the evolution equation

$$\frac{\partial u}{\partial t} = f \cdot \nabla \phi(x - u) \quad (42)$$

Here  $f$  will be called the “body force” or “force field” as this describes the direction along which the displacement field should follow. This equation should be interpreted in the sense of equation (37) rather than equation (38), due to the fact that the body force only acts on the objects being matched that are represented by zero level sets.

The extension of the body force outside of the zero level set depends on how the underlying displacement field is being modeled. We will essentially adopt the strategy of building diffeomorphisms through infinite dimensional group actions developed in this chapter. Starting from the next chapter, we will examine different cost functions that result in the corresponding force fields for matching different types of objects in both 2D and 3D. We will also examine minimization problems based on these cost functions that allow us to extend the body force based on the formulations discussed in this chapter and calculate the metric distance between objects.

## Chapter 5

# Two-Dimensional Object Matching using the Level Set Method

*Abstract-In this chapter, we examine the object matching problem in 2D using the level set method. Objects to be matched will be represented by level set functions and cost functions for matching different types of objects will be discussed. Semi-Lagrangian implementation of the level set method will be employed. In particular, we will link the level set based object matching to diffeomorphisms generated by infinite group actions. The modified Beg's algorithm based on variation with respect to the velocity field will be used to solve the corresponding time dependent partial differential equations.*

# 1. Shape Matching

Let us start with shape matching. A shape could be represented by a level set function with the boundary of the shape being the zero level curve of the level set function (positive value inside the shape and negative outside). Throughout this paper we will use the following notation. The shapes in the template image will be denoted by the level set functions  $\varphi_1, \varphi_2, \dots, \varphi_n$  and the corresponding shapes in the study image by  $\phi_1, \phi_2, \dots, \phi_n$ . Here  $n$  is the total number of pairs of shapes to be matched.

## One Pair of Overlapping Shapes

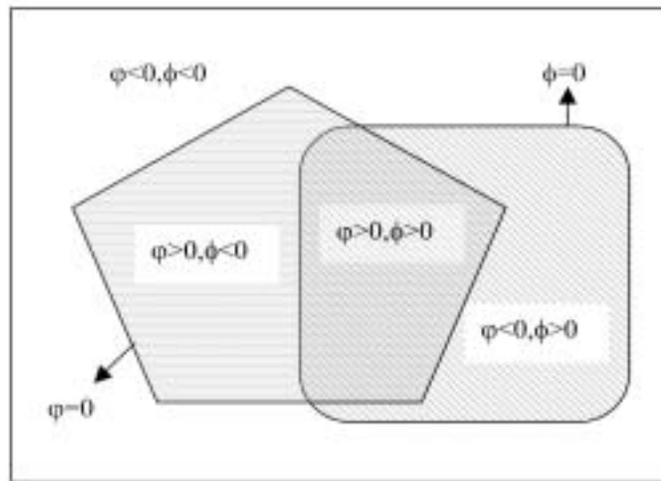
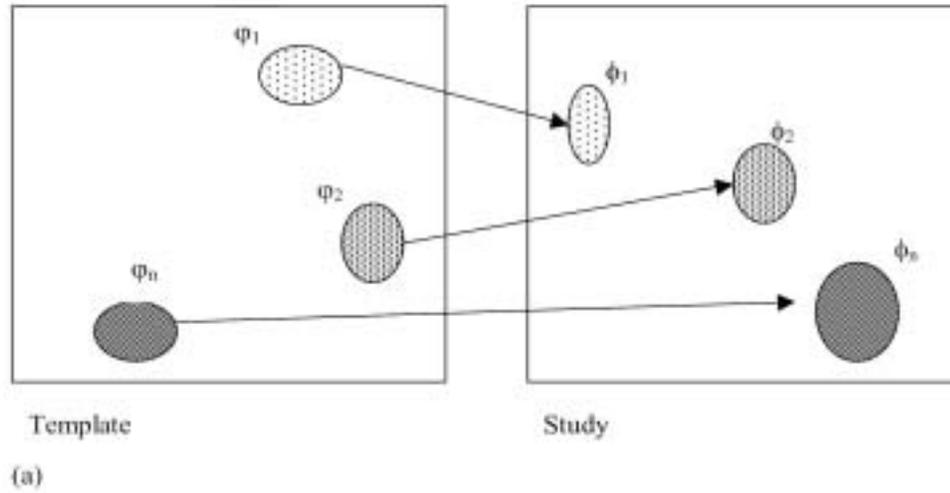
In order to derive a suitable distance measure that is always non-negative and only takes the value zero when the two level set functions match, we minimize the symmetric difference of the two level set functions (see [67, 68] for similar approaches).

$$D_{over}(T, S, t) = \int_{\Omega} H(\phi(x))[1 - H(\varphi(g_t^{-1}))] dx + \int_{\Omega} H(\varphi(g_t^{-1}))[1 - H(\phi(x))] dx. \quad (1)$$

The force field of this distance measure is

$$f_{over}(x, t) = [1 - 2H(\phi(x))] \delta(\varphi(g_t^{-1})) \nabla \varphi(g_t^{-1}). \quad (2)$$

Here  $H$  and  $\delta$  are the Heaviside and the delta function.



**Fig. 1: (a) Illustration of a general set-up for object matching with the level set based method. Objects in the template will be represented by level set functions  $\varphi$ 's. The corresponding objects in the study by level set functions  $\phi$ 's. (b): Illustration of four sub-regions divided by the level set functions, when the shapes in the template and study have overlap in space.**

## One Pair of Non-overlapping Shapes

The above distance measure does not work for non-overlapping shapes. The reason is that by minimizing the distance measure,  $\phi$  will simply shrink to a point and the cost function will reach a local minimum. To overcome this, we integrate  $-\phi$  in the area  $\phi > 0$ ,  $\phi < 0$ , and integrate  $-\phi$  in the  $\phi < 0$ ,  $\phi > 0$  and we now have to initialize the level set functions to be the signed distance function to their zero level sets.

$$D_{nonoverlapping}(T, S, t) = \int_{\Omega} -\phi(g_t^{-1})H(\phi(x))[1-H(\phi(g_t^{-1}))]dx + \int_{\Omega} -\phi(x)H(\phi(g_t^{-1})) [1-H(\phi(x))]dx. \quad (3)$$

In this case the force field is given by

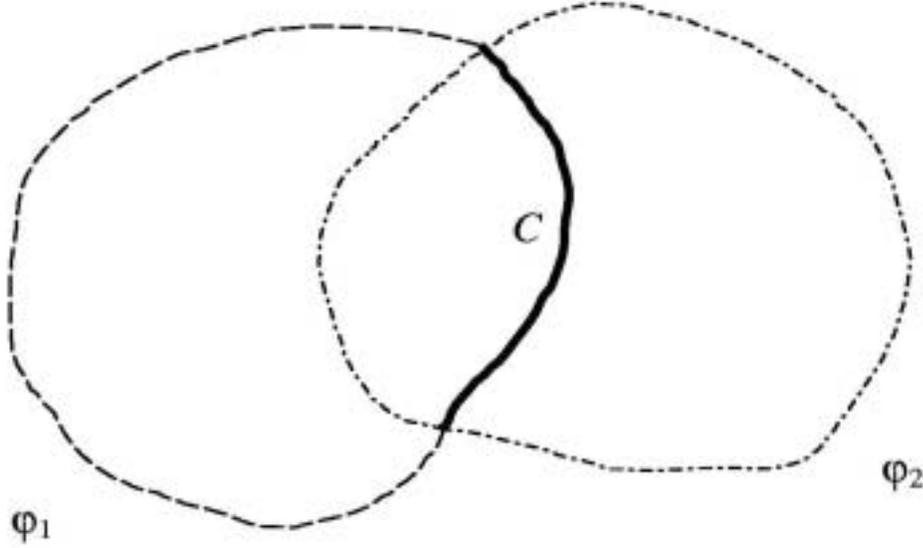
$$f_{nonoverlapping}(x, t) = -\{H(\phi(x))[1-H(\phi(g_t^{-1})) - \phi(g_t^{-1})\delta(\phi(g_t^{-1}))] + [1-H(\phi(x))]\phi(x)\delta(\phi(g_t^{-1}))\}\nabla\phi(g_t^{-1}). \quad (4)$$

## 2. Open Curve Matching

Now we will turn to the more interesting problem of matching open curves by level set functions. As before we will focus on only one pair of open curves, as matching multiple pairs is just a direct extension. Our task is then to find a deformation field that maps an open curve  $C$  in the template to another open curve  $C'$  in the study.

It has been well known that one of the disadvantages of level set approach is one level set function can not represent an open curve. Several remedies have been introduced. In this paper, we will follow the smart idea in [69] by appending a second level set function.

Please refer to figure 2. In order to find a representation of open curve  $C$ , we extend  $C$  to a closed curve (represented by the zero level set of  $\varphi_1$ ), and we further draw a second closed curve (represented by the zero level set of  $\varphi_2$ ), which crosses the zero level set of  $\varphi_1$  only at the two end points of  $C$ .



**Figure 2. Illustration of how to represent an open curve using level set functions.**

Then the open curve  $C$  could be written in the following way

$$C = \{x \mid \varphi_1(x) = 0 \text{ and } \varphi_2(x) > 0\}. \quad (5)$$

The open curve  $C$  in the study image could also be represented by two level set functions  $\phi_1$  and  $\phi_2$ . Let us further denote the distance functions of  $C$  and  $C'$  by  $D_S(x)$  and  $D_T(x)$ .

Inspired by the idea of geodesic active contours, we propose the following cost function

$$D_{curve}(T, S, t) = \int_{\Omega} D_s(x) \delta(\phi_1(g_t^{-1})) |\nabla \phi_1(g_t^{-1})| H(\phi_2(g_t^{-1})) dx + \int_{\Omega} D_T(g_t^{-1}) \delta(\phi_1(x)) |\nabla \phi_1(x)| H(\phi_2(x)) dx. \quad (6)$$

This cost function could be viewed as the sum of two line segment integrals with respect to the distance function of the other curve, and thus is nonnegative and zero only when the two open curves are equal. Notice that we could not omit either term of them and obtain the correct matching. For example, if we omit the second term, then the curve in the template will be matched to only a segment of the open curve in the study, and vice versa for omitting the first term. This discussion will be important later on when we proposed a second strategy for matching open curves in next chapter.

We will describe in detail how to obtain the Euler-Lagrange equation in this case. Note that all the derivations are in the sense of distributions. For the simplicity of derivation, we will work on the formulation with respect to the displacement field  $u$  instead of the path  $g$ . Thus we minimize the following cost function with respect to  $u$

$$\begin{aligned} D_{curve}(T, S, u) &= \int_{\Omega} D_s(x) \delta(\varphi_1(x-u)) |\nabla \varphi_1(x-u)| H(\varphi_2(x-u)) dx \\ &+ \int_{\Omega} D_r(x-u) \delta(\phi_1(x)) |\nabla \phi_1(x)| H(\phi_2(x)) dx. \end{aligned} \quad (7)$$

The force field contributed by the second term of equation is straightforward. We thus focus on how to obtain the force field given by the first term,  $G(u)$

$$G(u) = \int_{\Omega} D_s(x) \delta(\varphi_1(x-u)) |\nabla \varphi_1(x-u)| H(\varphi_2(x-u)) dx. \quad (8)$$

We want to compute the directional derivative of  $G$  along a test function  $v$

$$\lim_{\varepsilon \rightarrow 0} \frac{G(u + \varepsilon v) - G(u)}{\varepsilon}. \quad (9)$$

We first introduce the following notations

$$\delta_1 = \delta(\varphi_1(\mathbf{g}_t^{-1})), \quad (10)$$

$$H_2 = H(\varphi_2(\mathbf{g}_t^{-1})), \quad (11)$$

$$\delta_2 = \delta(\varphi_2(\mathbf{g}_t^{-1})). \quad (12)$$

By expanding the numerator in (8) up to first order, we get the following three terms

$$\int_{\Omega} (D_s \delta_1' H_2 |\nabla \varphi_1(x-u)|) (-\varepsilon \nabla \varphi_1|_{x-u} \cdot \mathbf{v}) dx \quad (13)$$

$$\int_{\Omega} D_s \delta_1 H_2 \frac{\nabla \varphi_1(x-u) \cdot \nabla (-\varepsilon \nabla \varphi_1|_{x-u} \cdot \mathbf{v})}{|\nabla \varphi_1(x-u)|} dx \quad (14)$$

$$\int_{\Omega} -\varepsilon (D_s \delta_1 \delta_2 |\nabla \varphi_1(x-u)|) \nabla \varphi_1|_{x-u} \cdot \mathbf{v} dx \quad (15)$$

Now we apply integration by parts on (14) and assume Neuman boundary conditions, we get

$$\begin{aligned} & - \int_{\Omega} \operatorname{div} (D_s \delta_1 H_2 \frac{\nabla \varphi_1(x-u)}{|\nabla \varphi_1(x-u)|}) (-\varepsilon \nabla \varphi_1|_{x-u} \cdot \mathbf{v}) dx \\ &= - \int_{\Omega} (D_s \delta_1' H_2 |\nabla \varphi_1(x-u)|) (-\varepsilon \nabla \varphi_1|_{x-u} \cdot \mathbf{v}) dx \\ & \quad - \int_{\Omega} \delta_1 \operatorname{div} (D_s H_2 \frac{\nabla \varphi_1(x-u)}{|\nabla \varphi_1(x-u)|}) (-\varepsilon \nabla \varphi_1|_{x-u} \cdot \mathbf{v}) dx. \end{aligned} \quad (16)$$

Thus, term (13) is canceled in the numerator of (8) by negative (13) from the expansion of (14). Let us further work on the integrand of the remaining term in (16)

$$\begin{aligned}
& \operatorname{div} \left( D_s H_2 \frac{\nabla \varphi_1(x-u)}{|\nabla \varphi_1(x-u)|} \right) \\
&= \frac{\{ H_2 \langle \nabla D_s, \nabla \varphi_1 |_{x-u} \rangle + D_s \delta_2 \langle \nabla \varphi_2 |_{x-u}, \nabla \varphi_1 |_{x-u} \rangle \}}{|\nabla \varphi_1(x-u)|} \\
&+ D_s H_2 \operatorname{div} \left( \frac{\nabla \varphi_1(x-u)}{|\nabla \varphi_1(x-u)|} \right).
\end{aligned} \tag{17}$$

Now we could put everything together and obtain the following force field given by this cost function

$$\begin{aligned}
f_{curve}(x, t) &= -\frac{\delta_1}{|\nabla \varphi_1(g_t^{-1})|} \{ H_2 \langle \nabla D_s, \nabla \varphi_1(g_t^{-1}) \rangle \\
&+ D_s \delta_2 \langle \nabla \varphi_2(g_t^{-1}), \nabla \varphi_1(g_t^{-1}) \rangle \} \nabla \varphi_1(g_t^{-1}) \\
&- \delta_1 D_s H_2 \operatorname{div} \left( \frac{\nabla \varphi_1(g_t^{-1})}{|\nabla \varphi_1(g_t^{-1})|} \right) \nabla \varphi_1(g_t^{-1}) \\
&+ D_s \delta_1 \delta_2 |\nabla \varphi_1(g_t^{-1})| \nabla \varphi_2(g_t^{-1}) + \delta(\phi_1) |\nabla \phi_1| H(\phi_2) \nabla D_T(g_t^{-1}).
\end{aligned} \tag{18}$$

### Matching Multiple Pairs of Shapes, Curves and Landmarks.

We should make a remark here that by combining different distance measures, we could solve matching problems in which multiple objects of different types are to be matched.

### 3. Incorporating the Level Set Method to Diffeomorphisms Generated by Infinite Dimensional Group Actions

#### Level Set Based Matching through Infinite Dimensional Group Actions

The level set based object matching could be integrated into the infinite dimensional group actions approach. Let us first recall the momentum  $p$  introduced in the previous chapter

$$p(x, t) = p_t(x) = L^+ L v_t(x) \quad (19)$$

Here  $v_t$  is the velocity field of the forward path  $g(x, t)$ . Then we have the following theorems that correspond to the inexact image matching problem and the space-time growth image matching problem as described in theorem 4.1 and 4.2 of [42]. We will refer to the following two matching problems as inexact level set based object matching and the space-time growth level set based object matching problem respectively.

#### Theorem 2 (Inexact Level Set Based Object Matching)

The path that minimizes the following inexact matching problem

$$\inf_{\frac{\partial}{\partial t} g_t^{-1} = -Dg_t^{-1} v_t, g_{t=0} = id} \int_{t=0}^1 \|v_t(x)\|_L^2 dt + D(T, S, t=1) \quad (20)$$

satisfies the geodesic relation in chapter 4 and the following boundary condition at  $t=1$

$$p(x, 1) - \left[ f(x, 1)^T Dg_{t=1}^{-1} \right]^T = 0 \quad (21)$$

### Theorem 3 (Space-Time Growth Level Set Based Object Matching)

The path that minimizes the following inexact matching problem

$$\inf_{\frac{\partial}{\partial t}g_t^{-1}=-Dg_t^{-1}v_t, g_{t=0}=id} \int_{t=0}^1 \|v_t(x)\|_L^2 + D(T, S, t)dt \quad (22)$$

satisfies the following Euler-Lagrange equation

$$\frac{dp_t}{dt} + (v_t \cdot \nabla)p_t + \text{div}(v_t)p_t + \sum_i \nabla((v_t)_i)(p_t)_i + [f(x, t)^T Dg_t^{-1}]^T = 0. \quad (23)$$

Here  $D(T, S, t)$  and  $f$  are the distance measure and force field defined in section 3.

For proof, simply combine the arguments in the previous chapter and [42] (also refer to [70]).

## 4. The Modified Beg's Algorithm

In this section, we will describe how to solve the inexact matching problem in 4.1 (the space-time growth problem could be handled similarly). Our algorithm is a modified version of the algorithm proposed by Faisal Beg (see [42]) for solving inexact image matching via variations with respect to the velocity field. The advantage of this approach is that we obtain the forward and inverse transformation and thus it allows us to do distance function re-initialization, which will be discussed in the following section (also refer to [70]).

## (Modified Beg's) Algorithm for Inexact Level Set Based Object Matching

Initialize  $v^{old}=0$ , choose a small number  $\varepsilon$ , for all  $t$  in  $[0 1]$ ,

### Step1 (fixed point iteration)

Solve  $g_t^{new}$  and  $g_t^{-1new}$  by

$$\frac{d}{dt} g_t^{new} = v_t^{old} (g_t^{new}), \quad \frac{d}{dt} g_t^{-1new}(x) = -Dg_t^{-1new}(x)v_t^{old}(x).$$

Compute  $g_1^{new}(g_t^{-1new}(\bullet))$ .

### Step2 (gradient decent)

$$v_t^{new} = v_t^{old} - \varepsilon \cdot grad,$$

$$grad = v_t^{old} - (L^\dagger L)^{-1} \left[ Dg_1^{new}(g_t^{-1new}) \left| \sum_i f_i(g_1^{new}(g_t^{-1new}(x)), t=1) \right. \right]^T.$$

Set  $v_t^{old} = v_t^{new}$ . Go back to step1 until convergence.

Here the algorithm is presented in the most general case in which multiple objects are to be matched. Thus, the force field has several components ( $f_i$ ) and each component is defined as in section 3 depending on the nature of the object. Notice that in step 2 the gradient descent direction is in the sense of the operator  $L^\dagger L$  instead of the usual  $L^2$  and thus could be interpreted as a smoothing step.

## 5. Distance Function Re-initialization

In the original level set method, the level set function has to be reinitialized as to prevent the slope from being too steep or flat. The same problem occurs in this semi-Lagrangian based implementation of the level set function. The signed distance function property of the level set functions used to represent the objects in the template is lost as the template is being carried by the forward path  $g$ . The problem is even worse when the information in level set functions are being used to calculate the decent direction that guides the flow of the template, as in the case of non-overlapping shapes and open curve matching. Thus, it would be desirable that we could re-initialize the level set functions in the template to be the signed distance function to their zero level sets. This gives a new interpretation of level set function re-initialization different from the definition of re-initialization in the original level set method based on Eulerian formulation. However, as in the original level set method, we only need to perform re-initialization once in several iterations, and thus does not increase the computation load by much.

Now let us describe how to re-initialize the level set function under the framework of the infinite group actions based on the theory of continuum mechanics. The goal is to ensure the signed distance function property of the level set functions in the template under the current flow. To this end, we re-initialize the level set function to its zero level set under the action of the backward path at time 1 ( $g^{-1}_{t=1}$ ), and update the level set function itself by composing back with the forward path  $g_{t=1}$ . To be more precise, we need to add the following step in 4.2 in the (modified) Beg's algorithm.

## **Additional step in the modified Beg's algorithm for re-initialization of the level set functions and distance functions**

*Compute  $\varphi_i^*(x) \triangleq \varphi_i(g_{t=1}^{-1new}(x))$*

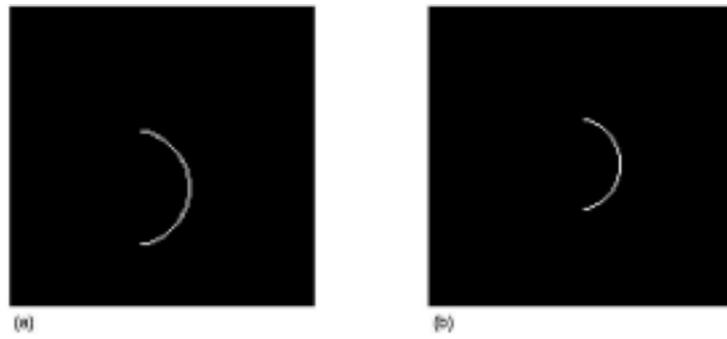
*Reinitialize  $\varphi_i^*(x)$  to its zero level set*

*Update  $\varphi_i \leftarrow \varphi_i^*(g_{t=1}^{new}(x))$*

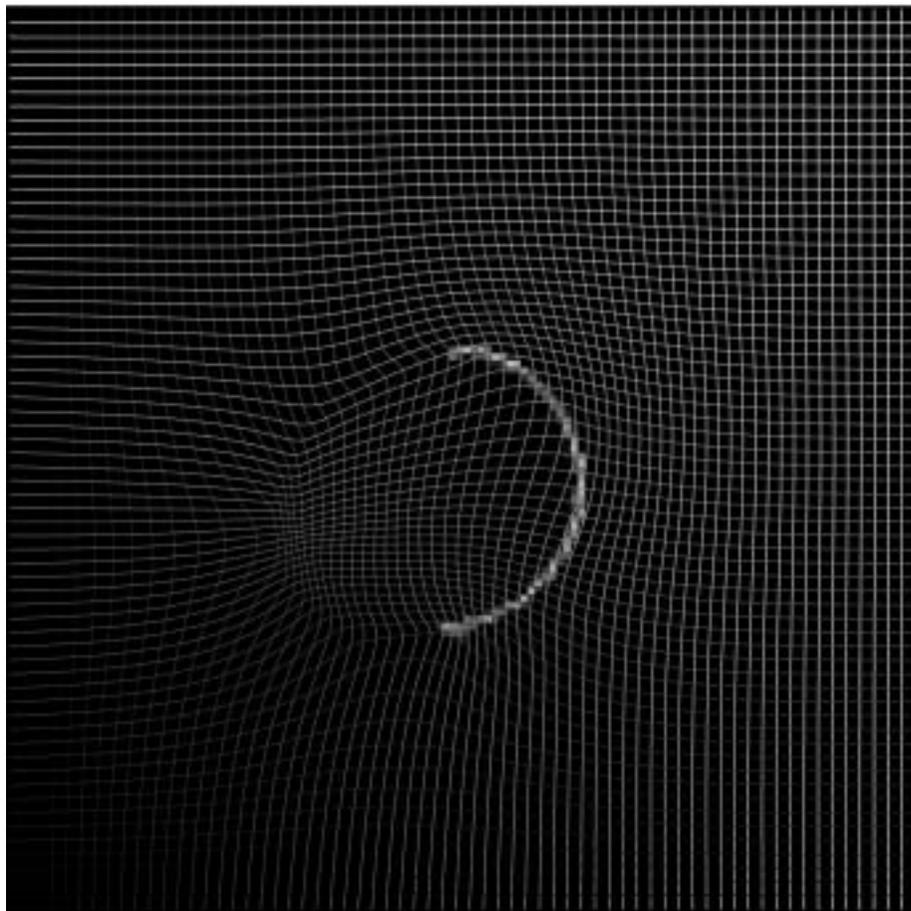
## **6. Results**

In this section, we present two simple numerical examples of matching two arcs and two pairs of oval shapes with the Horn and Schunck regularizer. More numerical examples based on the theory of continuum mechanics will be presented and discussed in the last chapter of this dissertation.

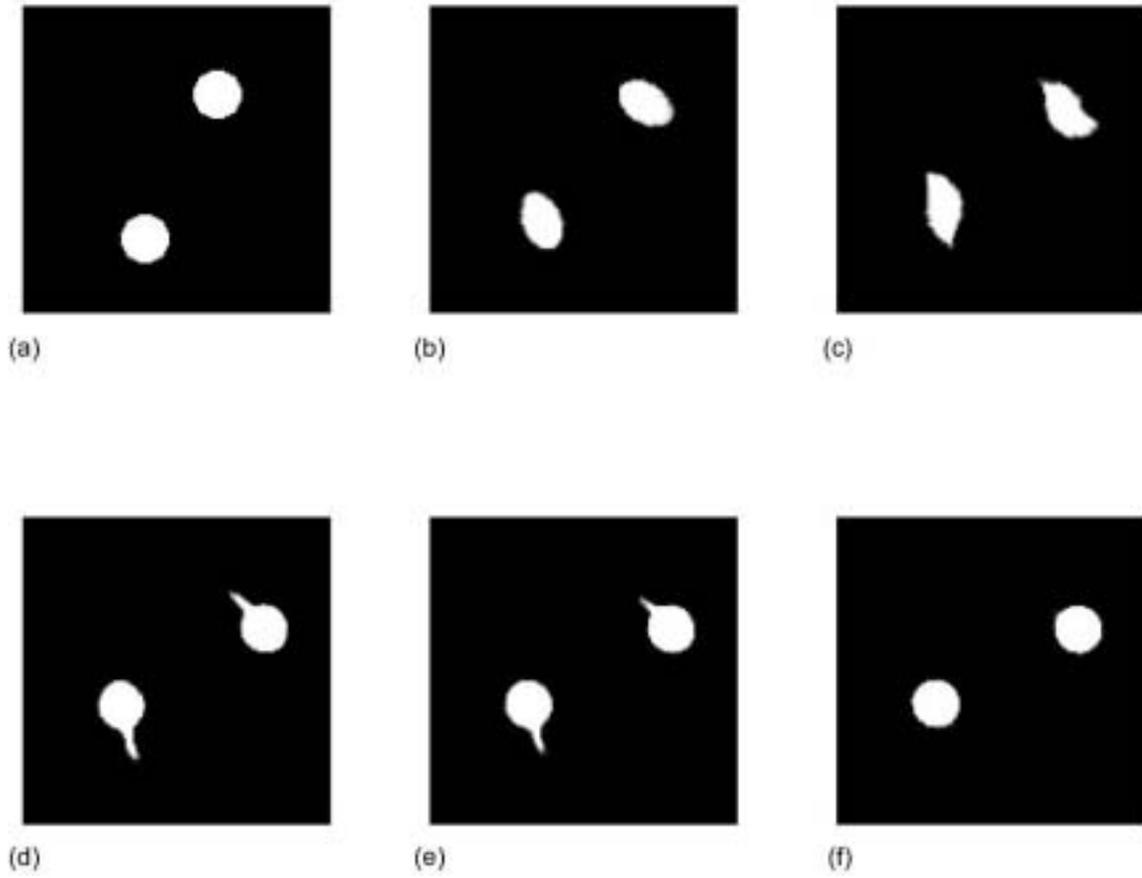
Figure 3 is a matching problem where (a) being the template and (b) the study. Figure 4 is the final deformed template with underlying grid deformation. Figure 5 is the transforming process from one pair of oval shapes to another non-overlapping pair, and the underlying grid deformation is shown in figure 6 (please refer to [70] for details on these two examples).



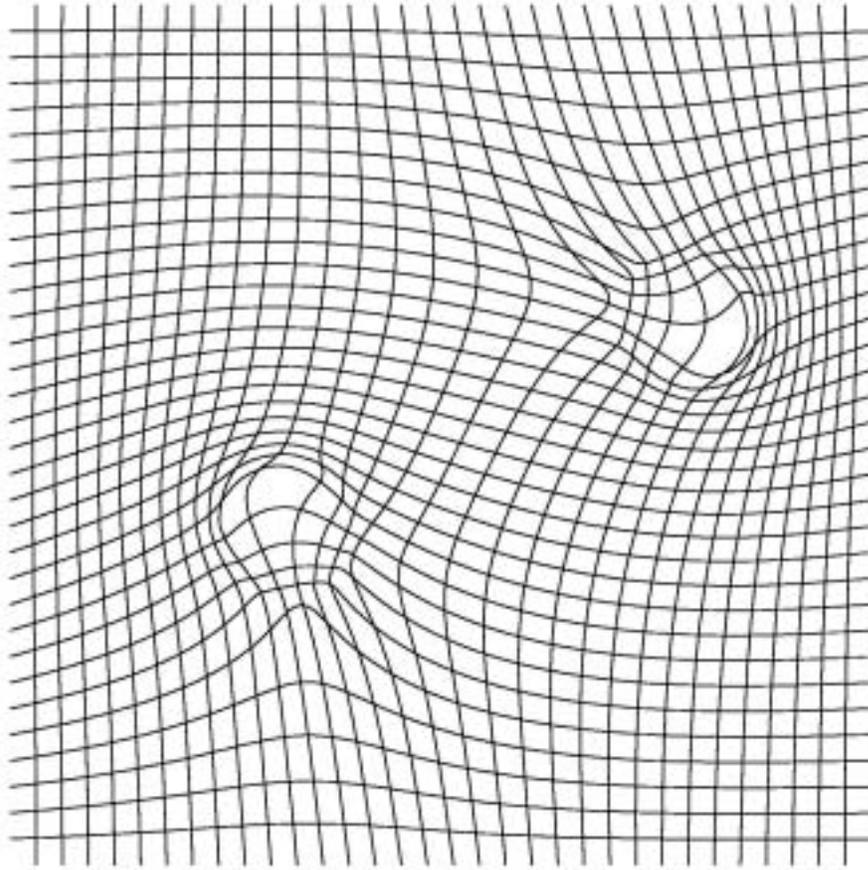
**Fig. 3.** A matching problem of two arcs in the template (a) and the study (b).



**Fig. 4.** The deformation field obtained by the propose method for matching figure 3(a) to 3(b)



**Fig. 5. The transformation process of the template (a) (one pair of oval shapes) to the study (f) (a pair of oval shapes without overlapping with the shapes in the template)**



**Fig. 6. The final deformation field of one pair of oval shapes to another non-overlapping pair in figure 5.**

## Chapter 6

# Length-Shortening Flow of Open Curves in Two-Dimensional Space

*Abstract-In this chapter, we will carefully re-examine the length shortening flow of open curves in 2D. We will address the issue of how to model the end points of an open curve as the intersection of the corresponding level set functions. The formulations presented are closely related to the representation and evolution of curves using the intersection of two level set functions in 3D. Moreover, a second approach for matching open curves will be presented.*

# 1. Basic Formulations for Length Shortening Flow of Curves in 2D

As in the previous chapter, let us use the intersection of two level set functions to represent an open curve (the zero level set of  $\phi$  enclosed inside of the second level set function  $\psi$ ). The Length minimizing flow is then computed in the following way

$$\min_{\phi, \psi} \int \delta(\phi) |\nabla \phi| H(\psi) dx \quad (1)$$

The corresponding Euler-Lagrange equation for  $\phi$  in this energy minimizing problem is

$$-\left[ \delta(\phi) \operatorname{div} \left( \frac{H(\psi) \nabla \phi}{|\nabla \phi|} \right) \right] = 0 \quad (2)$$

And the Euler-Lagrange equation for the level set function  $\psi$  is

$$-\delta(\psi) \delta(\phi) |\nabla \phi| = 0 \quad (3)$$

If we expand out the first equation, we get the following form

$$-\left[ \delta(\phi) \delta(\psi) \langle \nabla \phi / |\nabla \phi|, \nabla \psi \rangle + H(\psi) \delta(\phi) \operatorname{div} \left( \frac{\nabla \phi}{|\nabla \phi|} \right) \right] = 0 \quad (4)$$

We can also replace the delta function by the magnitude of the gradient to obtain the standard level set formulation

$$-\left[ \delta(\psi) \langle \nabla \phi, \nabla \psi \rangle + H(\psi) |\nabla \phi| \operatorname{div} \left( \frac{\nabla \phi}{|\nabla \phi|} \right) \right] = 0 \quad (5)$$

Let us examine the two terms in the above expansion. The second term is the usual mean curvature flow inside the level set function  $\psi$ . The first term only acts at the end points of the open curve. It points out the direction along which the length is locally decreased. An easy way to see this is the fact that the inner product between  $\nabla \phi$  and  $\nabla \psi$  becomes zero when the two gradients are orthogonal to each other, and thus locally no direction can be chosen to decrease the length.

Notice that the Euler-Lagrange equation in (3) also acts at the end points of the curve, and is scaled by the magnitude of the gradient of  $\phi$ .

## 2. How to Move the End Points of an Open Curve using Level Set Implicit Formulation

In this section, we will describe how to formulate and move the end points of an open curve in 2D using the level set implicit representation. The same technique allows us to do tracking of open curve that is the intersection between two level set functions in 3D.

Consider the following integral in 2D

$$\int f(x) \delta(\psi) \delta(\phi) |\nabla \phi \times \nabla \psi| dx \quad (6)$$

This integral will show up again in curve evolution in 3D as the (weighted) length of a curve (the intersection of two level set functions in 3D is a curve). Thus, we will treat the

technical details of how to derive the Euler-Lagrange equation for equation (6) in chapter 9 as the derivation is more general in 3D case and still holds in 2D.

Let us state the main result as follows

$$\int f(x)\delta(\psi)\delta(\phi)|\nabla\phi\times\nabla\psi|dx = \sum_i f(x = P_i) \quad (7)$$

Thus this integral recovers the sum of all the values of  $f$  evaluated at the intersection of level set functions  $\phi$  and  $\psi$ . This now gives us a new way to track the position of the intersections. Assume  $f$  is itself a distance function to a data set of points, then, by computing this integral, we now know how far the intersections are away from this set of data points.

### **3. A Second Strategy for Mapping Open Curves in 2D (When the End Points Are Known)**

In the previous chapter, we propose a cost function for matching two open curves in 2D. However, this approach requires that we need two closed curves for each open curve in the template and study along with the distance function to these two open curves. The proposed cost function relies on minimizing the sum of the weighted line integrals along the two open curves where the weights are the distance function with respect to the other open curve. This cost function can be reduced in the case when the end points on the study image are already known. Let us look at the cost function we propose for the open curve

$$\begin{aligned}
D_{curve}(T, S, t) = & \int_{\Omega} D_s(x) \delta(\phi_1(g_t^{-1})) |\nabla \phi_1(g_t^{-1})| H(\phi_2(g_t^{-1})) dx \\
& + \int_{\Omega} D_T(g_t^{-1}) \delta(\phi_1(x)) |\nabla \phi_1(x)| H(\phi_2(x)) dx.
\end{aligned} \tag{8}$$

Minimizing the first term in the above energy only ensures that the open curve in the template is matched to a segment of the open curve in the template (since all segments of a curve have line integral equal to zero with respect to this curve's distance function). The perfect matching is provided by the second matching term which requires that the open curve in the study will also be a segment of the final warped curve in the template. This ‘‘cross-validation’’ makes sure that the warped curve in the template and the curves in the study are perfectly matched.

However, if the end points are already know, then we can replace the second matching term by a term that enforces the matching of the end points using the techniques developed in the previous section

$$\begin{aligned}
D_{curve} = & \int_{\Omega} D_s \delta(\phi_1(g_t^{-1})) |\nabla \phi_1(g_t^{-1})| H(\phi_2(g_t^{-1})) dx \\
& + \lambda \int_{\Omega} D_{end} \delta(\phi_1(g_t^{-1})) \delta(\phi_2(g_t^{-1})) |\nabla \phi_1(g_t^{-1}) \times \nabla \phi_2(g_t^{-1})| dx
\end{aligned} \tag{9}$$

Here  $D_{end}$  is the distance function to the two end points of the curve in the study image. Notice if we use the above energy for matching curves, we no longer need the distance function for the open curve in the template  $D_T$  nor the two level set functions  $\phi_1$  and  $\phi_2$  for representing the open curve in the study image!!

The body force contributed by the first term in equation (9) is the same as before. We thus only have to obtain the body force contributed by the second term. Let us first define the orthogonal projector

$$P_v = I - \frac{v \otimes v}{|v|^2} \quad (10)$$

With this notation, we now state the body force contributed by the second term enforcing end point matching in the cost function. For

$$\begin{aligned} & -\operatorname{div} \left( \frac{P_{\nabla \varphi_2(\mathbf{g}_t^{-1})} \nabla \varphi_1(\mathbf{g}_t^{-1})}{|P_{\nabla \varphi_2(\mathbf{g}_t^{-1})} \nabla \varphi_1(\mathbf{g}_t^{-1})|} \Big| \nabla \varphi_2(\mathbf{g}_t^{-1}) \Big| D_{\text{end}} \right) \delta(\varphi_1(\mathbf{g}_t^{-1})) \delta(\varphi_2(\mathbf{g}_t^{-1})) \nabla \varphi_1(\mathbf{g}_t^{-1}) \\ & -\operatorname{div} \left( \frac{P_{\nabla \varphi_1(\mathbf{g}_t^{-1})} \nabla \varphi_2(\mathbf{g}_t^{-1})}{|P_{\nabla \varphi_1(\mathbf{g}_t^{-1})} \nabla \varphi_2(\mathbf{g}_t^{-1})|} \Big| \nabla \varphi_1(\mathbf{g}_t^{-1}) \Big| D_{\text{end}} \right) \delta(\varphi_1(\mathbf{g}_t^{-1})) \delta(\varphi_2(\mathbf{g}_t^{-1})) \nabla \varphi_2(\mathbf{g}_t^{-1}) \end{aligned} \quad (11)$$

However, we now have to identify the two end points of the open curve in the study as well as tune a parameter  $\lambda$  that adjusts the weight for the end point matching term.

### **Automatic initialization of open curves in 2D from data points**

Now if we are given an open curve in the form of some data points in 2D, and we want to generate the corresponding two level set functions that describe the open curve. The same cost function actually allows us to find these level set functions, since this cost function no longer requires the level set functions that describe the curve to be matched (now in the form of a collection of data points).

To achieve the goal, we can arbitrarily initialize the two level set functions and minimize the same cost function. All we need in the cost function is the distance function to these data points and identification of the two endpoints.

## Chapter 7

### Level Set based Landmark Matching

*Abstract-In this chapter, we will describe the formulation for landmark matching based on the level set method. The strategy is to represent landmarks by the intersection of two level set functions and use the formula we developed in the previous chapter to do matching. A simplified model is proposed that involves only one level set function for each landmark to be matched as compared to the first approach. The simplified model has a simpler force field that is easier to code, and numerically good results are obtained.*

## 1. Landmark Matching-First Approach

Landmark matching problem could also be formulated in the level set framework. Assume  $D_T$  is the distance function to the landmark in the template and the landmark is represented by the intersection of two level set functions  $\phi_1$ , and  $\phi_2$ . Similarly, we define  $D_S$ ,  $\phi_1$ , and  $\phi_2$  for the landmark in the study. With the formulation developed in the previous chapter, we can look at the following cost function for matching landmarks

$$D_{landmark}(T, S, t) = \int_{\Omega} D_S \delta(\phi_1(g_t^{-1})) \delta(\phi_2(g_t^{-1})) |\nabla \phi_1(g_t^{-1}) \times \nabla \phi_2(g_t^{-1})| dx + \int_{\Omega} D_T(g_t^{-1}) \delta(\phi_1) \delta(\phi_2) |\nabla \phi_1 \times \nabla \phi_2| dx \quad (1)$$

This cost function has the following force field

$$\begin{aligned} & -div \left( \frac{P_{\nabla \phi_2(g_t^{-1})} \nabla \phi_1(g_t^{-1})}{|P_{\nabla \phi_2(g_t^{-1})} \nabla \phi_1(g_t^{-1})|} |\nabla \phi_2(g_t^{-1})| D_S \right) \delta(\phi_1(g_t^{-1})) \delta(\phi_2(g_t^{-1})) \nabla \phi_1(g_t^{-1}) \\ & -div \left( \frac{P_{\nabla \phi_1(g_t^{-1})} \nabla \phi_2(g_t^{-1})}{|P_{\nabla \phi_1(g_t^{-1})} \nabla \phi_2(g_t^{-1})|} |\nabla \phi_1(g_t^{-1})| D_S \right) \delta(\phi_1(g_t^{-1})) \delta(\phi_2(g_t^{-1})) \nabla \phi_2(g_t^{-1}) \\ & + \delta(\phi_1) \delta(\phi_2) |\nabla \phi_1 \times \nabla \phi_2| \nabla D_T(g_t^{-1}) \end{aligned} \quad (2)$$

Now we need to pick the level set functions that describe the landmarks to be matched. For a landmark at position  $(x_0, y_0)$ , computationally the easiest way is to simply use the distance functions to the lines:  $x = x_0$  and  $y = y_0$  as the two level set functions. The advantages of this choice are that they are easy to generate and the gradient always has magnitude one and orthogonal to each other for the level set functions  $\phi_1$ , and  $\phi_2$  in the

study. With this in mind, we can drop the outer product of  $\phi_1$ , and  $\phi_2$  in both the cost function and the force field.

### **The reduction of the above formulation when distance functions on the template are re-initialized**

If the forward and backward mappings are both available, as in the case of the approach introduced in chapter 4 and 5, then equation (1) and (2) can be further reduced and has an intuitive interpretation at  $t = 1$ . In this case, at each step we re-initialize the level set functions  $\varphi_1$  and  $\varphi_2$  by

$$\begin{aligned}\varphi_1 &\equiv \varphi_1'(g_{t=1}(x)), \\ \varphi_2 &\equiv \varphi_2'(g_{t=1}(x)).\end{aligned}\tag{3}$$

Here the functions  $\varphi_1'$  and  $\varphi_2'$  are the signed distance functions to the lines:  $x = g_{t=1}(x_0)$  and  $y = g_{t=1}(y_0)$  respectively ( $g$  is the forward path). The force field at  $t=1$  now reads

$$\begin{aligned}& - \operatorname{div}(D_S \nabla \varphi_1') \delta(\varphi_1') \delta(\varphi_2') \nabla \varphi_1' \\ & - \operatorname{div}(D_S \nabla \varphi_2') \delta(\varphi_1') \delta(\varphi_2') \nabla \varphi_2' \\ & + \delta(\phi_1) \delta(\phi_2) \nabla D_T(g_t^{-1})\end{aligned}\tag{4}$$

The first two terms are simply a smoothed delta force source located at the current position of the landmark ( $g(x_0), g(y_0)$ ) pointing toward the target landmark; vice versa for the second term. Intuitively, equation (2) enforces the matching of the landmarks but looking at the current positions of the landmarks and apply forces that point from one landmark to the other.

## 2. Landmark Matching-Second Approach

Numerically, we notice that the first approach can be simplified by using one level set function to represent one landmark point. If  $\varphi$  is the distance function to the landmark in the template, and  $\phi$  the distance function to the corresponding landmark in the study, then we could consider the following energy that recovers the sum of the distance between these two landmarks under the action of the forward and inverse mapping

$$D_{landmark}(T, S, t) = \int_{\Omega} \delta(\varphi(g_t^{-1}))\phi(x)dx + \int_{\Omega} \delta(\phi(x))\varphi(g_t^{-1})dx \quad (5)$$

With this energy, the force field is then

$$F_{landmark}(x, t) = \{\delta'(\varphi(g_t^{-1}))\phi(x) + \delta(\phi(x))\}\nabla\varphi(g_t^{-1}). \quad (6)$$

This energy function deserves a closer look. At first sight, this function calculates the distance between landmarks with respect to two different level set functions. On second thought, it does not calculate the right quantity as in equation (1) since we are integrating the one dimensional delta function in the two dimensional integral (or three dimensional integral in the case of 3D landmark matching). In other words, we lose the proper scaling of the integrand. A good example of proper scaling is the equation we use to calculate the length of the zero level set

$$length = \int \delta(\phi)|\nabla\phi|dx. \quad (7)$$

Here the integrand is the one dimensional delta function (applied to the level set function) multiplied by the gradient of the level set function. Here the gradient of the level set

function serves as the right scaling, as the length of the zero level set depends neither on the scaling of the level set function nor on which level set function is used.

In the cost function (5) we propose for landmark matching, there is no such scaling. Yet numerically it performs well for small deformations.

If we examine the energy more carefully, the loss of proper scaling is balanced by the cross term. Imagine we apply this landmark matching energy to a pair of landmarks where the two landmarks are already at the same position (and thus  $g=\text{identity}$  is a stationary solution and the desired solution). Intuitively, by squeezing the grid toward the landmark in the template, we can make the first term smaller and smaller, but at the same time the second term will become larger, and vice versa if we dilate the grid away from the landmark in the template. This is why the energy function in theory is not correct yet numerically it works.

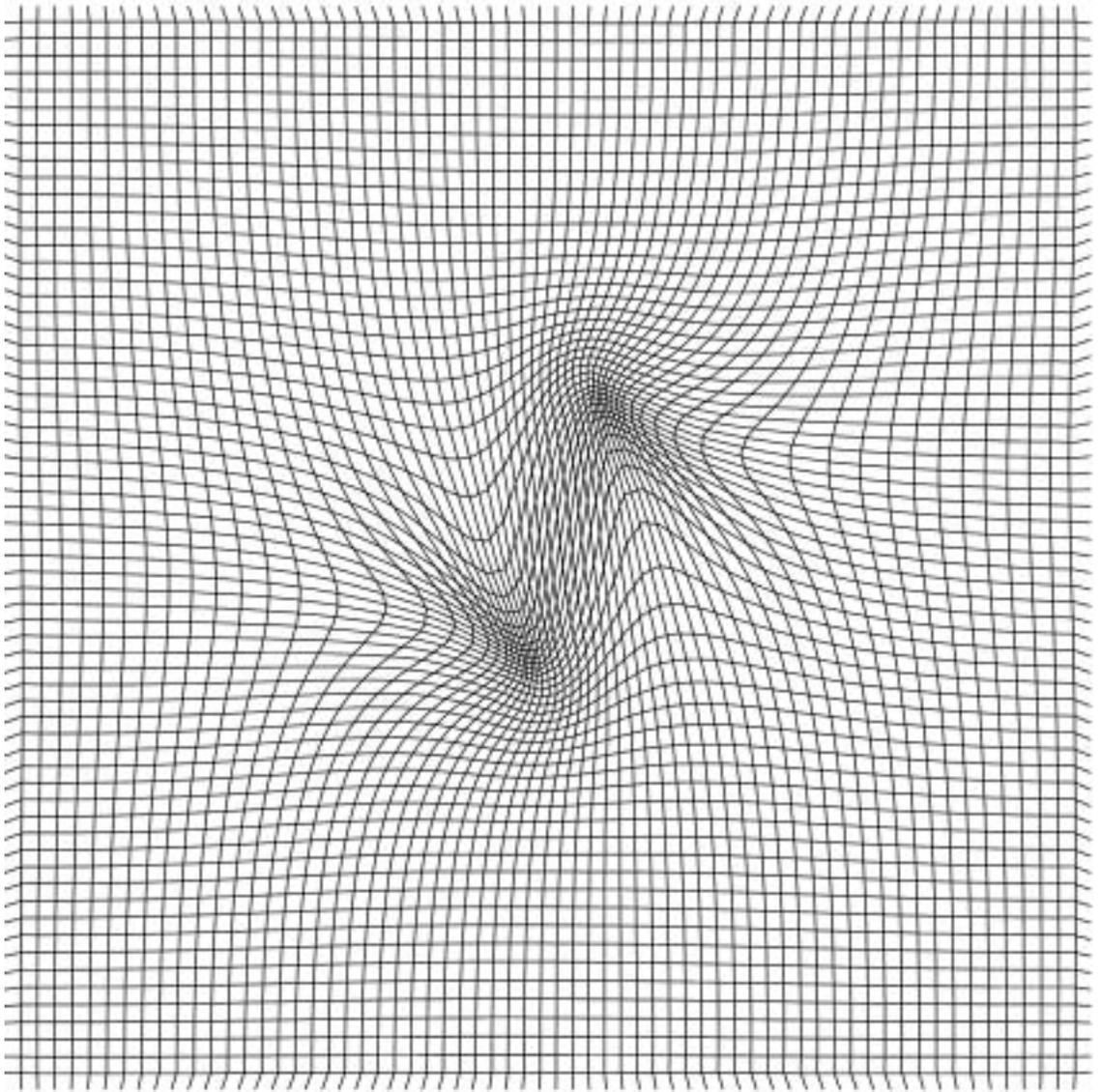
If we reinitialize distance function, then the situation is different. The energy always takes the same value no matter we squeeze or dilate the grid around the landmark as we always re-initialize  $\varphi$  to be the distance function under current grid deformation. Now the minimization problem will be governed by the regularization term and any grid deformation from the identity will be penalized. Thus,  $g$  being the identity remains a stationary solution and is the desired solution we are looking for.

Because of the above argument, we recommend that re-initialization should be carried out in the case of level set based landmark matching.

### 3. A numerical Example of Level Set based Landmark Matching

Figure 1 is the result of the level set based landmark matching generated by the second approach with distance function re-initialization. The landmarks to be matched are (25, 30) to (30, 25), and (40, 35) to (35, 40) on a 64 by 64 image. The regularizing operator  $L^+L$  is  $0.25(-\Delta+0.1id)^2$  and 2000 iterations are used to calculate the final result. Fourier method is used to calculate  $(L^+L)^{-1}$  by working on the Fourier transform of the force field and then applying the inverse transform. In order to examine how good the mapping is, we have to examine where the landmarks are carried by the forward path  $g$  at  $t=1$ . In this case, the forward path  $g_{t=1}$  carries the point (25, 30) to (30.0003, 25.0949) and the point (40, 35) to (34.9425, 39.9052) and thus is an “almost” exact matching.

For numerical results on the first approach for landmark matching, please refer to chapter 10 of this dissertation.



**Fig. 1.** The diffeomorphism given by the level set based landmark matching with distance re-initialization. The landmarks to be matched are  $(25, 30)$  to  $(30, 25)$ , and  $(40, 35)$  to  $(35, 40)$  on a 64 by 64 image.

## Chapter 8

# Equivalence of Objects under the Action of Translation, Scaling, and Rotation

*Abstract-In this chapter, we will briefly describe how to take into account the equivalence objects. In particular, we will focus on how to factor out the action of translation, scaling, and rotation. A modification to the cost functions will be presented and discussed. Moreover, a connection will be established between the modified cost functions and the Hausdorff metric between objects. This provides a solid foundation of the level set based object matching.*

## **1. The Equivalence Classes of Objects**

In the computational anatomy framework, the registration is to be carried out in a diffeomorphic way. The goal is to create a 1-to-1, onto, and differentiable map from one image back onto itself while at the same time all the important structures in the image are aligned. This is essentially a Lagrangian formulation.

On the other hand, pattern recognition in computer vision is being performed in a Eulerian formulation. The best way to understand the difference is to imagine we are to search in pictures for certain objects (for example, dogs), and we are given a priori some knowledge about the shapes of dogs. With this in mind, we will now match all possible shapes in the pictures, regardless of the scaling, position, and orientation to the known shapes of dogs. When a shape in a picture is very close to the known shapes of dogs, we will then say that this is possibly a dog.

From the above discussion, it is clear that the main difference between the framework we have discussed so far and the problem we consider in this chapter is the following

**1 Matching to an equivalent class of objects instead of the exact objects**

**2 Objects are considered to be independent from the underlying grid  
(Eulerian reference)**

## 2. The Invariance of Objects Represented by Their Signed Distance Functions

The equivalent class of an object is often defined by the invariance of objects subject to certain group actions. Some of the commonly considered group actions include translation, rotation, scaling, and stretching. In this chapter, we will focus on the first three group actions.

One of the first approach to register objects using level set method in this framework is in [68, 71] , where the authors proposed to match shapes by matching their signed distance functions. In order to take into account the invariance of objects under translation, rotation, and scaling, one of the distance functions being compared is then replaced by the following

$$\tilde{\phi}(a, b, r, \theta) = r\phi\left(\frac{(x-a)\cos\theta + (y-b)\sin\theta}{r}, \frac{-(x-a)\sin\theta + (y-b)\cos\theta}{r}\right) \quad (1)$$

Here  $a$  and  $b$  are the translation parameters,  $r$  is the scaling parameter, and  $\theta$  is the angle of rotation. The above function is simply the signed distance function of the shape under the same group actions with the same parameters. The cost functional is then minimized with respect to the above four parameters and experimental results are promising.

## Factoring out translation, rotation, and scaling

The above framework could be used to register objects by replacing the cost functions proposed in [68] with the cost functions proposed in the previous chapters. Moreover, it can be applied to all the formulations proposed in the previous chapters and allows us to factor out group actions under translation, rotation, and scaling.

We first notice that all the cost functions in previous chapters could be expressed in the following form as a function of the level set functions in the template and study and  $g$

$$D(T, S) = F(\varphi_i, \phi_i, g) \quad (2)$$

We now modify the cost function by inserting the modified level set function on the Study

$$D(T, S) = F(\varphi_i, \tilde{\phi}_i, g) \quad (3)$$

Thus, the following modified minimization problem is considered for factoring out actions under translation, dilation and rotation

$$\min_{a, b, r, \theta, g} F(\varphi_i, \tilde{\phi}_i(a, b, r, \theta), g) \quad (4)$$

This minimization problem is not easy to solve. However, approximate solution  $(a^*, b^*, r^*, \theta^*, g^*)$  could be obtained by minimizing with respect to the parameters of the group actions

$$(a^*, b^*, r^*, \theta^*) = \arg \min_{a, b, r, \theta} F(\varphi_i, \tilde{\phi}_i(a, b, r, \theta), g = id) \quad (5)$$

And then with respect to  $g$

$$g^* = \arg \min_g F(\varphi_i, \tilde{\phi}_i(a^*, b^*, r^*, \theta^*), g) \quad (6)$$

For the derivation of the gradient decent equations for  $a, b, r, \theta$ , we need the derivative of the cost function with respect to the level set functions:  $\frac{\partial F}{\partial \tilde{\phi}_i}$ , and also the derivatives of

the level set function with respect to the parameters:  $\frac{\partial \tilde{\phi}}{\partial a}$ ,  $\frac{\partial \tilde{\phi}}{\partial b}$ ,  $\frac{\partial \tilde{\phi}}{\partial r}$ , and  $\frac{\partial \tilde{\phi}}{\partial \theta}$ .

The formulation has already been given for  $\frac{\partial F}{\partial \tilde{\phi}_i}$  in each case when the corresponding cost function is derived (except that it is derived with respect to the deformation field).

The formulation for  $\frac{\partial \tilde{\phi}}{\partial a}$ ,  $\frac{\partial \tilde{\phi}}{\partial b}$ ,  $\frac{\partial \tilde{\phi}}{\partial r}$ , and  $\frac{\partial \tilde{\phi}}{\partial \theta}$  could be found in the paper [68], and thus we refer to this paper for more details.

### 3. Minimizing the Hausdorff Metric between Objects

The Hausdorff metric, which will be denoted by  $H$ , arises in geometric measure theory as a metric between two subsets  $A, B$  in  $R^n$  defined in the following way

$$H(A, B) = \max(h(A, B), h(B, A)), \quad (7)$$

$$h(X, Y) = \max_{x \in X} d(x, Y). \quad (8)$$

Here  $d(x, Y)$  is the usual Euclidean distance from the point  $x$  to its closest point restricted on the set  $Y$ . Please refer to the classic text in geometric measure theory [72] and also [73] for a detailed discussion on Hausdorff metric. In this chapter, let us state the fact that the family of all compact subsets of  $R^n$  is a complete metric space with respect to the Hausdorff metric  $H$ .

In the following, we will show that the Hausdorff metric is closely related to object matching using the level set method (This connection was originally suggested by Wei Zhu of the Mathematics Department at UCLA).

Let us use the cost function for open curves when factoring out equivalence classes as an example

$$\begin{aligned} & \min_{a,b,r,\theta} D_{curve}(T, S) \\ &= \min_{a,b,r,\theta} \left( \int_{\Omega} \tilde{D}_S \delta(\varphi_1(g_t^{-1})) |\nabla \varphi_1(g_t^{-1})| H(\varphi_2(g_t^{-1})) dx \right. \\ & \quad \left. + \int_{\Omega} D_T(g_t^{-1}) \delta(\tilde{\phi}_1) |\nabla \tilde{\phi}_1| H(\tilde{\phi}_2) dx \right) \end{aligned} \quad (9)$$

Let us modify the cost function by raising the distance functions to the curves to some power  $p$  and then take the  $p$ -th root

$$\begin{aligned} & \min_{a,b,r,\theta} D_{curve}(T, S) \\ &= \min_{a,b,r,\theta} \left( \int_{\Omega} \tilde{D}_S^p \delta(\varphi_1(g_t^{-1})) |\nabla \varphi_1(g_t^{-1})| H(\varphi_2(g_t^{-1})) dx \right)^{1/p} \\ & \quad \left( + \int_{\Omega} D_T^p(g_t^{-1}) \delta(\tilde{\phi}_1) |\nabla \tilde{\phi}_1| H(\tilde{\phi}_2) dx \right) \end{aligned} \quad (10)$$

Now if  $p$  goes to infinity, the limit of the above cost function is exactly the Hausdorff metric between the two curves in template and study!

In numerical calculation of equation (10), we could use a  $p$  much larger than one to approximate the  $L$  infinity norm and thus we can numerically find a choice of the parameter  $a$ ,  $b$ ,  $r$ , and  $\theta$  that minimizes the Hausdorff metric between the two curves. In fact, the same modification can also be used in the framework of chapter 4 and 5 to achieve non-linear matching through diffeomorphisms generated by infinite dimensional group actions.

We can modify the cost functions we proposed for matching non-overlapping shapes in 2D, surfaces in 3D, and also open curves in 3D in the same manner and re-interpret the cost functions as the Hausdorff metric between the shapes in 2D, surfaces in 3D, or open curves in 3D.

This means, by raising the distance functions to some higher power, a link could be established between the frame work of matching objects using the level set method and the theoretical results in geometric measure theory. This provides a strong theoretical support for the object matching using the level set method.

Numerical examples will be presented in the last chapter. However, it should be pointed out that from numerical experiments, better performance is noted when the cost function is modified in this manner.

## Chapter 9

# Extensions to Three Dimensional Object Matching

*Abstract- In this chapter, we will describe how to generalize the mathematical formulations developed in previous chapters to three-dimensional cases. In particular, we will describe how one can achieve closed and open curve matching in 3D. This requires new level set formulations dealing with the mathematical representation and numerical tools that allow us to evolve a 3D curve subject to certain velocity fields by numerically evolving its associated level set functions and tracing the crossing of the zero level sets.*

The representation and matching of volumes and open surfaces in 3D is straightforward extension of matching shapes and open curves in 2D. Thus in this chapter, we will focus on matching of curves in 3D.

## 1. Level Set Dictionary for 3 Dimensional Curves

Let vector valued level set function  $\phi = (\phi_1, \phi_2)$  be the implicit representation of a (closed) curve in 3D, then the tangent vector  $T$  is.

$$T = \frac{\nabla \phi_1 \times \nabla \phi_2}{|\nabla \phi_1 \times \nabla \phi_2|}. \quad (1)$$

Let us define the orthogonal projection matrix that projects vectors onto the plane with normal vector  $v$

$$P_v = I - \frac{v \otimes v}{|v|^2} \quad (2)$$

The length of the curve then could be calculated in the following way

$$\begin{aligned}
L(\Gamma) &= \int_{R^3} \delta(\phi_1) \delta(\phi_1) \left| P_{\nabla \phi_1} \nabla \phi_2 \right| |\nabla \phi_2| dx \\
&= \int_{R^3} \delta(\phi_1) \delta(\phi_1) \left| P_{\nabla \phi_2} \nabla \phi_1 \right| |\nabla \phi_1| dx \\
&= \int_{R^3} \delta(\phi_1) \delta(\phi_1) |\nabla \phi_1 \times \nabla \phi_2| dx.
\end{aligned} \tag{3}$$

This formulation for length could be found in [74]. An intuitive way to derive this expression is the following. Assume now we construct a “thin” tube along the curve with square cross section of side length  $\varepsilon$  (refer to figure 1). We will recover the length of the curve by shrinking  $\varepsilon$ . We first construct this tube by looking at the value of  $\phi_1$  with distance  $\varepsilon$  away from the curve (intersection of  $\phi_1$  and  $\phi_2$ ) along the outer normal direction

$$\phi_1\left(x + \varepsilon \frac{-\nabla \phi_1}{|\nabla \phi_1|}\right) = \phi_1(x) + \varepsilon \nabla \phi_1 \cdot \frac{-\nabla \phi_1}{|\nabla \phi_1|} = \phi_1(x) - \varepsilon |\nabla \phi_1| + O_1(\varepsilon^2) \tag{4}$$

In order to construct square cross section, we have to look at the value of  $\phi_2$  with distance  $\varepsilon$  away from the curve along the outer normal direction projected onto the plane orthogonal to  $\nabla \phi_1$

$$\phi_2\left(x + \varepsilon \frac{-P_{\nabla \phi_1} \nabla \phi_2}{\left|P_{\nabla \phi_1} \nabla \phi_2\right|}\right) = \phi_2(x) - \varepsilon \left|P_{\nabla \phi_1} \nabla \phi_2\right| + O_2(\varepsilon^2) \tag{5}$$

So the volume contained in this tube which has a cross section of  $\varepsilon^2$  is

$$\begin{aligned}
\text{volume} &= \int H[\phi_1(x) + \varepsilon|\nabla\phi_1| + O_1(\varepsilon^2)]H[\phi_2(x) + \varepsilon|P_{\nabla\phi_1}\nabla\phi_2| + O_2(\varepsilon^2)]dx \\
&\quad - \int H[\phi_1(x)]H[\phi_2(x) + \varepsilon|P_{\nabla\phi_1}\nabla\phi_2| + O_2(\varepsilon^2)]dx \\
&\quad - \int H[\phi_1(x) + \varepsilon|\nabla\phi_1| + O_1(\varepsilon^2)]H[\phi_2(x)]dx \\
&\quad + \int H[\phi_1(x)]H[\phi_2(x)]dx \\
&= \varepsilon^2 \int \delta(\phi_1(x))\delta(\phi_2(x))|\nabla\phi_1||P_{\nabla\phi_1}\nabla\phi_2|dx + O(\varepsilon^3)
\end{aligned} \tag{6}$$

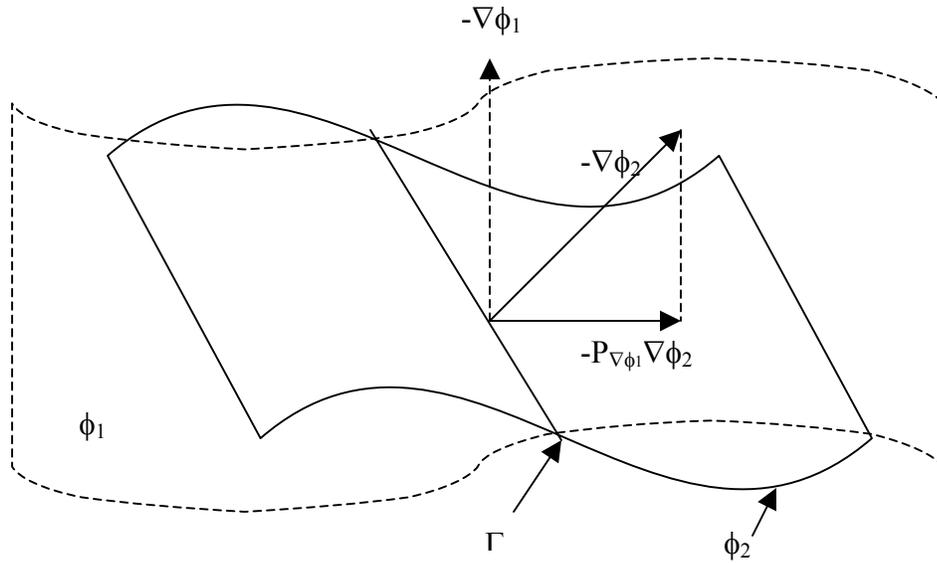
The length of the curve is obtained by letting  $\varepsilon$  goes to zero

$$\begin{aligned}
\text{length} &= \lim_{\varepsilon \rightarrow 0} \frac{1}{\varepsilon^2} \left\{ \int H[\phi_1(x) + \varepsilon|\nabla\phi_1| + O_1(\varepsilon^2)]H[\phi_2(x) + \varepsilon|P_{\nabla\phi_1}\nabla\phi_2| + O_2(\varepsilon^2)]dx \right. \\
&\quad - \int H[\phi_1(x)]H[\phi_2(x) + \varepsilon|P_{\nabla\phi_1}\nabla\phi_2| + O_2(\varepsilon^2)]dx \\
&\quad - \int H[\phi_1(x) + \varepsilon|\nabla\phi_1| + O_1(\varepsilon^2)]H[\phi_2(x)]dx \\
&\quad \left. + \int H[\phi_1(x)]H[\phi_2(x)]dx \right\} \\
&= \int \delta(\phi_1(x))\delta(\phi_2(x))|\nabla\phi_1||P_{\nabla\phi_1}\nabla\phi_2|dx
\end{aligned} \tag{7}$$

By switching the order of the two level set functions, we derive the second equality in equation 3. The third equality holds as

$$|\nabla\phi_1||P_{\nabla\phi_1}\nabla\phi_2| = |\nabla\phi_1 \times \nabla\phi_2| \tag{8}$$

Another intuitive way to understand the equation is by recognizing that the magnitude of the outer product of two vectors  $a$  and  $b$  can be geometrically interpreted as the area of the quadrilateral with edge vector  $a$  and  $b$ .



**Fig. 1: Illustration of how to represent a curve in 3D and calculate the length of the curve using the intersection of two level set functions**

## 2. Length Shortening Flow of Curves in 3D

We now will calculate the Euler-Lagrange equations that minimize this energy. In [74], the authors used a different approach by looking at the general problem of evolving a curve by Wulff flow on a surface. In this chapter, we will derive this Euler-Lagrange equation in a direct and intuitive approach.

The minimization problem we want to solve takes the following form

$$\begin{aligned}
 & \min \int \delta(\phi_1(x))\delta(\phi_2(x))|\nabla \phi_1| |P_{\nabla \phi_1} \nabla \phi_2| dx \\
 & = \min \int \delta(\phi_1(x))\delta(\phi_2(x))|\nabla \phi_2| |P_{\nabla \phi_2} \nabla \phi_1| dx
 \end{aligned} \tag{9}$$

Let us calculate the directional derivative of the above energy by perturbing  $\phi_1$  in the following way

$$\begin{aligned} & \int \delta(\phi_1 + \varepsilon\eta) \delta(\phi_2) |\nabla\phi_2| \left\| P_{\nabla\phi_2} \nabla(\phi_1 + \varepsilon\eta) \right\| dx \\ &= \int \varepsilon \delta'(\phi_1) \delta(\phi_2) |\nabla\phi_2| \left\| P_{\nabla\phi_2} \nabla\phi_1 \right\| dx + \int \delta(\phi_1) \delta(\phi_2) |\nabla\phi_2| \left\| P_{\nabla\phi_2} \nabla\phi_1 + \varepsilon P_{\nabla\phi_2} \nabla\eta \right\| dx \end{aligned} \quad (10)$$

We could expand the last term in equation (10) as follows

$$\begin{aligned} & \int \delta(\phi_1) \delta(\phi_2) |\nabla\phi_2| \left\| P_{\nabla\phi_2} \nabla\phi_1 + \varepsilon P_{\nabla\phi_2} \nabla\eta \right\| dx \\ &= \int \delta(\phi_1) \delta(\phi_2) |\nabla\phi_2| \left[ \left\| P_{\nabla\phi_2} \nabla\phi_1 \right\| + \frac{\left\langle P_{\nabla\phi_2} \nabla\phi_1, \varepsilon P_{\nabla\phi_2} \nabla\eta \right\rangle}{\left\| P_{\nabla\phi_2} \nabla\phi_1 \right\|} \right] dx \end{aligned} \quad (11)$$

Let us calculate the inner product in the numerator of the above equation

$$\begin{aligned} \left\langle P_{\nabla\phi_2} \nabla\phi_1, \varepsilon P_{\nabla\phi_2} \nabla\eta \right\rangle &= \varepsilon \left\langle P_{\nabla\phi_2} \nabla\phi_1, \nabla\eta - \frac{\left\langle \nabla\phi_2, \nabla\eta \right\rangle}{|\nabla\phi_2|^2} \nabla\phi_2 \right\rangle \\ &= \varepsilon \left\langle P_{\nabla\phi_2} \nabla\phi_1, \nabla\eta \right\rangle \end{aligned} \quad (12)$$

The second equality holds because  $\nabla\phi_1$  is being projected onto the plane with normal vector  $\nabla\phi_2$ , thus

$$\left\langle P_{\nabla\phi_2} \nabla\phi_1, \nabla\phi_2 \right\rangle = 0. \quad (13)$$

Now we can re-write the equation in the following way

$$\begin{aligned}
& \int \delta(\phi_1)\delta(\phi_2)|\nabla\phi_2| \left[ |P_{\nabla\phi_2}\nabla\phi_1| + \frac{\langle P_{\nabla\phi_2}\nabla\phi_1, \varepsilon P_{\nabla\phi_2}\nabla\eta \rangle}{|P_{\nabla\phi_2}\nabla\phi_1|} \right] dx \\
&= \int \delta(\phi_1)\delta(\phi_2)|\nabla\phi_2| |P_{\nabla\phi_2}\nabla\phi_1| dx \\
&- \int \operatorname{div} \left[ \delta(\phi_1)\delta(\phi_2)|\nabla\phi_2| \frac{P_{\nabla\phi_2}\nabla\phi_1}{|P_{\nabla\phi_2}\nabla\phi_1|} \right] \eta dx \\
&= \int \delta(\phi_1)\delta(\phi_2)|\nabla\phi_2| |P_{\nabla\phi_2}\nabla\phi_1| dx \tag{14} \\
&- \int \delta(\phi_1) \operatorname{div} \left[ \delta(\phi_2)|\nabla\phi_2| \frac{P_{\nabla\phi_2}\nabla\phi_1}{|P_{\nabla\phi_2}\nabla\phi_1|} \right] \eta dx - \int \delta'(\phi_1)\delta(\phi_2)|\nabla\phi_2| |P_{\nabla\phi_2}\nabla\phi_1| \eta dx \\
&= \int \delta(\phi_1)\delta(\phi_2)|\nabla\phi_2| |P_{\nabla\phi_2}\nabla\phi_2| dx \\
&- \int \delta(\phi_1)\delta(\phi_2) \operatorname{div} \left[ |\nabla\phi_2| \frac{P_{\nabla\phi_2}\nabla\phi_1}{|P_{\nabla\phi_2}\nabla\phi_1|} \right] \eta dx - \int \delta'(\phi_1)\delta(\phi_2)|\nabla\phi_2| |P_{\nabla\phi_2}\nabla\phi_1| \eta dx
\end{aligned}$$

Notice that the last equality in equation (14) holds due to equation (13), and the second last equality holds because of the following relation

$$\langle \nabla\phi_1, P_{\nabla\phi_2}\nabla\phi_1 \rangle = |P_{\nabla\phi_2}\nabla\phi_1|^2 \tag{15}$$

We now put everything together and state the following Euler-Lagrange equation for  $\phi_1$

$$-\delta(\phi_1)\delta(\phi_2) \operatorname{div} \left[ |\nabla\phi_2| \frac{P_{\nabla\phi_2}\nabla\phi_1}{|P_{\nabla\phi_2}\nabla\phi_1|} \right] = 0 \tag{16}$$

And similarly for  $\phi_2$

$$-\delta(\phi_1)\delta(\phi_2) \operatorname{div} \left[ |\nabla\phi_1| \frac{P_{\nabla\phi_1}\nabla\phi_2}{|P_{\nabla\phi_1}\nabla\phi_2|} \right] = 0 \tag{17}$$

The gradient descent PDE now takes the form

$$\begin{aligned}\frac{\partial \phi_1}{\partial t} &= \delta(\phi_1)\delta(\phi_2)\operatorname{div}\left[\frac{P_{\nabla\phi_2}\nabla\phi_1}{|P_{\nabla\phi_2}\nabla\phi_1|}\right] \\ \frac{\partial \phi_2}{\partial t} &= \delta(\phi_1)\delta(\phi_2)\operatorname{div}\left[\frac{P_{\nabla\phi_1}\nabla\phi_2}{|P_{\nabla\phi_1}\nabla\phi_2|}\right]\end{aligned}\tag{18}$$

And thus if we look at the following weighted length minimization problem with respect to the function  $f(x)$

$$\min_{R^3} \int f(x)\delta(\phi_1)\delta(\phi_2)|\nabla\phi_1 \times \nabla\phi_2|dx.\tag{19}$$

The time dependent gradient descent PDE takes the form

$$\begin{aligned}\frac{\partial \phi_1}{\partial t} &= \delta(\phi_1)\delta(\phi_2)\operatorname{div}\left[f\frac{P_{\nabla\phi_2}\nabla\phi_1}{|P_{\nabla\phi_2}\nabla\phi_1|}\right] \\ \frac{\partial \phi_2}{\partial t} &= \delta(\phi_1)\delta(\phi_2)\operatorname{div}\left[f\frac{P_{\nabla\phi_1}\nabla\phi_2}{|P_{\nabla\phi_1}\nabla\phi_2|}\right]\end{aligned}\tag{20}$$

### 3. Open Curve Matching in 3D

Now let us turn to the problem of how to match an open curve in 3D as we did before for open curves in 2D.

By introducing 3 level set functions  $\phi_1, \phi_2, \phi_3$ , an open curve  $\Gamma$  in 3D can be implicitly represented in the following way

$$\Gamma = \{x \mid \phi_1(x) = 0, \phi_2(x) = 0, \phi_3(x) > 0\}\tag{21}$$

As in the case of open curve matching in 2D, we also propose two different approaches for 3D curve matching, namely, using the sum of two line integrals as in equation (8) of chapter 6 or using only one line integral coupled with end point constraints as in equation (9) of chapter 6. In either formulation, the first term of the cost functions is the same. With notations similarly defined as in equation (8) and (9) in chapter 6, the first term in the cost function is in the following form

$$\int_{R^3} D_S(x) \delta(\varphi_1(\mathbf{g}_t^{-1})) \delta(\varphi_2(\mathbf{g}_t^{-1})) H(\varphi_3(\mathbf{g}_t^{-1})) \left| \nabla \varphi_1(\mathbf{g}_t^{-1}) \times \nabla \varphi_2(\mathbf{g}_t^{-1}) \right| dx. \quad (22)$$

Using the result in equation (20), the force field contributed by this part of the cost function is then

$$\begin{aligned} & -\delta(\varphi_1(\mathbf{g}_t^{-1})) \delta(\varphi_2(\mathbf{g}_t^{-1})) \operatorname{div} \left[ D_S H(\varphi_3(\mathbf{g}_t^{-1})) \left| \nabla \varphi_2(\mathbf{g}_t^{-1}) \right| \frac{P_{\nabla \varphi_2(\mathbf{g}_t^{-1})} \nabla \varphi_1(\mathbf{g}_t^{-1})}{\left| P_{\nabla \varphi_2(\mathbf{g}_t^{-1})} \nabla \varphi_1(\mathbf{g}_t^{-1}) \right|} \right] \nabla \varphi_1(\mathbf{g}_t^{-1}) \\ & -\delta(\varphi_1(\mathbf{g}_t^{-1})) \delta(\varphi_2(\mathbf{g}_t^{-1})) \operatorname{div} \left[ D_S H(\varphi_3(\mathbf{g}_t^{-1})) \left| \nabla \varphi_1(\mathbf{g}_t^{-1}) \right| \frac{P_{\nabla \varphi_1(\mathbf{g}_t^{-1})} \nabla \varphi_2(\mathbf{g}_t^{-1})}{\left| P_{\nabla \varphi_1(\mathbf{g}_t^{-1})} \nabla \varphi_2(\mathbf{g}_t^{-1}) \right|} \right] \nabla \varphi_2(\mathbf{g}_t^{-1}) \\ & + D_S \delta(\varphi_1(\mathbf{g}_t^{-1})) \delta(\varphi_2(\mathbf{g}_t^{-1})) \delta(\varphi_3(\mathbf{g}_t^{-1})) \left| \nabla \varphi_1(\mathbf{g}_t^{-1}) \times \nabla \varphi_2(\mathbf{g}_t^{-1}) \right| \nabla \varphi_3(\mathbf{g}_t^{-1}) \end{aligned} \quad (23)$$

## Landmark matching in 3D

Now the remaining formulation we need to complete the discussion in 3D is the formula for level set based landmark matching in 3D. The same formulation also allows us to control the end points of an open curve  $\Gamma$  in 3D.

Given any continuous function  $g$ , the main formulation we want to derive is the following

$$\begin{aligned}
& \int g \delta(\phi_1) \delta(\phi_2) \delta(\phi_3) |\nabla \phi_1 \times \nabla \phi_2 \cdot \nabla \phi_3| dx \\
&= \int g \delta(\phi_1) \delta(\phi_2) \delta(\phi_3) |\nabla \phi_3 \times \nabla \phi_2 \cdot \nabla \phi_1| dx \\
&= \int g \delta(\phi_1) \delta(\phi_2) \delta(\phi_3) |\nabla \phi_1 \times \nabla \phi_3 \cdot \nabla \phi_2| dx \\
&= \sum_i g(P_i)
\end{aligned} \tag{24}$$

Here the points  $P_i$  are the landmark points represented by the intersection of the three level set functions. Thus, in 3D, the integration factor needed to recover the point values at the intersection of level set functions is the absolute value of the mixed product (or scalar triple product) of the gradient vectors.

$$|\nabla \phi_1 \times \nabla \phi_2 \cdot \nabla \phi_3| \tag{25}$$

This formula can be derived similarly as in the derivation of the length of a curve in 3D by integrating  $g$  in a small cube with side length  $\varepsilon$  located at the intersection of the three level set functions and letting  $\varepsilon$  goes to zero. The three adjustments we need to make for the level set functions  $\phi_1$ ,  $\phi_2$ , and  $\phi_3$  are

$$\varepsilon |\nabla \phi_1| \tag{26}$$

$$\varepsilon |P_{\nabla \phi_1} \nabla \phi_2| \tag{27}$$

$$\varepsilon \left| \nabla \phi_3 \cdot \frac{\nabla \phi_1 \times \nabla \phi_2}{|\nabla \phi_1 \times \nabla \phi_2|} \right| = \varepsilon \frac{|\nabla \phi_3 \cdot \nabla \phi_1 \times \nabla \phi_2|}{|\nabla \phi_1 \times \nabla \phi_2|} \tag{28}$$

Recalling the relation in equation (8), we thus recover the integration factor

$$|\nabla \phi_1| \|P_{\nabla \phi_1} \nabla \phi_2\| \frac{|\nabla \phi_3 \cdot \nabla \phi_1 \times \nabla \phi_2|}{|\nabla \phi_1 \times \nabla \phi_2|} = |\nabla \phi_3 \cdot \nabla \phi_1 \times \nabla \phi_2| \quad (29)$$

Another intuitive way is to recognize that the absolute value of the mixed product of three vectors  $a$ ,  $b$ , and  $c$  can be geometrically interpreted as the volume of a parallelepiped with edge vectors  $a$ ,  $b$ , and  $c$ .

Using similar notations as in equation (1) of chapter 7 (landmark matching in 2D), we propose the following cost function for level set based landmark matching in 3D

$$\min \left( \int D_S \delta(\phi_1(g_t^{-1})) \delta(\phi_2(g_t^{-1})) \delta(\phi_3(g_t^{-1})) |\nabla \phi_1(g_t^{-1}) \times \nabla \phi_2(g_t^{-1}) \cdot \nabla \phi_3(g_t^{-1})| dx \right. \\ \left. + \int D_T(g_t^{-1}) \delta(\phi_1) \delta(\phi_2) \delta(\phi_3) |\nabla \phi_1 \times \nabla \phi_2 \cdot \nabla \phi_3| dx \right) \quad (30)$$

Now let us compute the Euler-Lagrange equation for equation (30). The force field contributed by the second term of equation (30) is straightforward, and thus we will focus on the derivation of the force field for the first term. Instead of working on the Lagrangian formulation directly, we work on the Euler-Lagrangian equation for the following variational problem in a Eulerian formulation

$$\min_{\phi_1, \phi_2, \phi_3} \int D \delta(\phi_1) \delta(\phi_2) \delta(\phi_3) |\nabla \phi_1 \times \nabla \phi_2 \cdot \nabla \phi_3| dx \quad (31)$$

Let us compute the Euler-Lagrange equation for (31) with respect to  $\phi_1$  using similar arguments as before by perturbing  $\phi_1$

$$\begin{aligned}
& \int D \delta(\phi_2) \delta(\phi_3) \delta(\phi_1 + \varepsilon \eta) |\nabla \phi_2 \times \nabla \phi_3 \cdot \nabla(\phi_1 + \varepsilon \eta)| dx \\
&= \int D \delta(\phi_2) \delta(\phi_3) \delta(\phi_1) |\nabla \phi_2 \times \nabla \phi_3 \cdot \nabla \phi_1| dx \\
&+ \int D \delta(\phi_2) \delta(\phi_3) \delta'(\phi_1) |\nabla \phi_2 \times \nabla \phi_3 \cdot \nabla \phi_1| \varepsilon \eta dx \\
&+ \int D \delta(\phi_1) \delta(\phi_2) \delta(\phi_3) \operatorname{sgn}(\nabla \phi_2 \times \nabla \phi_3 \cdot \nabla \phi_1) \varepsilon \nabla \phi_2 \times \nabla \phi_3 \cdot \nabla \eta dx
\end{aligned} \tag{32}$$

We further expand the last term in the above equation

$$\begin{aligned}
& \int D \delta(\phi_1) \delta(\phi_2) \delta(\phi_3) \operatorname{sgn}(\nabla \phi_2 \times \nabla \phi_3 \cdot \nabla \phi_1) \varepsilon \nabla \phi_2 \times \nabla \phi_3 \cdot \nabla \eta dx \\
&= - \int \operatorname{div}[D \delta(\phi_1) \delta(\phi_2) \delta(\phi_3) \operatorname{sgn}(\nabla \phi_2 \times \nabla \phi_3 \cdot \nabla \phi_1) \nabla \phi_2 \times \nabla \phi_3] \varepsilon \eta dx \\
&= - \int [\delta'(\phi_1) D \delta(\phi_2) \delta(\phi_3) \operatorname{sgn}(\nabla \phi_2 \times \nabla \phi_3 \cdot \nabla \phi_1) \nabla \phi_2 \times \nabla \phi_3 \cdot \nabla \phi_1] \varepsilon \eta dx \\
&\quad - \int \delta(\phi_1) \operatorname{div}[D \delta(\phi_2) \delta(\phi_3) \operatorname{sgn}(\nabla \phi_2 \times \nabla \phi_3 \cdot \nabla \phi_1) \nabla \phi_2 \times \nabla \phi_3] \varepsilon \eta dx \\
&= - \int \delta'(\phi_1) D \delta(\phi_2) \delta(\phi_3) |\nabla \phi_2 \times \nabla \phi_3 \cdot \nabla \phi_1| \varepsilon \eta dx \\
&\quad - \int \delta(\phi_1) \delta(\phi_2) \delta(\phi_3) \operatorname{div}[D \operatorname{sgn}(\nabla \phi_2 \times \nabla \phi_3 \cdot \nabla \phi_1) \nabla \phi_2 \times \nabla \phi_3] \varepsilon \eta dx
\end{aligned} \tag{33}$$

The last equality holds because

$$\nabla \phi_2 \cdot (\nabla \phi_2 \times \nabla \phi_3) = 0 \tag{34}$$

and similarly

$$\nabla \phi_3 \cdot (\nabla \phi_2 \times \nabla \phi_3) = 0 \tag{35}$$

Thus, the Euler-Lagrange equation for  $\phi_1$  is

$$\begin{aligned}
& - \operatorname{div}[D \operatorname{sgn}(\nabla \phi_2 \times \nabla \phi_3 \cdot \nabla \phi_1) \nabla \phi_2 \times \nabla \phi_3] \delta(\phi_1) \delta(\phi_2) \delta(\phi_3) \\
&= - \operatorname{div} \left[ D \frac{\nabla \phi_2 \times \nabla \phi_3 \cdot \nabla \phi_1}{|\nabla \phi_2 \times \nabla \phi_3 \cdot \nabla \phi_1|} \nabla \phi_1 \times \nabla \phi_2 \right] \delta(\phi_1) \delta(\phi_2) \delta(\phi_3) \\
&= 0
\end{aligned} \tag{36}$$

And similarly for  $\phi_2$  and  $\phi_3$

$$\begin{aligned}
& - \operatorname{div}[D \operatorname{sgn}(\nabla \phi_3 \times \nabla \phi_1 \cdot \nabla \phi_2) \nabla \phi_3 \times \nabla \phi_1] \delta(\phi_1) \delta(\phi_2) \delta(\phi_3) \\
& = 0
\end{aligned} \tag{37}$$

$$\begin{aligned}
& - \operatorname{div}[D \operatorname{sgn}(\nabla \phi_1 \times \nabla \phi_2 \cdot \nabla \phi_3) \nabla \phi_1 \times \nabla \phi_2] \delta(\phi_1) \delta(\phi_2) \delta(\phi_3) \\
& = 0
\end{aligned} \tag{38}$$

We now can go back to the force field contributed by the first term in equation (30). Using similar derivations as in (36), (37), and (38), the force field contributed by  $\varphi_1$  in (30) has the following form

$$\begin{aligned}
& - \operatorname{div}\{D_s \operatorname{sgn}(\nabla \varphi_2(\mathbf{g}_t^{-1}) \times \nabla \varphi_3(\mathbf{g}_t^{-1}) \cdot \nabla \varphi_1(\mathbf{g}_t^{-1})) \nabla \varphi_2(\mathbf{g}_t^{-1}) \times \nabla \varphi_3(\mathbf{g}_t^{-1})\} \times \\
& \delta(\varphi_1(\mathbf{g}_t^{-1})) \delta(\varphi_2(\mathbf{g}_t^{-1})) \delta(\varphi_3(\mathbf{g}_t^{-1})) \nabla \varphi_1(\mathbf{g}_t^{-1})
\end{aligned} \tag{39}$$

The force field contributed by  $\varphi_2$  and  $\varphi_3$  are obtained similarly. Putting all three terms together plus the contribution from the second term in equation (30), we have the force field for level set based landmark matching in 3D.

## Automatic initialization of open curves in 3D from data points

Now if we are given an open curve in the form of some data points in 3D, and we want to automatically generate the corresponding three level set functions that describe the open curve. A similar cost function as in equation (9) in chapter 6 can be used to find these level set functions as in the case of 2D open curves.

$$\min \left( \begin{aligned} & \int D_s(x) \delta(\varphi_1(\mathbf{g}_t^{-1})) \delta(\varphi_2(\mathbf{g}_t^{-1})) H(\varphi_3(\mathbf{g}_t^{-1})) |\nabla \varphi_1(\mathbf{g}_t^{-1}) \times \nabla \varphi_2(\mathbf{g}_t^{-1})| dx \\ & + \lambda \int D_{\text{end}} \delta(\varphi_1(\mathbf{g}_t^{-1})) \delta(\varphi_2(\mathbf{g}_t^{-1})) \delta(\varphi_3(\mathbf{g}_t^{-1})) |\nabla \varphi_1(\mathbf{g}_t^{-1}) \times \nabla \varphi_2(\mathbf{g}_t^{-1}) \cdot \nabla \varphi_3(\mathbf{g}_t^{-1})| dx \end{aligned} \right) \tag{40}$$

To achieve the goal, we can arbitrarily initialize the three level set functions and minimize the same cost function. All we need in the cost function is the distance function to these data points and identification of the two endpoints.

# Chapter 10

## Numerical Experiments

*Abstract-In this chapter, we present numerical results for the level set based object matching in 2D using formulations proposed in previous chapters. All numerical examples are computed using either the Horn and Schunck functional or the Modified Beg's algorithm. Results for all formulations in 2D will be presented. Moreover, we will address the issue of the equivalence classe of objects with an open curve matching example. The equivalence class of the open curve in the study that minimizes the Hausdorff metric between the two curves is presented and the geodesic path that links the two curves with or without taking into consideration the invariance of objects will be compared.*

In this chapter, all the results are computed on the unit square  $[0, 1]^2$  discretized to a 64 by 64 grid when the modified Beg's algorithm is used to generate diffeomorphic mapping. The remaining results are computed using the regularizer proposed by Horn and Schunck on a 128 by 128 grid with grid size 0.1.

The numerical approximations for Heaviside and delta function introduced in chapter 1 are used for numerical computation. Notice that in chapter 1, two types of approximations for Heaviside function (equation (21) and (23) in chapter1) and delta function (equation (22) and (24) in chapter) are proposed. We have tested both types of approximations and the results show that numerically they perform similarly.

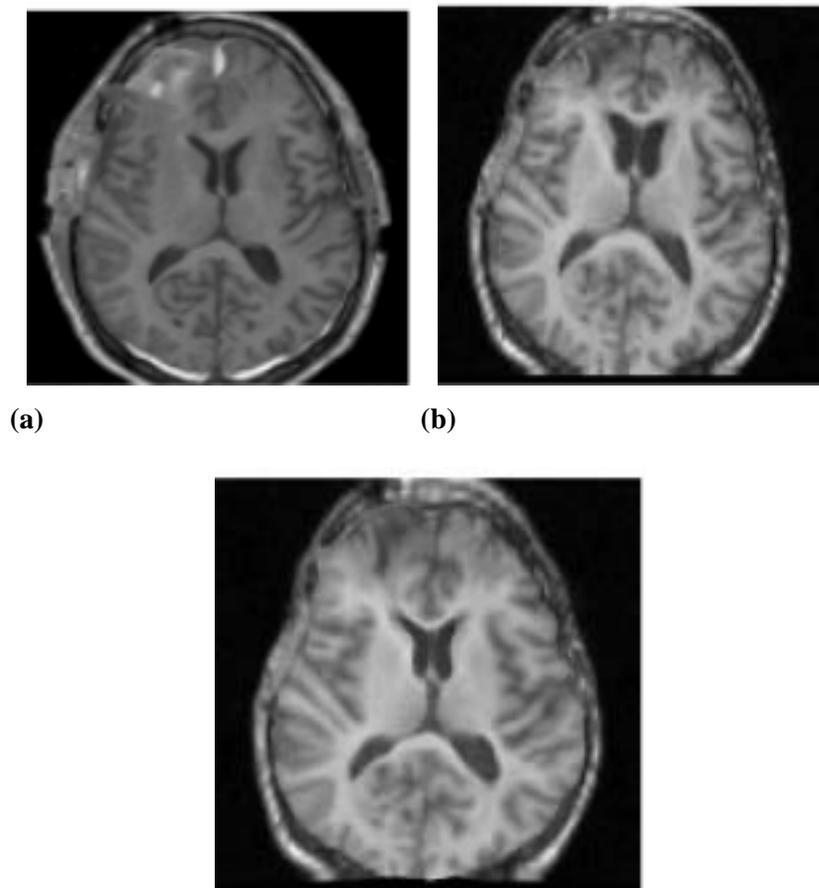
Zero boundary condition is used for both the modified Beg's algorithm and the Horn and Schunck type regularization, although other types of boundary condition could be used. For example, in the original Beg's algorithm in [42], periodic boundary condition was used.

## **1. Shape Matching**

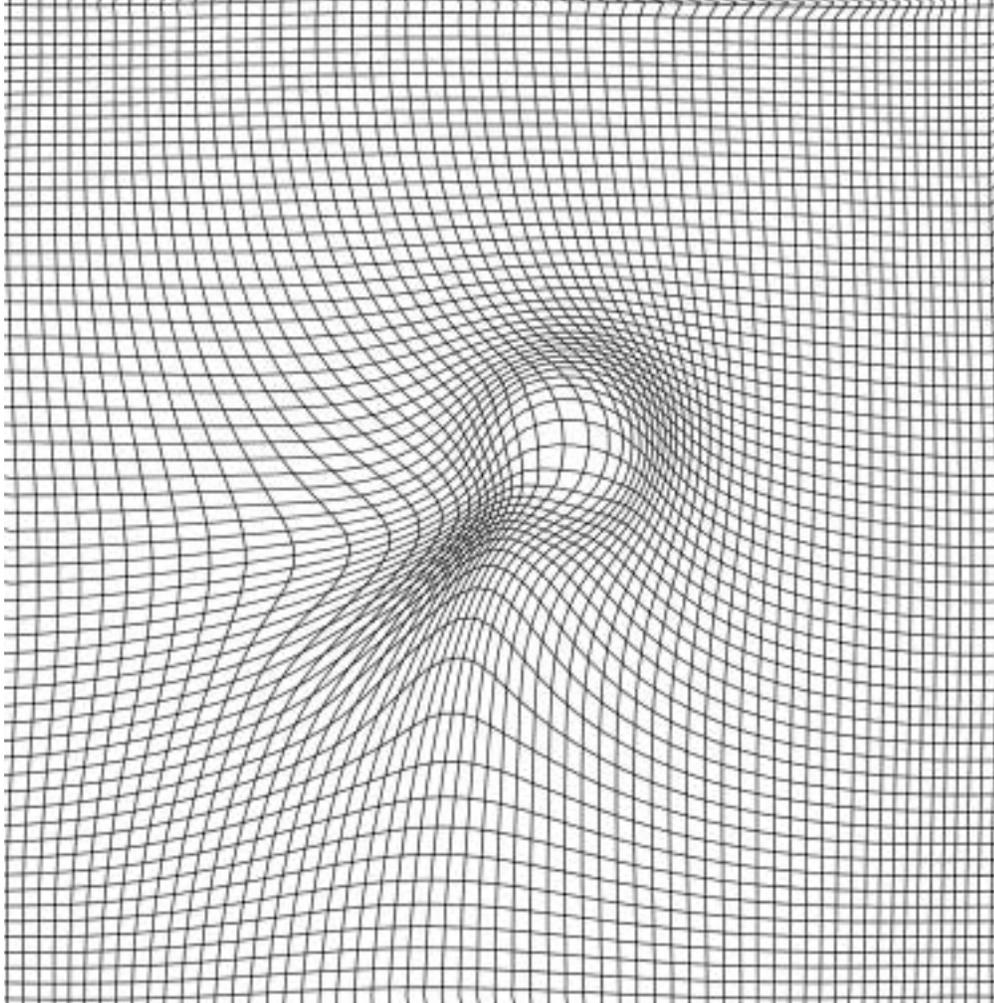
We show two examples of shape matching. In figure 1, we warped two brain images by aligning the contours around the brain tissue and the ventricles using the Horn and Schunck type regularizer. The two images were from the same patient with a traumatic lesion in the frontal lobe. Figure 1(a) was acquired right after admission, and 1(b) was

taken three months after trauma. Rigid co-registration was first performed based on the mutual information before non-linear shape matching was performed. Figure 1(c) shows the warping of figure 1(b) to figure 1(a).

The second example is, without factoring out invariance of objects under translation, scaling, and rotation, to match a circle with radius 5 centered at the grid point (35, 35) to (45, 45) using the modified Beg's algorithm. The regularizing operator  $L^+L$  in this case is  $0.05(-2.5\Delta+0.1id)^2$ .



**Fig. 1 (a): a brain MRI image right after traumatic brain injury. (b): MRI image of the same patient three months later. (c): warped image of (b) to match (a).**



**Fig. 2:** The diffeomorphism of moving the circle centered at grid point  $(35,35)$  with radius 5 to  $(45,45)$  using the modified Beg's algorithm calculated on the unit square discretized to a 64 by 64 grid.

## 2. Open Curve Matching

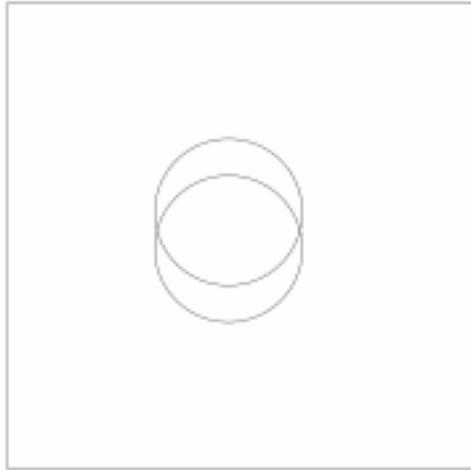
In this experiment, we validate the second matching approach for matching open curves. We generate two level set functions to implicitly represent a line segment joining grid points  $(30, 60)$  and  $(90, 60)$  on a 128 by 128 grid. We arbitrarily initialize two level set functions and minimize the proposed cost function (equation (9) in chapter 6). Figure 3 shows the results using the Horn and Schunck regularizer. Figure 3(a) is the initial level set functions and 3(b) shows the final result. Figure 3(c) shows the result without the end point constraint, and thus the result in this case does not match the whole line segment as in figure 3(b). The deformation field is shown in figure 4(a).

Most of the current methods for matching curves are based on matching corresponding landmarks on the curves. The landmarks are chosen to be equidistant from each other and the matching is obtained by assuming that landmarks stay equidistant when warped to another curve. Figure 4(b) shows how this assumption can fail. We add oscillations to the line segment described above joining the grid points  $(30, 60)$  and  $(90, 60)$ , and the displacement field is computed with the same initial conditions and regularizer. The corresponding deformation field is shown in figure 4(b). It is noted that equidistant points on the arc are not mapped to equidistant points on the final warped curve anymore. Thus we argue that representing curves by the intersections of level set functions is a better strategy for matching open curves.

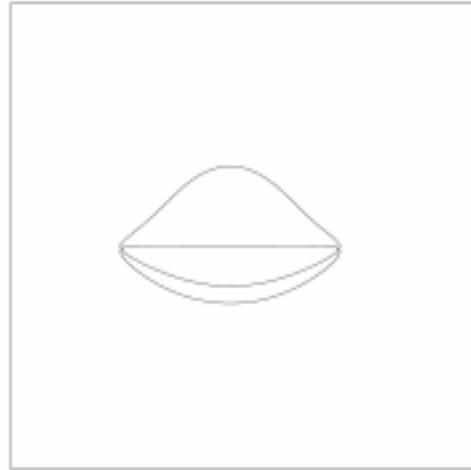
In order to compare the effect of the regularizer, the same testing problem is computed using the modified Beg's algorithm, and the deformation field is shown in figure 5. There is a difference in the deformation created by these two formulations. The regularizing operator  $L^+L$  in this case is  $0.025(-1.5\Delta+0.1id)^2$ .

### **3. Landmark Matching**

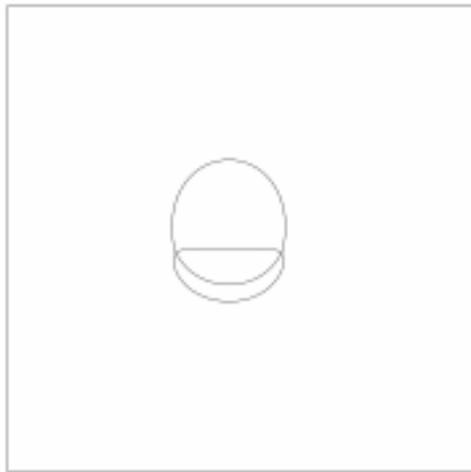
Figure 6 is the result of the level set based landmark matching with six pairs of landmarks generated by the modified Beg's algorithm with distance function re-initialization. The landmarks to be matched are (16, 32) to (16, 27), (29, 35) to (29, 40), (29, 29) to (29, 34), (35, 35) to (35, 40), (35, 29) to (35, 24), and (50, 32) to (50, 37) on a 64 by 64 grid. The regularizing operator  $L^+L$  in this case is  $0.025(-\Delta+0.1id)^2$ .



(a)

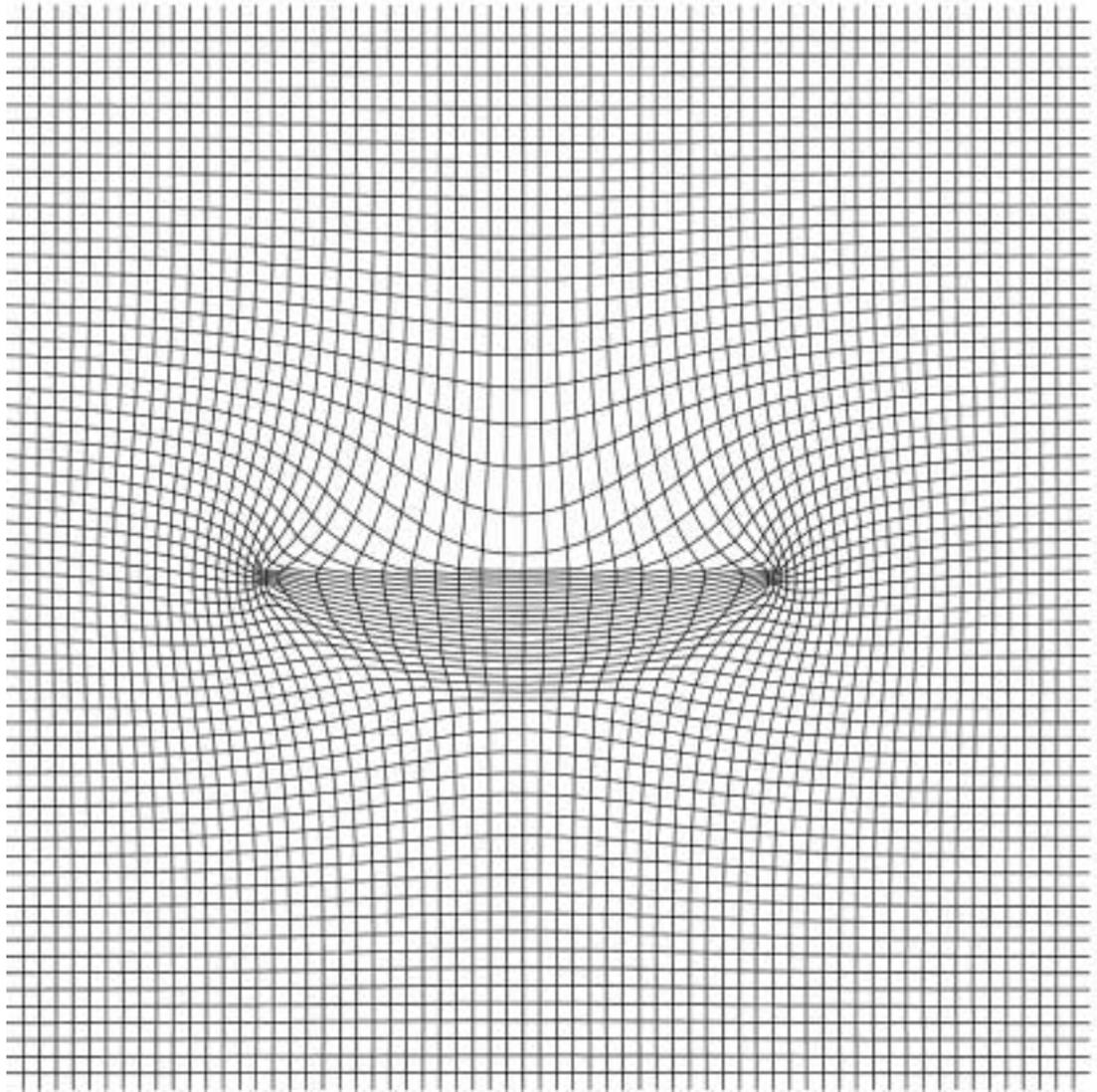


(b)

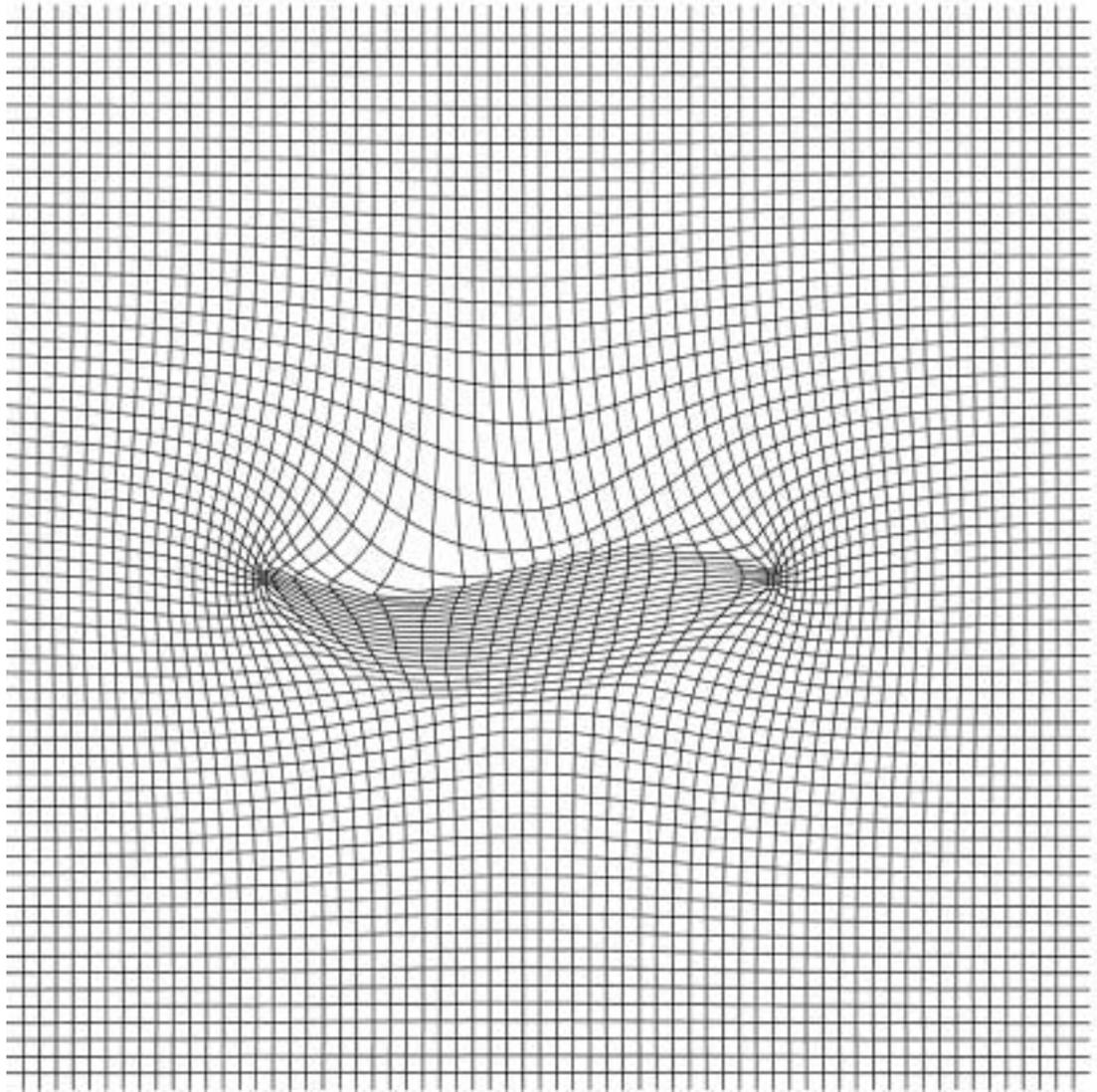


(c)

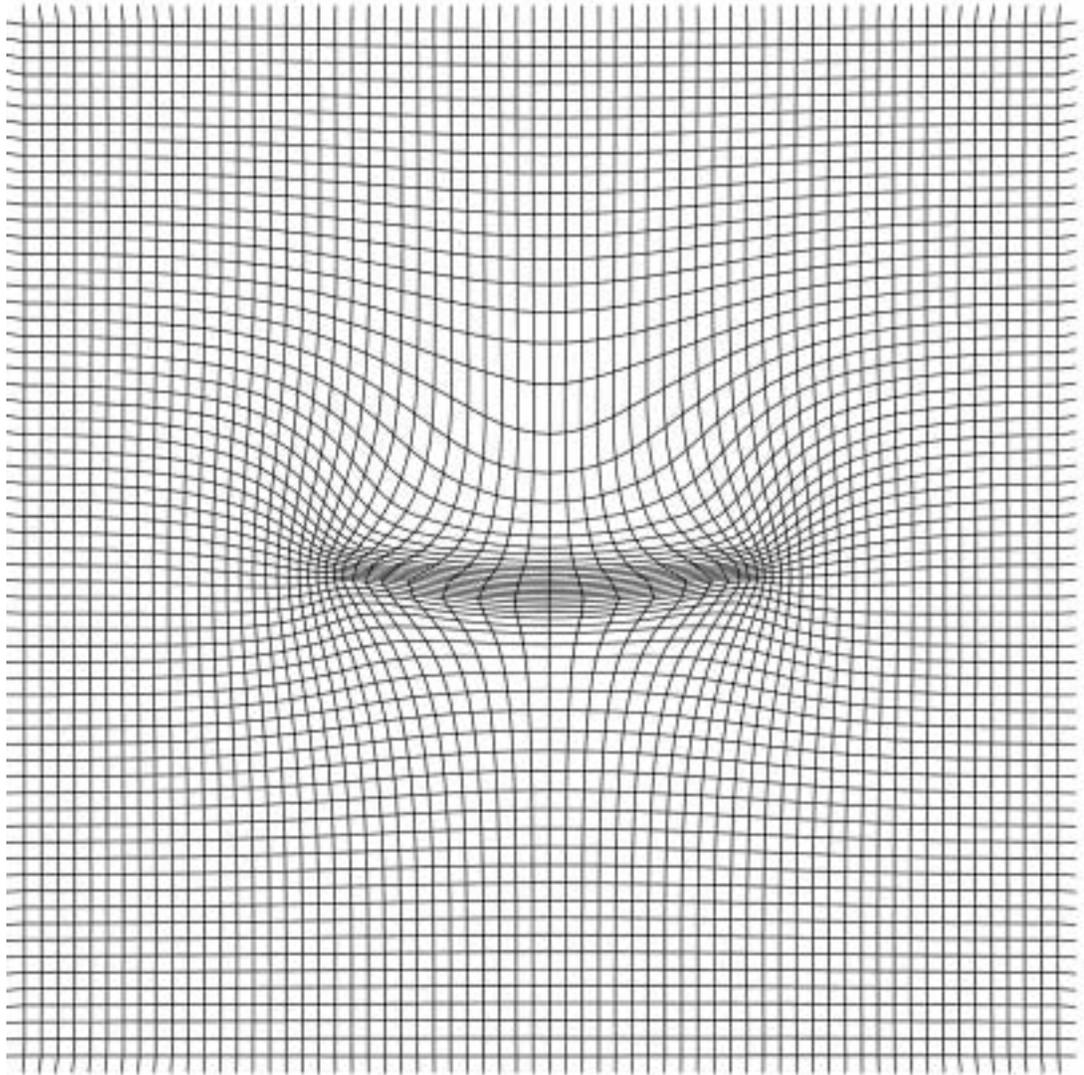
**Fig. 3: (a) the zero level sets of two level set functions arbitrarily initialized. (b) the warped zero level sets of the level set functions with the arc in the middle matched to the line segment joining grid points (30, 60) and (90, 60). (c) the warped zero level sets of the level set functions without enforcing matching of the end points.**



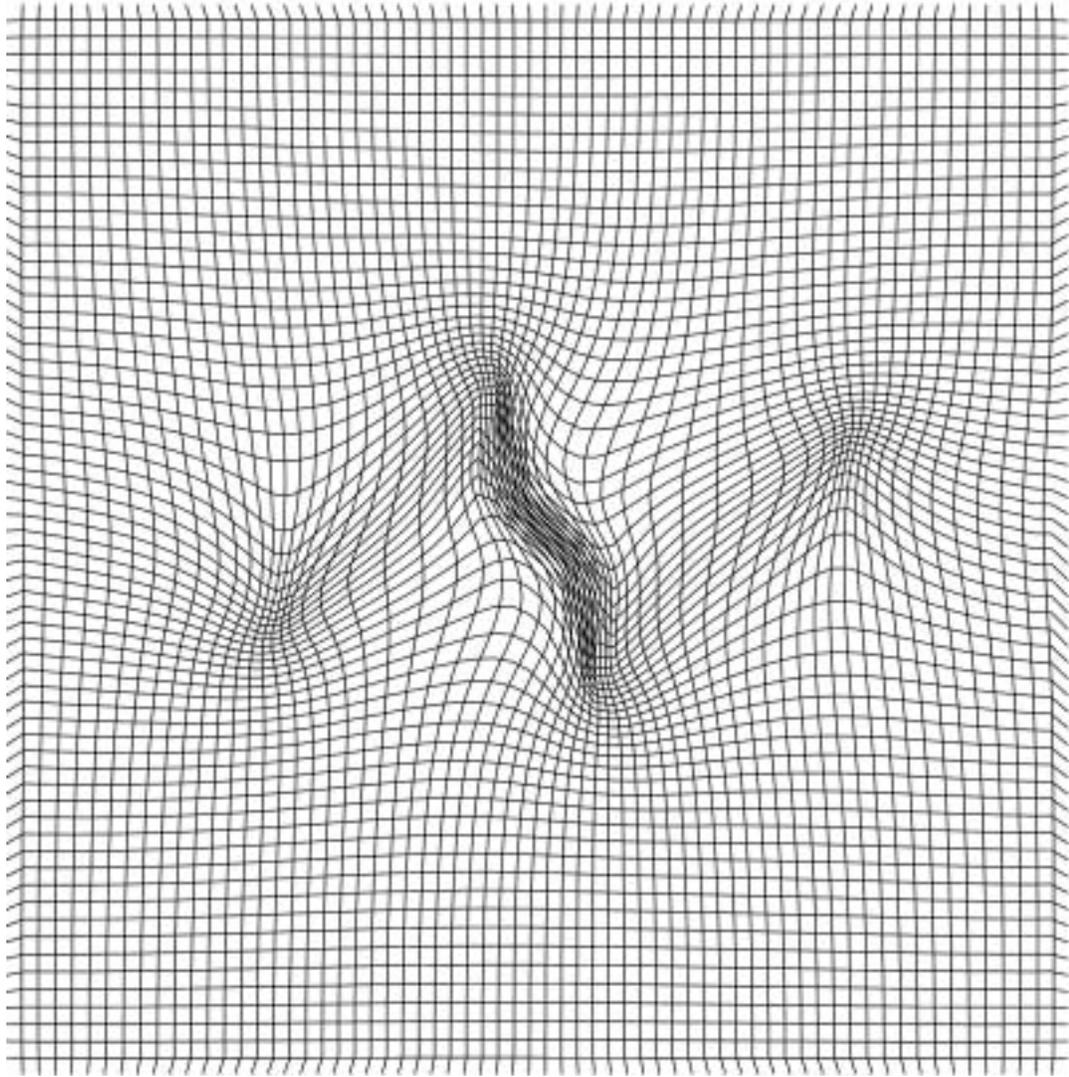
**Fig. 4(a): the deformation field of figure 3(b).**



**Fig. 4(b): the deformation field of matching the line segment in (a) with oscillations added**



**Fig. 5:** The same testing problem as in figures 3 and 4(a) is computed using the modified Beg's algorithm. The deformation field obtained in this case is different from figure 5(a).

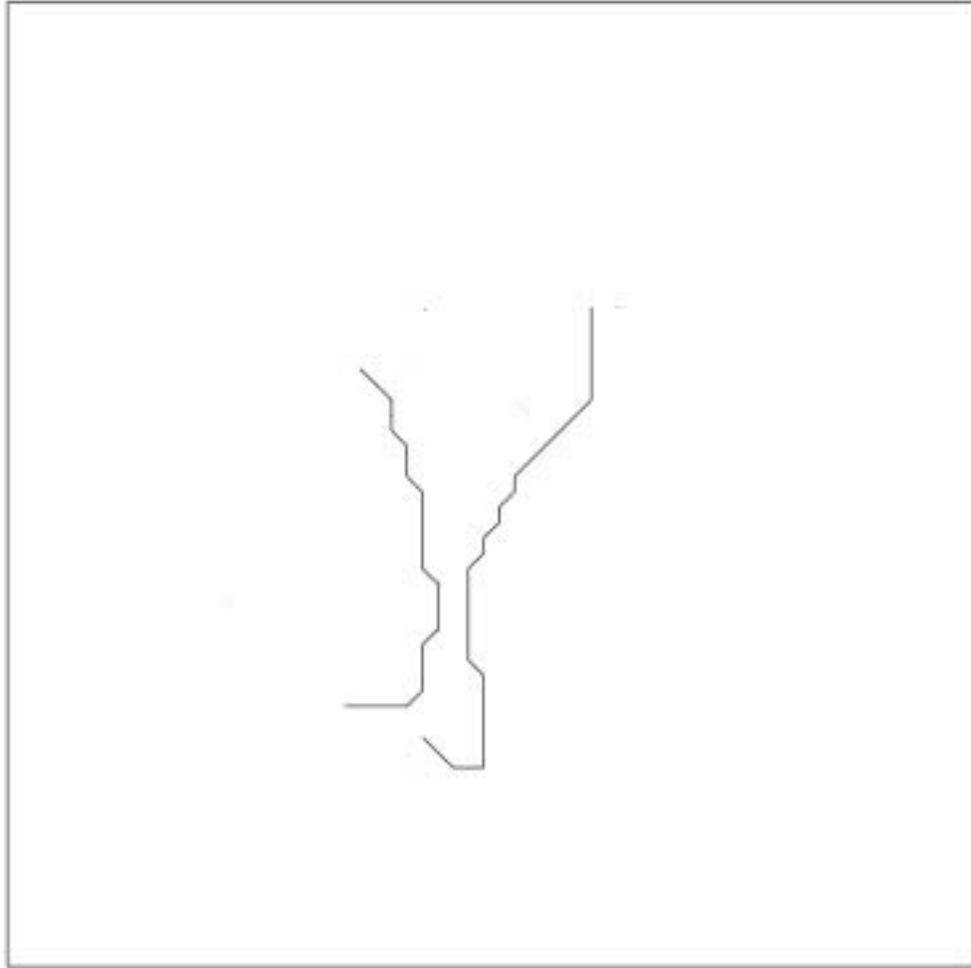


**Fig. 6: The diffeomorphism given by the level set based landmark matching with distance re-initialization. The landmarks to be matched are  $(16, 32)$  to  $(16, 27)$ ,  $(29, 35)$  to  $(29, 40)$ ,  $(29, 29)$  to  $(29, 34)$ ,  $(35, 35)$  to  $(35, 30)$ ,  $(35, 29)$  to  $(35, 24)$ , and  $(50, 32)$  to  $(50, 37)$  on a 64 by 64 image**

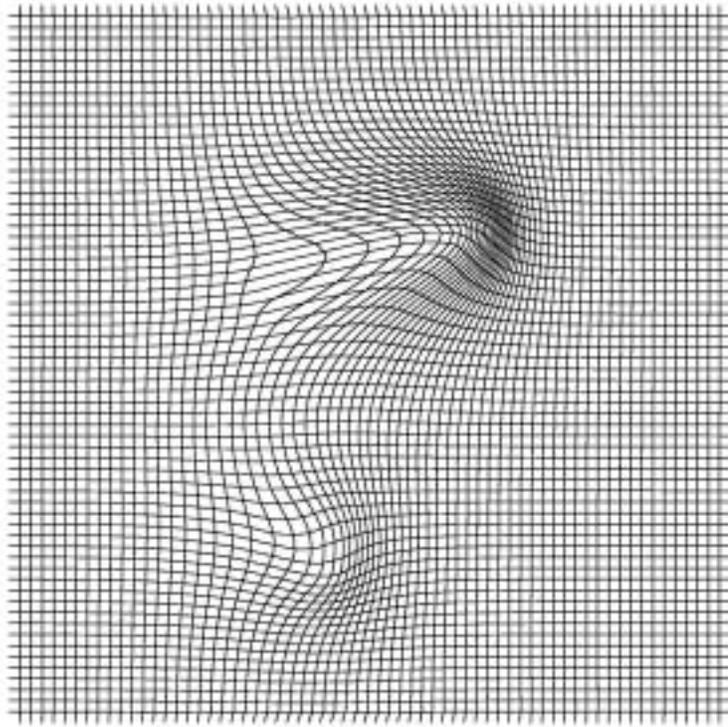
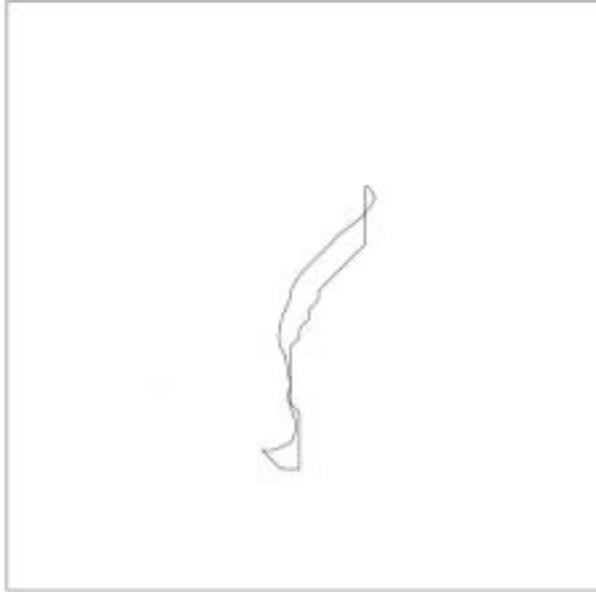
The following example serves for three purposes. Two random open curves are drawn on a 64 by 64 grid (shown in figure 7). The corresponding level set function representations are generated. The differential operator  $L^+L$  used in these numerical examples is  $50(-\Delta + 0.1 \times id)^2$ . Level set functions and distance functions in the template are re-initialized as described before.

First, landmark matching is applied to match the two pairs of the end points of the two curves using the first approach (equation (1) and (2)) introduced in chapter 7. The positions of the corresponding two open curves after end point matching is shown in the upper panel of figure 8, and the deformation field in the lower panel of figure 8. The length of the geodesic path in this case is 8.97. Notice that matching the end point alone does not ensure whole curve matching.

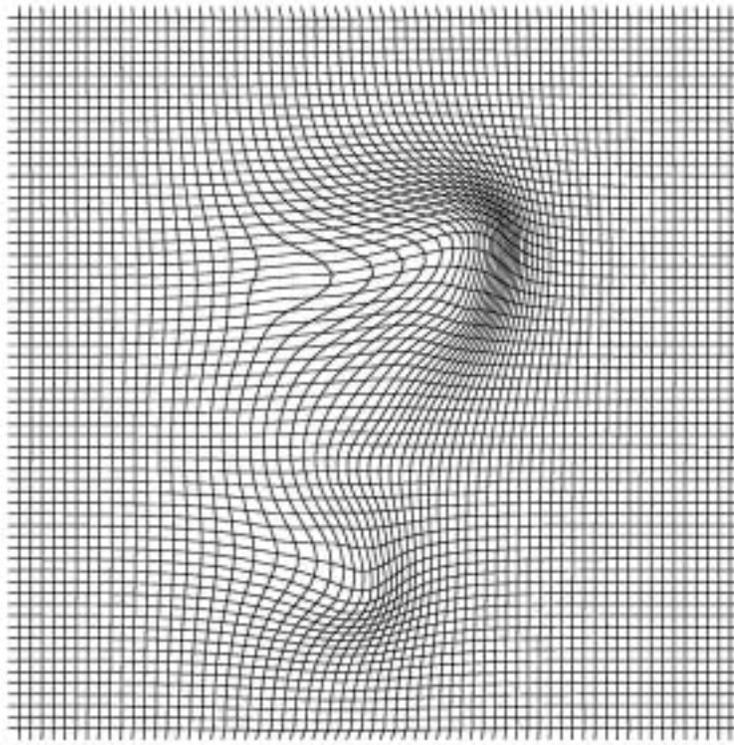
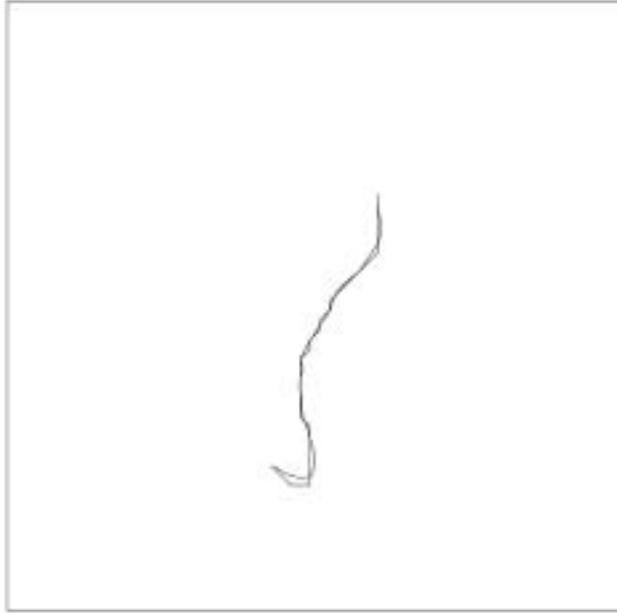
Figure 9 shows the whole matching of the two curves using the method introduced in chapter 5. The upper panel shows the corresponding two curves under the final deformation field that is shown in the lower panel of figure 9. The length of the geodesic path in the case of whole curve matching is 10.87.



**Fig. 7: Two random curves are drawn by hand on a 64 by 64 grid for illustrating landmark matching, curve matching, and the equivalence class of images under the action of translation, scaling, and rotation**



**Fig. 8: Upper panel: the positions of the two open curves after matching the end points of the curves. Lower panel: the corresponding deformation field.**

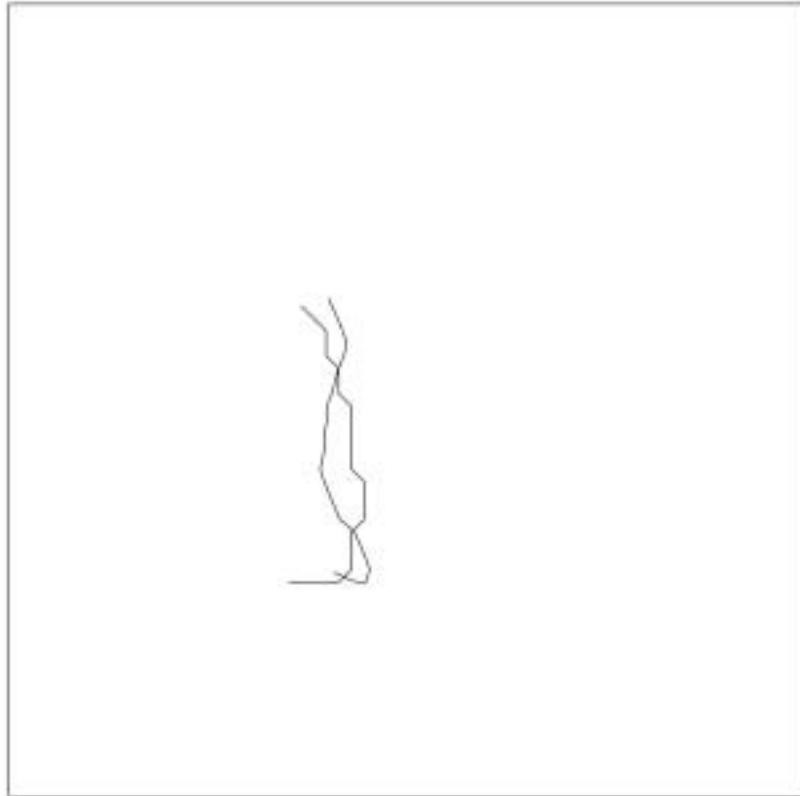


**Fig. 9: Upper panel: the positions of the two open curves after matching the two curves. Lower panel: the corresponding deformation field.**

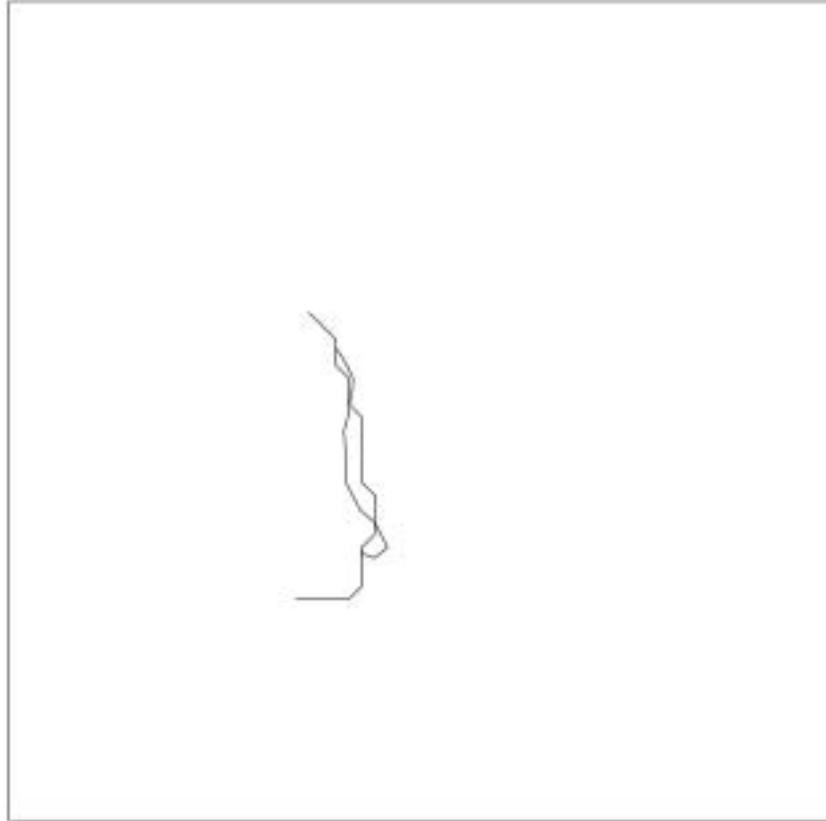
In the next part of this numerical example, we try to take into account the equivalence classes of the study by factoring out translation, scaling, and rotation. In other words, the curve in the template is now matched to an equivalent class of the curve in the study using the formulations in chapter 8. The cost function used is the Hausdorff metric between two curves approximated by raising the power of the distance function to 7. The final result is shown in figure 10. Figure 11 shows the result when the original cost function (equation (9) in chapter 8) is used. It is noticed in this figure that, due to the presence of the scaling factor in equation (1) of chapter 8, the open curve in the study will be shrunk to a smaller scale since the second term in the cost function (9) of chapter 8 will decrease as the scaling factor decreases. However, it is no longer the case when the Hausdorff measure is used, since the measure itself does not depend directly on the scale of the curve in the study.

The four parameters in figure 10 are  $a = -2.442$ ,  $b = -6.978$ ,  $r = 0.7340$ , and  $\theta = -0.3865$ , while the four parameters in figure 11 are  $a = -2.582$ ,  $b = -5.582$ ,  $r = 0.5314$ , and  $\theta = -0.3828$ . The unit for the translation parameters  $a$  and  $b$  is pixel. The corresponding deformation field of matching the curve in the template to this equivalent class of the curve in the study is shown in the lower panel of figure 12. The upper panel of figure 12 shows the relative positions of the two curves under this deformation field. The length of the geodesic that links the object in the template to the equivalence class of the object in the study is now 0.7187.

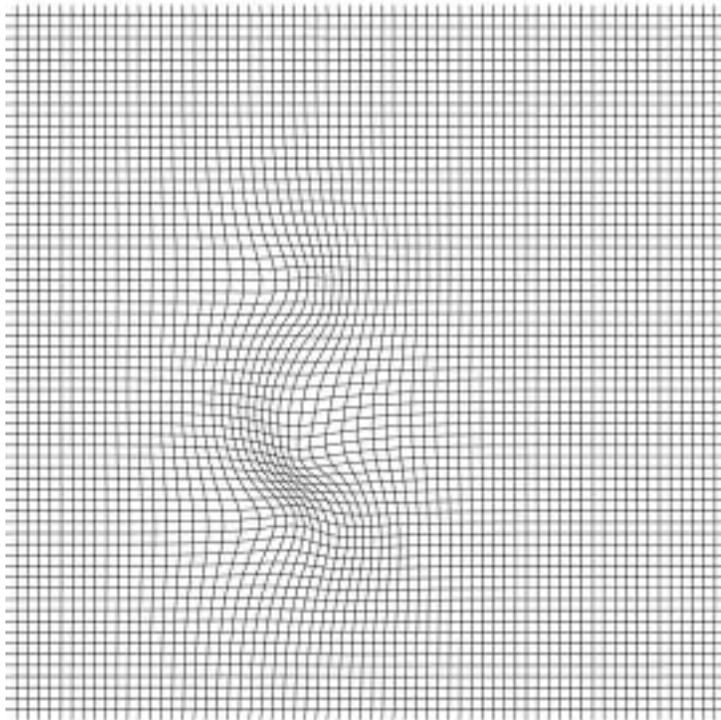
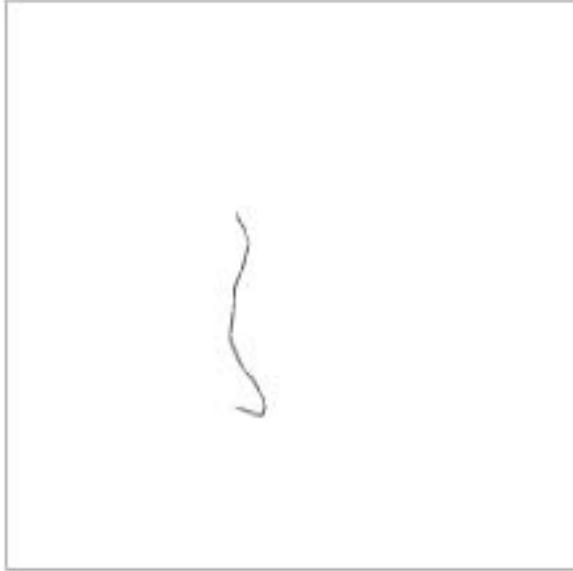
I would like to thank Wei Zhu for carrying out the computation in figure 10 and 11, and I am grateful for his help in preparing the corresponding level set functions and the corresponding distance functions.



**Fig. 10:** The positions of the two open curves after taking into account the equivalence class of the curve in the study by factoring out actions of translation, scaling, and rotation. The Hausdorff metric between two curves (approximated by raising the power of the distance function to the 7-th power) is used as the cost function. The four parameters in this example are  $a = -2.442$ ,  $b = -6.978$ ,  $r = 0.7340$ , and  $\theta = -0.3865$ .

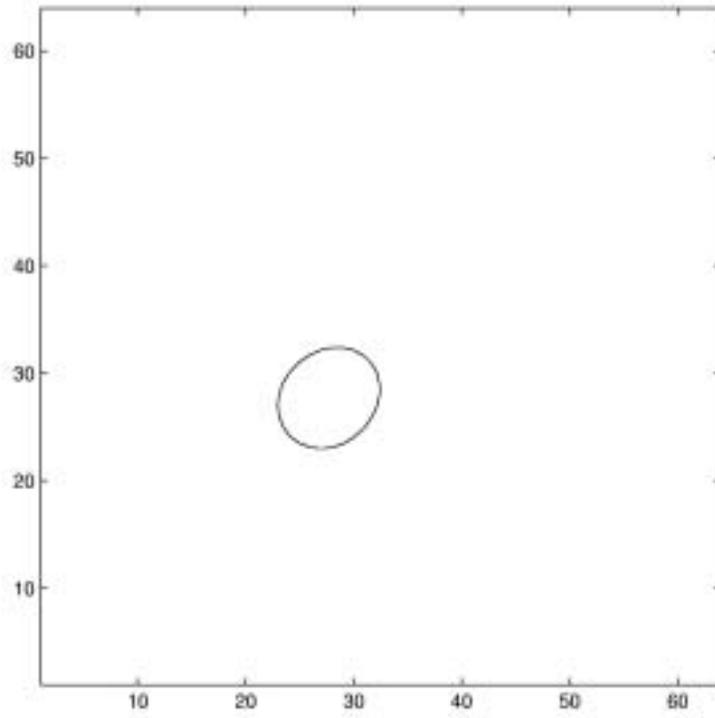


**Fig. 11: The positions of the two open curves after taking into account the equivalence class of the curve in the study by factoring out actions of translation, scaling, and rotation. The original cost function between two curves is used without raising the distance function to higher power than one. The four parameters in this example are  $a = -2.582$ ,  $b = -5.582$ ,  $r = 0.5314$ , and  $\theta = -0.4828$ .**

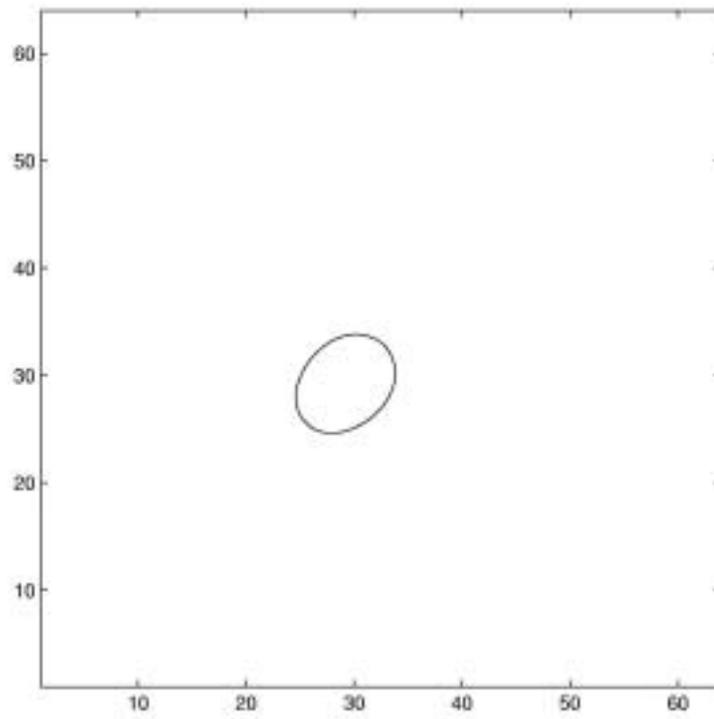


**Fig. 12: Upper panel: the positions of the two open curves after matching the two curves using the target in figure 10 where translation, scaling and rotation are factored out. Lower panel: the corresponding deformation field.**

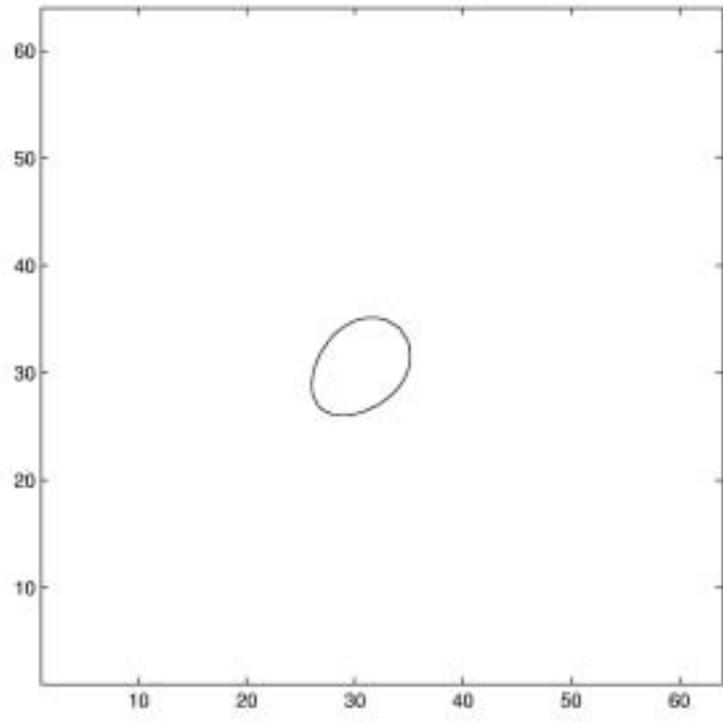
The next two figures show the geodesic path that links the object in the template to the object in the study. Figure 13 and 14 are the geodesic path that links the two oval shapes in the second example of shape matching. Figure 13 shows the position of shape in the template being carried along the geodesic path and figure 14 shows the underlying grid deformation of the geodesic path. Figure 15 shows the geodesic path that links the two open curves in figure 7 without taking into account the equivalence class. The time zero to one is discretized to 10 time steps and each time step is 0.1.



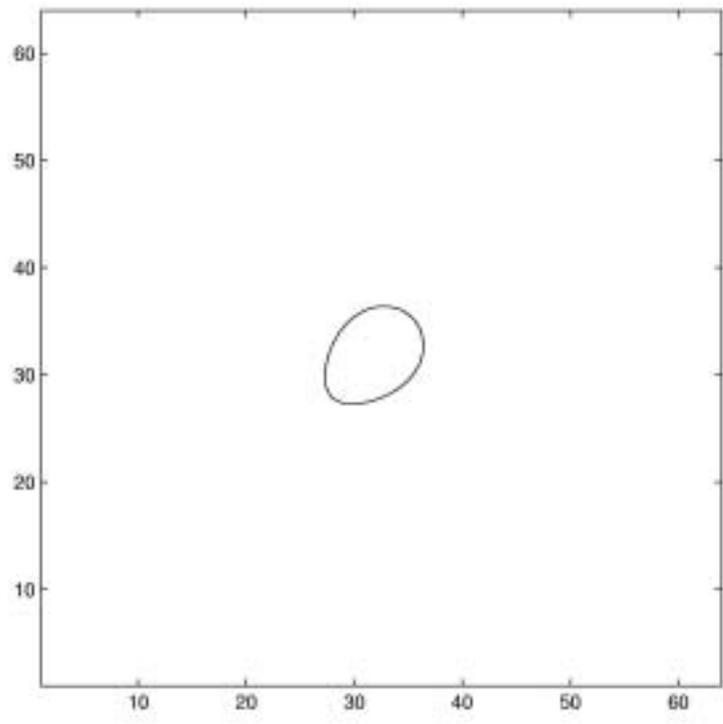
(a)



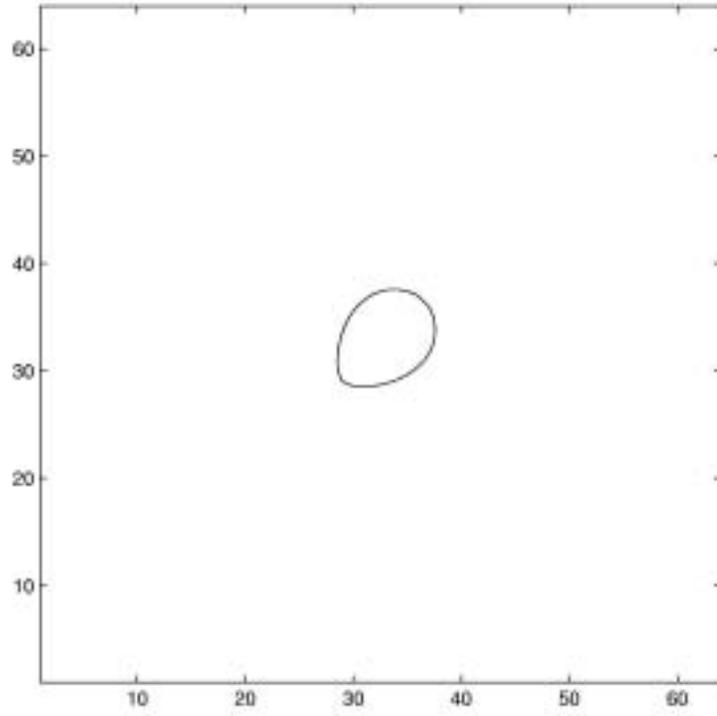
(b)



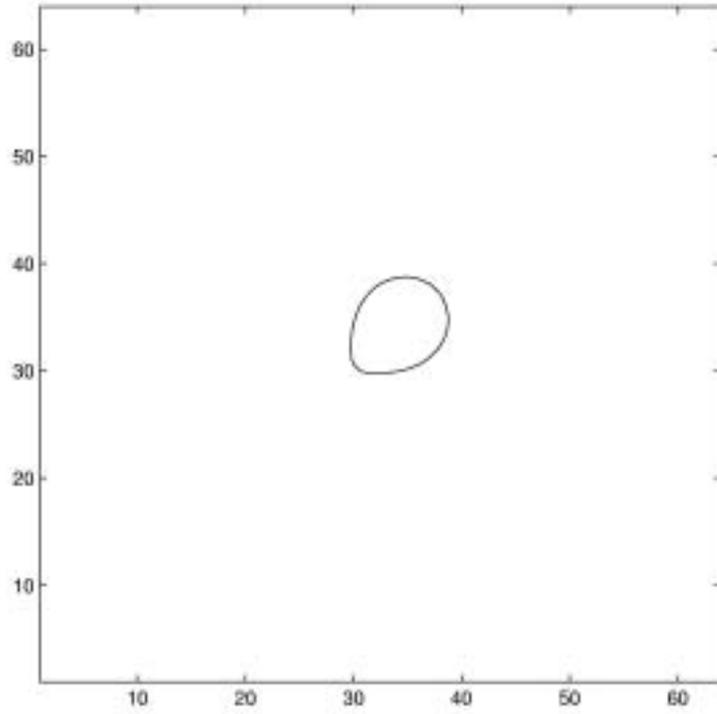
(c)



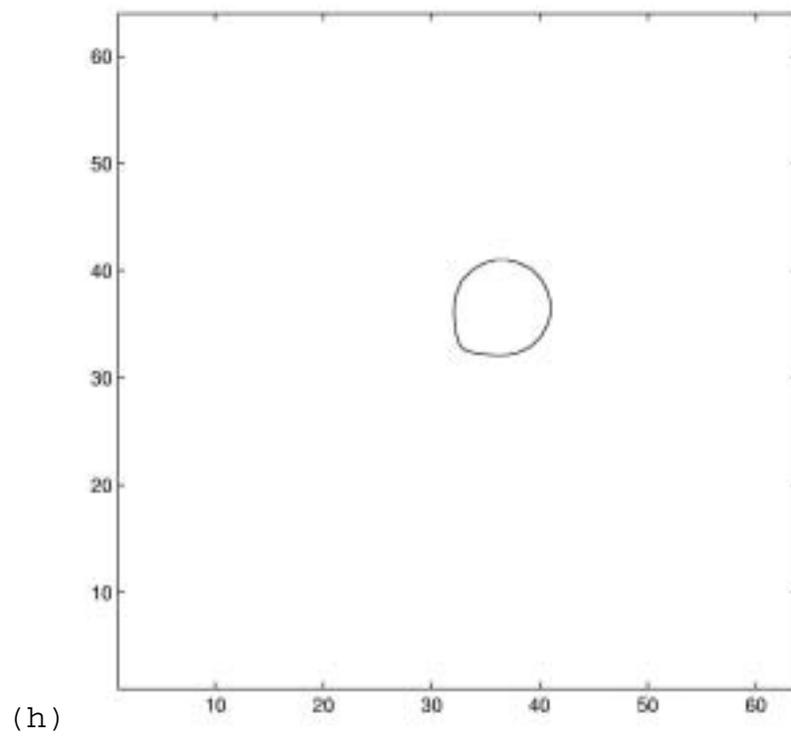
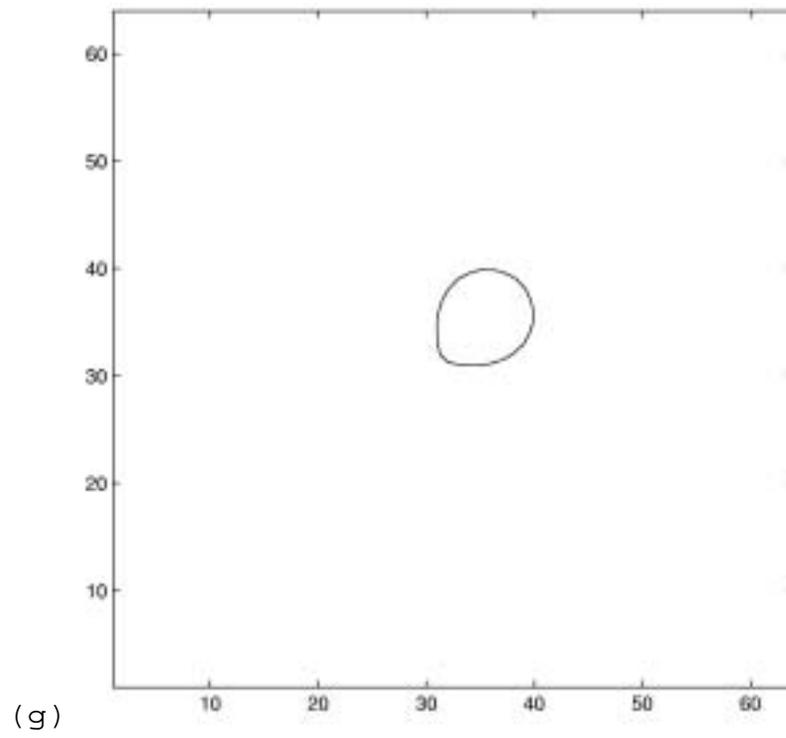
(d)

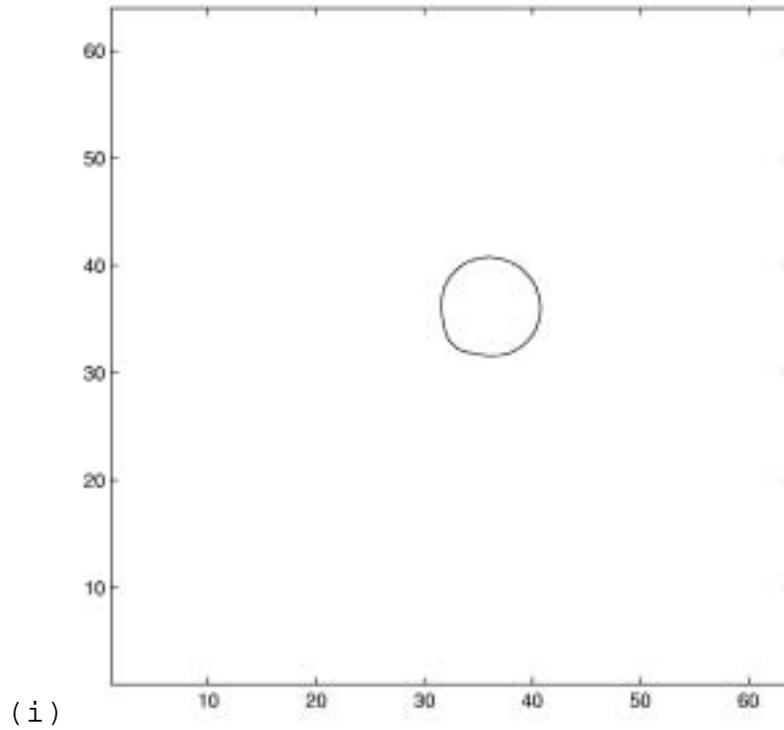


(e)



(f)





**Fig. 13: The oval shape carried by the geodesic flow that links the two oval shapes in figure 2 at time 0.1 (a), time 0.2 (b), time 0.3(c), 0.4 time (d), time 0.5 (e), time 0.6 (f), time 0.7 (g), time 0.8(h), and time 0.9 (i).**

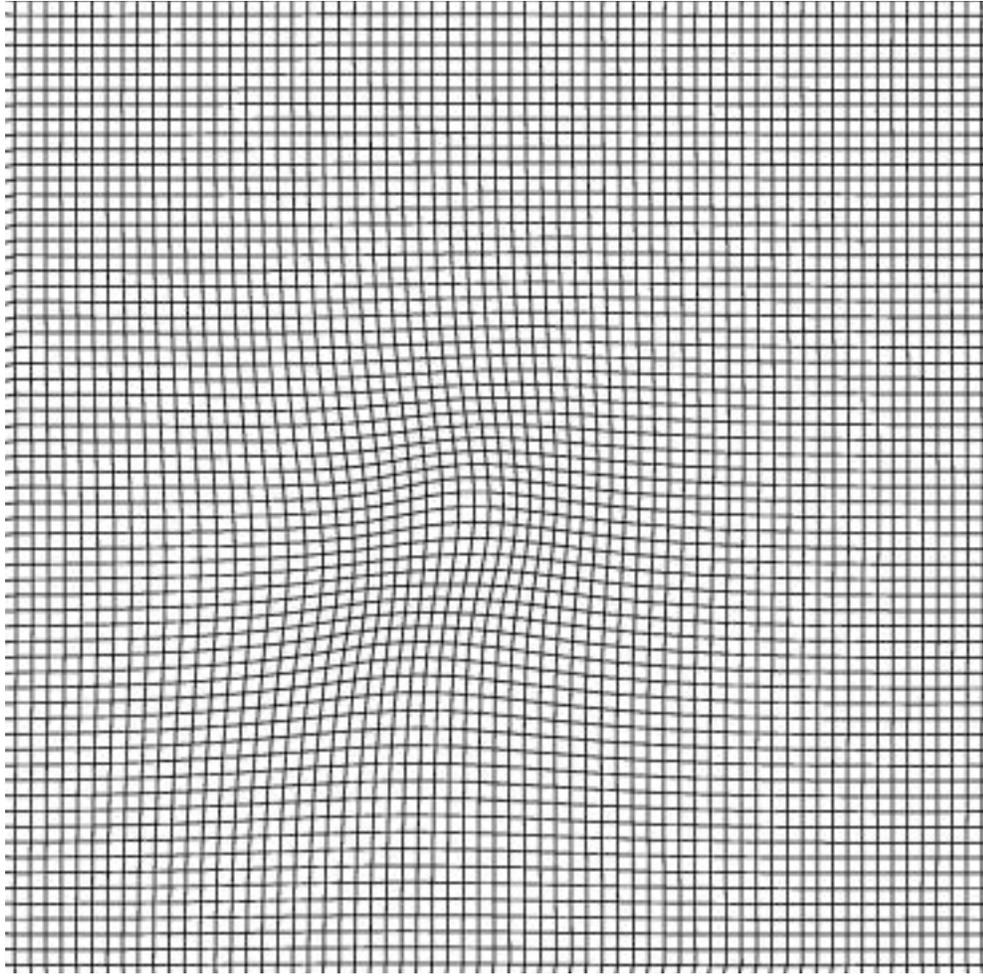


Fig.14 (a)

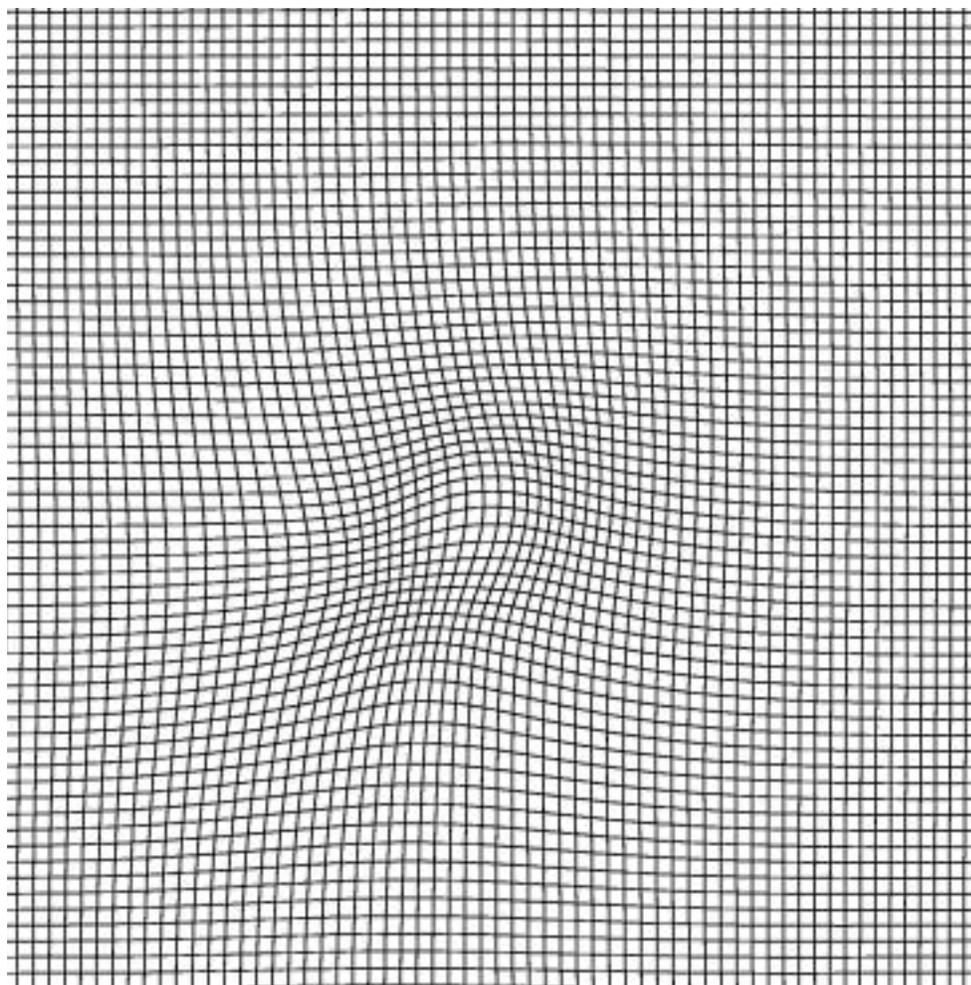


Fig.14 (b)

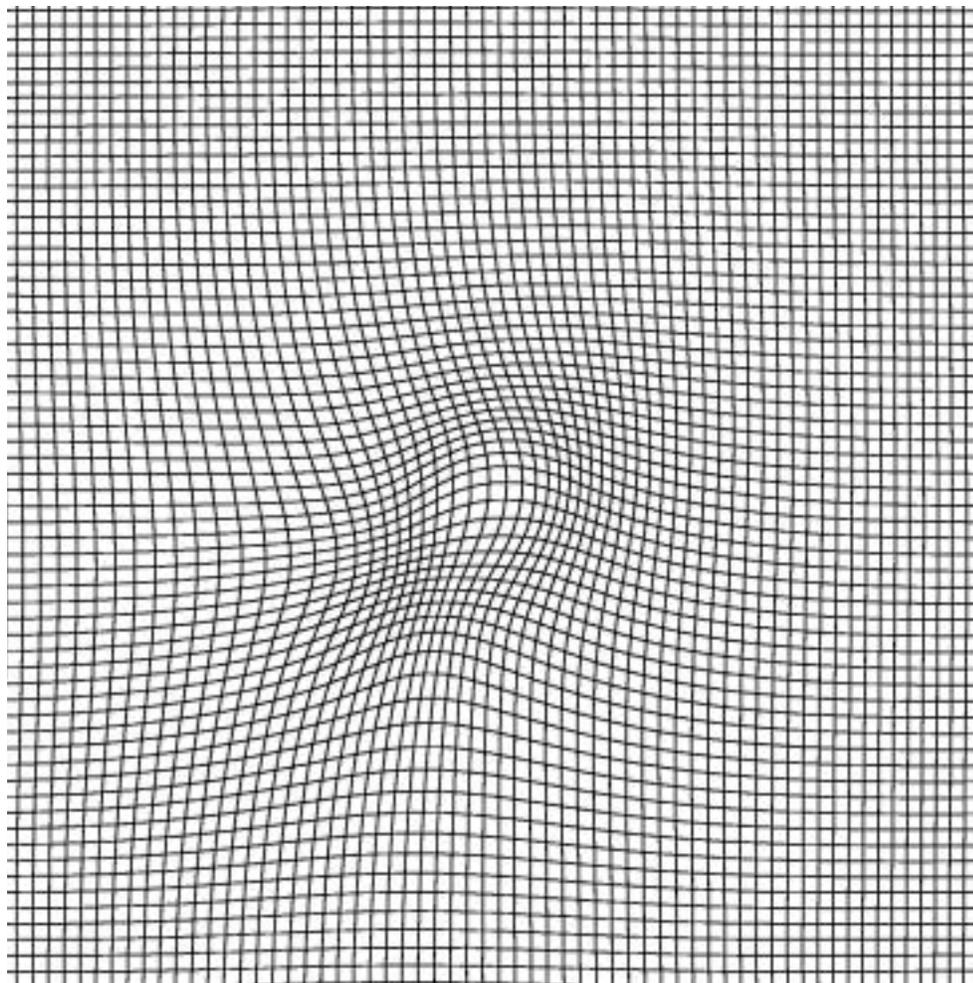


Fig.14 (C)

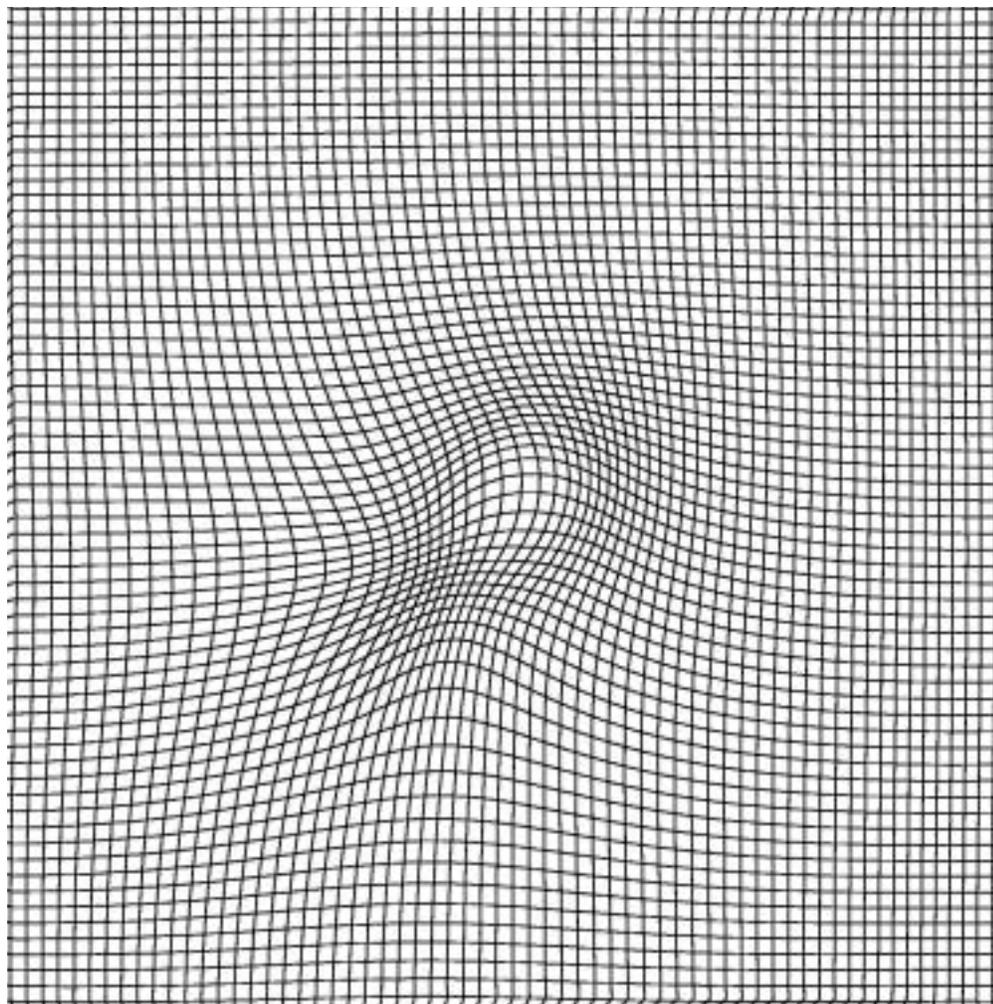


Fig.14 (d)

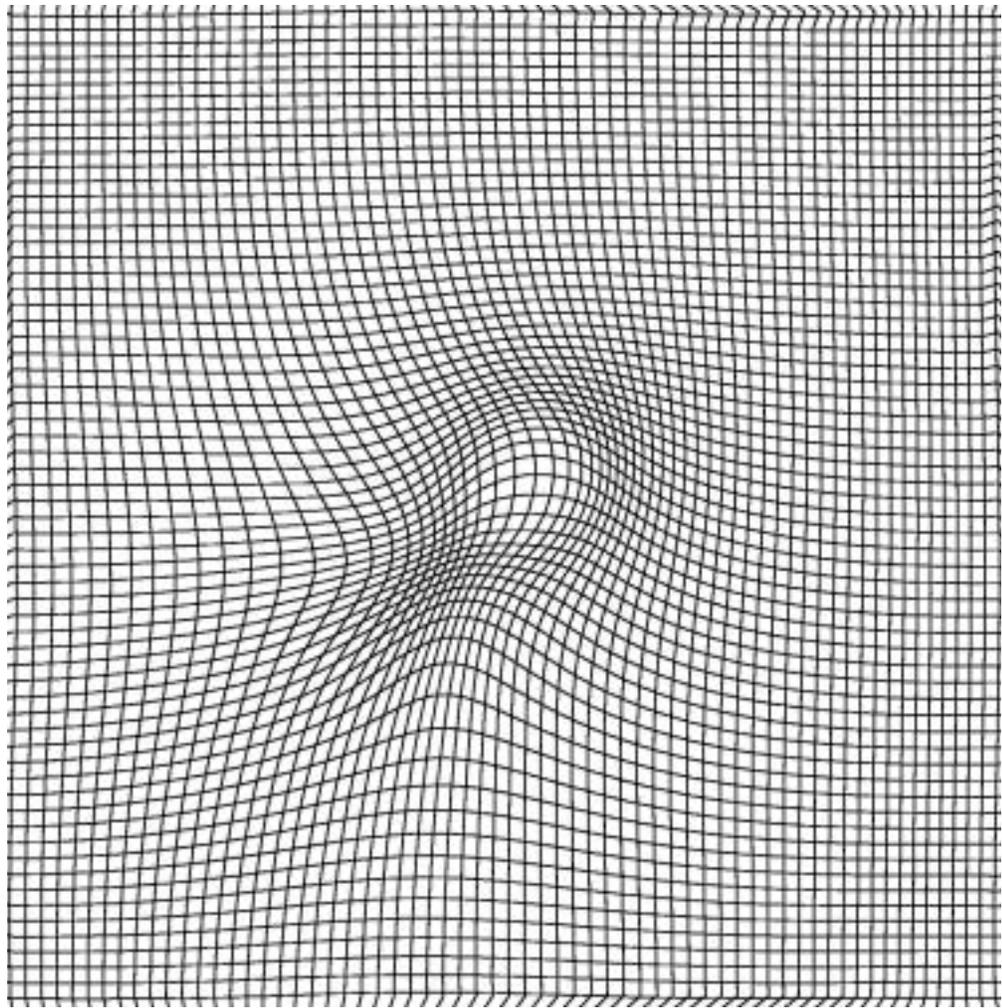


Fig.14 (e)

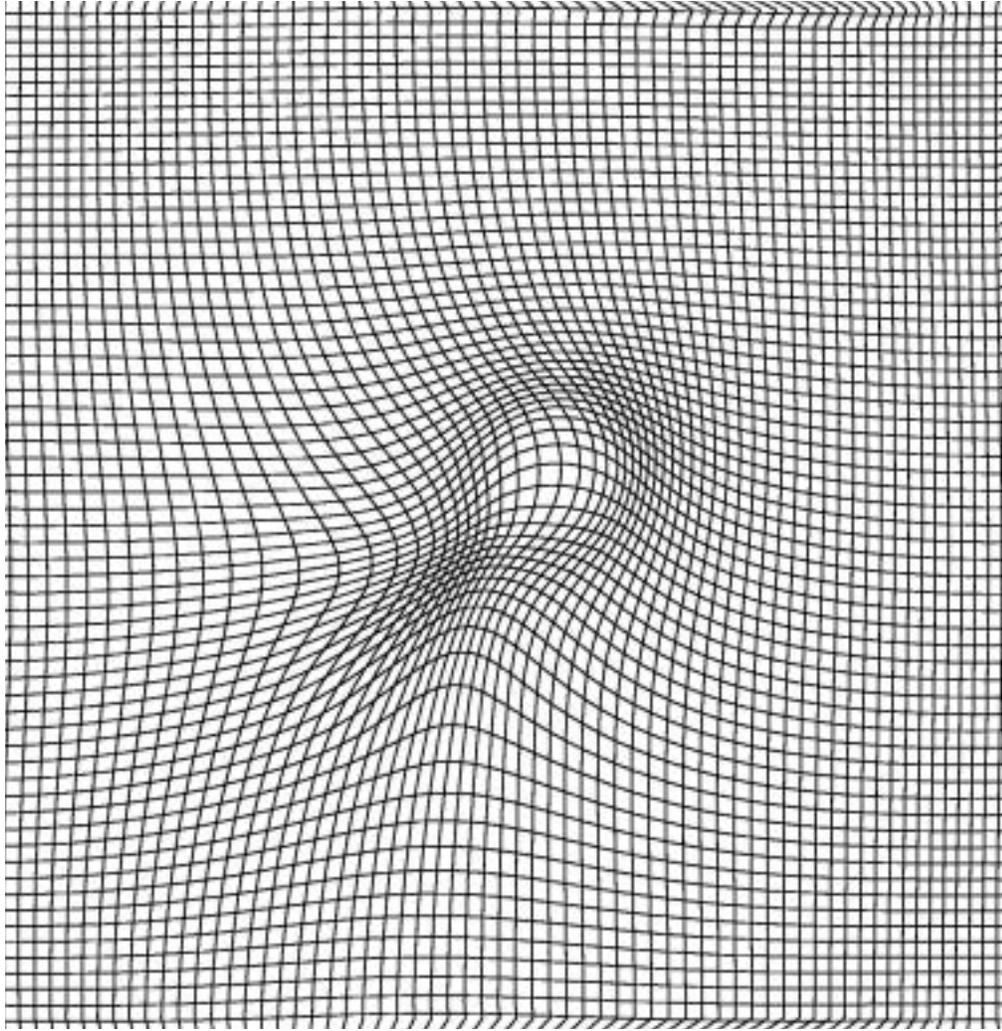


Fig.14 (f)

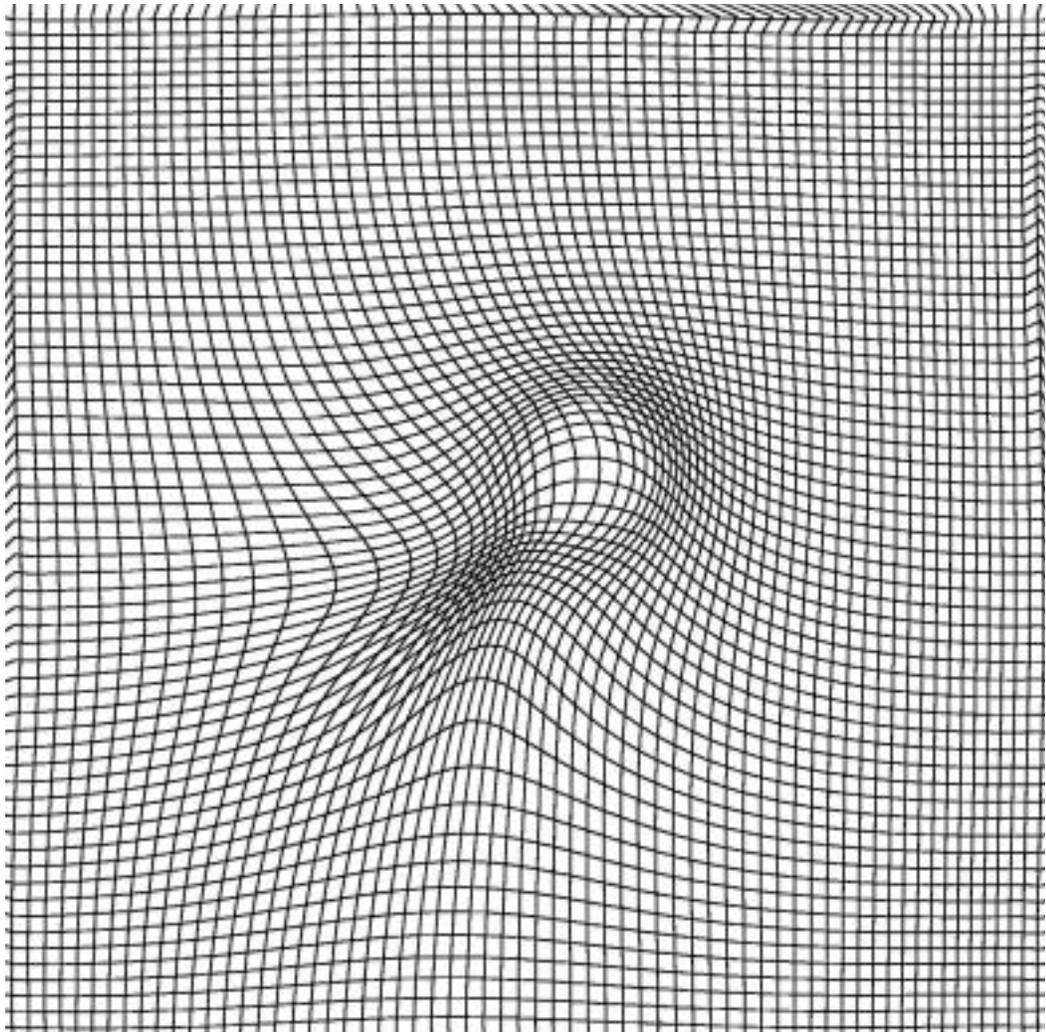


Fig.14 (g)

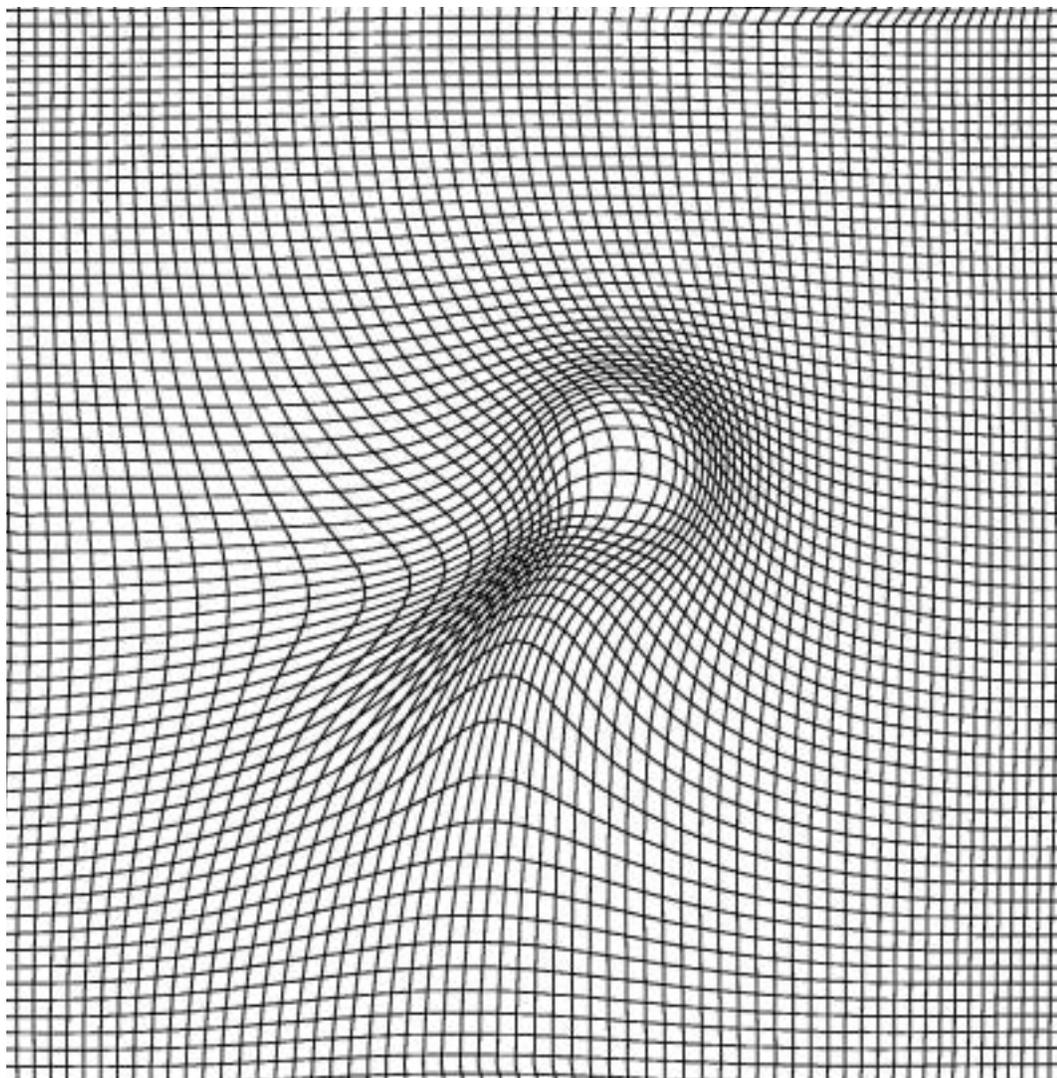


Fig.14 (h)

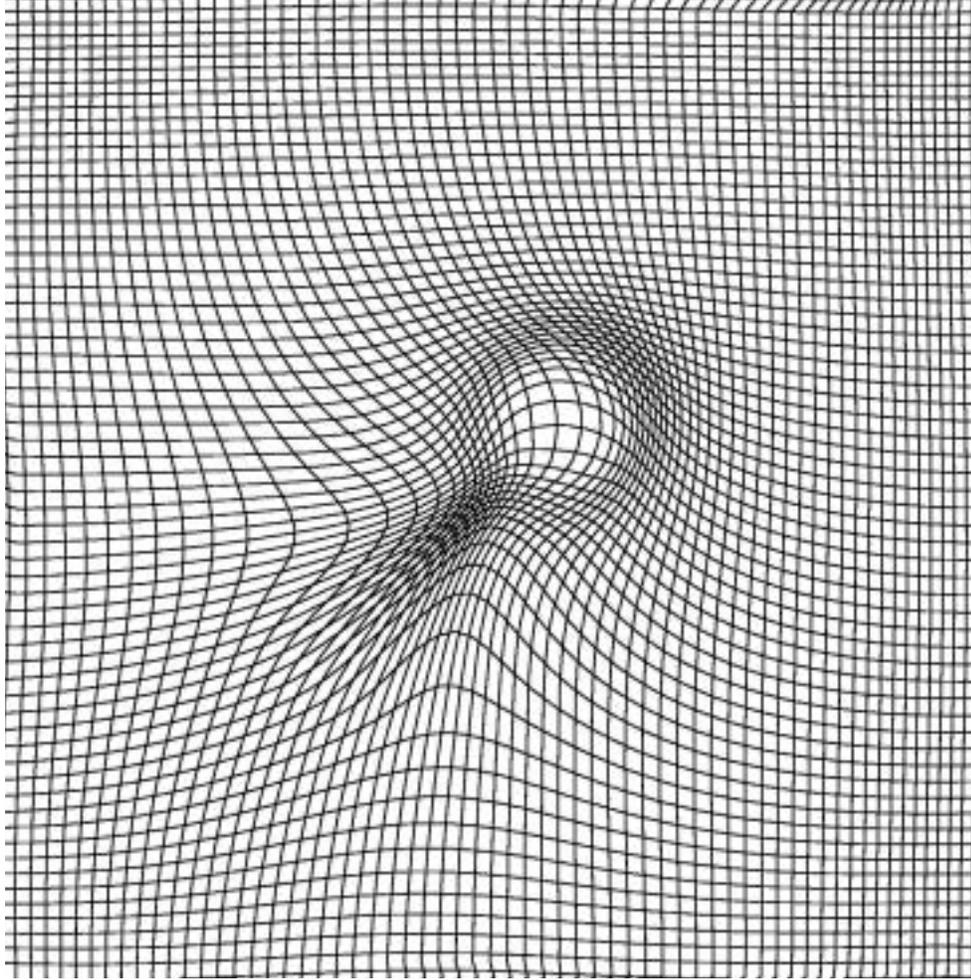


Fig.14 (i)

**Fig. 14: The geodesic flow that links the two oval shapes in figure 2 at time 0.1 (a), time 0.2 (b), time 0.3(c), 0.4 time (d), time 0.5 (e), time 0.6 (f), time 0.7 (g), time 0.8(h), and time 0.9 (i).**

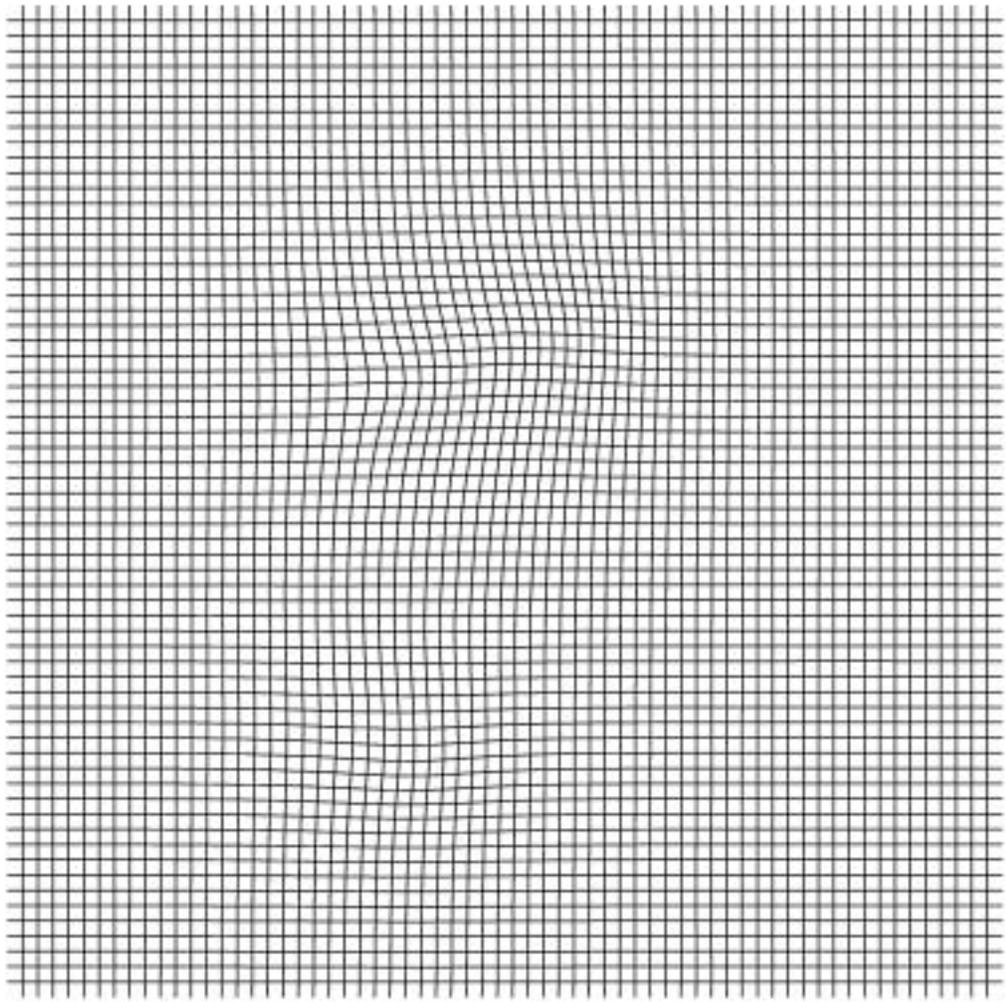


Fig.15 (a)

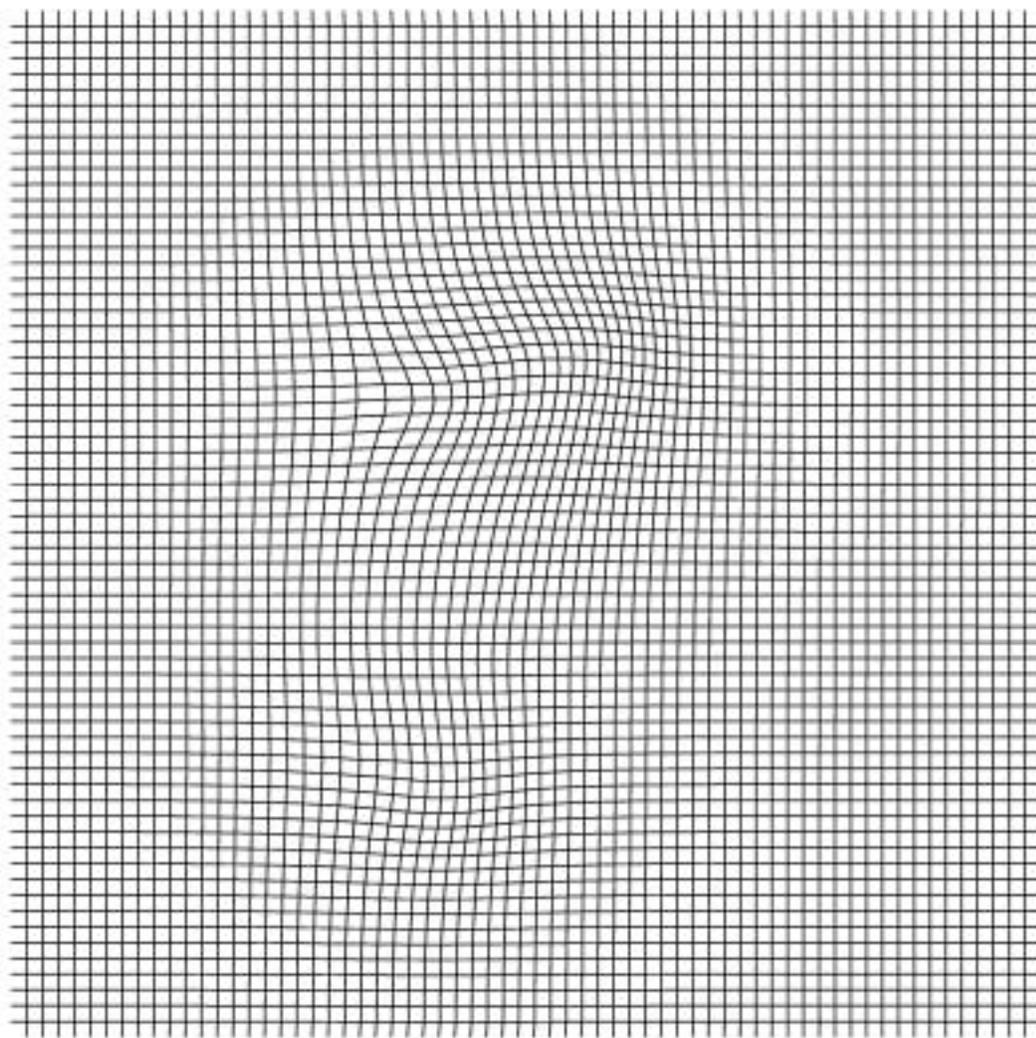


Fig.15 (b)

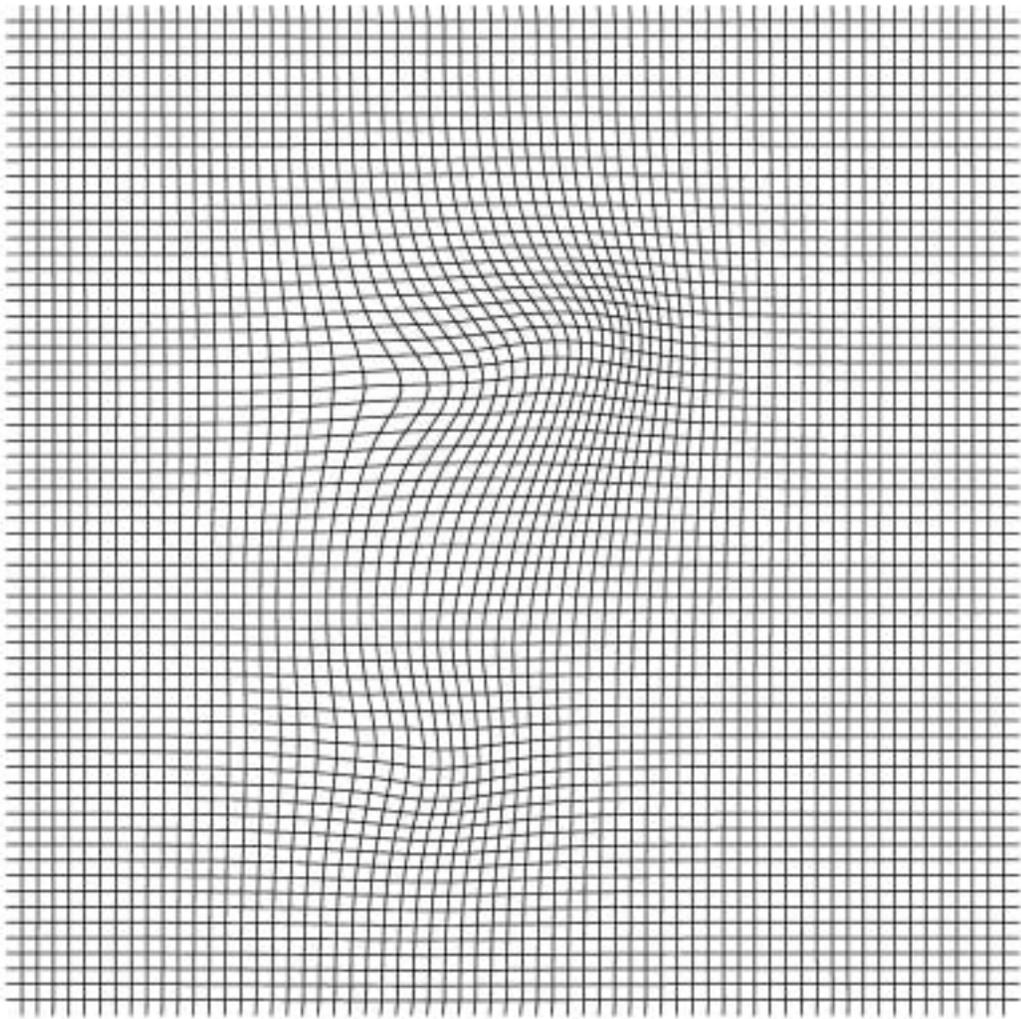


Fig.15 (c)

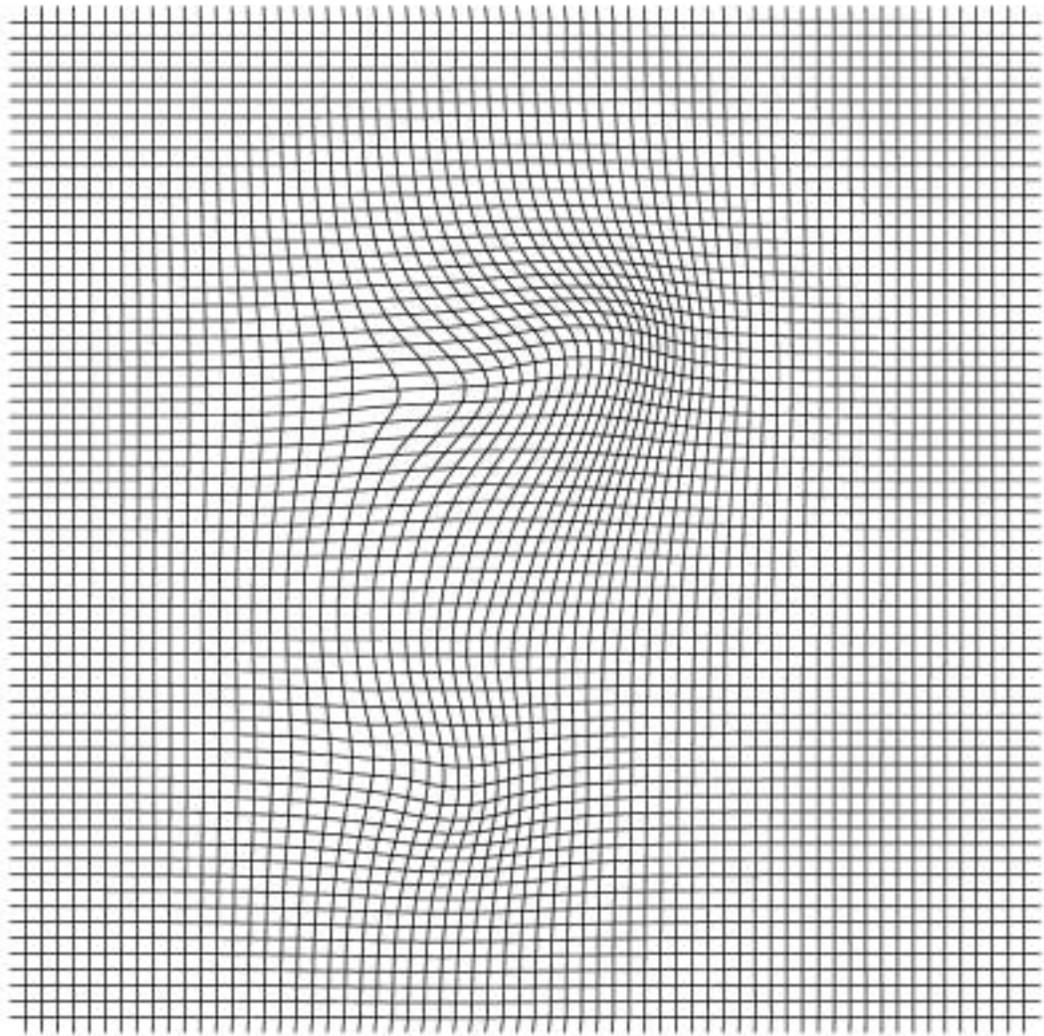


Fig.15 (d)

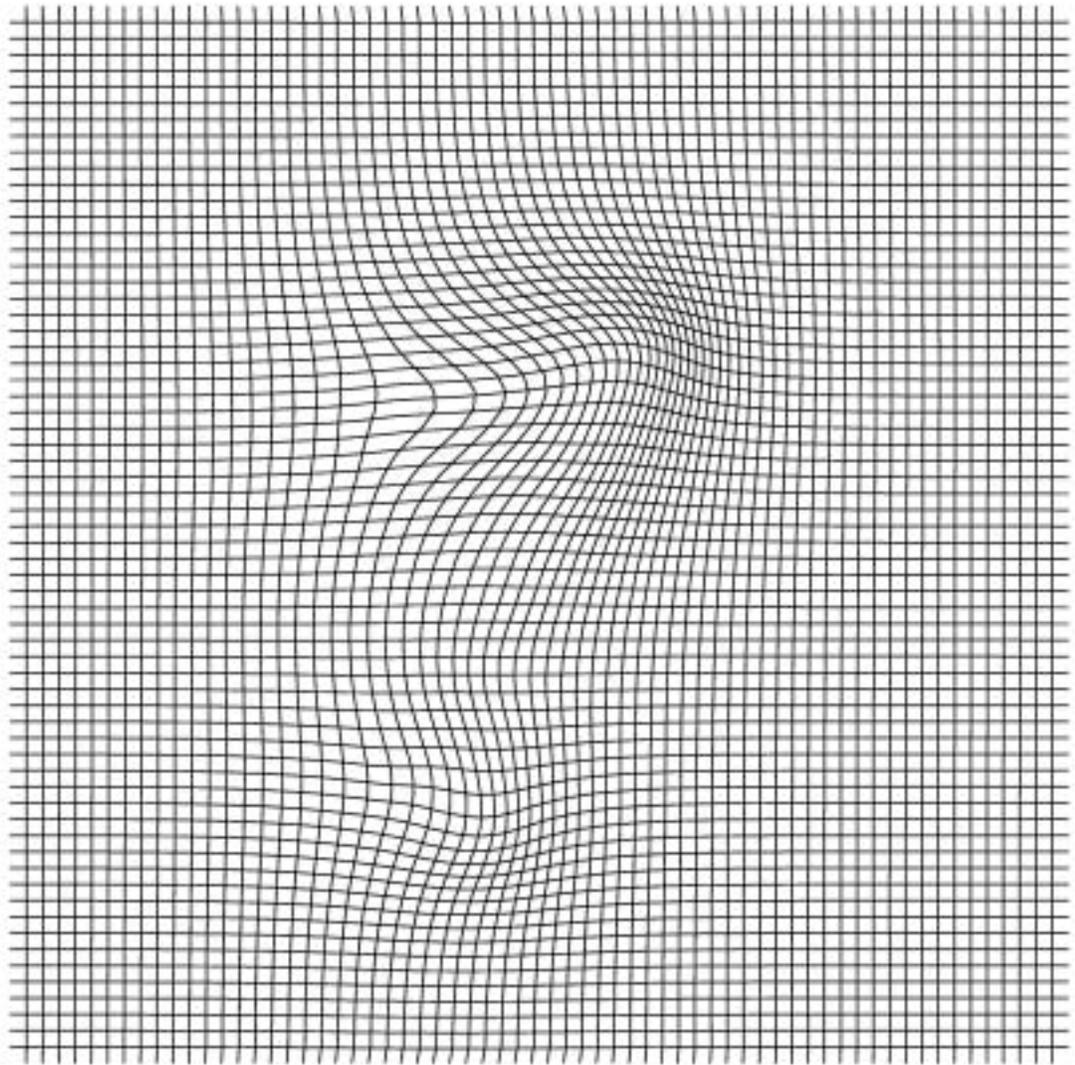


Fig.15 (e)

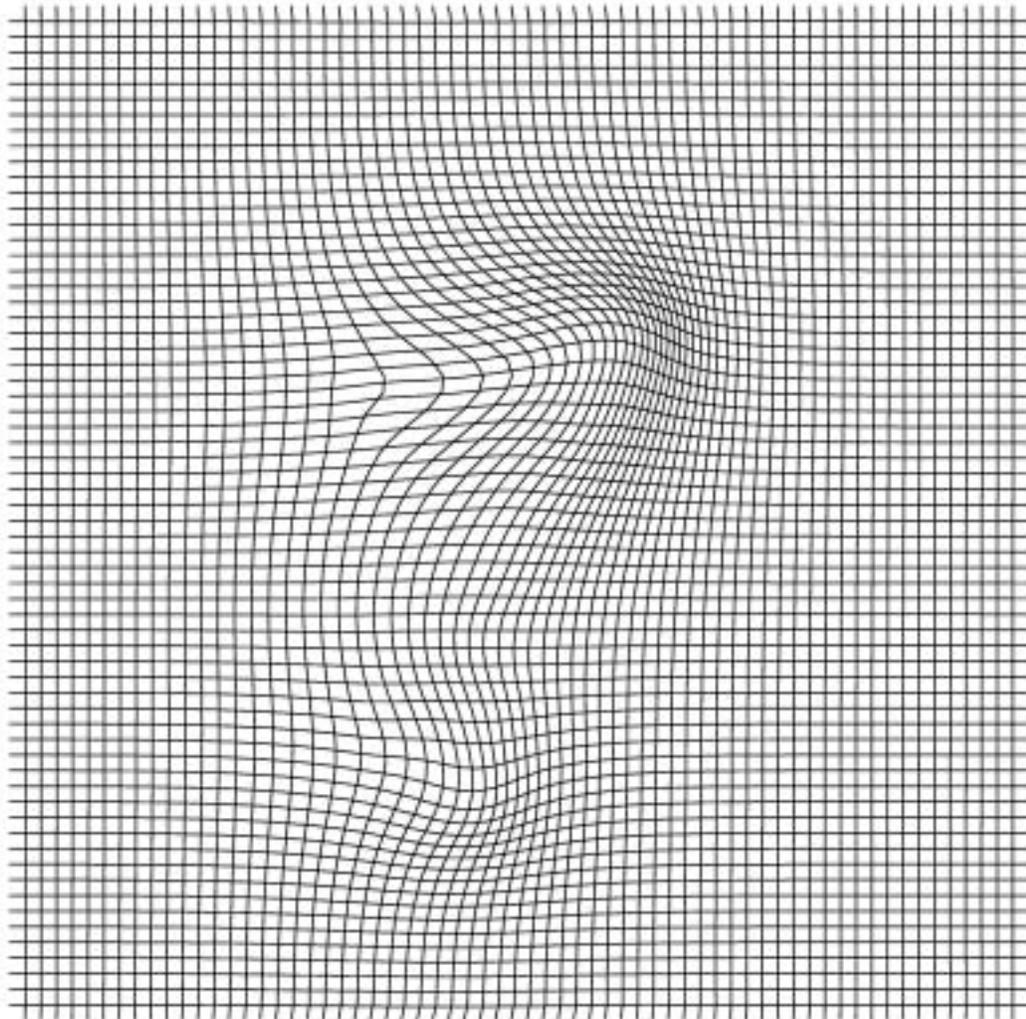


Fig.15 (f)

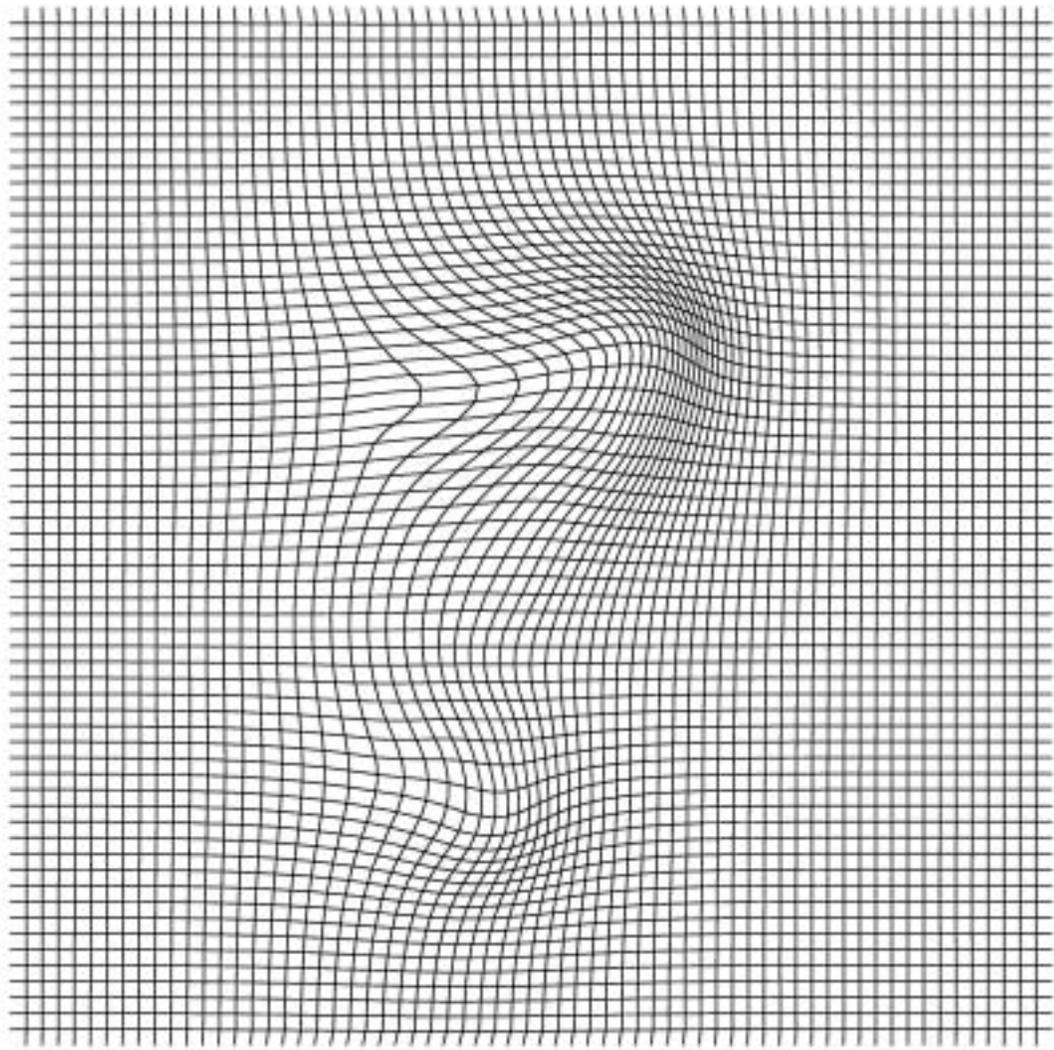


Fig.15 (g)

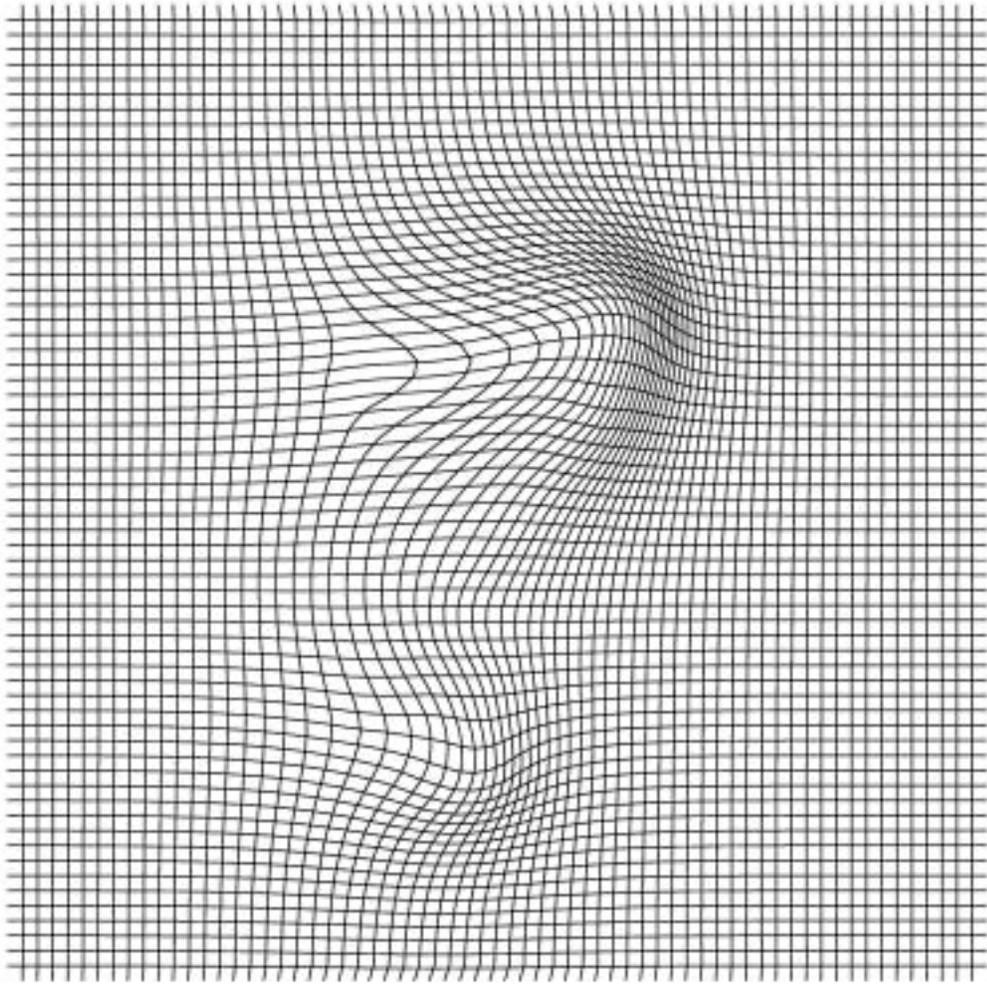


Fig.15 (h)

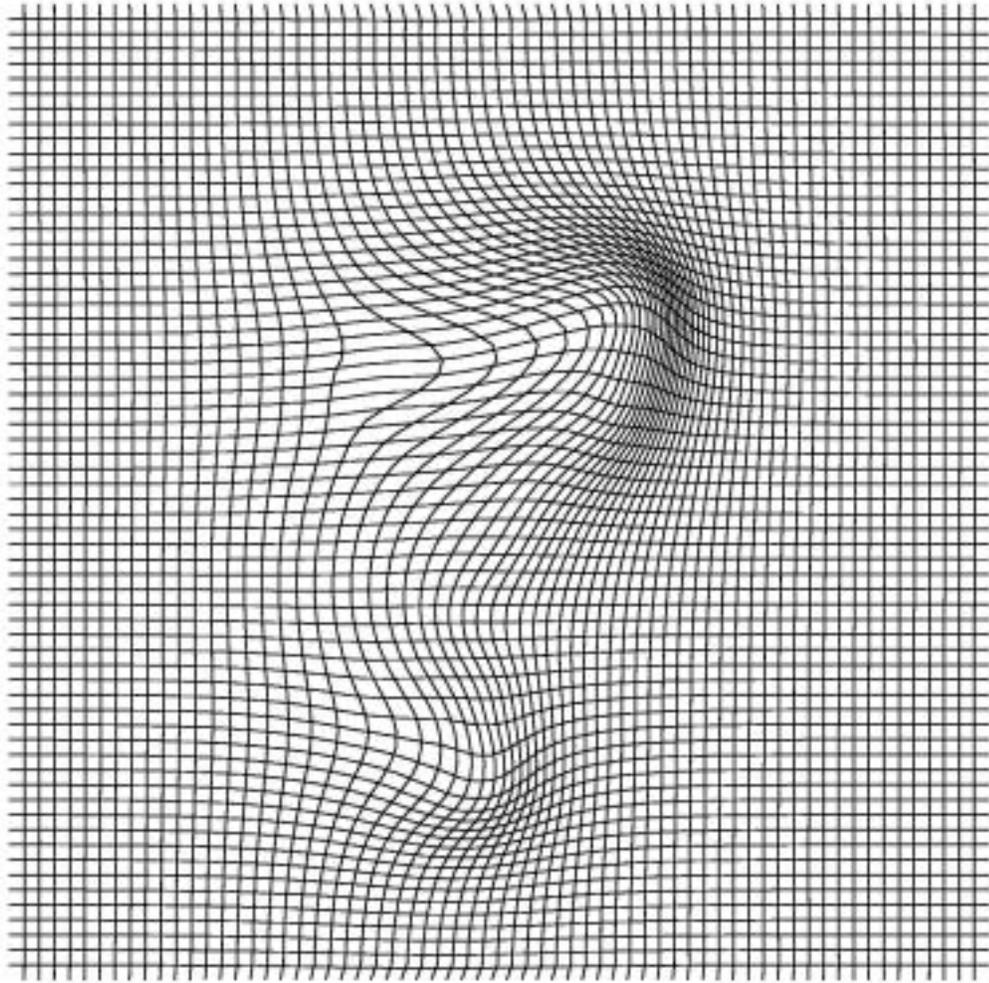


Fig.15 (i)

**Fig. 15: The geodesic flow that links the two open curves in figure 7 without taking into account the equivalence class at time 0.1 (a), time 0.2 (b), time 0.3(c), 0.4 time (d), time 0.5 (e), time 0.6 (f), time 0.7 (g), time 0.8 (h), and time 0.9 (i).**

# Chapter 11

## Conclusion and Future Directions

*Abstract- In this final chapter, we summarize what have been achieved and the contributions of this dissertation. We also outline few interesting directions that are worth exploring.*

## **1. Conclusion**

In this dissertation, we bring together the framework of the level set method and computational anatomy. The following is a summary of what have been achieved.

### **The semi-Lagrangian level set method**

The original level set method is a Eulerian framework. The front that we wish to track is being implicitly represented as the zero level set of the corresponding level set function of one higher dimension. The Eulerian nature of the original level set method can be viewed intuitively as if we are standing on the grid points and updating the values of the level set function on the grid points regardless of the current position of the front. In order to recover the position of the front, we trace back the zero crossing of the level set function. The main advantage of this approach is to treat topological changes automatically. Due to this nature, it is well suited for tasks that involve the tracking of interface commonly encountered in many disciplines. Thus, since the introduction of the level set method, a lot of image processing techniques have been designed (or re-interpreted) and implemented in this manner.

However, the main advantage of the level set method becomes the main drawback as well when it is applied to the computational anatomy in which the objects are to be compared along with quantifying the underlying grid deformation. First of all, in computational anatomy, topological changes are not allowed since they are not physiologic. Secondly,

due to the Eulerian nature of the original level set method, we could never track how the particles move since we do not stand on the interface and move along with it. However, it would be desirable if the level set method could be applied in computational anatomy since it provides a strong tool of representing fronts using implicit representation.

The contribution of this dissertation is, by re-interpreting the level set method in a semi-Lagrangian reference formulation, it now can be applied to the framework of computational anatomy. In the first few chapters of this dissertation, we give an overview of the level set method and the computational anatomy through diffeomorphisms generated by infinite dimensional group actions based on formulations borrowed from continuum mechanics. The semi-Lagrangian implementation is introduced and discussed at the end of chapter 4. We should point out that the semi-Lagrangian implementation introduced in this dissertation is slightly different from [64-66]. In these references, the interpolation is being done at each iteration while the semi-Lagrangian implementation in this dissertation keeps track of the displacement field all the way back to the initial image, thus allows the tracking of the underlying grid deformation. Because of this, the term semi-Lagrangian in this dissertation is not exactly in the usual semi-Lagrangian sense, and a novel name “quasi-Lagrangian” might be more appropriate to describe this new technique.

## **The incorporation of the semi-Lagrangian level set method and computational anatomy**

In chapters 5, 6, 7, 8, and 9, we re-formulate all object matching problems in terms of the level set method. In the past [42, 51, 64-66], objects in the computational anatomy were being compared either in terms of the whole image (sum of least square or other global intensity measure), or in terms of landmarks placed on the objects beforehand. In this dissertation, we derive novel and efficient strategies of comparing overlapping and non-overlapping shapes, open curves, and landmarks in 2D. Furthermore, the same techniques are then generalized to deal with 3D objects. With the help of the level set method, we now no longer need to restrict ourselves to certain types of objects. Moreover, due to the same variational nature of these techniques, different types of objects can be compared easily within the same image. This gives us a whole new world of mathematical tools that allow us to explore and tackle the challenging problems encountered in today's computational anatomy.

## **2. Future Directions**

In what follows, we point out some interesting directions worth exploring that have not been treated or only treated slightly in this dissertation.

## **Narrow banding implementation of the semi-Lagrangian level set method**

As in the original level set method, only the zero level set is the interest and thus computing the corresponding level set function on the whole image is unnecessary and increases the computation load. It remains unanswered in this dissertation as to how to construct and implement the narrow banding technique in the semi-Lagrangian implementation that allows us to reduce the computation load and storage.

## **Theoretical and rigorous justification for the level set based object matching**

It should be noted that in this dissertation, all derivations are done in a formal way without rigorously looking at the problem of well posedness of the proposed methods and existence and uniqueness of the solution. Due to the highly non-linear nature of the approaches introduced in this dissertation, it remains an open question as to how to build rigorous and sound foundations for them.

## **Efficient and practical numerical techniques for 3D object comparison**

Although formulations and strategies are being discussed in this dissertation for matching 3D objects. Numerically, we have not implemented them yet due to the tremendous demand on storage and computation time.

## **Fast and efficient auto-initialization and representation of open curves and open surfaces**

Last but not least, as pointed out in [21], one level set function could not represent any object with co-dimension higher than one. Thus introducing more level set functions and using the intersection to represent open curves in 2D and 3D and open surfaces in 3D are used in this dissertation. However, this increases storage requirement for the level set functions and a proper auto-initialization technique is then needed for practical purpose that allows automatic initialization of the level set functions of which the intersection is the open curve or surface we are interested in.

Although in this dissertation, strategies are discussed as to how to initialize the level set functions by enforcing the end points matching, this remains an open question as the proposed techniques are too slow due to the fact that semi-Lagrangian implementation is used along with diffeomorphisms generated through infinite dimensional group actions. A possible answer to this is to implement the proposed strategy of enforcing the matching of endpoints back in the original level set method in the Eulerian reference. The big challenge remains in that how we prevent the level set functions from breaking into different pieces and undergo topological changes.



# Bibliography

- [1] S. Osher and J. A. Sethian, "Fronts Propagating with Curvature-Dependent Speed - Algorithms Based on Hamilton-Jacobi Formulations," *Journal of Computational Physics*, vol. 79, pp. 12-49, 1988.
- [2] J. Sethian, *Level set methods and fast marching methods*: Cambridge University Press, 1999.
- [3] S. Osher and R. Fedkiw, *Level set methods and dynamic implicit surfaces*: Springer, 2002.
- [4] E. Rouy and A. Tourin, "A Viscosity Solutions Approach to Shape-from-Shading," *Siam Journal on Numerical Analysis*, vol. 29, pp. 867-884, 1992.
- [5] J. A. Sethian, "A fast marching level set method for monotonically advancing fronts," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 93, pp. 1591-1595, 1996.
- [6] J. A. Sethian, "Fast marching methods," *Siam Review*, vol. 41, pp. 199-235, 1999.
- [7] J. A. Sethian and A. M. Popovici, "3-D traveltime computation using the fast marching method," *Geophysics*, vol. 64, pp. 516-523, 1999.
- [8] R. Kimmel and J. A. Sethian, "Computing geodesic paths on manifolds," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 95, pp. 8431-8435, 1998.
- [9] Y. H. Tsai, L. T. Cheng, S. Osher, and H. K. Zhao, "Fast Sweeping Algorithms for a Class of Hamilton-Jacobi Equations," *UCLA CAM Report*, vol. 01-27, 2001.
- [10] M. Gage and R. S. Hamilton, "The Heat-Equation Shrinking Convex Plane-Curves," *Journal of Differential Geometry*, vol. 23, pp. 69-96, 1986.
- [11] M. E. Gage, "Curve Shortening Makes Convex Curves Circular," *Inventiones Mathematicae*, vol. 76, pp. 357-364, 1984.
- [12] M. A. Grayson, "The Heat-Equation Shrinks Embedded Plane-Curves to Round Points," *Journal of Differential Geometry*, vol. 26, pp. 285-314, 1987.
- [13] H. K. Zhao, T. Chan, B. Merriman, and S. Osher, "Variational level set approach to multiphase motion," *Journal of Computational Physics*, vol. 127, pp. 179-195, 1996.

- [14] Y. C. Chang, T. Y. Hou, B. Merriman, and S. Osher, "A level set formulation of eulerian interface capturing methods for incompressible fluid flows," *Journal of Computational Physics*, vol. 124, pp. 449-464, 1996.
- [15] M. Sussman, P. Smereka, and S. Osher, "A Level Set Approach for Computing Solutions to Incompressible 2-Phase Flow," *Journal of Computational Physics*, vol. 114, pp. 146-159, 1994.
- [16] D. Adalsteinsson and J. A. Sethian, "A Fast Level Set Method for Propagating Interfaces," *Journal of Computational Physics*, vol. 118, pp. 269-277, 1995.
- [17] T. F. Chan and L. A. Vese, "Active contours without edges," *Ieee Transactions on Image Processing*, vol. 10, pp. 266-277, 2001.
- [18] T. E. Chan, B. Y. Sandberg, and L. A. Vese, "Active contours without edges for vector-valued images," *Journal of Visual Communication and Image Representation*, vol. 11, pp. 130-141, 2000.
- [19] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic active contours," 1995.
- [20] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic active contours," *International Journal of Computer Vision*, vol. 22, pp. 61-79, 1997.
- [21] G. Aubert and P. Kornprobst, *Mathematical problems in image processing : partial differential equations and the calculus of variations*. New York: Springer, 2001.
- [22] P. Perona and J. Malik, "Scale-Space and Edge-Detection Using Anisotropic Diffusion," *Ieee Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, pp. 629-639, 1990.
- [23] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear Total Variation Based Noise Removal Algorithms," *Physica D*, vol. 60, pp. 259-268, 1992.
- [24] J. Weickert, "Foundations and applications of nonlinear anisotropic diffusion filtering," *Zeitschrift Fur Angewandte Mathematik Und Mechanik*, vol. 76, pp. 283-286, 1996.
- [25] J. Weickert, "A review of nonlinear diffusion filtering," *Scale-Space Theory in Computer Vision*, vol. 1252, pp. 3-28, 1997.
- [26] J. Weickert, "Recursive separable schemes for nonlinear diffusion filters," *Scale-Space Theory in Computer Vision*, vol. 1252, pp. 260-271, 1997.

- [27] J. Weickert, "Coherence-enhancing diffusion filtering," *International Journal of Computer Vision*, vol. 31, pp. 111-127, 1999.
- [28] J. Weickert, "Coherence-enhancing diffusion of colour images," *Image and Vision Computing*, vol. 17, pp. 201-212, 1999.
- [29] J. Weickert, B. M. T. Romeny, and M. A. Viergever, "Efficient and reliable schemes for nonlinear diffusion filtering," *Ieee Transactions on Image Processing*, vol. 7, pp. 398-410, 1998.
- [30] F. Maes, A. Collignon, D. Vandermeulen, G. Marchal, and P. Suetens, "Multimodality image registration by maximization of mutual information," *Ieee Transactions on Medical Imaging*, vol. 16, pp. 187-198, 1997.
- [31] P. Viola and W. M. Wells, "Alignment by maximization of mutual information," *International Journal of Computer Vision*, vol. 24, pp. 137-154, 1997.
- [32] U. Grenander and M. I. Miller, "Computational anatomy: An emerging discipline," *Quarterly of Applied Mathematics*, vol. 56, pp. 617-694, 1998.
- [33] P. Thompson and A. W. Toga, "A framework for computational anatomy," *Computing and Visualization in Science*, vol. 5, pp. 13-34, 2002.
- [34] R. Dann, J. Hoford, S. Kovacic, M. Reivich, and R. Bajcsy, "Evaluation of Elastic Matching System for Anatomic (Ct, Mr) and Functional (Pet) Cerebral Images," *Journal of Computer Assisted Tomography*, vol. 13, pp. 603-611, 1989.
- [35] R. Bajcsy and S. Kovacic, "Multiresolution Elastic Matching," *Computer Vision Graphics and Image Processing*, vol. 46, pp. 1-21, 1989.
- [36] B. K. P. Horn and B. G. Schunck, "Determining Optical-Flow - a Retrospective," *Artificial Intelligence*, vol. 59, pp. 81-87, 1993.
- [37] B. K. P. Horn and B. G. Schunck, "Determining Optical-Flow," *Proceedings of the Society of Photo-Optical Instrumentation Engineers*, vol. 281, pp. 319-331, 1981.
- [38] B. Fischer and J. Modersitzki, "Fast Diffusion Registration," *Institute of Mathematics, Medical University of Lubeck*, vol. Preprint A-01-18, 2001.
- [39] H. H. Nagel, "On the Estimation of Optical-Flow - Relations between Different Approaches and Some New Results," *Artificial Intelligence*, vol. 33, pp. 299-324, 1987.

- [40] H. H. Nagel and W. Enkelmann, "An Investigation of Smoothness Constraints for the Estimation of Displacement Vector-Fields from Image Sequences," *Ieee Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, pp. 565-593, 1986.
- [41] J. Weickert and C. Schnorr, "A theoretical framework for convex regularizers in PDE-based computation of image motion," *International Journal of Computer Vision*, vol. 45, pp. 245-264, 2001.
- [42] M. I. Miller, A. Troune, and L. Younes, "On the metrics and Euler-Lagrange equations of computational anatomy," *Annual Review of Biomedical Engineering*, vol. 4, pp. 375-405, 2002.
- [43] L. Younes, "Deformations, warping and object comparison -A tutorial," CMLA, ENS Cachan, France., technical report 2000.
- [44] L. Younes, "Optimal matching between shapes via elastic deformations," *Image and Vision Computing*, vol. 17, pp. 381-389, 1999.
- [45] L. Younes, "Computable elastic distances between shapes," *Siam Journal on Applied Mathematics*, vol. 58, pp. 565-586, 1998.
- [46] L. Younes, "A distance for elastic matching in object recognition," *Comptes Rendus De L Academie Des Sciences Serie I-Mathematique*, vol. 322, pp. 197-202, 1996.
- [47] A. Troune, "Infinite-Dimensional Group Action and Pattern-Recognition," *Comptes Rendus De L Academie Des Sciences Serie I-Mathematique*, vol. 321, pp. 1031-1034, 1995.
- [48] A. Troune, "Diffeomorphisms groups and pattern matching in image analysis," *International Journal of Computer Vision*, vol. 28, pp. 213-221, 1998.
- [49] A. Troune and L. Younes, "Diffeomorphic matching problems in one dimension: Designing and minimizing matching functionals," *Computer Vision - Eccv 2000, Pt I, Proceedings*, vol. 1842, pp. 573-587, 2000.
- [50] P. M. Thompson, R. P. Woods, M. S. Mega, and A. W. Toga, "Mathematical/computational challenges in creating deformable and probabilistic atlases of the human brain," *Human Brain Mapping*, vol. 9, pp. 81-92, 2000.
- [51] M. I. Miller and L. Younes, "Group actions, homeomorphisms, and matching: A general framework," *International Journal of Computer Vision*, vol. 41, pp. 61-84, 2001.

- [52] S. C. Joshi and M. I. Miller, "Landmark matching via large deformation diffeomorphisms," *Ieee Transactions on Image Processing*, vol. 9, pp. 1357-1370, 2000.
- [53] G. E. Christensen, S. C. Joshi, and M. I. Miller, "Volumetric transformation of brain anatomy," *Ieee Transactions on Medical Imaging*, vol. 16, pp. 864-877, 1997.
- [54] G. E. Christensen, M. I. Miller, M. W. Vannier, and U. Grenander, "Individualizing neuroanatomical atlases using a massively parallel computer," *Computer*, vol. 29, pp. 32-&, 1996.
- [55] G. E. Christensen, R. D. Rabbitt, and M. I. Miller, "3d Brain Mapping Using a Deformable Neuroanatomy," *Physics in Medicine and Biology*, vol. 39, pp. 609-618, 1994.
- [56] G. E. Christensen, R. D. Rabbitt, and M. I. Miller, "Deformable templates using large deformation kinematics," *Ieee Transactions on Image Processing*, vol. 5, pp. 1435-1447, 1996.
- [57] P. Dupuis, U. Grenander, and M. I. Miller, "Variational problems on flows of diffeomorphisms for image matching," *Quarterly of Applied Mathematics*, vol. 56, pp. 587-600, 1998.
- [58] D. Mumford, "Pattern Theory: the Mathematics of Perception," in *ICM 2002*, vol. 3, 2002.
- [59] V. I. Arnold, *Mathematical methods of classical mechanics*: Springer, 1989.
- [60] V. I. Arnold, *Ordinay differential equations*: The MIT press, 1973.
- [61] G. E. Christensen, R. D. Rabbitt, and M. I. Miller, "Deformable templates using large deformation kinematics," *IEEE Transactions on Image Processing*, vol. 5, pp. 1435-47, 1996.
- [62] G. E. Christensen, R. D. Rabbitt, and M. I. Miller, "3D brain mapping using a deformable neuroanatomy," 1994.
- [63] G. E. Christensen, S. C. Joshi, and M. I. Miller, "Volumetric transformation of brain anatomy," *IEEE Transactions on Medical Imaging*, vol. 16, pp. 864-77, 1997.
- [64] J. Strain, "Semi-Lagrangian methods for level set equations," *Journal of Computational Physics*, vol. 151, pp. 498-533, 1999.

- [65] J. Strain, "A fast modular semi-Lagrangian method for moving interfaces," *Journal of Computational Physics*, vol. 161, pp. 512-36, 2000.
- [66] J. Strain, "A fast semi-Lagrangian contouring method for moving interfaces," *Journal of Computational Physics*, vol. 170, pp. 373-94, 2001.
- [67] R. T. Whitaker, "A level-set approach to image blending," *IEEE Transactions on Image Processing*, vol. 9, pp. 1849-61, 2000.
- [68] N. Paragios, M. Rousson, and V. Ramesh, "Matching Distance Functions: A Shape-to-Area Variational Approach for Global-to-Local Registration," *ECCV*, 2002.
- [69] P. Smereka, "Spiral crystal growth," *Physica D*, vol. 138, pp. 282-301, 2000.
- [70] W. H. Liao, A. Khuu, M. Bergsneider, L. A. Vese, S. C. Huang, and S. Osher, "From Landmark Matching to Shape and Open Curve Matching: A Level Set Approach," in *UCLA CAM report*, 2002.
- [71] L. Younes, "Synchronous random fields and image restoration," *Ieee Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, pp. 380-390, 1998.
- [72] H. Federer, *Geometric measure theory*: Springer, 1969.
- [73] P. Mattila, *Geometry of sets and measures in Euclidean spaces*: Cambridge university press, 1995.
- [74] 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*, vol. 175, pp. 604-644, 2002.