

**ON VISUALIZING BRANCHED SURFACE:
AN ANGLE/AREA PRESERVING APPROACH**

A Thesis
Presented to
The Academic Faculty

by

Lei Zhu

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
Department of Biomedical Engineering

Georgia Institute of Technology
September 2004

ON VISUALIZING BRANCHED SURFACE: AN ANGLE/AREA PRESERVING APPROACH

Approved by:

Dr. Allen R. Tannenbaum, Advisor

Dr. James Gruden

Dr. Oskar Skrinjar

Dr. May D. Wang

Dr. Anthony J. Yezzi

Date Approved: September 13, 2004

To my grandfather,

Shanquan Zhu,

for his love, support and selfless sacrifices.

ACKNOWLEDGEMENTS

I would like to thank my advisor Dr. Allen Tannenbaum, not only for his wonderful understanding and guidance of technical topics, but also for his kindness and generosity. I would like to thank Dr. Steven Haker for his help and suggestions. I would like to thank my thesis committee members, Dr. James Gruden, Dr. Oskar Skrinjar, Dr. May Wang and Dr. Anthony Yezzi, for their helpful comments. I would also like to thank my colleagues, Delphine Nain, Marc Niethammer, Eric Pichon, Gallagher Pryor, Yogesh Rathi, Yan Yang and Amanda Wake, for their help throughout my study. Finally, special thanks to my family, including my girl friend, for their support and encouragement from overseas.

TABLE OF CONTENTS

DEDICATION	iii
ACKNOWLEDGEMENTS	iv
LIST OF TABLES	vii
LIST OF FIGURES	viii
SUMMARY	xi
Chapter 1 INTRODUCTION	1
1.1 Texture Mapping and Surface Flattening	2
1.2 Optical Flow	3
1.3 Image Registration	4
1.4 Image Morphing (Image Interpolation in Time Domain)	7
1.5 Organization of the Thesis	9
Chapter 2 CONFORMAL MAPPING OF BRANCHED SURFACES	10
2.1 Introduction to Conformal Mapping	10
2.1.1 Harmonic Functions	11
2.1.2 Analytic Function	11
2.1.3 Conformal Mapping	12
2.2 Harmonic Skeleton	13
2.3 Conformal Flattening of a Y-shaped surface	16
2.4 Conformal Flattening of a Multi-branched Surface	18
2.5 Numerical Implementation	19
2.6 Medical Examples	22
Chapter 3 AREA-PRESERVING MAPPINGS: A CORRECTION OF CONFORMAL MAPPINGS	33
3.1 Optimal Mass Transportation	34
3.1.1 Introduction	34
3.1.2 Mathematical Formulation	35

3.2	Area-preserving Mappings Based on the L^2 Monge-Kantorovich Problem	45
3.3	Medical Examples	47
Chapter 4	IMAGE MORPHING BASED ON OPTIMAL MASS TRANSPORTATION	58
4.1	Image Morphing Based on the L^2 Monge-Kantorovich Problem	58
4.1.1	The Sum of Squared Difference as Comparison	59
4.1.2	Mutual Information as Comparison	62
4.1.3	Image Morphing on a Doubly-connected Domain	67
4.2	Numerical Implementation	71
4.2.1	Rectangular Regions	72
4.2.2	Doubly-connected Regions	72
4.3	Examples	74
4.3.1	Circle to triangle example	74
4.3.2	Cloud example	76
4.3.3	Flame example	80
4.3.4	The first brain example	89
4.3.5	The second brain example	94
4.3.6	Heart example	97
Chapter 5	CONCLUSIONS AND FUTURE RESEARCH	103
	REFERENCES	107
	VITA	115

LIST OF TABLES

Table 1	Notation used in Chapter 3.	35
---------	-------------------------------------	----

LIST OF FIGURES

Figure 1	Mapping a Y-shaped vessel onto a plane	17
Figure 2	The branch point and its neighbors	20
Figure 3	Triangle geometry	22
Figure 4	The original gray scaled image (for <i>coronary</i> artery example).	24
Figure 5	Classification result with MAP labels (for <i>coronary</i> artery example).	24
Figure 6	Segmented left main coronary artery with marked boundaries.	24
Figure 7	A segmented coronary artery	25
Figure 8	The harmonic skeleton for Figure 7	25
Figure 9	A coronary artery and its harmonic skeleton	26
Figure 10	Cross-sectional area measurement	26
Figure 11	A coronary artery (the same one as in Figure 7)	27
Figure 12	Conformal mapping of Figure 11	27
Figure 13	A frontal cerebral artery (the <i>cerebral</i> example).	28
Figure 14	The harmonic skeleton of figure 13	29
Figure 15	Conformal mapping of figure 13	29
Figure 16	A <i>carotid</i> artery	30
Figure 17	Conformally flattened <i>carotid</i> artery	31
Figure 18	Harmonic skeleton with camera direction for Figure 16	31
Figure 19	Some views in fly-throughs of Figure 16	32
Figure 20	Area-preserving mapping for the front cerebral artery (<i>cerebral</i> example).	48
Figure 21	The histogram for triangle area changing ratios in conformal mapping (<i>cerebral</i> example).	49
Figure 22	The histogram for triangle area changing ratios in area-preserving mapping (<i>cerebral</i> example).	49
Figure 23	The pseudo mass density μ_0 for the <i>coronary</i> example.	50
Figure 24	The area-preserving flattening for the <i>coronary</i> example.	51

Figure 25	The histogram for triangle area changing ratios in conformal mapping (<i>coronary</i> example).	52
Figure 26	The histogram for triangle area changing ratios in area-preserving mapping (<i>coronary</i> example).	52
Figure 27	The pseudo mass density μ_0 for the <i>carotid</i> example.	53
Figure 28	The area-preserving flattening for the <i>carotid</i> example.	54
Figure 29	The histogram for triangle area changing ratios in conformal mapping (<i>carotid</i> example).	54
Figure 30	The histogram for triangle area changing ratios in area-preserving mapping (<i>carotid</i> example).	55
Figure 31	Original carotid artery shaded by wall shear stress.	56
Figure 32	Flattened carotid artery shaded by wall shear stress.	56
Figure 33	Statistics of area changing in conformal mapping for the second <i>carotid</i> example.	57
Figure 34	Statistics of area changing in area-preserving mapping for the second <i>carotid</i> example.	57
Figure 35	A doubly-connected domain Σ with two boundaries.	69
Figure 36	Harmonic parametrization of a heart image	70
Figure 37	The diagram of constructing u^0 between two doubly-connected domains.	71
Figure 38	Image morphing from a circle to a triangle	75
Figure 39	The starting <i>cloud</i> image	76
Figure 40	In-between <i>cloud</i> image at $t=0.2$	77
Figure 41	In-between <i>cloud</i> image at $t=0.4$	77
Figure 42	In-between <i>cloud</i> image at $t=0.6$	78
Figure 43	In-between <i>cloud</i> image at $t=0.8$	78
Figure 44	The ending <i>cloud</i> image.	79
Figure 45	The deformed grid in the <i>cloud</i> example.	79
Figure 46	Two given <i>flame</i> images	80
Figure 47	Interpolation of <i>flame</i> images without comparison term.	82
Figure 48	Interpolation of <i>flame</i> images using SSD as the comparison term.	83

Figure 49	Interpolation of <i>flame</i> images using MI as the comparison term.	84
Figure 50	Comparison of pure L^2 MKP, SSD, and MI at $t = 0.5$	85
Figure 51	The deformed grid in the <i>flame</i> example without comparison term.	86
Figure 52	The deformed grid in the <i>flame</i> example with SSD similarity measure.	86
Figure 53	The deformed grid in the <i>flame</i> example with MI similarity measure.	87
Figure 54	Decision of α	89
Figure 55	The starting T1 brain image ($t = 0$).	90
Figure 56	Interpolated T1 brain image ($t = 0.2$).	90
Figure 57	Interpolated T1 brain image ($t = 0.4$).	91
Figure 58	Interpolated T1 brain image ($t = 0.6$).	91
Figure 59	Interpolated T1 brain image ($t = 0.8$).	92
Figure 60	The ending T1 brain image ($t = 1$).	92
Figure 61	The deformed grid in the <i>brain</i> example.	93
Figure 62	<i>Second brain</i> example ($t = 0$).	94
Figure 63	<i>Second brain</i> example ($t = 0.2$).	95
Figure 64	<i>Second brain</i> example ($t = 0.4$).	95
Figure 65	<i>Second brain</i> example ($t = 0.6$).	96
Figure 66	<i>Second brain</i> example ($t = 0.8$).	96
Figure 67	<i>Second brain</i> example ($t = 1.0$).	97
Figure 68	The mask for starting <i>heart</i> image	98
Figure 69	The mask for ending <i>heart</i> image	98
Figure 70	The deformed grid on the systolic heart image.	99
Figure 71	The starting heart image ($t = 0$)	99
Figure 72	In-between <i>heart</i> image ($t = 0.2$)	100
Figure 73	In-between <i>heart</i> image ($t = 0.4$)	100
Figure 74	In-between <i>heart</i> image ($t = 0.6$)	101
Figure 75	In-between <i>heart</i> image ($t = 0.8$)	101
Figure 76	The ending heart image ($t = 1.0$)	102
Figure 77	Two multi-connected domains	105

SUMMARY

The techniques of surface deformation and mapping are useful tools for the visualization of medical surfaces, especially for highly undulated or branched surfaces. In this thesis, two algorithms are presented for flattened visualizations of multi-branched medical surfaces, such as vessels. The first algorithm is an angle preserving approach, which is based on conformal analysis. The mapping function is obtained by minimizing two Dirichlet functionals. On a triangulated representation of vessel surfaces, this algorithm can be implemented efficiently using a finite element method. The second algorithm adjusts the result from conformal mapping to produce a flattened representation of the original surface while preserving areas. It employs the theory of optimal mass transport via a gradient descent approach.

A new class of image morphing algorithms is also considered based on the theory of optimal mass transport. The mass moving energy functional is revised by adding an intensity penalizing term, in order to reduce the undesired “fading” effects. It is a parameter free approach. This technique has been applied on several natural and medical images to generate in-between image sequences.

CHAPTER 1

INTRODUCTION

In this thesis, we discuss two surface mapping approaches for visualizing multi-branched surfaces. The first algorithm is based on conformal analysis and the second one is based on the theory of optimal mass transportation. Both of them involve using partial differential equations (PDEs). We will also describe several other related interesting applications, including image morphing, flying-through for tubular structures, and harmonic parametrization.

During the past a few years, many three dimensional visualization techniques have been applied in surgical planning, noninvasive diagnosis and treatment, and image-guided surgery. Among them, the techniques of surface deformation and mapping can be useful tools for the visualization of surfaces, especially for those surfaces which are highly undulated or branched. For example, the brain cortical surface can be extracted from magnetic resonance images (MRI) [50] and flattened for better visualization of neural activity deep within the folds or sulci of the brain [3]. Virtual colonoscopy provides radiologists a minimally invasive screening method for the detection of small polyps. However, it has a fundamental problem that some areas are not inspected at all, leading to incomplete examinations. An alternative approach is to simulate the action by pathologists to cut open the tube represented by a colon and lay it out for a comprehensive inspection [36]. A flattened representation of the colon surface can provide an unobstructed view of the entire surface. Recently, there has been a demand for evaluation of atherosclerosis diagnosis method using CT images, by comparing with traditional catheter method. We believe that flattened presentations of cardiac vessels can be helpful in these evaluations. There have been many

approaches for flattening representations of medical surfaces. Paik et al. [69] proposed a visualization technique, which uses cylindrical and planar map projections. Among others, methods based on quasi-isometric and quasi-conformal flattening of brain surfaces have been considered [17, 79]. Typically, these methods can distort the shape or do not guarantee bijectivity (one-to-one) of the mappings. Wang et al. [89, 88] present an algorithm for unravelling colon surfaces based on an artificial electromagnetic field. This algorithm requires a central line as an input and sometimes needs to make compromises between large distortions and bijectivity.

In this introductory chapter, we briefly review some of the related topics in computer vision and image processing.

1.1 Texture Mapping and Surface Flattening

Texture mapping is a computer visualization technique that maps a 2D image, either synthesized or digitized, onto a given 3D object to make the object look more natural or to achieve some special effects. This process is similar to applying wall-paper on a wall. Texture mapping can be used to define many surface parameters besides color. These include the perturbation of surface normal vectors to simulate bumpy surface (bump mapping), the opacity of a translucent surface (transparency mapping), glossiness of a surface (specularity mapping), and the distribution of incoming light in all directions (illumination mapping) [44, 45, 37].

No matter what kind of image mapping is used, the parametrization function that relates the geometry of an object to the image plane must be determined, i.e. the association of a coordinate on the image plane with a location on the surface of the geometric object. For complex objects, finding a usable parametrization function can be a very difficult task. Lévy and Mallet proposed an algorithm for nondistorted mappings on triangle meshes, which can be made continuous through cuts [61]. Lee et al. [62] have developed a multiresolution approach based on loop subdivision. Haker et

al. [37] proposed an explicit method based on conformal mapping for parametrization on a genus zeros surface. Gu et al. [34] extended this conformal parameterization method to nonzero genus closed surfaces.

Another closely related technique is surface flattening (or surface warping), e.g. flattening the surface of a globe to be the map of the world. Once a reasonable parametrization of the object surface is given, it is very straightforward to generate a flattened presentation of the surface of original object.

1.2 *Optical Flow*

Optical flow is the apparent motion of brightness pattern in a series of images [49]. It can be used to approximate the velocity field since the motion of the brightness pattern will be the same as the motion of the object in most cases. It can also be used for tracking targets and has been applied to medical images as a tool for segmentation.

The Optical Flow Constraint (OFC) is defined as

$$E_x u + E_y v + E_t = 0, \quad (1)$$

where E_x , E_y and E_t are the intensity gradients in the x , y , and the temporal direction, respectively. u and v are the two components of the apparent motion of brightness pattern. Equation (1) is based on a constant intensity assumption. However, there is only one function but two unknowns for each pixel. Additional constraints, such as a smoothness constraint, must be added to the OFC in order to find a unique solution. Hence an energy functional is defined to be the sum of two parts as shown in equation (2):

$$e = \lambda \left[\int \int (E_x u + E_y v + E_t)^2 \right] + [(u_x^2 + u_y^2) + (v_x^2 + v_y^2)]. \quad (2)$$

The first term comes from the OFC and the second term is the smoothness constraint. After applying the calculus of variation, the corresponding Euler equations are

$$\Delta u = \lambda(E_x u + E_y v + E_t)E_x \quad (3)$$

$$\Delta v = \lambda(E_x u + E_y v + E_t)E_y \quad (4)$$

where λ is a positive constant,

An L^1 constraint can also be applied as the smoothing term [60, 59, 23], which has the inherent advantages of not smoothing the flow velocity across the edges and hence preserving edge information. Various of other methods for solving this problem can be found in [7].

OFC assumes that the pixel intensity does not change during the movement, which is very restrictive. A weakened constraint, called the Extend Optical Flow Constraint (EOFC), will give a more accurate computation in some cases [11, 12]. In EOFC, the intensity preserving constraint is replaced by a mass preserving constraint as shown in equation (5):

$$E_x u + E_y v + E u_x + E v_y + E_t = 0, \quad (5)$$

if we regard the intensity image as a mass density map.

Multi-grid methods can be used to speed up the calculation of optical flow with both OFC and EOFC [23, 16].

1.3 Image Registration

Image registration is a process of aligning images so that corresponding features can be easily related [35, 66]. The images may come from different modalities or from the same modality at different times. Sometimes, the term of “registration” is also used to indicate aligning images with a computer model or an atlas. It is common now for a patient to be imaged several times, either by repeated imaging using a single modality (e.g., before the surgery and during the surgery), or by imaging with different modalities (e.g., PET and MRI). In either case, there exists a task for physicians to mentally combine or “fuse” all the information to make diagnoses. This procedure requires visual compensation for changes in subject positions. The process of “fusing” is tedious and time consuming if done manually. Automatic

image registration algorithms align the images and show the correspondence between features seen on different images (which may or may not be from the same modality). Registration of an image to an atlas or a computer model aids in the delineation of anatomical and pathological structures, making it a useful tool for detailed analysis. Automatic registration methods are not yet fully developed, and they are a major goal of current research.

Basically, image registration can be divided into two categories, intensity-driven or model-driven, depending on the types of features that drive the mapping of one image onto the other. In intensity-driven approaches, parameters of the mapping are successively adjusted until the value of similarity measure is maximized (or minimized). The applied similarity measures include normalized cross-correlation [6], squared difference [20], mutual information [57], etc. In model-driven approaches, explicit geometric models are built corresponding to identifiable anatomic elements in the original image (volume in 3D case). These models typically include point landmarks [14], curves and surfaces [85]. And there are also some hybrid models, which combine the advantages of both intensity-driven and model-driven algorithms.

Image registration can also be classified as rigid, affine, projective, and non-rigid. This classification is based on the nature of the transformation, more specifically the degrees of freedom of the transformation. 3D rigid registration has six degrees of freedom, i.e. three translations and three rotations. 3D affine registration has 12 degrees of freedom. Nonrigid registration has an infinite number of degrees of freedom. Point-based rigid registration is also known as the Procrustes problem, which can be solved by Singular Value Decomposition (SVD) [53, 71]. Surface-based rigid registration can be solved by Head and Hat algorithm [70], distance transforms algorithm and iterative closest point algorithm [10]. Splines are commonly used for point-based nonrigid registration. Thin-plate splines, which were initially formulated for the surface interpolation of scattered data, have been widely used for image registration in

recent years [14, 33]. In algorithms using thin-plates splines, radial basis functions have infinite support, which is sometimes undesirable since it becomes difficult to model local deformations. Another type of splines, B-splines, are used to model local deformations [26, 77]. The landmarks (often called control points) for thin-plates splines can be selected arbitrarily, which are often anatomical or geometrical landmarks that can be identified on both images. In contrast, B-splines require a regular mesh of control points with uniform spacing.

The elastic registration technique models the deformation of the source image as a physical process which resembles the stretching of an elastic material. The movement of the rubber-like material is governed by two forces. The first force is the internal force which is caused by the deformation of elastic material (i.e., stress). This internal force counteracts any force that deforms the elastic body from its equilibrium shape. The second force is the external force that acts on the elastic body. The deformation of the elastic body is the result of the equilibrium between the internal force and the external force. The behavior of an elastic material is described by the Navier equation.

$$\mu \nabla^2 u + (\lambda + \mu) \nabla(\nabla \cdot u) + f = 0 \quad (6)$$

where u is the displacement field, f is the external force acting on the elastic body, λ and μ are Lamé's elasticity constants. The external force can be chosen as the gradient of local correlation measure based on intensities [6], intensity differences, or intensity features such as edges and curvatures [32]. An alternative choice is the displacements of control points on the corresponding curves and surfaces between two images (volumes) [25]. Registration based on the elastic model usually cannot model highly localized deformations, since the deformation energy increases proportionally to the strength of the deformation.

Optical flow technique can also be used for solving the registration problem [42].

In an early registration algorithm by Bajcsy et al. [6], normalized local cross-correlation measure is used as the comparison terms. Intensity neighborhoods to be correlated in each volume are first projected onto a truncated three dimensional Hermite polynomial basis [21]. The normalized cross-correlation value is then calculated based on the calculated coefficients. This procedure enhances the response of edge features and accelerates computation. Interestingly, correlation can also be carried in spatial frequency domain (k-space) [55]. In particular, a translation in spatial domain becomes a phase change in k-space and a rotation in spatial domain is a rotation by the same angle in k-space. In an algorithm proposed by Christensen et al. [20], squared differences in pixel intensities are used as measure of similarity. Informational theoretic techniques are also widely used in registration algorithm. Joint entropy [80] measures the amount of information in the combined image. The goal of registration is to minimize the information contained in the combined image. A mutual information metric is also used to measure the statistical dependency between intensity patterns in the original and target images. In Kim et al.'s work [57], mutual information is calculated from a two-dimensional gray scale histogram of an image pair and its negative value provided a matching cost function. Combined with a thin-plate-spline warping method, autoradiographic image slices of a rat brain are registered with video reference images of the uncut block face. In Maes et al.'s work [65], normalized mutual information is also proposed as a measure of similarity.

1.4 Image Morphing (Image Interpolation in Time Domain)

Image interpolation in time domain, which is also called *image morphing*, deals with the metamorphosis of one image to another [63]. This technique has been widely used in television commercials, music videos and movies such as *Willow* [91]. It is also important for facial recognition [96]. When a starting image I_0 and an ending image I_1 are given, the goal of image interpolation problem is to find a sequence of

in-between images $I(t)$ such that $I(0) = I_0$ and $I(1) = I_1$. As t varies from 0 to 1 the starting image continuously changes to the ending image. The key problem in this technique is to find a reasonable warping function from the starting image to the ending image. It is closely related to the task of image registration, although the requirement for a good warping is slightly different. In registration, the quality of the warping is evaluated by the mapping error of corresponding features, while in image morphing the transient images should appear natural. There have been many algorithms proposed to find the warping function.

The first category of warping is mesh warping [92], where features are specified by a nonuniform control mesh, and the warp is usually generated by a spline interpolation. The mesh warping method shows a good distortion behavior but has a critical drawback in specifying features since the features on the control mesh may have an arbitrary structure. Also, using a user interface to define the feature correspondence is very difficult and time consuming. The second category is field morphing [8], where a pair of corresponding lines in the starting and ending image define a coordinate mapping between the two images. The mapping of a point in the vicinity of a line can be determined by its distance from the line. Since there are multiple line pairs, the mapping of a point is given by a weighted sum of the mappings due to all the line pairs. This method gives an easy-to-use and expressive method in specifying features. However, unexpected distortions appear (referred to as “ghosts”) which means a part of the image shows up in some unrelated regions in the interpolated images. The reason for this undesired result is caused by possibly wrong combinations of the specified line segments. The last category of morphing is energy-minimization based warpings. This type of technique usually guarantees the one-to-one property of the warp function, which prevents the warped image from folding back upon itself. For example, in Lee et al.’s work [63], all primitives, including points, polylines and curves, are sampled and reduced to a collection of points. These points are then used

to generate a warp by minimizing an energy functional that is the sum of difference penalty, elastic energy and “glue” energy. Each feature is distinguished by a set of points in the image. Hassanien et al. [41] also proposed a similar method based on a Navier elastic body spline.

1.5 Organization of the Thesis

We now briefly outline the contents for this thesis. In Chapter 2, a new technique is developed for flattening multi-branched vessel surfaces in an angle-preserving way. In this chapter, we also present a new algorithm for extracting central line from a branched surface, based on harmonic analysis. In Chapter 3, we develop a new flattening algorithm that preserves area. This algorithm is based on the theory of optimal mass transportation by making correction on the result from the angle-preserving mapping. In Chapter 4, we describe another new technique for image interpolation between image pairs (also called image morphing), also based on the theory of optimal mass transportation. Finally, in Chapter 5, we conclude this thesis and outline some possible future research directions.

CHAPTER 2

CONFORMAL MAPPING OF BRANCHED SURFACES

This chapter is devoted to a conformal mapping technique that flattens tubular structures with multi-branches. The method is based on inverting a discrete Laplace-Beltrami operator to flatten a branched tubular surface onto a planar polygonal region in an angle-preserving manner. The contents of this chapter is structured as follows. Section 2.1 introduces related theories of analytic functions and conformal mappings. Section 2.2 describes the algorithm of constructing a harmonic skeleton and its applications. In Section 2.3 we present the construction of a conformal mapping of a Y-shaped open-ended surface, and in Section 2.4 we show how to extend this algorithm to multi-branched surfaces. In Section 2.5 we discuss the numerical details when applying this algorithm on a triangulated surface. Section 2.6 gives some examples of this conformal flattening technique, as well as the applications of harmonic skeletons.

2.1 Introduction to Conformal Mapping

Conformal mapping [68] is a very important concept in complex analysis, as well as many areas of physics and engineering. A *conformal mapping* is a transformation $\omega = f(z)$ in the complex domain, which preserves local angles and hence local geometry. An analytic function is conformal at any point with nonzero derivative. And any conformal mapping of a complex variable with continuous partial derivatives is analytic.

2.1.1 Harmonic Functions

In 2D case, a real function $u(x, y)$ is said to be *harmonic* in a domain D if the first order partial derivatives $\frac{\partial u}{\partial x}$, $\frac{\partial u}{\partial y}$, and second order partial derivatives $\frac{\partial^2 u}{\partial x^2}$, $\frac{\partial^2 u}{\partial y^2}$ exist and are continuous, and if

$$\Delta u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0 \quad (7)$$

at all points of D . Equation (7) can be expressed as

$$\frac{\partial^2 u}{\partial r^2} + \frac{1}{r} \frac{\partial u}{\partial r} + \frac{1}{r^2} \frac{\partial^2 u}{\partial \theta^2} = 0 \quad (8)$$

in polar coordinates. If another function $v = v(x, y)$ can be associated with u by means of the following system of partial differential equations:

$$\frac{\partial u}{\partial x} = \frac{\partial v}{\partial y} \quad (9)$$

$$\frac{\partial u}{\partial y} = -\frac{\partial v}{\partial x}, \quad (10)$$

v is called the *harmonic conjugate* of u . This system of equations is known as the *Cauchy-Riemann equations*. It plays a fundamental role in the theory of conformal mapping. We can take the derivatives of equation (9) with respect to y and equation (10) with respect to x . By adding them together, we obtain:

$$\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} = 0, \quad (11)$$

which indicates v is also a harmonic function. It can also be easily proved that if two functions $u(x, y)$ and $v(x, y)$ both have two continuous derivatives and satisfy the *Cauchy-Riemann equations*, u and v are both harmonic.

2.1.2 Analytic Function

A complex function $f(z)$ is differentiable at point $z_0 \in D$ if

$$f'(z_0) = \lim_{h \rightarrow 0} \frac{f(z_0 + h) - f(z_0)}{h} \quad (12)$$

exists. Here h is an arbitrary complex parameter which tends to zero, and so $z_0 + h$ tends to z_0 . By “exists”, we mean that the limit exists, and is the same, regardless of the path along which h is approaching zero.

An analytic function is defined as a complex function which is single-valued in some domain D of the complex plane, and which has a derivative in the sense of equation (12) at each point of D . The analytic function $f(z)$ is called *regular* in the domain D if it is single-valued and has a derivative at each point of D ; $f(z)$ is called *regular* at the point z_0 if it is regular in a small neighborhood of z_0 (i.e. $|z - z_0| < \epsilon$, ϵ being a small positive number). For an analytic function, the following theorem holds for its real part and imaginary part:

At a point at which an analytic function $f(z)=u+iv$ is regular, its real part $u=u(x,y)$ and its imaginary part $v=v(x,y)$ are connected by the Cauchy-Riemann differential equations.

The converse statement also holds:

If $u=u(x,y)$ is a harmonic function and $v=v(x,y)$ is its harmonic conjugate, then $u(x,y)+iv(x,y)$ is an analytic function of the complex variable $z=x+iy$.

The proof of these theorems can be found in [68].

2.1.3 Conformal Mapping

In order to have a geometric representation of an analytic function $\omega = f(z)$, z and ω can be regarded as points in two different planes – the z -plane and the ω -plane, each being a complex plane. Hence, the function $\omega = f(z)$ can be interpreted as a mapping of points in the z -plane onto points in the ω -plane.

A mapping is conformal if it preserves the angle between two differentiable arcs. The conformality is a characteristic property of the mapping affected by regular analytic functions. If the associated analytic function $f(z)$ is regular in the neighborhood of a point $z = z_0$ at which $f'(z_0) \neq 0$, the mapping is conformal in the neighborhood

of $z = z_0$. And no continuously differentiable mapping of one plane onto another can be conformal unless it is the mapping associated with a regular analytic function. There is an important theorem on the conformal correspondence of two domains [68]:

Let C and C^ be two simple closed contours in the z -plane and ω -plane, respectively, and let $\omega = f(z)$ be regular on and within C . If $\omega = f(z)$ maps C onto C^* in such a way that C^* is traversed by ω exactly once in the positive sense. If z describes C in the positive sense, then $\omega = f(z)$ maps the domain bounded by C onto the domain bounded by C^* .*

2.2 Harmonic Skeleton

The skeleton, or medial axis, is defined as the set of centers of maximal balls in a region [43]. The way of getting a skeleton, called *skeletonization*, is a process for reducing foreground regions in a binary image to a one pixel width skeletal remnant. The resulting skeleton should largely preserve the extent and connectivity of the original region while throwing away most of the original foreground pixels. We can imagine that a binary object is made of some slow-burning material and fire is lighted simultaneously along the whole boundary of the object. The front of the fire moves into the interior of the object at a uniform speed. At points where the fire front meets another part of the fire front travelling from a different part, the fire will extinguish and the those points form the skeleton.

The skeleton was first introduced for studying biological shape and later was found having applications in pattern recognition and solid modeling [13]. More recently, researchers have started to use skeleton as a tool for navigation of human organs, such as virtual endoscopy of colons and vessels.

By definition, a skeleton is easy to imagine. But numerically, it is a very difficult task to obtain a skeleton. There have been many algorithms proposed to address this problem. Most of them can be divided into two classes. The first class is based on

boundary peeling, where the basic idea is to iteratively peel off the boundary layer-by-layer, identifying “simple points” (which means the removal of this point does not affect the topology). For example, Siddiqi et al. [81] introduced a new algorithm rooted in Hamiltonian physics and a very efficient algorithm for shock detection based on the violation of energy conservation principle. This class of technique often faces the problem of smoothness and connectivity. Post processing is usually necessary. The second class is so called distance coding methods. A typical algorithm in this category first does a distance transform, then detects all local maxima in terms of distance value and finally reconnects the local maxima to generate a skeleton. The typical problem involved in transform-based methods is still the connectivity. In [97], Zhou et al. proposed a novel 3D voxel-coding-based skeletonization algorithm. Our algorithm is similar to Zhou’s method. But our approach is based on the solution of a Laplace equation instead of the single-seed-coding used in [97]. The skeleton thus generated is named *Harmonic Skeleton*, which is connected, smooth and logically structured. It can automatically provide a viewing vector when used as a central line for fly-throughs. In addition, it can automatically provide boundary conditions needed for the conformal flattening problem.

Now we describe in detail the steps for generating a harmonic skeleton of a multi-branched surface [99]. Let Σ denote a closed embedded surface (no self-intersections), which is topologically a tube with several branches. The first step of calculating central line is to identify several boundaries on the surface according to the topology, with one boundary on each branch. In our approach, we are considering $PL(\Sigma)$, which is a triangulated representation of Σ . A starting point (*seed*) is manually selected close to the tip of the main branch. Then triangles close to this *seed* are removed to make a boundary σ_0 , which is referred to as *root* in this section. The boundary values of points on the *root* are set to be zero. We then perform the following region growing algorithm on the surface on a triangulated mesh, in order to identify other boundaries

and their values:

1. Mark the points on the *root* as “used” points;
2. Mark all the neighbors of “used” points as “used” points, and divide them into several groups according to connectivity;
3. $stepNumber = stepNumber + 1$;
4. Repeat steps 2 & 3 until a group of “used” points cannot find any new neighbors, which means the “tip” of a new branch is found;
5. Do a reverse region growing on the surface from this “tip” for m steps, record a new boundary as the front of the reverse region growing;
6. Set the value of target boundary as $stepNumber - m$;
7. Continue to do region growing on the surface until all points are visited.

In summary, we do a region growing algorithm on the triangulated surface to discover the remaining boundaries and set a boundary’s value to be the number of triangles between this boundary and the *root* boundary.

The second step of calculating central line is to solve a Dirichlet problem $\Delta u = 0$ on $\Sigma \setminus \sigma_i (i = 0 \dots N)$, given the boundary condition determined in the first step. We use a Finite Element Method (FEM) to solve it. Numerical details are explained in Section 2.5.

The third step is to build a tree-like structure of the harmonic skeleton. For a given u , we find all points on the surface with values equal to u (by interpolation). Then these points are partitioned into several circles according to their connectivity on the surface. The centroid of each circle corresponds to a point on the harmonic skeleton. By increasing u from 0 to the maximal value, we can build a structured tree. Each node on the tree keeps its own record of its ancestor and offspring(s). The

locations of the bifurcations become obvious since it is easy to track where one node has two or more offsprings.

The final step is to refine the harmonic skeleton. In the second step, the boundary values are decided by triangle counts between the *root* and other boundaries, which is a very coarse estimation. Now, we update those boundary values by measuring the actual length of the harmonic skeleton obtained in the third step, which is a much more reasonable way of deciding boundary conditions. Then the Dirichlet problem $\Delta u = 0$ is solved again and the third step is repeated to get a refined harmonic skeleton. An example of harmonic skeleton is shown in Figure 8.

In applications such as virtual endoscopy [48], we may also be interested in getting a forward direction for the camera. One straightforward way is to use the position differences of an ordered set of points on the skeleton. However, this often causes troubles near bifurcations, where the direction vector changes rapidly. This problem can be handled by using Singular Value Decomposition (SVD) [71]. In step two, when calculating the centroid, we can also do an SVD for each circle and choose the eigenvector associated with the smallest eigenvalue as the axis of the camera in fly-through applications. The remaining two eigenvectors form a cutting plane, which is useful in the application of vessel cross-area measure. The cross-sectional area of a vessel is defined as the area of the cross section of this cutting plane and the vessel surface.

2.3 Conformal Flattening of a Y-shaped surface

In this section, we present the outline of the conformal mapping algorithm on a Y-shaped tubular structure. The relevant results from the theory of partial differential equations (PDEs) can be found in [74].

Assume $\Sigma \subset R^3$ represents an embedded surface (having no self-intersections), which is a tube with two branches (Figure 1). The tube has three boundaries, which

are circles in topology. We denote these boundaries as σ_0 , σ_1 and σ_2 . We want to construct a conformal mapping [68], $f = u + iv : \Sigma \rightarrow C$, which maps Σ to a planar polygonal-shaped region.

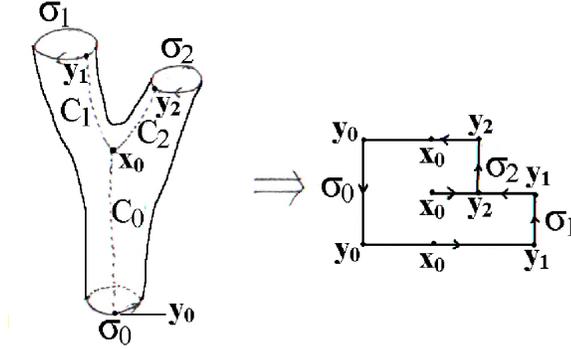


Figure 1: Mapping a Y-shaped vessel onto a plane

There is a well-known classical mathematical method for flattening a multiply-connected surface; see [68] and [34]. The method involves solving a Dirichlet problem for each boundary circle, and then solving a simple linear system of equations to find the correct linear combination u of these functions, together with boundary conditions for a conjugate harmonic function v . However, while mathematically elegant, we have found that the resulting mapping induces extreme areal distortion. We have adjusted this algorithm to provide a mapping which is more practical for the purposes of medical visualization.

The construction begins by finding the real part of $f = u + iv$. This is done by solving a Dirichlet problem $\Delta u = 0$ on $\Sigma(\sigma_0 \cup \sigma_1 \cup \sigma_2)$, with appropriate boundary conditions on σ_0 , σ_1 and σ_2 .

Analytically, there is a saddle point (also called a *branch point* here) x_0 on the surface where $u'(x_0) = 0$. We define three smooth curves C_0 , C_1 and C_2 running from x_0 to σ_0 , σ_1 , and σ_2 , respectively (see Figure 1), such that C_1 and C_2 are along the gradient direction of u and C_0 is opposite to the gradient direction of u (with the

assumption that the boundary value of σ_0 is smaller than σ_1 and σ_2). The curve C_i meets the boundary σ_i at point y_i ($i = 0, 1, 2$). Since $u'(x_0) = 0$, we can make C_1 and C_2 lie on a line in the neighborhood of x_0 , while C_0 is perpendicular to the line.

These curves define a cut on Σ . The cut and the original boundaries σ_0 , σ_1 and σ_2 define an oriented boundary B of the cut surface:

$$B : y_0 \xrightarrow{\sigma_0} y_0 \xrightarrow{-C_0} x_0 \xrightarrow{C_1} y_1 \xrightarrow{\sigma_1} y_1 \xrightarrow{-C_1} x_0 \xrightarrow{C_2} y_2 \xrightarrow{\sigma_2} y_2 \xrightarrow{-C_2} x_0 \xrightarrow{C_0} y_0$$

where $-C_i$ means running in the opposite direction of C_i .

The second step of constructing the mapping function is to calculate the harmonic conjugate of u by solving another Dirichlet problem $\Delta v = 0$. From Cauchy-Riemann equations, we have

$$\frac{\partial v}{\partial s} = \frac{\partial u}{\partial n} \quad (13)$$

on B and so the boundary conditions of v satisfy

$$v(\zeta) = \int_{\zeta_0}^{\zeta} \frac{\partial v}{\partial s} ds = \int_{\zeta_0}^{\zeta} \frac{\partial u}{\partial n} ds \quad (14)$$

where ζ_0 is a given starting point and ζ is any point on the boundary. By deciding the boundary in this way, it is guaranteed that the mapping function transverse the domain boundary only once after mapping.

2.4 Conformal Flattening of a Multi-branched Surface

When the harmonic skeleton is built in Section 2.2, the “tree” structure is constructed from the *root* to the other branches. Each point on the tree has information for its connected neighbors. Curve points, branch points and end points are thus easily classified. Normal curve points are connected to *two* points (one ancestor and one offspring), branch points are connected to *three or more* other points and end points are connected to only *one* other point. This “tree” is then cut into several segments, each containing a Y-shaped structure. By using this partitioned skeleton as a reference, we can easily divide the whole vessel into several parts, each being a Y-shaped

tube (hence one branch point) with three boundaries.

The harmonic function u on the whole surface has then been found when building the harmonic skeleton. Next, cuts are made on each Y-shaped segment and v is solved (see Section 2.3). Some modifications are made to cuts where two adjacent segments meet in order to make the cut continuous through boundaries between two segments. These modified cuts may not follow the gradient of u in some region, but they should not void the monotonic (increasing or decreasing) property of u along the cuts. These modifications guarantee that the mapping result is continuous on the whole surface.

2.5 Numerical Implementation

The Laplace equations in Section 2.2 and Section 2.3 are solved by a FEM numerically. The theory of the finite element method can be found in [51]. In this section, we will discuss the problem for solving a Laplace equation on a Y-shaped tube only. The numerical method of solving the harmonic skeleton in Section 2.2 is very similar to this, just by given more boundary conditions. More details can be found in [100]. In [3] and [38] there are similar methods for brain flattening and colon flattening, respectively. However, due to the differences in topology, the boundary conditions have to be changed.

In the previous section, a cut is found through a saddle point with $u'(x) = 0$. However, when we are working with a triangulated surface, we may not find such a point. And the point with minimum absolute value of derivative ($\min|u'(x)|$) may be far away from the real “branch point”. The algorithm applied here tries to find a point with the difference of its value and its neighbors’ value changing sign for multiple times. For example, in Figure 2, x_0 has 8 neighbors with the difference of value (between x_0 and its neighbors) written beside. If a point is a saddle point, the sign of the difference value will change 4 times around x_0 as shown in figure 2. Otherwise, it changes only twice.

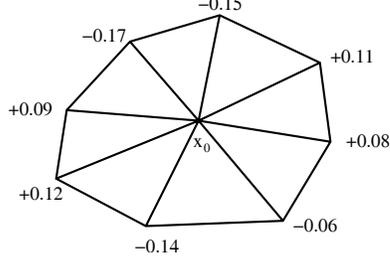


Figure 2: The branch point and its neighbors

Now we will present the finite element method used in our algorithm. In this section, Σ is assumed to be a triangulated surface of a Y-shaped tube with three boundaries σ_0 , σ_1 and σ_2 . Let $PL(\Sigma)$ denote the finite dimensional space of piecewise linear functions on Σ . Then we define a basis $\{\phi_V\}$ for $PL(\Sigma)$. For each vertex, there is a corresponding piecewise linear basis function which is 1 on this vertex and 0 on all other vertices, i.e.

$$\begin{aligned} \phi_V(V) &= 1, \\ \phi_V(W) &= 0, \quad W \neq V, \end{aligned} \quad (15)$$

ϕ_V is linear on each triangle.

Any function $u \in PL(\Sigma)$ can be approximated as the linear combination of these basis functions, in which the coefficients are the values on vertices:

$$u = \sum_V u_V \phi_V. \quad (16)$$

The u we are seeking is continuous on Σ and piecewise linear on each triangle. From the theory of calculus of variations [28, 74], it is known that the solution of the Laplace function $\Delta u = 0$ is a harmonic function u which minimizes the Dirichlet functional

$$\begin{aligned} D(u) &= \frac{1}{2} \int \int_{\Sigma} |\nabla u|^2 dS \\ u|_{\partial u_0} &= \alpha_0, \quad u|_{\partial u_1} = \alpha_1, \quad u|_{\partial u_2} = \alpha_2 \end{aligned} \quad (17)$$

Let $F = |\nabla u|^2$. The Euler equation of the energy functional (17) is

$$F_u - \frac{\partial}{\partial x} F_{u_x} - \frac{\partial}{\partial y} F_{u_y} = 0 \quad (18)$$

Since $F_u = 0$, $\frac{\partial}{\partial x} F_{u_x} = u_{xx}$ and $\frac{\partial}{\partial y} F_{u_y} = u_{yy}$, equation (18) is exactly the Laplace equation

$$\Delta u = 0 \quad (19)$$

According to FEM theory, u should satisfy

$$\int \int \Delta u \cdot v = 0 \quad (20)$$

for any testing function v satisfying $v = 0$ on all boundaries. By doing integration by parts, we have

$$- \int \int \nabla u \cdot \nabla v = 0 \quad (21)$$

Since we have known the value of u on the boundaries, for any given point on the domain, u can be written as

$$u = \sum_{W \in \Sigma \setminus (\sigma_0 \cup \sigma_1 \cup \sigma_2)} u_W \phi_W + \sum_{i=0,1,2} \alpha_i \sum_{W \in \sigma_i} u_W \quad (22)$$

And v can be chosen to be ϕ_V with $V \in \Sigma \setminus (\sigma_0 \cup \sigma_1 \cup \sigma_2)$. Hence, u is the minimizer of the Dirichlet functional, if for each vertex $\Sigma \setminus (\sigma_0 \cup \sigma_1 \cup \sigma_2)$,

$$\sum_{W \in \Sigma \setminus (\sigma_0 \cup \sigma_1 \cup \sigma_2)} D_{VW} u_W = - \sum_{i=0,1,2} \alpha_i \sum_{W \in \sigma_i} D_{VW} \quad (23)$$

D_{VW} is defined as

$$D_{VW} = \int \int \nabla \phi_V \cdot \nabla \phi_W dS \quad (24)$$

for any pair of vertices V and W .

It is easy to see that $D_{VW} = 0$ unless V and W are connected by an edge in the triangulation. As shown in [3], assume VW is an edge belonging to two triangles VWX and VWY . From the theory of finite element methods, we know that for $V \neq W$

$$D_{VW} = -\frac{1}{2}(\cot \angle X + \cot \angle Y) \quad (25)$$

where $\angle X$ is the angle at the vertex X in the triangle VWX and $\angle Y$ is the angle at the vertex Y in the triangle VWY , as shown in Figure 3. Moreover,

$$D_{VV} = - \sum_{W \neq V} D_{VW} \quad (26)$$

for $V = W$.

What we are seeking is a flattening function $f = u + iv$. The computation for v is similar to that of u , using the boundary conditions obtained from (14), which is derived from the Cauchy-Riemann equations.

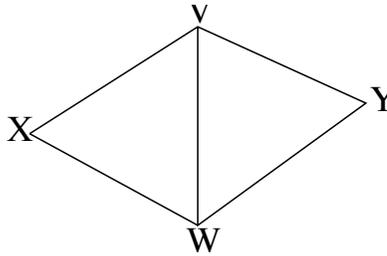


Figure 3: Triangle geometry

2.6 Medical Examples

In this section, we show some medical examples based on the algorithms introduced in this chapter. The first example is a coronary artery example. The dataset is a 3D CT dataset provided by the Department of Radiology of Emory Hospital as shown in Figure 4, which has an isotropic voxel size of $0.6mm \times 0.6mm \times 0.6mm$. The left coronary artery was segmented and the surface was triangulated, using method described in [93], which is similar to the method by Hernandez et al. [47]. In the segmentation algorithm, each voxel is assumed to belong to one of three classes. A priori knowledge of each class is introduced by Bayes' rule. Posterior probabilities obtained from Bayes' rule are anisotropically smoothed. And the segmentation is obtained via the *maximum a posteriori* (MAP) classifications of the smoothed posteriors as shown in Figure 5, in which all the voxels are classified into blood, myocardium and lung, with

blood being the brightest. An active contour model [18, 56] is then applied to the blood class and the coronary arteries are extracted with subvoxel accuracy as shown in Figure 6. The Visualization Toolkit (VTK) [78] is used to generate a triangulated surface. This surface is then smoothed by using a version of the mean curvature flow. The vessel caps are removed to generate an open-ended vessel surface as shown in Figure 7, in which boundary points are marked by blue stars. The vessel surface is painted by the solution to the Laplace equation as described in section 2.2, given appropriate boundary conditions. Figure 8 shows the harmonic skeleton extracted from this coronary artery. Red stars indicate end points and blue stars indicate branchpoints. There are 4 end points and 2 branchpoints in Figure 8. Figure 9 shows the coronary artery and its harmonic skeleton together. And Figure 10 presents the measure of vessel cross-area. Since the images are upsampled by a factor of 2 before segmentation, the LAD has a cross area of $130 \times 0.6 \times 0.6 \times \frac{1}{4} = 11.7mm^2$ at the indicated position (shown by the blue circle) and the LCX has a cross area of $105 \times 0.6 \times 0.6 = 9.5mm^2$ (shown by the red circle).

The harmonic skeleton is then divided into several parts, each containing a Y-shaped structure. By using this partitioned skeleton as a reference, several sections of Y-shaped tubes are obtained. (In this data set, the original vessel is partitioned into two Y-shaped sections.) Then, as described in Section 2.3, the mapping function is solved for each Y-shaped section. The mapping results of all segments are put together to give a global view of the vessel surface. We made some corrections where two Y-shaped sections meet as described in Section 2.4, in order to make the flattened surface look continuous. Figure 11 shows the same coronary artery, but in a non-transparent view. Figure 12 is the result from conformal mapping, whose points have been assigned outward normals according to their original positions in Figure 11.



Figure 4: The original gray scaled image (for *coronary* artery example).



Figure 5: Classification result with MAP labels (for *coronary* artery example).

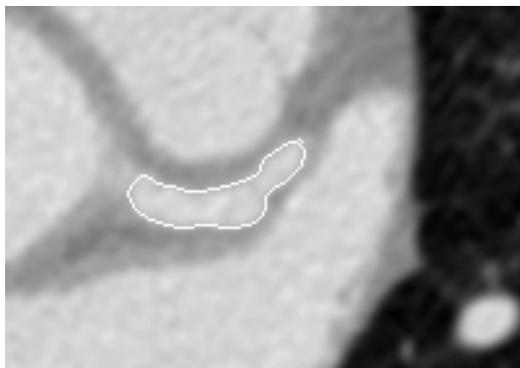


Figure 6: Segmented left main coronary artery with marked boundaries.

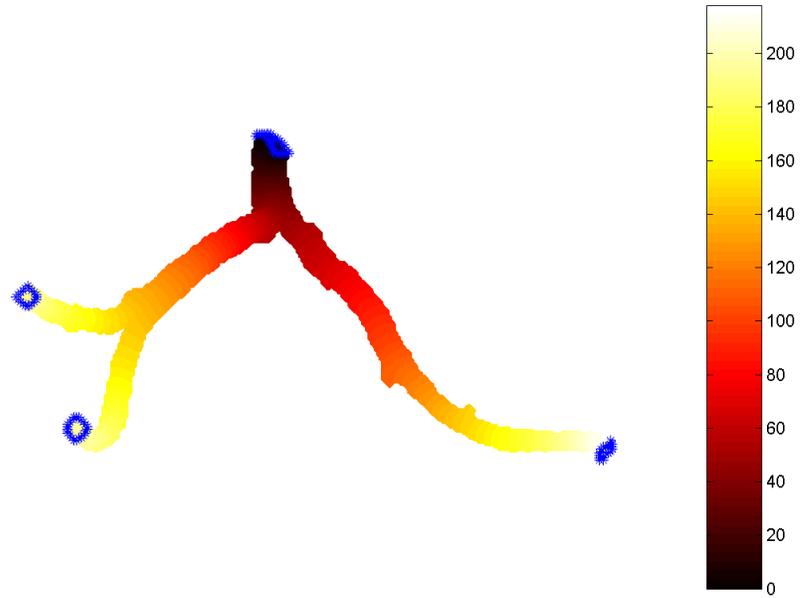


Figure 7: A segmented coronary artery

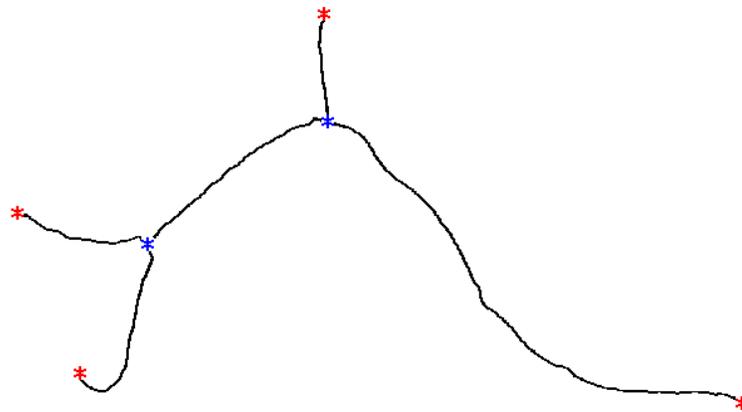


Figure 8: The harmonic skeleton for Figure 7

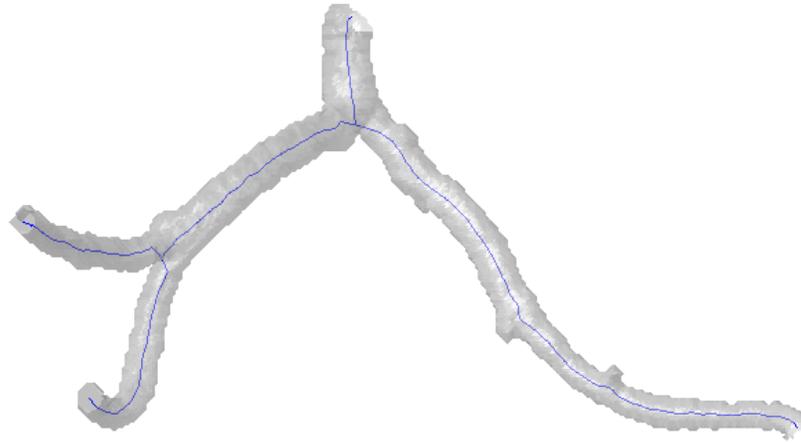


Figure 9: A coronary artery and its harmonic skeleton

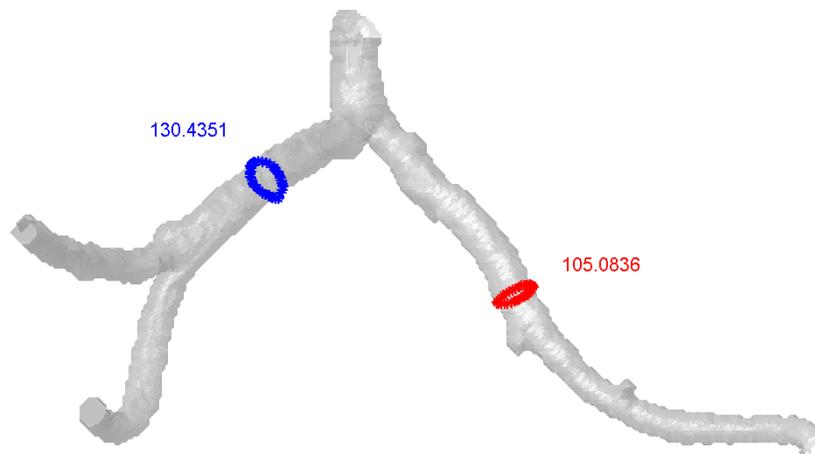


Figure 10: Cross-sectional area measurement

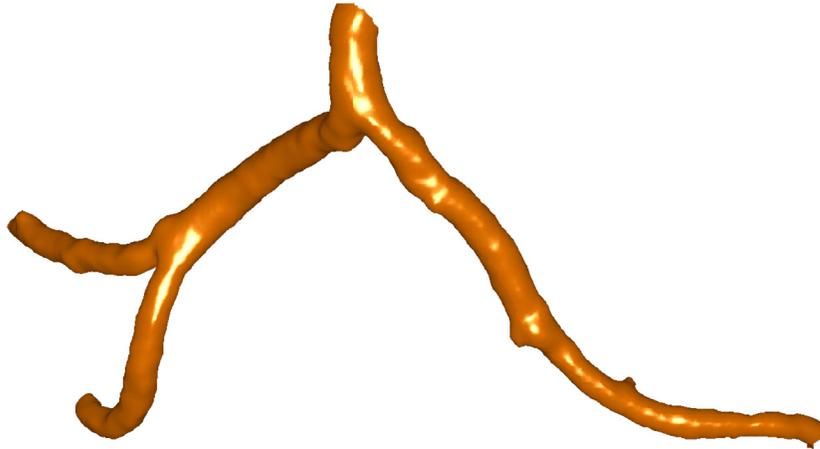


Figure 11: A coronary artery (the same one as in Figure 7)

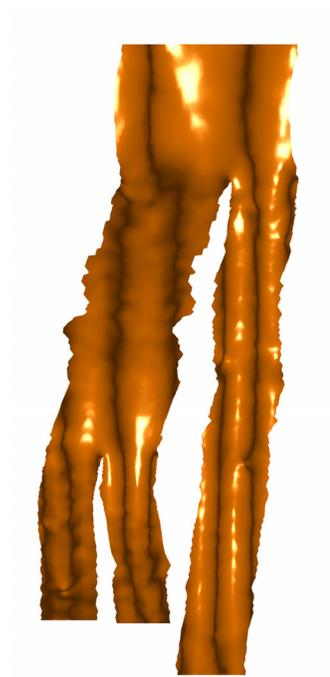


Figure 12: Conformal mapping of Figure 11

The second dataset is a brain MRA imagery provided by the Surgical Planning Lab of Brigham and Women's Hospital. The dataset has a dimension of $256 \times 256 \times 47$. By using a similar approach, we extracted the surface of the frontal cerebral artery, as shown in Figure 13. Following the same procedure as we did on the previous example, a harmonic skeleton was constructed as shown in Figure 14. The whole vessel was then cut into three pieces accordingly and conformally flattened onto a plane as shown in Figure 15.

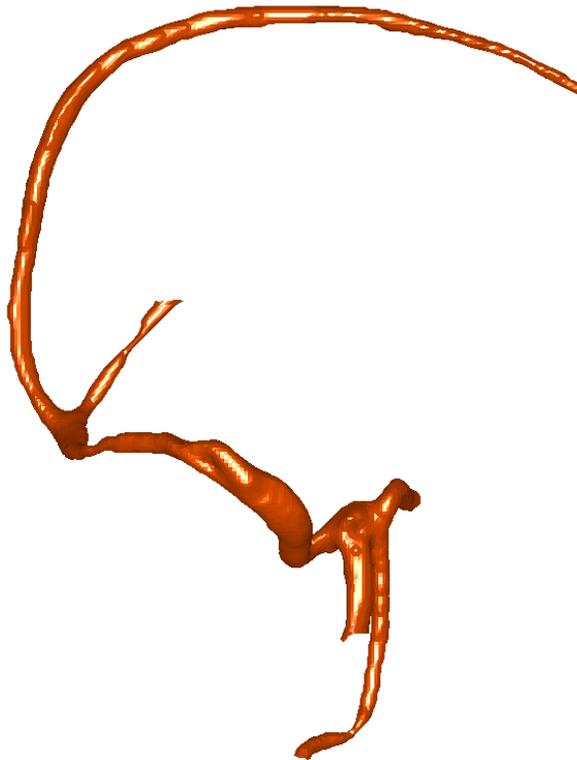


Figure 13: A frontal cerebral artery (the *cerebral* example).

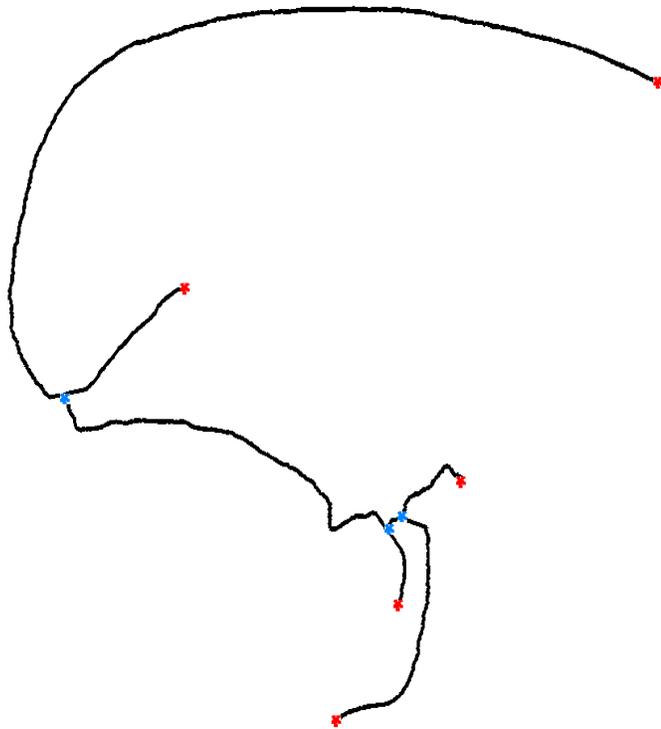


Figure 14: The harmonic skeleton of figure 13



Figure 15: Conformal mapping of figure 13

The third dataset is a CT carotid artery dataset provided by the Department of Radiology of Emory Hospital, which has a dimension of $512 \times 512 \times 91$. In this case, there is only one Y-shaped structure. Figure 16 shows the original 3D surface and Figure 17 shows the result from conformal mapping. Again, the flattened surface is shaded by outward normals from the original 3D surface. Figure 18 shows the calculated harmonic skeleton, which can be used to define the path of the camera in fly-through applications. The little green arrows indicate the directions of the camera's axis, calculated from the SVD algorithm described in Section 2.2. Figure 19 shows some views in a preliminary example for fly-through.

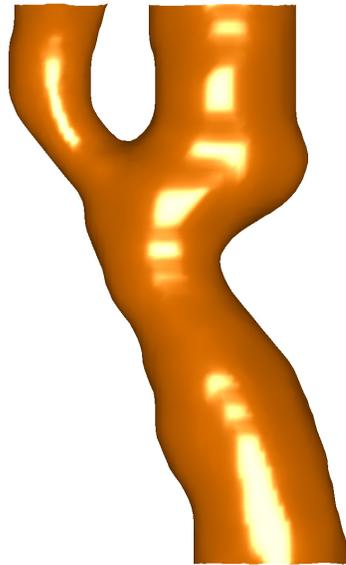


Figure 16: A *carotid* artery

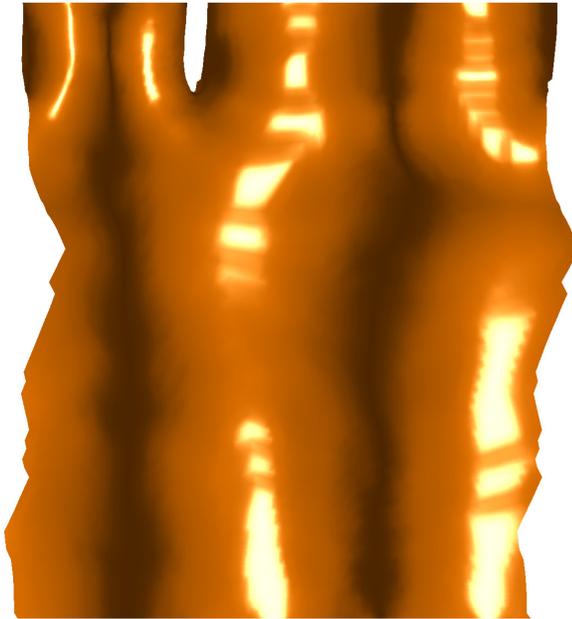


Figure 17: Conformally flattened *carotid* artery

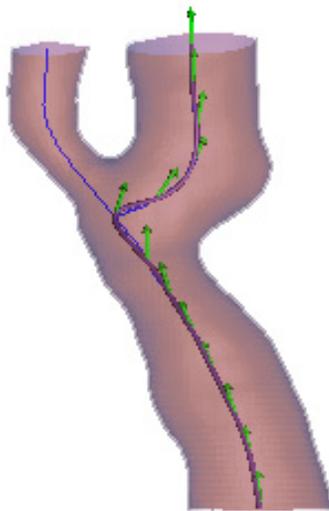


Figure 18: Harmonic skeleton with camera direction for Figure 16

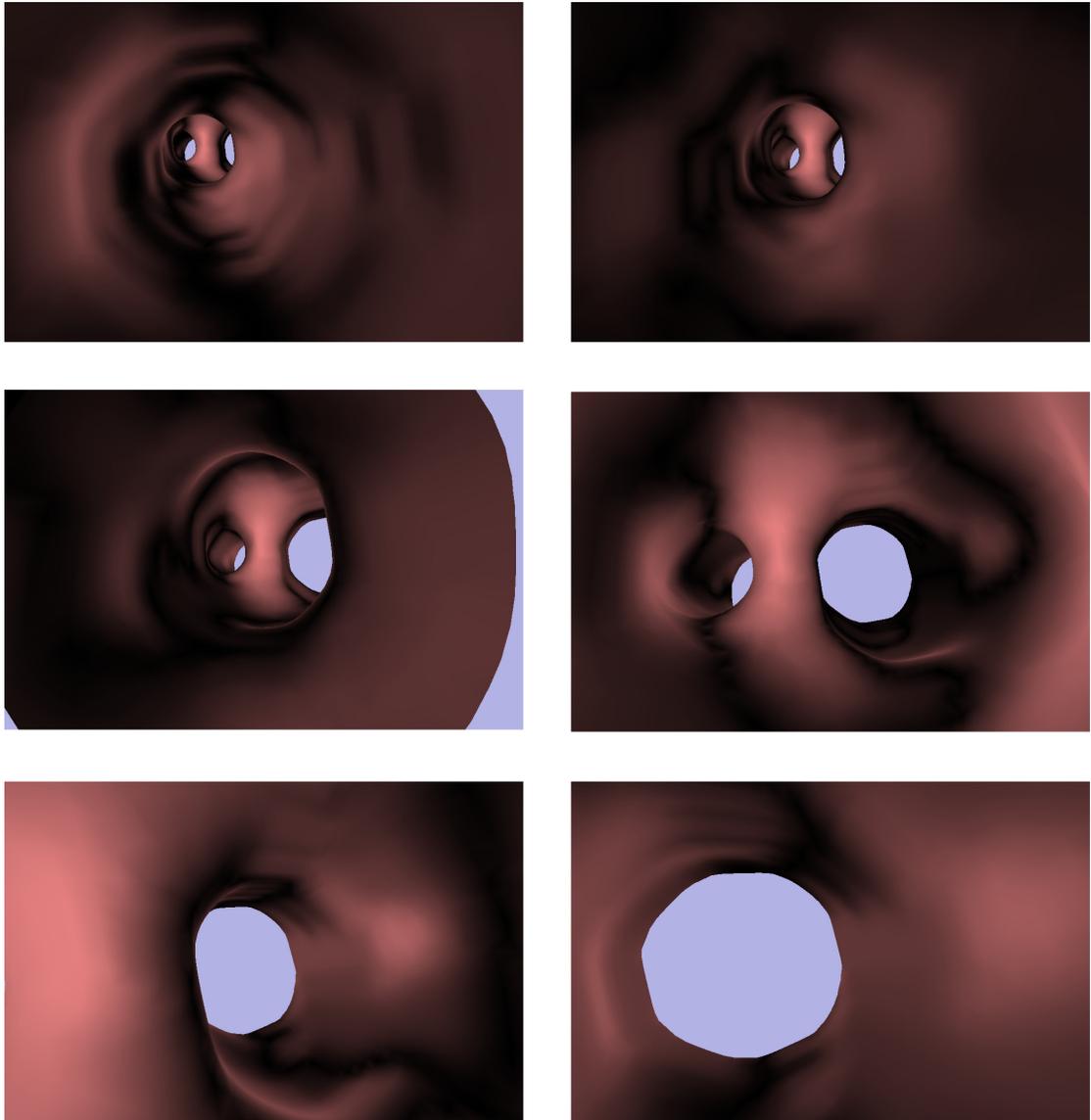


Figure 19: Some views in fly-throughs of Figure 16

CHAPTER 3

AREA-PRESERVING MAPPINGS: A CORRECTION OF CONFORMAL MAPPINGS

The conformal mappings presented in Chapter 2 preserve angles and hence preserve local geometry, but they are not area-preserving in general, since a flattening cannot be both angle-preserving and area-preserving unless the original surface has zero gaussian curvature. Some areas on the original surface may be greatly enlarged or shrunk after flattening. This problem becomes particularly pronounced when we wish to construct a flattened representation for a multi-branched surface. In a conformal representation, each time a vessel passes a branch point, it approximately narrows by a factor of two, as shown in Figure 1. In other words, the vessel at any point has approximately narrowed by a factor of 2^N , where N is the number of branch-points from the *root*. Hence, it might be interesting and useful to consider another type of one-to-one mapping that preserves area [99, 98].

In this chapter, we will present an algorithm for area correction based on the result of conformal flattening. This algorithm used the theory of optimal mass transportation. Section 3.1 gives the introduction of optimal mass transportation and our general approach to the problem by using a gradient descent method. Section 3.2 briefly presents the algorithm of making area correction based on the results of conformal mapping. And Section 3.3 shows area-corrected flattenings of the vessels presented in Chapter 2.

3.1 *Optimal Mass Transportation*

3.1.1 Introduction

The mass transport problem, which is also known as the Monge-Kantorovich Problem (MKP), was first formulated by a French Mathematician Gaspar Monge in 1781 as the following [73]:

Split two equally large volumes into infinitely small particles and then associate them with each other so that the sum of products of these paths of the particles to a volume is least. Along what paths must the particles be transported and what is the smallest transportation cost?

It was given a modern formulation in the work of Kantorovich [54]. The original problem is concerned with finding the optimal way, in the sense of minimal transportation cost, of moving a pile of soil from one place to another. The total amount of soil, or the mass, is conserved in the process.

Optimal transport methods have appeared in econometrics, fluid dynamics, automatic control, transportation, statistical physics, shape optimization, expert systems, and meteorology [72]. They also naturally fit into certain problems in computer vision [29]. In particular, for tracking problem, a robust and reliable object and shape recognition system is of major importance. A key way to carry this out is via *template matching*, which is the matching of one object to another within a given category of objects. Typically, the matching will not be exact and some *shape metric* is necessary for measuring the similarity or “goodness of fit” between objects [40].

The optimal transport problem has also been studied within the context of certain imaging applications, in particular for content-based image retrieval [75, 76, 64]. In these works, pixels in an image are divided into several bins (called “*signatures*”) according to their positions in color and/or spatial locations. The Earth Mover’s Distance (EMD) is calculated between the *signatures* of two images and then used

for image retrieval. However, the EMD method cannot give a warped grid defined on every pixel, which is essential for image registration and image morphing. Some interesting discrete computational techniques have also been proposed in [52] and applied to images. Using the Kantorovich–Wasserstein distance for image registration and warping has a number of advantages. First, it is parameter free. Second, it utilizes all of the grayscale data in both images, and places the two images on equal footing. It is thus symmetrical (if no comparison term is added), the optimal mapping from image A to image B being the inverse of the optimal mapping from B to A , which is a desired property for registration [82]. Hence, it is not necessary to use an extra constraint to guarantee the symmetrical property, as in [19]. Third, it does not require landmarks to be specified. The minimizer of the distance functional involved is unique; there are no other local minimizers. Finally, it is specifically designed to take into account changes in density that result from changes in area or volume. This last point is essential. In fact, as we will show this method leads to area-preserving diffeomorphisms which have very nice regularity properties since they are naturally derived from conformal angle-preserving ones.

3.1.2 Mathematical Formulation

Table 1: Notation used in Chapter 3.

Ω_0	source domain
Ω_1	target domain
μ_0	a density map in domain Ω_0 , which is positive everywhere
μ_1	a density map in domain Ω_1 , which is positive everywhere
u^0	the initial mapping function
s^{-1}	an MP mappings at given time t from (Ω_0, μ_0) to (Ω_0, μ_0) ; for simplicity we usually omit t
s	the inverse function of s^{-1} , which is also an MP mapping from (Ω_0, μ_0) to (Ω_0, μ_0)
\tilde{s}^{-1}	the MP mappings from (Ω_0, μ_0) to (Ω_0, μ_0) at $t \rightarrow \infty$, as well as its inverse function \tilde{s}
u	$u = u^0 \circ s^{-1}$ an MP mapping function at given time t ; t is omitted for simplicity
\tilde{u}	$\tilde{u} = u^0 \circ \tilde{s}^{-1}$, the optimal mapping function, which is u at time $t \rightarrow \infty$
w	a scalar field
χ	a divergence-free vector field, such that $u = \nabla w + \chi$
ζ	a divergence-free vector field

A modern formulation of Monge–Kantorovich problem is now given as follows: assume Ω_0 and Ω_1 to be two subdomains of \mathbf{R}^d , with smooth boundaries, each with a positive density function, μ_0 and μ_1 , respectively. A further assumption is made to make both domains contain same total amount of mass:

$$\int_{\Omega_0} \mu_0 = \int_{\Omega_1} \mu_1 \quad (27)$$

If not, we can always scale one of them to make the total amount of mass equal.

We consider diffeomorphisms u from Ω_0 to Ω_1 , which maps one density to the other in the sense that

$$\mu_0 = |Du| \mu_1 \circ u, \quad (28)$$

which is called the *mass preserving* (MP) property, and written as $u \in MP$. Equation (28) is called the *Jacobian equation*. Here $|Du|$ denotes the determinant of the Jacobian Du , and \circ denotes the composition of two functions. In particular, Equation (28) implies, for example, that if a small region in Ω_0 is mapped to a larger region in Ω_1 , there must be a corresponding decrease in density in order for the mass to be preserved. A mapping u that satisfies this property thus defines a redistribution of a mass of material from one distribution (Ω_0, μ_0) to another (Ω_1, μ_1) .

There exist infinite number of such mappings. A criteria, or a metric, must be defined first in order to solve an optimal mapping. A typical metric, called the L^p Kantorovich–Wasserstein metric, is defined as follows:

$$d_p(\mu_0, \mu_1) := \inf_{u \in MP} \int \|u(x) - x\|^p \mu_0(x) dx. \quad (29)$$

This metric can be seen as to place a penalty on the L^p of each bit of material moved by the map u , weighted by the material’s mass. An optimal mass preserving mapping \tilde{u} is the one that minimizes this functional, while satisfying equation (28). From another point of view, the Kantorovich–Wasserstein metric defines the distance between two mass densities, by computing the “cheapest” way to transport the mass from one domain to the other with respect to equation (29).

The optimal mass-preserving mapping places the two images on equal footing and is symmetrical: the optimal MP mapping from (Ω_0, μ_0) to (Ω_1, μ_1) is the inverse of the optimal mapping from (Ω_1, μ_1) to (Ω_0, μ_0) .

Assume u be an MP mapping from (Ω_0, μ_0) to (Ω_1, μ_1) and $y = u(x)$, we have

$$\begin{aligned} \int_{\Omega_0} \|u(x) - x\|^p \mu_0(x) dx &= \int_{\Omega_1} \|u \circ u^{-1}(y) - u^{-1}(y)\|^p \mu_0 \circ u^{-1}(y) du^{-1}(y) \\ &= \int_{\Omega_1} \|y - u^{-1}(y)\|^p \mu_0 \circ u^{-1}(y) |Du^{-1}(y)| dy \\ &= \int_{\Omega_1} \|y - u^{-1}(y)\|^p \mu_1(y) dy, \end{aligned}$$

Where we use the MP property of u^{-1} : $\mu_1 = |Du^{-1}| \mu_0 \circ u^{-1}$.

Noticing that this statement holds for any $u \in MP$, we have

$$d_p(\mu_0, \mu_1) = d_p(\mu_1, \mu_0),$$

and that if \tilde{u} denotes the optimal mass-preserving map from Ω_0 to Ω_1 then the optimal from Ω_1 to Ω_0 is precisely \tilde{u}^{-1} , i.e. optimal mass preserving mapping is symmetrical.

The case $p = 2$ has been extensively studied and will be the one used here for image morphing. The L^2 Monge–Kantorovich problem has been studied in statistics, functional analysis, and the atmospheric sciences [9]. A fundamental theoretical result [15, 31, 58], shows that there is a unique optimal mapping $\tilde{u} \in MP$ transporting (Ω_0, μ_0) to (Ω_1, μ_1) , and that this \tilde{u} is characterized as the gradient of a convex function w , i.e., $\tilde{u} = \nabla w$. Note that from Equation (28), we have that w satisfies the *Monge–Ampère* equation

$$|Hw| \mu_1 \circ (\nabla w) = \mu_0, \tag{30}$$

where $|Hw|$ denotes the determinant of the Hessian of w . In 2D, the Hessian matrix is

$$Hw = \begin{bmatrix} \omega_{xx} & \omega_{xy} \\ \omega_{yx} & \omega_{yy} \end{bmatrix} \tag{31}$$

There have been a number of algorithms considered for computing an optimal

transport map. For example, methods have been proposed based on linear programming [72], and on Lagrangian mechanics closely related to ideas from the study of fluid dynamics [9]. An interesting geometric method has been formulated by Cullen and Purser [22]. Among them the most common approach is to reduce the L^2 optimal transport to a linear programming problem. However, a fundamental difficulty for doing so is the computational complexity. Even in 2D case, for a pair of 256×256 images we may have $256^2 \times 256^2$ possibilities, and the linear programming problem can get to be quite unwieldy [52].

In our method, we will employ a natural solution to the optimal transport problem based on the equivalent problem of polar factorization [15, 30, 67]. Assume we have an initial mapping $u^0 : (\Omega_0, \mu_0) \rightarrow (\Omega_1, \mu_1)$ with the mass preserving (MP) property (the details for construction such an initial mapping will be described later). According to the generalized results of [15, 30], u^0 has a unique decomposition of the form

$$u^0 = (\nabla w) \circ s, \tag{32}$$

where w is a convex function and s is an MP mapping $s : (\Omega_0, \mu_0) \rightarrow (\Omega_0, \mu_0)$. This is called the *polar factorization* of u^0 with respect to μ_0 .

We will find the polar factorization of the MP mapping u , in an iterative way. We consider the family of MP mappings in the form of $u = u^0 \circ s^{-1}$ as s varies over diffeomorphic MP mappings from (Ω_0, μ_0) to itself. If we consider u as a vector field, we can always find a function w and another vector field χ , with $\text{div}(\chi) = 0$, such that

$$u = \nabla w + \chi, \tag{33}$$

which is called the Helmholtz decomposition, i.e., we can decompose u into the sum of a curl-free and a divergence-free vector field [83]. Once the divergence-free part is zero, the polar factorization is completed. Thus, what we try to do is to find a mapping \tilde{s} which will yield a \tilde{u} without any curl, such that $\tilde{u} = \nabla w$. Once such an \tilde{s}

is found, we will have $u^0 = \tilde{u} \circ \tilde{s} = (\nabla w) \circ \tilde{s}$ and so we will find the polar factorization as equation (30) of our given function u^0 .

As we have discussed above, the unique optimal solution of the L^2 Monge–Kantorovich problem has the form of $\tilde{u} = \nabla w$, so the problem of finding the unique polar factorization of u^0 and finding the optimal Monge–Kantorovich mapping \tilde{u} are equivalent. More mathematical details describing this connection can be found in [1, 15, 31]. In essence, to solve the Monge–Kantorovich problem we create a *rearrangement* of an initial vector field u^0 using an MP mapping \tilde{s} , so that the resulting vector field $\tilde{u} = u^0 \circ \tilde{s}^{-1}$ has no curl. $\tilde{u} : \Omega_0 \rightarrow \Omega_1$ is an MP mapping of the form $\tilde{u} = \nabla w$, with w convex. Uniqueness follows from the theory of the Monge–Ampère equation (30) (see [84], page 251). We can now give the technical details of our construction.

In our following discussion, we intend to minimize a more general form of the energy functional in the form of

$$M[u] = \int_{\Omega_0} \Phi(u(x) - x) \mu_0(x) dx \quad (34)$$

where Φ is a non-negative C^1 function. In the case of $\Phi(x) = \|x\|^2$, it is exactly the L^2 Kantorovich–Wasserstein functional. The algorithm is a two-step approach. The first step is to find an initial MP mapping u^0 . And the second step is to minimize the energy functional over $u = u^0 \circ s^{-1}$ by varying s over MP mappings from Ω_0 to Ω_1 , starting with s equal to the identity map.

- **Finding an initial mapping u^0**

For a general domain, this initial mapping can be solved using a method by Moser [24]. Since we are working with regular domains (in 2D case, two rectangular domains or in 3D case, two cubic domains), a simpler algorithm is implemented here. For simplicity, assume we are working in \mathbf{R}^2 and $\Omega_0 = \Omega_1 = [0, 1]^2$, the generalization to higher dimensions being straightforward. The idea of this construction is that we

solve a one-dimensional mass transport problem in one direction and then solve for a family of one-dimensional mass transport problems in the other direction. In 1D, the optimal transport map can be found by simple quadrature. We assume mass is first transported along lines parallel to the x -axis, and then transported along lines parallel to the y -axis.

Accordingly, we define a function $a = a(x)$ by equation

$$\int_0^{a(x)} \int_0^1 \mu_1(\eta, y) dy d\eta = \int_0^x \int_0^1 \mu_0(\eta, y) dy d\eta. \quad (35)$$

By differentiation with respect to x , we have

$$a'(x) \int_0^1 \mu_1(a(x), y) dy = \int_0^1 \mu_0(x, y) dy. \quad (36)$$

We may now define a function $b = b(x, y)$ by equation

$$a'(x) \int_0^{b(x, y)} \mu_1(a(x), \rho) d\rho = \int_0^y \mu_0(x, \rho) d\rho, \quad (37)$$

and set $u^0(x, y) = (a(x), b(x, y))$. Since $a_y = 0$, $|Du| = a_x b_y = a'(x) b_y(x, y)$, by differentiating (37) with respect to y we find

$$a'(x) b_y(x, y) \mu_1(a(x), b(x, y)) = \mu_0(x, y) \quad (38)$$

$$|Du^0| \mu_1 \circ u^0 = \mu_0, \quad (39)$$

which is the MP property we need. In practice, a and b can be found with simple numerical integration techniques. Given our assumption that μ_0 and μ_1 are positive everywhere, $a(x)$ is monotonically increasing from (35), and $b(x, y)$ is also monotonically increasing with respect to y from (37) since $a'(x)$ is always positive. Hence, there is no space folding problem with the initial mapping m^0 .

It should be pointed out that the energy functional to be minimized is not involved when finding u^0 . Hence, this initial mapping approach is suitable for any energy functional.

- **Finding the minimizer \tilde{u}**

Once an initial MP u^0 is found, we need to apply a process to minimize the energy functional. Since u^0 is an MP, while $u = u^0 \circ s^{-1}$ we see that u is an MP mapping if and only if s is an MP map from (Ω_0, μ_0) to itself, that is, if and only if

$$\mu_0 = |Ds| \mu_0 \circ s. \quad (40)$$

The same argument applies to \tilde{s} (the optimal rearrangement of initial mapping) also.

Next, rather than working with s directly, we solve for the energy minimizing problem iteratively via a gradient descend method. If the energy functional is the L^2 Kantorovich-Wasserstein metric, this process is equivalent to finding the polar factorization and is guaranteed to converge to a global optimum [2]. We assume that s is a function of time, initially being the identity map. Then the evolution of s : $s_t := \frac{d}{dt}s$ should decrease the energy functional. This will give us an evolution equation for s_t and in turn an equation for u_t as well, the latter being the most important for implementation. In what follows the t subscript denotes differentiation with respect to time t , D and ∇ refer to spatial derivatives, and div is the spacial divergence operator.

Since s is an MP mapping from (Ω_0, μ_0) to itself, we have that $\mu_0 = |Ds| \mu_0 \circ s$. By differentiating it with respect to time t , we get

$$\begin{aligned} 0 &= |Ds|_t \mu_0 \circ s + |Ds| (\mu_0 \circ s)_t \\ 0 &= |Ds| \left(\text{div} \left(s_t \circ s^{-1} \right) \circ s \right) \mu_0 \circ s + |Ds| \langle (\nabla \mu_0) \circ s, s_t \rangle, \\ 0 &= \left(\mu_0 \text{div} \left(s_t \circ s^{-1} \right) \right) \circ s + \langle (\nabla \mu_0) \circ s, s_t \rangle, \\ 0 &= \mu_0 \text{div} \left(s_t \circ s^{-1} \right) + \left\langle \nabla \mu_0, s_t \circ s^{-1} \right\rangle \\ 0 &= \text{div}(\mu_0 s_t \circ s^{-1}), \end{aligned} \quad (41)$$

Hence s_t should have the following form

$$s_t = \left(\frac{1}{\mu_0} \zeta \right) \circ s, \quad (42)$$

for some vector field ζ on Ω_0 , with $\operatorname{div}(\zeta) = 0$ and $\langle \zeta, \vec{n} \rangle = 0$ on $\partial\Omega_0$, \vec{n} being the normal to the boundary of Ω_0 . Since u and s satisfy $u \circ s = u^0$, by taking the derivative of it respective to t , we derive

$$\begin{aligned} (Du \circ s) s_t + u_t \circ s &= 0 \\ u_t \circ s &= -(Du \circ s) s_t \\ u_t &= -Du s_t \circ s^{-1}. \end{aligned}$$

By equation (42) we have

$$u_t = -\frac{1}{\mu_0} Du \zeta, \quad (43)$$

Now let us recall the energy functional (34), which is repeated here for clarity.

$$M[u] = \int_{\Omega_0} \Phi(u(x) - x) \mu_0(x) dx \quad (44)$$

A change of variable is applied here by substituting x with $y = s^{-1}(x)$. Due to the MP property of s and s^{-1} , the following is obvious:

$$\mu_0(x) dx = \mu_0 \circ s(y) ds(y) = \mu_0 \circ s(y) |Ds(y)| dy = \mu_0(y) dy, \quad (45)$$

and also $u(x) = u \circ s(y) = u^0(y)$. Hence, functional (44) equals

$$M = \int_{\Omega_0} \Phi(u^0(y) - s(y)) \mu_0(y) dy \quad (46)$$

By taking the derivative of (46) respect to t , we get,

$$\frac{dM}{dt} = - \int_{\Omega_0} \langle \nabla \Phi(u^0(y) - s(y)), \frac{\partial s}{\partial t}(y) \rangle \mu_0(y) dy \quad (47)$$

Then we do another change of variable by substituting y back with $x = s(y)$ and get

$$\frac{dM}{dt} = - \int_{\Omega_0} \langle \nabla \Phi(u(x) - x), \mu_0(x) s_t \circ s^{-1}(x) \rangle dx \quad (48)$$

Clearly, were it not for the constraint $\operatorname{div}(\mu_0 s_t \circ s^{-1}) = 0$, we could take $\mu_0 s_t \circ s^{-1} = \nabla\Phi(u(x) - x)$ to decrease energy. However, considering this mass-preserving constraint, $\mu_0 s_t \circ s^{-1}$ should be a divergence-free vector field. Hence, we define

$$\zeta = \mu_0 s_t \circ s^{-1} \quad (49)$$

Assume $\nabla\Phi(u(x) - x)$ can be decomposed into a curl-free part and a divergence-free part, i.e.

$$\nabla\Phi(u(x) - x) = \nabla w + \chi, \quad (50)$$

where $\operatorname{div}(\chi) = 0$ and $\langle \chi, \vec{n} \rangle = 0$ on $\partial\Omega_0$. Then equation (48) can be rewritten as,

$$\begin{aligned} -M_t &= \int_{\Omega_0} \langle \nabla w + \chi, \zeta \rangle \\ &= \int_{\Omega_0} \langle \nabla w, \zeta \rangle + \int_{\Omega_0} \langle \chi, \zeta \rangle \\ &= \int_{\Omega_0} (\operatorname{div}(w\zeta) - w \operatorname{div}(\zeta)) + \int_{\Omega_0} \langle \chi, \zeta \rangle \\ &= \int_{\partial\Omega_0} w \langle \zeta, \vec{n} \rangle + \int_{\Omega_0} \langle \chi, \zeta \rangle \\ &= \int_{\Omega_0} \langle \chi, \zeta \rangle, \end{aligned} \quad (51)$$

where ζ is chosen to be χ , and can be found through Helmholtz decomposition.

Now we give the evolving equation for u in general \mathbf{R}^d case, as well as in 2D case which has a simpler expression and will be used in our algorithm.

Gradient Descent: \mathbf{R}^d :

By taking the divergence of equation (50) on both sides, it is easy to see that w should be a solution of the following Neumann-type boundary problem

$$\begin{aligned} \operatorname{div}(\nabla\Phi(u(x) - x)) &= \Delta w \\ \langle \nabla w, \vec{n} \rangle &= \langle \nabla\Phi(u(x) - x), \vec{n} \rangle \text{ on } \partial\Omega_0, \end{aligned} \quad (52)$$

and we can set $\chi = \nabla\Phi(u(x) - x) - \nabla w$. It is then easily seen that χ satisfies the necessary requirements. This PDE can be solved using a number of methods, e.g.

finite volume as in [5]. Thus, by (43), we have the following evolution equation for u :

$$u_t = -\frac{1}{\mu_0} Du \left\{ \nabla \Phi(u(x) - x) - \nabla \Delta^{-1} \operatorname{div} [\nabla \Phi(u(x) - x)] \right\}, \quad (53)$$

which is a first order non-local scheme for u_t , if we count Δ^{-1} as minus 2 derivatives.

If $\Phi(u(x) - x)$ has the form of $\|u(x) - x\|^2$, which is the L^2 Monge-Kantorovich problem, equation (53) has the form of,

$$u_t = -\frac{2}{\mu_0} Du \left\{ u(x) - x - \nabla \Delta^{-1} \operatorname{div} [u(x) - x] \right\}. \quad (54)$$

Note that this flow is consistent with the L^2 Monge-Kantorovich theory in the following sense. If \tilde{u} is the optimal MP mapping, then it is given as the gradient of a scalar function. The divergence-free part of the Helmholtz decomposition with respect to \tilde{u} is zero and the curl-free part is \tilde{u} itself. Hence,

$$\tilde{u} - \nabla \Delta^{-1} \operatorname{div}(\tilde{u}) = 0, \quad (55)$$

Similarly,

$$x = \nabla \Delta^{-1} \operatorname{div}(x). \quad (56)$$

We get $\tilde{u}_t = 0$.

Gradient Descent: \mathbf{R}^2 :

The situation is somewhat simpler in the \mathbf{R}^2 case, due to the fact that a divergence free vector field ζ can in general be written as $\zeta = \nabla^\perp h$ for a scalar function h , where \perp represents rotation by $\pi/2$ counter clockwise, so that $\nabla^\perp h = (-h_y, h_x)$. In this case, equation (51) becomes

$$-M_t = \int_{\Omega_0} \langle \nabla^\perp f, \nabla^\perp h \rangle = \int_{\Omega_0} \langle \nabla f, \nabla h \rangle \quad (57)$$

where the decomposition of $\nabla \Phi(u(x) - x)$ is $\nabla \Phi(u(x) - x) = \nabla w + \nabla^\perp f$, and we can let h equal f . Hence,

$$\nabla \Phi(u(x) - x)^\perp = \nabla^\perp w - \nabla f. \quad (58)$$

Considering

$$\operatorname{div}(\nabla^\perp w) = \operatorname{div}(-w_y, w_x) = -w_{yx} + w_{xy} = 0, \quad (59)$$

the function f can be solved by the following Dirichlet-type boundary problem

$$\begin{aligned} -\operatorname{div}(\nabla\Phi(u(x) - x)^\perp) &= \Delta f, \\ f &= 0 \text{ on } \partial\Omega_0, \end{aligned} \quad (60)$$

which gives us the evolution equation

$$u_t = \frac{1}{\mu_0} Du \nabla^\perp \Delta^{-1} \operatorname{div}(\nabla\Phi(u(x) - x)^\perp). \quad (61)$$

In the L^2 Monge-Kantorovich problem, equation (61) can be rewritten as,

$$u_t = \frac{2}{\mu_0} Du \nabla^\perp \Delta^{-1} \operatorname{div}((u - \underline{id})^\perp) \quad (62)$$

where \underline{id} is an identity map.

3.2 Area-preserving Mappings Based on the L^2 Monge-Kantorovich Problem

It is assumed that there is a uniform mass density on the original 3D surface. After conformal mapping f , the original uniform density is deformed to μ_0 , by assuming that the mass is preserved during the mapping. If we can find another mapping u that maps μ_0 back to a uniform distribution and preserves mass, then the composition of f and u will be an area-preserving mapping, i.e.

$$g = u \circ f.$$

Since both f and u are one-to-one mappings, the resulting area-preserving mapping g is also a one-to-one mapping.

Assume we have the result of the conformal mapping f , which has a range of $\operatorname{range}(f)$ on the plane. We then define a pseudo-density μ_0 on $\operatorname{range}(f)$ to be

$$\mu_0 = |Df|^{-1}. \quad (63)$$

If a region is enlarged after the conformal mapping, $|Df|$ will be a value greater than unity and hence μ_0 will be a value smaller than one. Numerically, μ_0 can be defined as the area of a triangle on the original surface divided by the area of the triangle once flattened, and then μ_0 is interpolated on a rectangular grid.

In order to solve the problem on a rectangular domain, μ_0 is extended to a rectangular region Ω_0 surrounding $\text{range}(f)$ by setting μ_0 to a constant outside of $\text{range}(f)$. The constant can be the mean of μ_0 inside $\text{range}(f)$ or it can be several times of the mean.

The target density map μ_1 is one everywhere on another rectangular region Ω_1 , such that

$$\int_{\Omega_0} \mu_0 = \int_{\Omega_1} \mu_1 \quad (64)$$

We are going to find an MP mapping function \tilde{u} from (Ω_0, μ_0) to (Ω_1, μ_1) such that

$$\mu_0 = |D\tilde{u}| \mu_1 \circ \tilde{u} \quad (65)$$

And we want to find an optimal one in the L^2 MKP sense, i.e. We minimize the functional

$$d(\mu_0, \mu_1)^2 = \int \|\tilde{u}(x) - x\|^2 \mu_0(x) dx. \quad (66)$$

As we have discussed in the previous section, the unique optimal solution \tilde{u} is characterized as the gradient of a convex function w , i.e., $\tilde{u} = \nabla w$.

Since $\mu_1 = 1$, the constraint equation (65) reduces to $\mu_0 = |D\tilde{u}|$. Considering equation (63), we get

$$|Df| \cdot |D\tilde{u}| = 1. \quad (67)$$

Hence, the mapping \tilde{u} compensates exactly for the distortion in area that occurred during conformal flattening. Further, since we minimize the functional (66) to find the minimizing \tilde{u} , our solution differs minimally from the identity.

The algorithm for finding the optimal mapping begins with finding an initial mapping u^0 followed by removing the curl of u^0 . The details can be found in Section 3.1.

3.3 *Medical Examples*

Now we show some examples by using the same datasets as we used in Chapter 2. Figure 20(a) shows the pseudo mass density μ_0 for the *cerebral* artery example, defined by equation 63. We get the mean of μ_0 inside $range(f)$ and multiple it by three to fill the regions out of $range(f)$, as shown in the light gray region surrounding the vessel. A factor three is chosen here since it is desired to have enough space between different branches in the constructed area-preserving mapping. The dark color represents enlarged areas and the light color represents shrunk areas. Figure 20(b) is the initial mapping $u^0 \circ f$ for area correction, where f is the conformal mapping. Figure 20(c) shows the optimal mapping $\tilde{u} \circ f$. Both Figure 20(b) and (c) are painted by vessel's normal on the original 3D surface. Although corrected in the sense of area, surface structures are still clearly discernible. The curl-free nature of the Monge-Kantorovich mapping avoids distortion effects often associated with area preserving maps. The area changing ratio in the conformal mapping can be calculated for each triangle. Figure 21 shows the histogram of area-changing ratios in conformal mapping. It can be clearly seen from Figure 21 that a conformal mapping can cause large area distortions, especially for highly branched surfaces. Figure 22 shows the histogram of area-changing ratio after area correction by using the L^2 MKP. It can be clearly seen that the area of triangles has been greatly corrected. The reason why the area cannot be completely corrected is due to numerical errors. Remember we use the area changing ratio of triangles to approximate $|Df|^{-1}$ (and thus defines μ_0), which can cause some numerical errors. Further, there are two resamplings that will cause more numerical errors. The first one is the resampling of μ_0 on a rectangular grid for easy computation. And the second one is to decide the final positions of vertices by

interpolation on a deformed rectangular grid. There are also some numerical errors which occur when solving for the L^2 MKP.



(a) The pseudo density μ_0 . (b) The initial mapping $u^0 \circ f$. (c) The final mapping $\tilde{u} \circ f$.

Figure 20: Area-preserving mapping for the front cerebral artery (*cerebral* example).

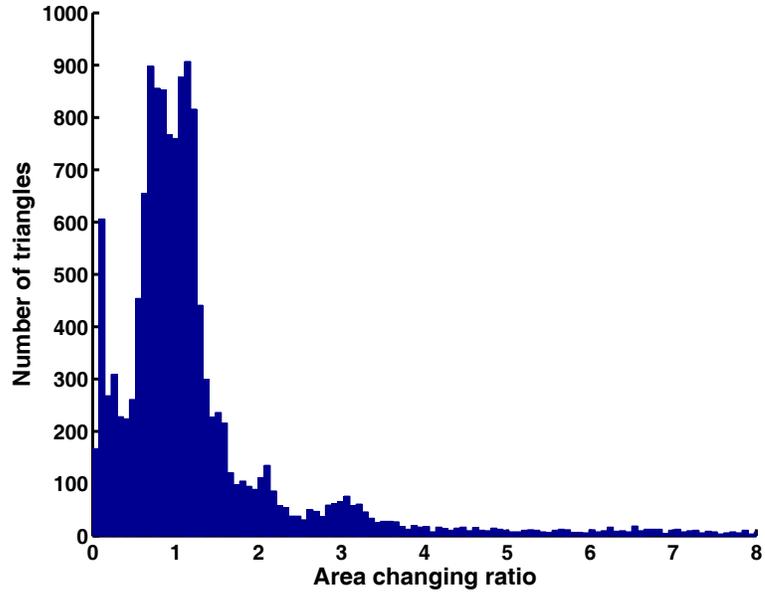


Figure 21: The histogram for triangle area changing ratios in conformal mapping (*cerebral* example).

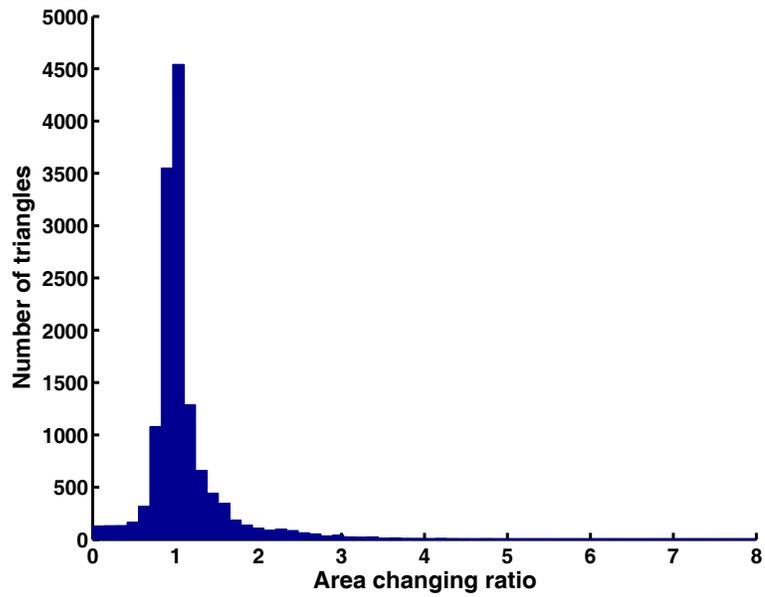


Figure 22: The histogram for triangle area changing ratios in area-preserving mapping (*cerebral* example).

Figure 23 is the pseudo mass density defined for the *coronary* artery example. The original 3D surface of the coronary artery is shown in Figure 11, whose conformal flattened version is shown in Figure 12. Figure 24 is the final area-preserving mapping. Figure 25 and Figure 26 show the histogram of area-changing ratios for all triangles after conformal mapping and after area correction, respectively.

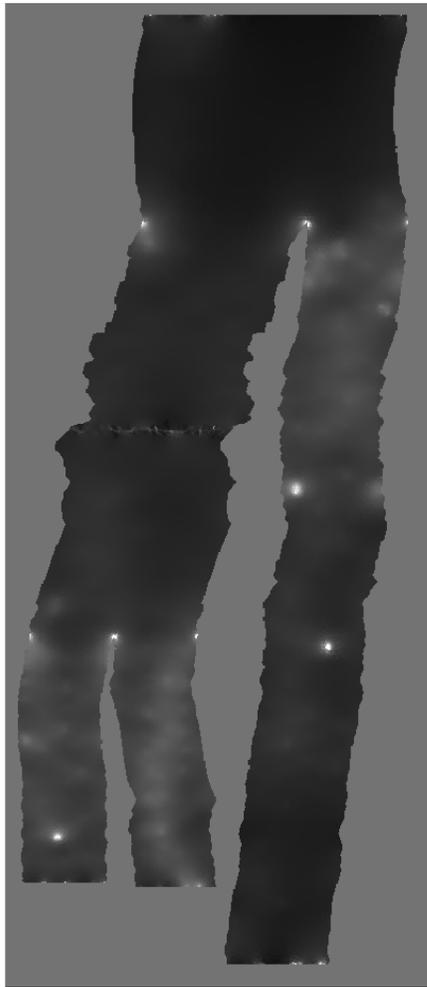


Figure 23: The pseudo mass density μ_0 for the *coronary* example.

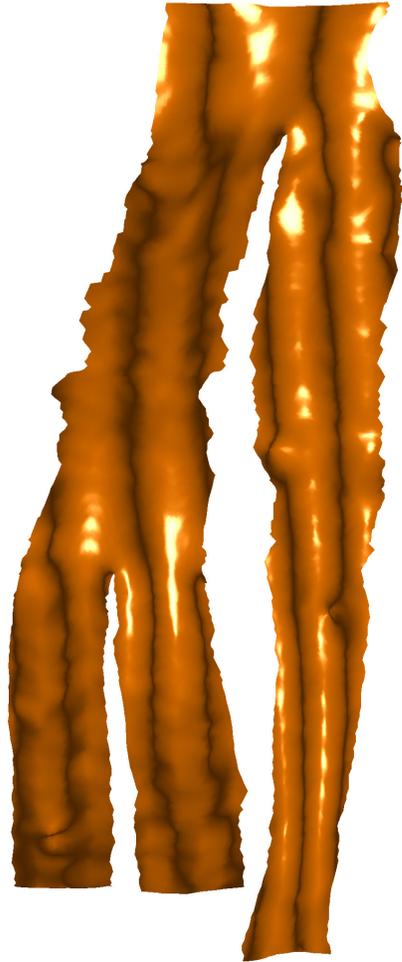


Figure 24: The area-preserving flattening for the *coronary* example.

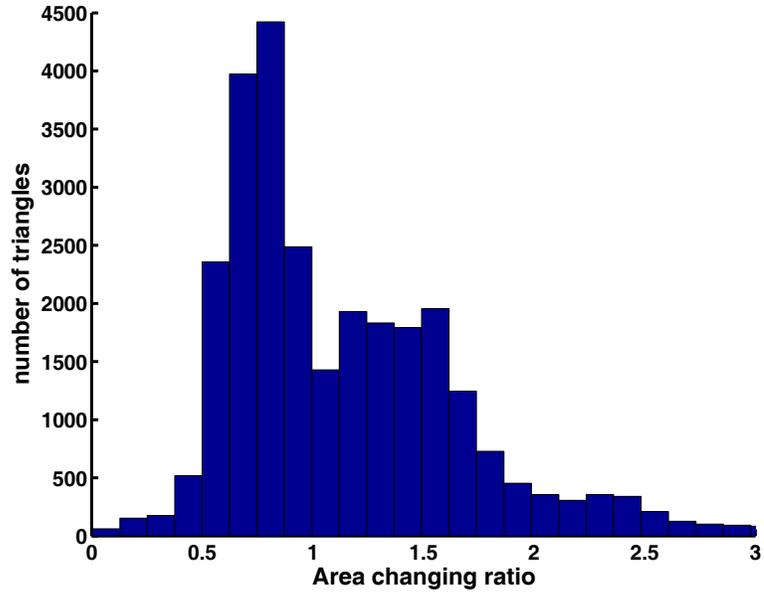


Figure 25: The histogram for triangle area changing ratios in conformal mapping (*coronary* example).

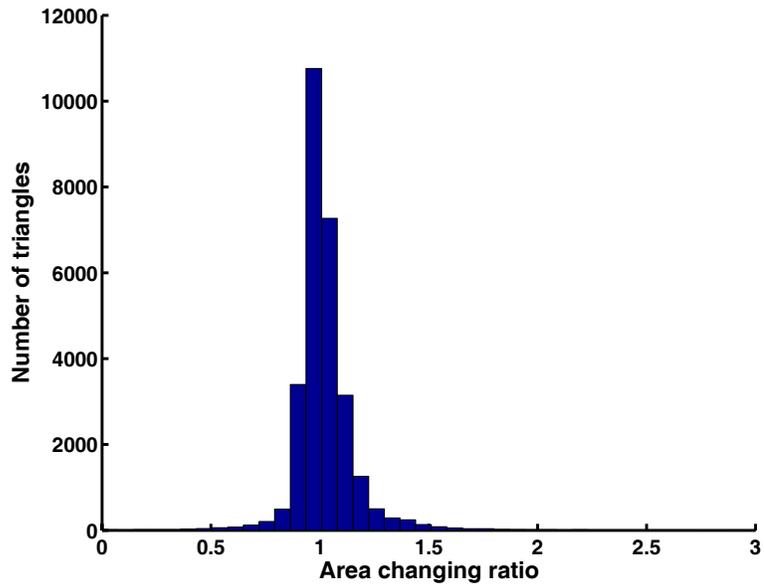


Figure 26: The histogram for triangle area changing ratios in area-preserving mapping (*coronary* example).

Figure 27 through Figure 30 are the area corrections made for a carotid vessel as shown in Figure 16, whose conformal flattening is shown in Figure 17. Figure 27 is the pseudo mass density μ_0 . Figure 28 is the area-preserving flattening. Figure 29 and Figure 30 are the histograms of the area-changing ratios before and after area correction, respectively.

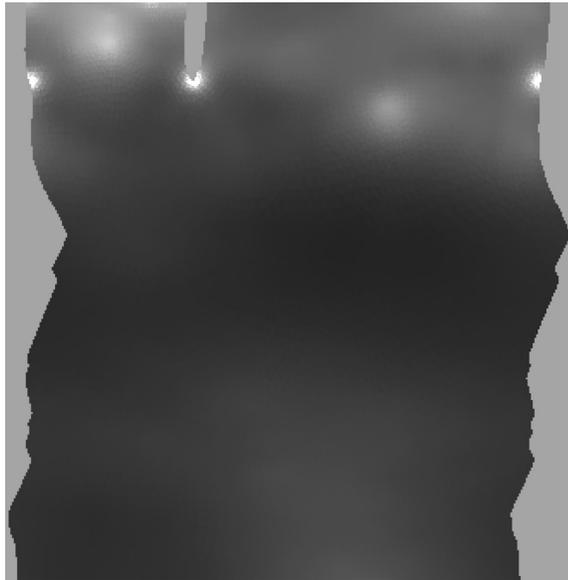


Figure 27: The pseudo mass density μ_0 for the *carotid* example.



Figure 28: The area-preserving flattening for the *carotid* example.

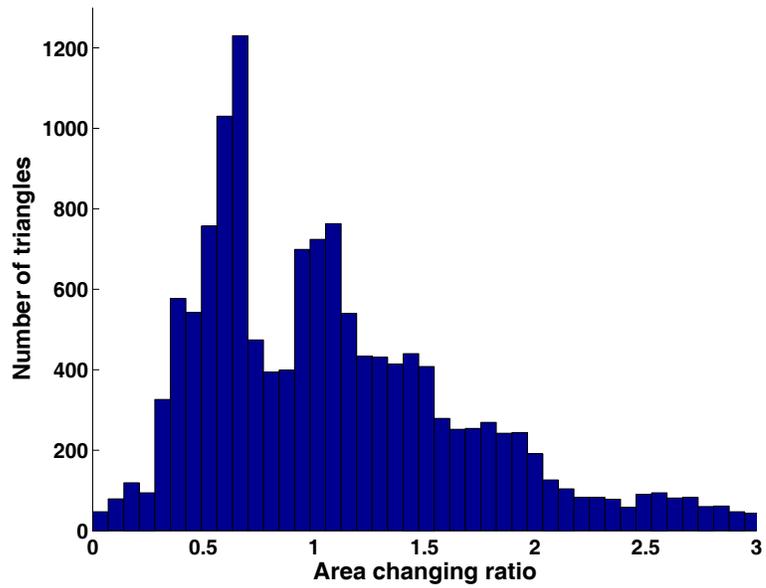


Figure 29: The histogram for triangle area changing ratios in conformal mapping (*carotid example*).

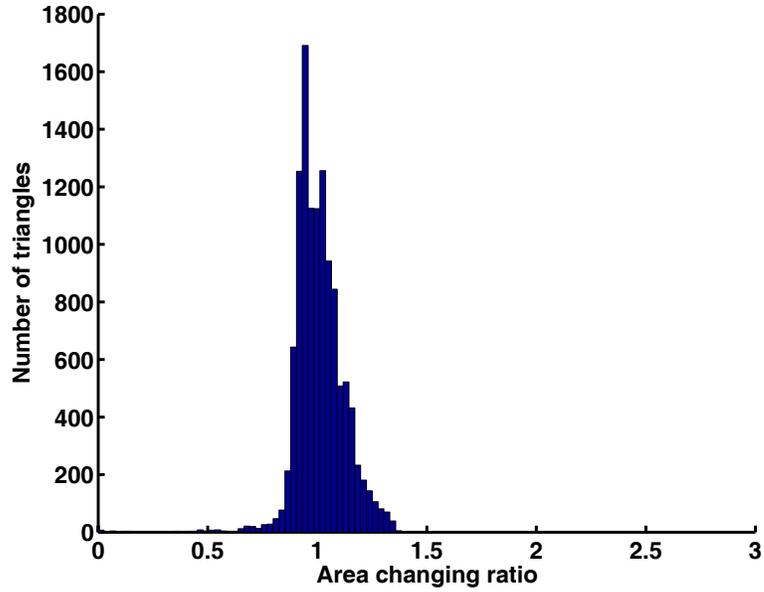


Figure 30: The histogram for triangle area changing ratios in area-preserving mapping (*carotid example*).

The last example is also a carotid artery dataset. Figure 31 shows the original 3D surface, and Figure 32 shows the results from the two types of mappings. The surfaces are shaded by the computed axial wall shear stress (WSS) in dyne/cm^2 . Regions with dark blue colors (indicating low axial WSS) correlate with atherosclerosis [95]. Figure 33 is the histogram of area change in the conformal mapping and Figure 34 is the histogram of area changing after area corrections. Other geometric quantities, such as the cross-sectional area of vessels or the level of calcification can also be visualized in this manner.

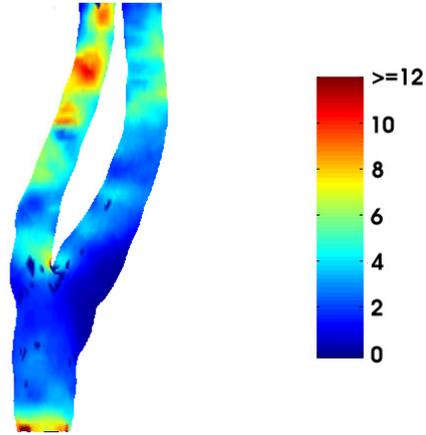
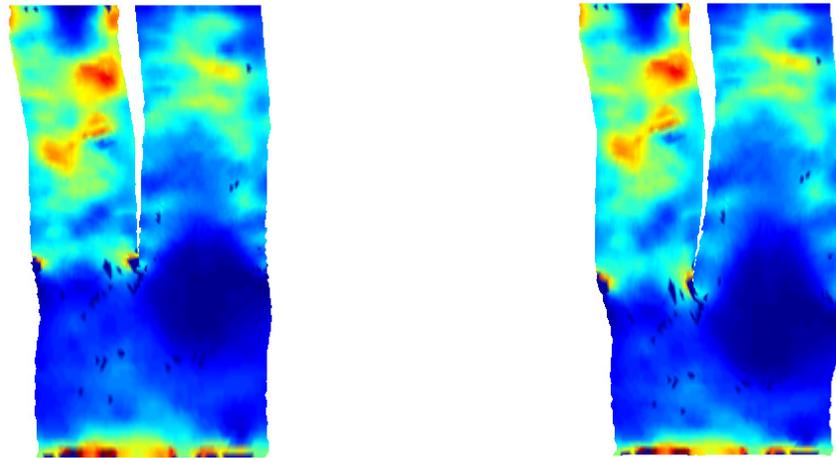


Figure 31: Original carotid artery shaded by wall shear stress.



left: Conformal mapping.

right: Area-preserving mapping.

Figure 32: Flattened carotid artery shaded by wall shear stress.

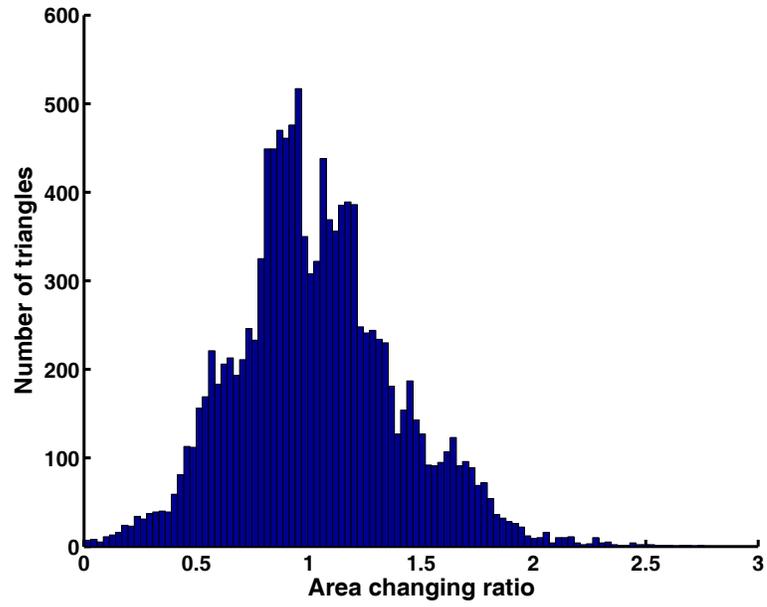


Figure 33: Statistics of area changing in conformal mapping for the second *carotid* example.

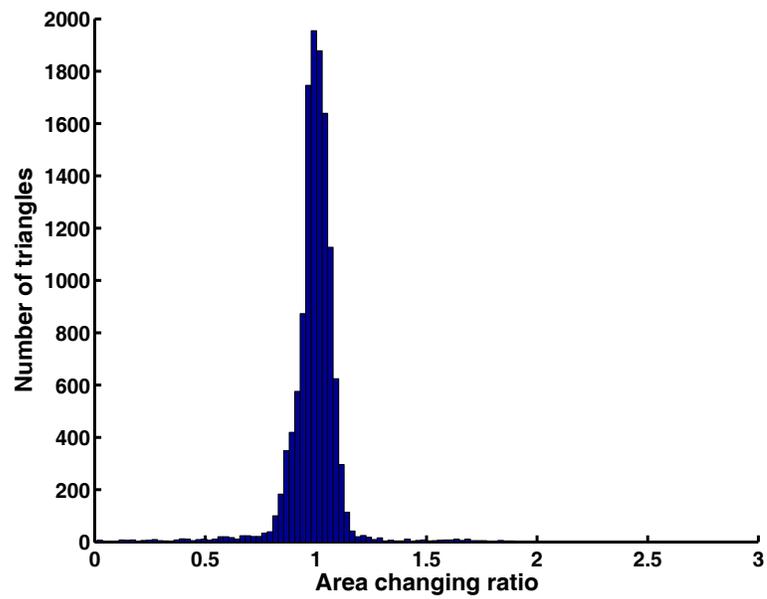


Figure 34: Statistics of area changing in area-preserving mapping for the second *carotid* example.

CHAPTER 4

IMAGE MORPHING BASED ON OPTIMAL MASS TRANSPORTATION

Our interest in the optimal mass transportation problem originally arose in our work in medical applications. As we show in Chapter 3 where it has applications on constructing volume or area preserving mappings. This chapter discusses the applications of optimal mass transportation for image morphing (image interpolation in the time domain) between a pair of related images [39]. The assumption or the constraint of the two images is that they obey a mass preservation property. Thus, we will be matching mass densities in the method, which may be thought of as density weighted areas in 2D or density weighted volumes in 3D.

We now briefly outline the structure of this chapter. Subsection 4.1.1 presents a morphing algorithm by applying the Sum of Squared Differences (SSD) as the similarity measure. Subsection 4.1.2 presents another variation of the algorithm by applying Mutual Information (MI) measure as the comparison term [102]. Subsection 4.1.3 addresses the issue of solving the problem on a doubly-connected domain. Section 4.2 discusses numerical details in the above algorithms. Finally, Section 4.3 illustrates our algorithms with some examples.

4.1 Image Morphing Based on the L^2 Monge-Kantorovich Problem

We can directly apply the optimal MP algorithm on two related images to generate a deformed grid, and calculate the in-between images for image morphing. However, a mapping that maps a small high intensity region to a large low intensity region is not desirable. It will cause undesirable *fade in* and *fade out* effects when doing

morphing. The Monge-Kantorovich metric has penalty only on the “work” spent on moving mass from one shape to another, without concerning the change of mass density. Hence, we add a comparison term to the energy functional to account for the change of intensity.

The idea is to minimize a functional of the following form over MP mappings $u : \Omega_0 \rightarrow \Omega_1$:

$$M_\alpha[u] := \int C(I_0, I_1 \circ u) dx + \alpha \int \|u(x) - x\|^2 \mu_0 dx, \quad (68)$$

for a fixed positive number $\alpha \in \mathbf{R}$. Here the first term controls the “goodness of fit” between the (intensity) images $I_0 : \Omega_0 \rightarrow \mathbf{R}$ and $I_1 \circ u : \Omega_1 \rightarrow \mathbf{R}$, and the second Monge-Kantorovich term controls the the warping of the map. The function μ_0 is the mass density of the source image defined on Ω_0 , which could be the same as I_0 or a smoothed version of I_0 . It could also be any scalar field that is appropriate for the underlying physical model. Similarly, μ_1 is assumed to be the mass density of the target image defined on Ω_1 . By adjusting α , we control the tradeoff between minimal mass transportation and minimal intensity change. In Section 4.3, we will show how to adjust this parameter semi-automatically.

The comparison term $C(I_0, I_1 \circ u)$ can be any metric that measures the similarity between the transformed source and the target image, e.g. SSD, likelihood measurement, correlation ratio, normalized correlation and MI. In this section, SSD and MI are introduced as the similarity measures.

4.1.1 The Sum of Squared Difference as Comparison

If SSD is used as the similarity measure [101], we are minimizing the following energy functional,

$$M_\alpha[u] = \int (I_1 \circ u - I_0)^2 dx + \alpha \int \|u(x) - x\|^2 \mu_0 dx, \quad (69)$$

with the MP constraint $\mu_0 = |Du| \mu_1 \circ u$. Similar to pure MKP, there are two steps in solving this problem. The first step is to build an initial MP mapping u^0 which is the

same as described in the previous chapter, since the construction of u^0 is independent of the energy functional. The second step is to iteratively evolve it to minimize the energy functional.

We now rewrite the energy functional to be the sum of two terms,

$$M_\alpha = M_1 + \alpha M_2 \quad (70)$$

with

$$M_1 = \int (I_1 \circ u - I_0)^2 dx, \quad (71)$$

and

$$M_2 = \int \|u(x) - x\|^2 \mu_0 dx. \quad (72)$$

The M_2 term is exactly the L^2 MKP, whose derivative has been discussed in Section 3.1. Now we consider the derivative of the first term M_1 with respect to time t . Before taking the derivative, we play a trick on M_1 by multiplying it with μ_0 and then dividing it with μ_0 , i.e.

$$M_1 = \int \left[\frac{1}{\mu_0} (I_1 \circ u - I_0)^2 \right] \mu_0 dx. \quad (73)$$

The purpose for doing so is to have a term $\mu_0 dx$, which make the change of variables easier. By setting $y = s^{-1}(x)$ and considering equation (45): $\mu_0(x)dx = \mu_0(y)dy$, we get

$$\begin{aligned} M_1 &= \int (I_1 \circ u(x) - I_0(x))^2 dx \\ M_1 &= \int \left(\frac{1}{\mu_0(x)} (I_1 \circ u(x) - I_0(x))^2 \mu_0(x) \right) dx \\ M_1 &= \int \left(\frac{1}{\mu_0 \circ s(y)} (I_1 \circ u^0(y) - I_0 \circ s(y))^2 \mu_0(y) \right) dy \end{aligned}$$

Then we take the derivative of M_1 with respect to t and get

$$\begin{aligned} M_{1t} &= \int \left\langle -\frac{1}{\mu_0^2 \circ s(y)} (I_1 \circ u^0(y) - I_0 \circ s(y))^2 \nabla \mu_0(y) \circ s(y) \right. \\ &\quad \left. - \frac{2}{\mu_0 \circ s(y)} (I_1 \circ u^0(y) - I_0 \circ s(y)) \nabla I_0 \circ s(y), \frac{\partial s^t}{\partial t} \right\rangle \mu_0(y) dy \quad (74) \end{aligned}$$

$$-M_{1t} = \int \left\langle \frac{1}{\mu_0^2(x)} (I_1 \circ u(x) - I_0(x))^2 \nabla \mu_0(x) \right. \quad (75)$$

$$\left. + \frac{2}{\mu_0(x)} (I_1 \circ u(x) - I_0(x)) \nabla I_0(x), \mu_0 \frac{\partial s}{\partial t} \circ s^{-1}(x) \right\rangle dx \quad (76)$$

From M_2 term, we get

$$-M_{2t} = \int \left\langle 2(u(x) - x), \mu_0 \frac{\partial s}{\partial t} \circ s^{-1}(x) \right\rangle dx \quad (77)$$

We now set

$$P = \frac{1}{\mu_0^2} (I_1 \circ u - I_0)^2 \nabla \mu_0 + \frac{2}{\mu_0} (I_1 \circ u - I_0) \nabla I_0 + 2\alpha (u - \underline{id}) \quad (78)$$

where \underline{id} again denotes an identical mapping. Similar to L^2 MKP, we use a gradient descent algorithm to update the mapping function. In the case of subdomains of \mathbf{R}^d , the following Neumann-type Poisson equation needs to be solved.

$$\operatorname{div}(P) = \Delta w \quad (79)$$

$$\langle \nabla w, \vec{n} \rangle = \langle P, \vec{n} \rangle \text{ on } \partial\Omega_0. \quad (80)$$

From the result in the previous chapter (equation (53)), the evolution of u should be determined by

$$u_t = -\frac{1}{\mu_0} Du (P - \nabla w), \quad (81)$$

or

$$u_t = -\frac{1}{\mu_0} Du \left(P - \nabla \Delta^{-1} \operatorname{div}(P) \right). \quad (82)$$

providing P is defined in equation (78).

In 2D case, a Dirichlet-type boundary problem is solved,

$$-\operatorname{div}(P^\perp) = \Delta f, \quad (83)$$

$$f = 0 \text{ on } \partial\Omega_0. \quad (84)$$

The evolution of u can be written as,

$$u_t = \frac{1}{\mu_0} Du \nabla^\perp f. \quad (85)$$

It may be expressed in a simpler form of

$$u_t = \frac{1}{\mu_0} Du \nabla^\perp \Delta^{-1} \operatorname{div}(P^\perp). \quad (86)$$

4.1.2 Mutual Information as Comparison

The definition of a random variable makes no mention of how random the variable is. Entropy is a statistic that measures the amount of “disorder” of a system (or a random variable). The more random a variable is, the more entropy it will have. In mathematics, the *entropy* of a variable X is defined as [80, 87, 86],

$$H(X) = -E_X[\log(P(X))] = - \sum_{x_i \in \Omega_X} \log(P(X = x_i))P(X = x_i). \quad (87)$$

The classic entropy definition can be extended to continuous variables as differential entropy,

$$h(X) = -E_X[\log(P(X))] = - \int_{-\infty}^{+\infty} p(x) \log(p(x)) dx. \quad (88)$$

Entropy by itself is not sufficient to provide an absolute measure of the randomness, instead it only provides a relative measure of the randomness. Entropy can also be used to relate the predictability of two random variables. *Conditional entropy* of Y given X can be defined as,

$$H(Y|X) = E_X[E_Y[\log(p(Y|X))]]. \quad (89)$$

which is a measure of the randomness of Y given knowledge of X . *Joint entropy* of two random variables X and Y can be defined as,

$$H(Y, X) = E_X[E_Y[\log(p(Y, X))]]. \quad (90)$$

Joint entropy and conditional entropy have the following relation:

$$H(Y|X) = H(X, Y) - H(X) \quad (91)$$

X and Y are considered to be *independent* if

$$H(Y|X) = H(Y) \quad (92)$$

which means that knowing the knowledge of X does not help to predict Y . Considering $\log(p(Y, X)) = \log(p(Y)) + \log(p(X))$, we have

$$H(X, Y) = H(X) + H(Y), \quad (93)$$

if X and Y are independent of each other. When Y becomes more dependent on X , $H(Y|X)$ is smaller. However, a small value of $H(Y|X)$ may happen when $H(Y)$ is small, although X and Y are independent. Hence, it is more reasonable to measure the decrease of randomness of Y given the knowledge of X . The *Mutual Information* (MI) between these two random variables is thus given by

$$I(X, Y) = H(Y) - H(Y|X). \quad (94)$$

By equation (91), MI has a symmetric expression:

$$\begin{aligned} I(X, Y) &= H(Y) - H(Y|X) \\ &= H(Y) + H(X) - H(X, Y) \\ &= H(X) - H(X|Y) \\ &= I(Y, X) \end{aligned} \quad (95)$$

When the Mutual Information is used as the similarity measure, the energy functional to be minimized has the form of

$$\begin{aligned} M_\alpha : &= - \int_{i_0 \times i_1} p_u^{I_0, I_1 \circ u}(i_0, i_1) \log \frac{p_u^{I_0, I_1 \circ u}(i_0, i_1)}{p^{I_0}(i_0) p_u^{I_1 \circ u}(i_1)} di_0 di_1 \\ &+ \alpha \int_{\Omega_0} \|u(x) - x\|^2 \mu_0(x) dx, \end{aligned} \quad (96)$$

Again, we can rewrite the functional to be the sum of two terms as $M_\alpha = M_1 + \alpha M_2$ with

$$M_1 = - \int_{i_0 \times i_1} p_u^{I_0, I_1 \circ u}(i_0, i_1) \log \frac{p_u^{I_0, I_1 \circ u}(i_0, i_1)}{p^{I_0}(i_0) p_u^{I_1 \circ u}(i_1)} di_0 di_1 \quad (97)$$

and

$$M_2 = \int \|u(x) - x\|^2 \mu_0(x) dx. \quad (98)$$

The M_1 term measures the Mutual Information between the transformed source image and the target image. There is a minus sign in front of the MI term, since the more similar between the two images the larger the MI measure. It should also be pointed out that the integral is taken on the domain of $i_0 \times i_1$. And the M_2 term is still the L^2 MKP.

To solve for the optimal MP mapping \tilde{u} , we use the same two step approach. The first step is to find an initial MP mapping u^0 , which is exactly the same as described in Section 3.1.2. And the second step is to evolve u iteratively by decreasing the energy functional, starting with $u = u^0$.

Now we take the derivative of equation (97) respect to t [46],

$$\begin{aligned}
\frac{\partial M_1}{\partial t} &= -\frac{\partial}{\partial t} \int_{i_0 \times i_1} p_u^{I_0, I_1 \circ u}(i_0, i_1) \log \frac{p_u^{I_0, I_1 \circ u}(i_0, i_1)}{p^{I_0}(i_0) p_u^{I_1 \circ u}(i_1)} di_0 di_1 \\
&= -\int_{i_0 \times i_1} \log \frac{p_u^{I_0, I_1 \circ u}(i_0, i_1)}{p^{I_0}(i_0) p_u^{I_1 \circ u}(i_1)} \frac{\partial p_u^{I_0, I_1 \circ u}(i_0, i_1)}{\partial t} di_0 di_1 \\
&\quad - \int_{i_0 \times i_1} \frac{\partial p_u^{I_0, I_1 \circ u}(i_0, i_1)}{\partial t} di_0 di_1 \\
&\quad + \int_{i_0 \times i_1} \frac{p_u^{I_0, I_1 \circ u}(i_0, i_1)}{p_u^{I_1 \circ u}(i_1)} \frac{\partial p_u^{I_1 \circ u}(i_1)}{\partial t} di_0 di_1
\end{aligned} \tag{99}$$

Considering

$$\int_{i_0} p_u^{I_0, I_1 \circ u}(i_0, i_1) di_0 = p_u^{I_1 \circ u}(i_1) \tag{100}$$

It is easy to see that the last term equals zero,

$$\begin{aligned}
&\int_{i_0 \times i_1} \frac{p_u^{I_0, I_1 \circ u}(i_0, i_1)}{p_u^{I_1 \circ u}(i_1)} \frac{\partial p_u^{I_1 \circ u}(i_1)}{\partial t} di_0 di_1 \\
&= \int_{i_1} \frac{p_u^{I_1 \circ u}(i_1)}{p_u^{I_1 \circ u}(i_1)} \frac{\partial p_u^{I_1 \circ u}(i_1)}{\partial t} di_1 \\
&= \frac{\partial}{\partial t} \int_{i_1} p_u^{I_1 \circ u}(i_1) di_1 \\
&= \frac{\partial}{\partial t} (1) \\
&= 0
\end{aligned} \tag{101}$$

Hence equation (99) can be written as the following,

$$\frac{\partial M_1}{\partial t} = - \int_{i_0 \times i_1} \left[1 + \log \frac{p_u^{I_0, I_1 \circ u}(i_0, i_1)}{p^{I_0}(i_0) p_u^{I_1 \circ u}(i_1)} \right] \frac{\partial p_u^{I_0, I_1 \circ u}(i_0, i_1)}{\partial t} di_0 di_1 \quad (102)$$

The probability density function (PDF) $p^{I_0}(i_0)$, $p_u^{I_1 \circ u}(i_1)$ can be estimated by one dimensional non-parametric Parzen-Rozenblatt density models [27]:

$$p^{I_0}(i_0) = \frac{1}{V} \int_{\Omega_0} \psi_{1D}(I_0(x) - i_0) dx, \quad (103)$$

and

$$p_u^{I_1 \circ u}(i_1) = \frac{1}{V} \int_{\Omega_0} \psi_{1D}(I_1(u(x)) - i_1) dx. \quad (104)$$

Here ψ_{1D} is a one dimensional Gaussian window, whose standard deviation can be chosen to be 10% of the standard deviation of i_0 or i_1 , respectively. V is the area of Ω_0 or number of pixels in Ω_0 in discrete case. From another point of view, this Parzen-window algorithm is equal to smoothing the histogram by a Gaussian filter.

In a similar way, $p_u^{I_0, I_1 \circ u}(i_0, i_1)$ can also be estimated using a two dimensional non-parametric Parzen-Rozenblatt density model as the follows,

$$p_u^{I_0, I_1 \circ u}(i_0, i_1) = \frac{1}{V} \int_{\Omega_0} \psi_{2D}(I_0(x) - i_0, I_1(u(x)) - i_1) dx. \quad (105)$$

in which ψ_{2D} is a two dimensional window whose covariance is decided by the covariance matrix of the paired random variables (i_0, i_1) . For simplicity, we write ψ_{2D} as ψ in the remaining part of this section.

Finally, the most difficult part $\frac{\partial}{\partial t} p_u^{I_0, I_1 \circ u}(i_0, i_1)$ in equation (102) can be computed as following. First, equation (105) can be rewritten as

$$p_u^{I_0, I_1 \circ u}(i_0, i_1) = \frac{1}{V} \int_{\Omega_0} \frac{1}{\mu_0(x)} \psi(I_0(x) - i_0, I_1(u(x)) - i_1) \mu_0(x) dx, \quad (106)$$

where the same trick of multiplying and dividing μ_0 has been applied. Then by setting $y = s^{-1}(x)$ (s^{-1} is an MP mapping from Ω_0 onto itself) and noticing $u(x) = u^0 \circ s^{-1}$, we get,

$$p_u^{I_0, I_1 \circ u}(i_0, i_1) = \frac{1}{V} \int_{\Omega_0} \frac{1}{\mu_0 \circ s(y)} \psi(I_0(s(y)) - i_0, I_1(u^0(y)) - i_1) \mu_0(y) dy \quad (107)$$

The derivative of $p_u^{I_0, I_1 \circ u}(i_0, i_1)$ is given as,

$$\begin{aligned} \frac{\partial p_u^{I_0, I_1 \circ u}}{\partial t}(i_0, i_1) &= \frac{1}{V} \int_{\Omega_0} < \frac{\psi_\alpha(I_0(s(y)) - i_0, I_1(u^0(y)) - i_1)}{\mu_0 \circ s(y)} \nabla I_0(s(y)) \\ &\quad - \frac{\psi(I_0(s(y)) - i_0, I_1(u^0(y)) - i_1)}{\mu_0^2 \circ s(y)} \nabla \mu_0(s(y)), \frac{\partial s(y)}{\partial t} > \mu_0(y) dy, \end{aligned} \quad (108)$$

in which ψ_α is the partial derivative of ψ with respect to its first component. Now, we do change of variable again by substituting y with $s^{-1}(x)$ and get,

$$\begin{aligned} \frac{dp_u^{I_0, I_1 \circ u}}{dt}(i_0, i_1) &= \frac{1}{V} \int < \frac{\psi_\alpha(I_0(x) - i_0, I_1(u(x)) - i_1)}{\mu_0(x)} \nabla I_0(x) \\ &\quad - \frac{\psi(I_0(x) - i_0, I_1(u(x)) - i_1)}{\mu_0^2(x)} \nabla \mu_0(x), \frac{\partial s}{\partial t} \circ s^{-1}(x) > \mu_0(x) dx \end{aligned} \quad (109)$$

Hence, the derivative of M_1 with respect to t is given by,

$$\begin{aligned} \frac{\partial M_1}{\partial t} &= - \int_{i_0 \times i_1} \int_{\Omega_0} < \frac{1}{V} \left[1 + \log \frac{p_u^{I_0, I_1 \circ u}(i_0, i_1)}{p^{I_0}(i_0) p^{I_1 \circ u}(i_1)} \right] \\ &\quad \left[\frac{\psi_\alpha(I_0(x) - i_0, I_1(u(x)) - i_1)}{\mu_0(x)} \nabla I_0(x) - \frac{\psi(I_0(x) - i_0, I_1(u(x)) - i_1)}{\mu_0^2(x)} \nabla \mu_0(x) \right], \\ &\quad \frac{\partial s}{\partial t} \circ s^{-1}(x) > \mu_0(x) dx di_0 di_1 \end{aligned} \quad (110)$$

Equation (110) is a quadruple integral, which could be rewritten in a concise format by using convolution,

$$\begin{aligned} \frac{\partial M_1}{\partial t} &= \int_{\Omega_0} < -\frac{1}{V} [(1 + \log \frac{p_u^{I_0, I_1 \circ u}}{p^{I_0} p^{I_1 \circ u}}) * \psi_\alpha(I_0(x), I_1 \circ u(x)) \frac{\nabla I_0(x)}{\mu_0(x)} \\ &\quad - (1 + \log \frac{p_u^{I_0, I_1 \circ u}}{p^{I_0} p^{I_1 \circ u}}) * \psi(I_0(x), I_1 \circ u(x)) \frac{\nabla \mu_0(x)}{\mu_0^2(x)}], \frac{\partial s}{\partial t} \circ s^{-1}(x) > \mu_0(x) dx \end{aligned} \quad (111)$$

Now it looks like a double integral on Ω_0 domain, and could be combined with the derivative from M_2 part. According to previous discussion for SSD as similarity measure in Subsection 4.1.1, P can be chosen as

$$\begin{aligned} P &= -\frac{1}{V} [(1 + \log \frac{p_u^{I_0, I_1 \circ u}}{p^{I_0} p^{I_1 \circ u}}) * \psi_\alpha(I_0(x), I_1 \circ u(x)) \frac{\nabla I_0(x)}{\mu_0(x)} \\ &\quad - (1 + \log \frac{p_u^{I_0, I_1 \circ u}}{p^{I_0} p^{I_1 \circ u}}) * \psi(I_0(x), I_1 \circ u(x)) \frac{\nabla \mu_0(x)}{\mu_0^2(x)}] + 2\alpha(u - \underline{id}) \end{aligned} \quad (112)$$

And as before, the evolution of u is determined by

$$u_t = -\frac{1}{\mu_0} Du \left(P - \nabla \Delta^{-1} \operatorname{div}(P) \right). \quad (113)$$

In 2D case, the evolution of u can be simplified as,

$$u_t = \frac{1}{\mu_0} Du \nabla^\perp \Delta^{-1} \operatorname{div}(P^\perp). \quad (114)$$

4.1.3 Image Morphing on a Doubly-connected Domain

The algorithms presented in the previous subsections are only capable of finding an optimal warping function between two single-connected domains, or more specifically two rectangular regions. Our assumption is that the mass is preserved on the whole domain. However, this is not always the case. Sometimes, the mass-preserving assumption is valid only between part of the two images. For example, we have two CT images from the same position of a heart. The first is a diastolic phase image and the other is a systolic phase image. During the cardiac cycle, the MP assumption is only valid on myocardium but invalid on ventricles since the volume of the blood varies from time to time. We may do image segmentation to separate the myocardium from the rest part of the image, and then find a warping function between them using MP assumption. Here we need to deal with the warping between two multi-connected domains. In this subsection, we will present an algorithm based on a Finite Element Method (FEM) to solve for the optimal MP mapping for a pair of doubly-connected domains, i.e. a pair of images with a hole in each of them.

In the approach for finding a MP mapping on two rectangular regions, there are two steps. The first step is to find an initial MP mapping u^0 from (Ω_0, μ_0) to (Ω_1, μ_1) , and the second step is to update this initial mapping iteratively to minimize an energy functional. For doubly-connected domains, once an initial MP mapping u^0 is found, the same updating algorithm (the second step) can be used to find the optimal warping function. Hence, the remain task is how to construct an initial MP

mapping that preserves mass between two doubly-connected domains. We cannot use the approach of solving a 1D MKP along one axis followed by a family of 1D MKPs in the other axis, as we did for rectangular domains. In this subsection, we present an algorithm for constructing an initial MP mapping u^0 by using the harmonic parametrization. The initial MP mapping u^0 is constructed by finding a 1D MKP along one harmonic coordinates followed by a family of 1D MKPs along the other harmonic coordinate.

- **Harmonic Parametrization**

In Chapter 2, we have presented an algorithm for flattening a multi-connected domain onto a plane with the angle-preserving property. The same technique can be applied here for generating a parametrization for a doubly-connected domain. The only difference is that there is no need for finding a “branch point”. Here we will only sketch the steps for construction an analytic function $f^c = u^c + iv^c$ for harmonic parametrization. Similar techniques has been applied for measuring tissue thickness [94]. For more details and numerical implementations, please read Section 2.5.

Assume we have a triangulated doubly-connected domain Σ , which has an inner boundary of σ_0 and an outer boundary of σ_1 as shown in Figure 35.

First, we want to construct u^c , which is the real part of f . It is assumed that u satisfies

$$\begin{aligned} \Delta u^c &= 0 \\ \text{with } u^c(\sigma_0) &= 0 \quad \text{and} \quad u^c(\sigma_1) = 1 \end{aligned} \tag{115}$$

The Laplace equation can be solved using an FEM presented in Section 2.5. And a cut C is then defined from σ_0 to σ_1 from an arbitrary point x_0 along the gradient of u^c to another point x_1 on σ_1 . The cut C and two original boundaries σ_0 and σ_1 form an oriented boundary B for the cut surface,

$$B : x_0 \xrightarrow{\sigma_0} x_0 \xrightarrow{C} x_1 \xrightarrow{\sigma_1} x_1 \xrightarrow{-C} x_0$$

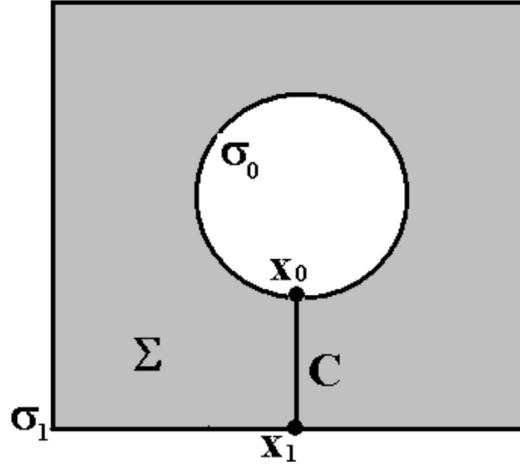


Figure 35: A doubly-connected domain Σ with two boundaries.

The boundary condition of the imaginary part v^c can be decided by equation (14),

$$v^c(\zeta) = \int_{\zeta_0}^{\zeta} \frac{\partial v}{\partial s} ds = \int_{\zeta_0}^{\zeta} \frac{\partial a}{\partial n} ds$$

according to the Cauchy-Riemann equations. And v^c inside the cut surface is given by the solution of $\Delta v^c = 0$.

Once the analytic function $f^c = u^c + iv^c$ is constructed, a harmonic coordinate system can be defined by using u^c as one axis and v^c as the other axis. Figure 36 shows such a parametrization on a heart CT image without involving the ventricle area.

- **Finding the Initial Mapping u^0**

By doing the harmonic parametrization, the first doubly-connected domain (Ω_0, μ_0) is cut and mapped onto a rectangle region (Ω_0^c, μ_0^c) via a conformal mapping $f_0^c = u_0^c + iv_0^c$. μ_0^c is given by

$$\mu_0^c = |Df_0^c|^{-1} \mu_0, \quad (116)$$

which guarantees the mapping from Ω_0 to Ω_0^c is not only angle-preserving, but

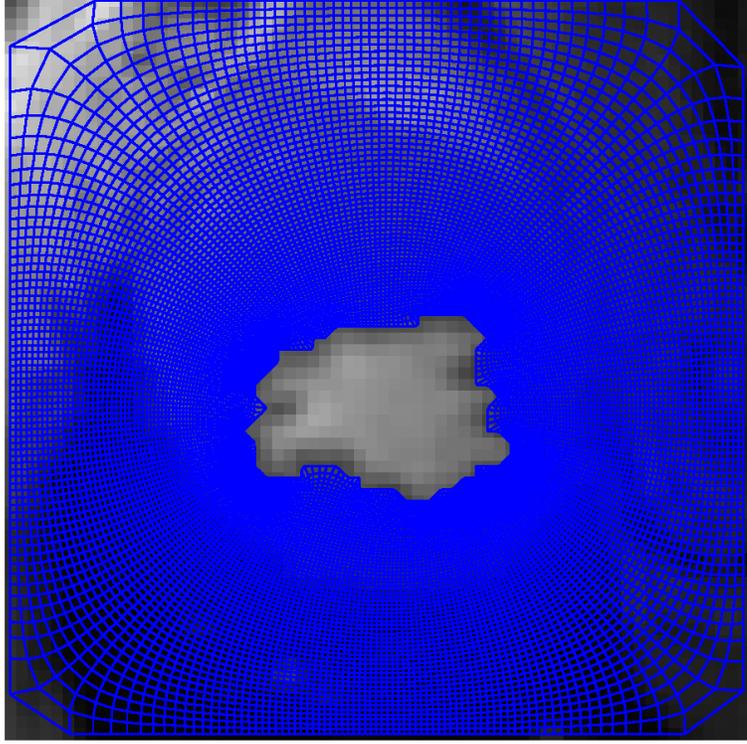


Figure 36: Harmonic parametrization of a heart image

also mass-preserving (which is not area-preserving). Similarly, the second doubly-connected domain (Ω_1, μ_1) is mapped onto another rectangle region (Ω_1^c, μ_1^c) via $f_1^c = u_1^c + iv_1^c$. And μ_1^c is given by

$$\mu_1^c = |Df_1^c|^{-1} \mu_1. \quad (117)$$

Then the remain task is to find an MP mapping from (Ω_0^c, μ_0^c) to (Ω_1^c, μ_1^c) . Since Ω_0 and Ω_1 are rectangles now, we can use the algorithm presented in Section 3.1 to find an MP mapping u_{ini} between them. The whole process can be illustrated by Figure 37.

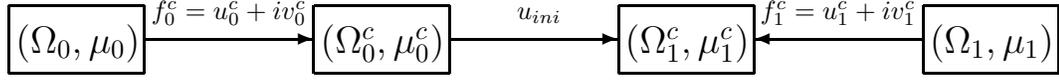


Figure 37: The diagram of constructing u^0 between two doubly-connected domains.

And the resulting initial mapping u^0 is the composition f_0^c , u_{ini} and $(f_1^c)^{-1}$ such that

$$u^0 = (f_1^c)^{-1} \circ u_{ini} \circ f_0^c. \quad (118)$$

Since f_0^c , f_1^c and u_{ini} are all MP mapping, u^0 is also an MP mapping, due to the facts that the composition of two MP mappings is an MP mapping and the inverse of an MP mapping is also an MP mapping.

- **Finding the minimizer \tilde{u}**

The equation for evolution of u is the same as for rectangular regions. An FEM is used to solve the Poisson equation on an irregular domain. More numerical details will be provided in the next section.

4.2 Numerical Implementation

In this section, we will discuss some numerical details in solving for the optimal mass transportation problem. For 2D rectangular regions, a finite difference method is applied. And for doubly-connected domains, a finite element based method is applied.

4.2.1 Rectangular Regions

In order to solve for the Poisson equation (83) on a rectangular region, we use Matlab solver *poicalc*, which is a Fourier transform based algorithm. The solution is obtained by sine transforms in one direction and tridiagonal matrix solution in the other direction. For the evolution equation of u (equation (86)), we use an upwind scheme for computing Du and standard centered differences for the other spatial derivatives. The time step dt can be chosen to be less than

$$\min_{x,i} \left| \frac{1}{\mu_0} (\nabla^\perp \Delta^{-1} \operatorname{div}(P^\perp))_i \right|^{-1}.$$

Once we have numerically solved the right hand side of equation (86), we can use the result to update u . The optimal map is obtained as $t \rightarrow \infty$. In practice, we iterate until convergence with respect to a specified tolerance.

The complexity of the method during each iteration scales is $N \log_2 N$, where N is the number of pixels in the image, if fast sine transforms are used.

4.2.2 Doubly-connected Regions

We use an FEM method to solve for the Poisson equation (83) on an triangulated irregular region $PL(\Sigma)$ with an inner boundary σ_0 and an outer boundary σ_1 . We define a basis $\{\phi_V\}$ on the domain as we did in Section 2.5.

According to the FEM theory, f should satisfy

$$\int \int \Delta f \cdot v = \int \int -\operatorname{div}(P^\perp) \cdot v \quad (119)$$

for any testing function v satisfying $v = 0$ on all boundaries. For convenience, basis functions ϕ_V are used as testing function, for $V \in \Sigma \setminus (\sigma_0 \cup \sigma_1)$. By doing integration by parts, we have,

$$\int \int \nabla f \cdot \nabla v = \int \int \operatorname{div}(P^\perp) \cdot v. \quad (120)$$

Considering the zero boundary condition of f , f can be approximated as

$$f = \sum_{W \in \Sigma \setminus (\sigma_0 \cup \sigma_1)} f_W \phi_W. \quad (121)$$

Then equation (120) can be written as

$$\sum_{W \in \Sigma \setminus (\sigma_0 \cup \sigma_1)} D_{VW} f_W = \sum_{W \in \Sigma \setminus (\sigma_0 \cup \sigma_1)} Q_{VW}. \quad (122)$$

Where D_{VW} is decided by equation (24). According to the second order approximation [90], when $V = W$, Q_{VW} can be decided by

$$Q_{WW} = \frac{1}{6} \operatorname{div}(P_W^\perp) \sum_i A_i, \quad (123)$$

where A is the area of a triangle, and the sum is over all triangles containing W .

When V and W are neighbors, Q_{VW} is given by

$$Q_{VW} = \frac{1}{12} \operatorname{div}(P_V^\perp)(A_1 + A_2), \quad (124)$$

where A_1 and A_2 are the area of two triangles containing both V and W . $Q_{VW} = 0$ if V and W are not neighbors.

In the evolution equation of u (equation (86)), we use an upwind scheme for computing Du . For all other derivatives, we use a least mean square method for deciding the spacial derivatives. For example, assume that a given point (x_0, y_0) has N neighbors $(x_i, y_i), i = 1 \dots N$, and a function f is defined such that $f(x_i, y_i) = f_i$ for $i = 0 \dots N$. It is easy to see that the derivatives of f should satisfy

$$\begin{pmatrix} f_1 - f_0 \\ \dots \\ f_N - f_0 \end{pmatrix} = \begin{pmatrix} x_1 - x_0, & y_1 - y_0 \\ \dots \\ x_N - x_0, & y_N - y_0 \end{pmatrix} \begin{pmatrix} f_x \\ f_y \end{pmatrix} \quad (125)$$

According to the least square method, the derivatives of f are given by

$$\begin{pmatrix} f_x \\ f_y \end{pmatrix} = (A^T A)^{-1} A^T \begin{pmatrix} f_1 - f_0 \\ \dots \\ f_N - f_0 \end{pmatrix}, \quad (126)$$

where A is the position difference matrix given by

$$A = \begin{pmatrix} x_1 - x_0, & y_1 - y_0 \\ \dots \\ x_N - x_0, & y_N - y_0 \end{pmatrix} \quad (127)$$

4.3 Examples

We now demonstrate our image interpolation method with some examples. These examples can be divided into three categories: synthetic imagery, natural (color) imagery, and medical imagery. In each example, we have two given images. One of them is used as the starting image and the other as the ending image. Then an optimal MP mapping minimizing the defined energy functional is found using the gradient descent algorithm.

Once the optimal mapping \tilde{u} is solved, the remaining task for the image morphing problem is to generate a sequence of in-between images $I(t)$, such that $I(0) = I_0$ and $I(1) = I_1$. It is assumed that when the time t varies from 0 to 1, the starting image I_0 continuously changes to the ending image I_1 . We further require that the same transition rate is applied to all points on the in-between images. Hence image warping map at any time t ($t \in [0, 1]$) is simply given by

$$X^t(x) = (1 - t)x + t\tilde{u}(x), \quad (128)$$

The corresponding cross-dissolved image at time t is given by

$$I^t(X^t(x)) = (1 - t)I_0(x) + tI_1(u(x)). \quad (129)$$

I_0 and I_1 could also be color images and then equation (129) is applied to three color components separately.

The warp function (128) guarantees the continuous transformation of the source image to target image, and t is the transition rate. One can always guarantee that the intermediate frames are mass-preserving simply by shading the pixels in the in-between images according to $|D(X^t)^{-1}| \mu_0 \circ (X^t)^{-1}$.

4.3.1 Circle to triangle example

The first example is based on a pair of synthetic images. The first image is a solid white circle, as shown in Figure 38 (a). And the second image is a solid triangle, as

shown in Figure 38 (f). Both of them are on a black background. Figure 38 (b) to (e) show the in-between images from (a) to (f) at pseudo time $t = 0.2, 0.4, 0.6$ and 0.8 , respectively. The energy functional used is a pure L^2 MKP, i.e. no comparison term added.

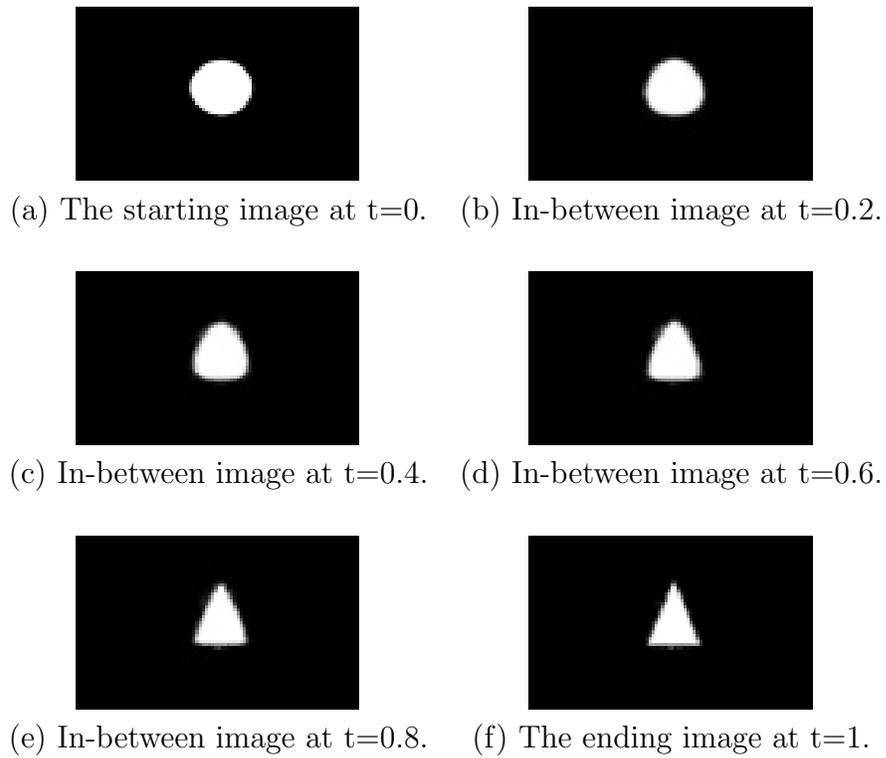


Figure 38: Image morphing from a circle to a triangle

4.3.2 Cloud example

The second example is an image morphing example between two cloud images. Figures 39 and 44 are the starting and ending images, respectively. Figures 40 to 43 are generated in-between images by using SSD as the comparison term. Figure 45 shows the optimal deformed grid between the two images.



Figure 39: The starting *cloud* image

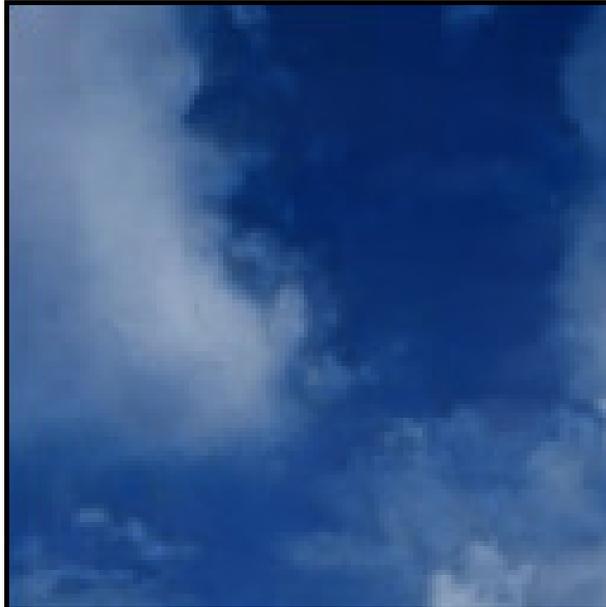


Figure 40: In-between *cloud* image at $t=0.2$.



Figure 41: In-between *cloud* image at $t=0.4$.

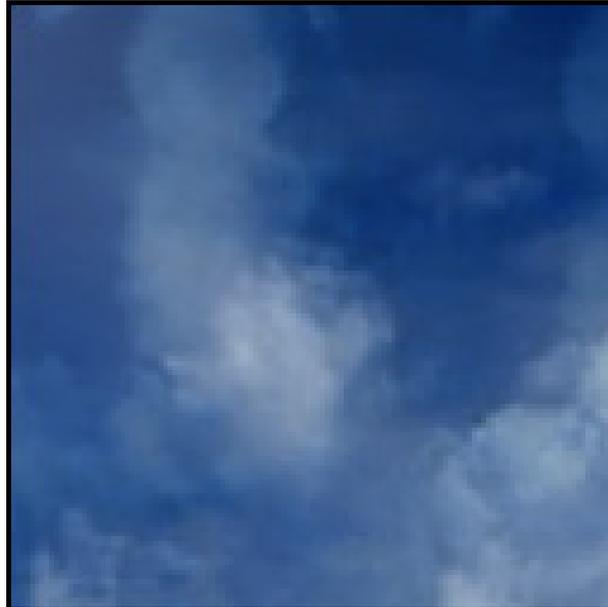


Figure 42: In-between *cloud* image at $t=0.6$.

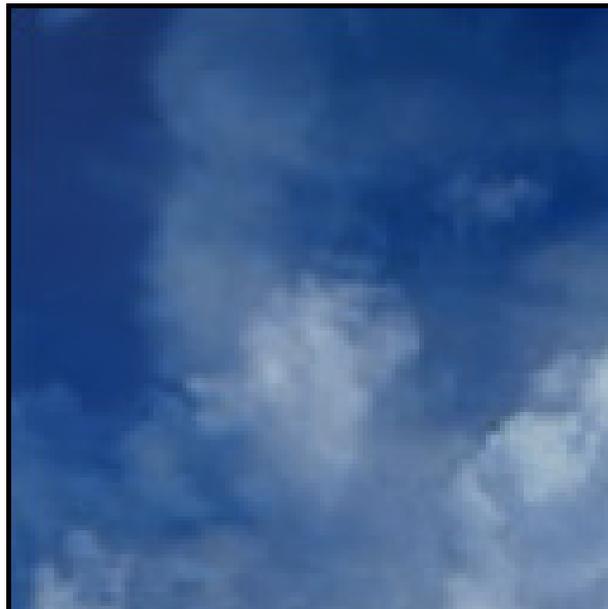


Figure 43: In-between *cloud* image at $t=0.8$.

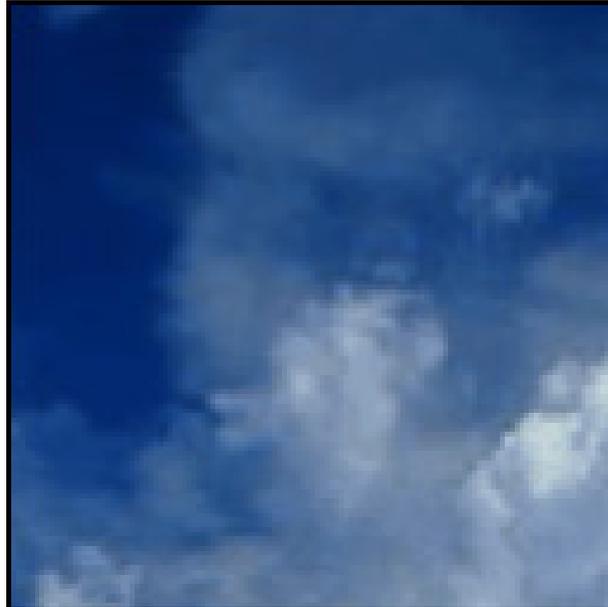


Figure 44: The ending *cloud* image.

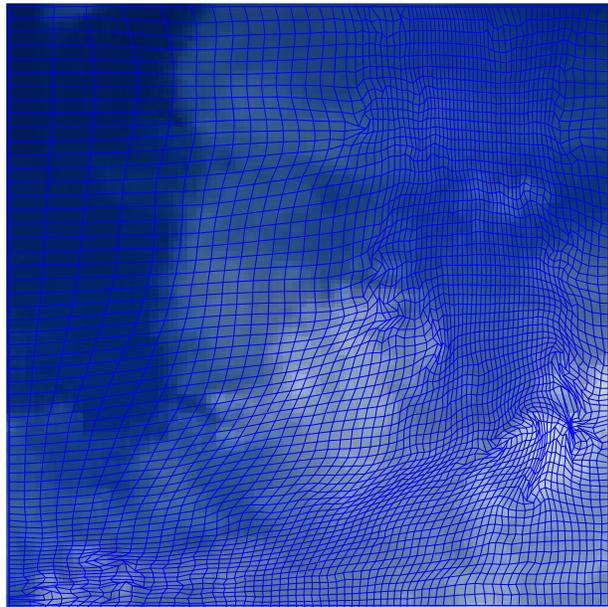


Figure 45: The deformed grid in the *cloud* example.

4.3.3 Flame example

The third example is the image morphing between two *flame* images. The starting image (Figure 46(a)) and the ending image (Figure 46(b)) are from a *flame* video sequence in the *Artbeats Digital Film Library* (www.artbeats.com). The two images are the 24th and the 29th frames of the video, respectively. Considering the frame rate of 30fps, there is about 0.2 second between the two frames in the original sequence. We will use this example to show the effect of comparison term and how to choose the parameter α which control the trade off between the intensity disparity and the moving distance.

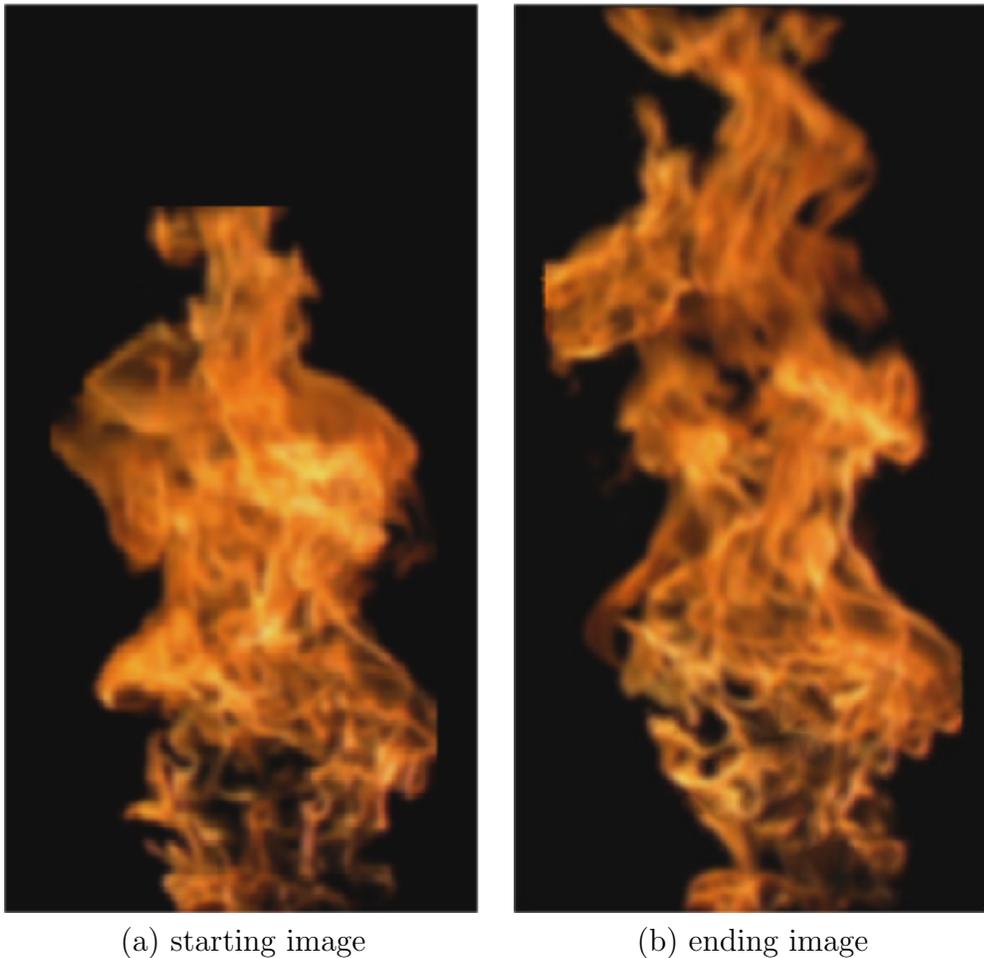
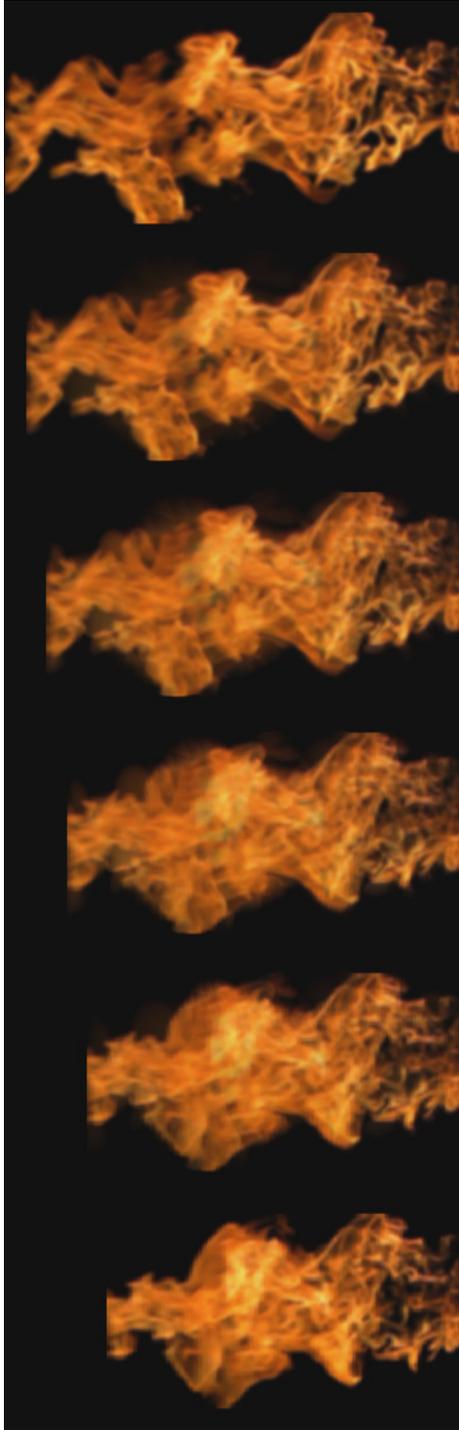


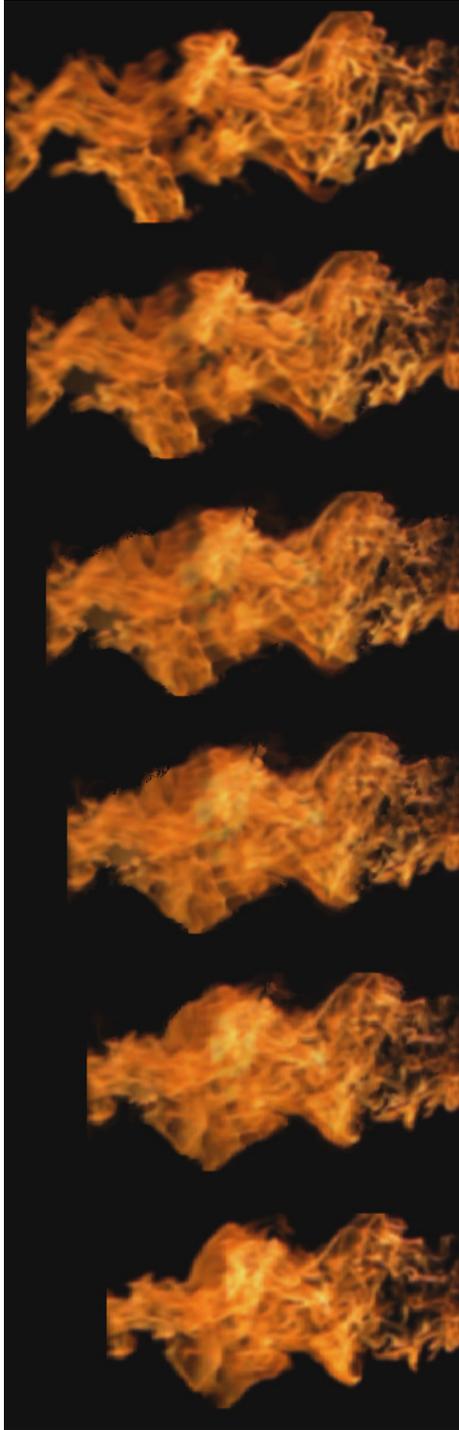
Figure 46: Two given *flame* images

Figure 47 shows the result of implementing pure L^2 MKP without a comparison term. Figures 48 and 49 are the results of SSD and MI as the comparison term, respectively. All the figures are rotated by 90 degrees counter-clockwise. Actually, the result from pure L^2 MKP is used as the initial mapping u^0 for Figures 48 and 49. It has been pointed out before that adding a comparison term in the energy functional leads to some curl remaining in the final deformed grid. If the algorithm starts with an initial mapping as constructed by equations (35) through (37), the remaining curl is mostly in the “bright” region of the image and it will cause unnatural effects (including spacial folding) when we generate in-between images. By starting with a curl-free initial mapping, the algorithm makes corrections mainly close to the edges between high intensity regions and low intensity regions. Thus the remaining curl is mostly in the “dark” region of the image and it is unnoticeable (since a small high intensity region is mapped to a large low intensity region).



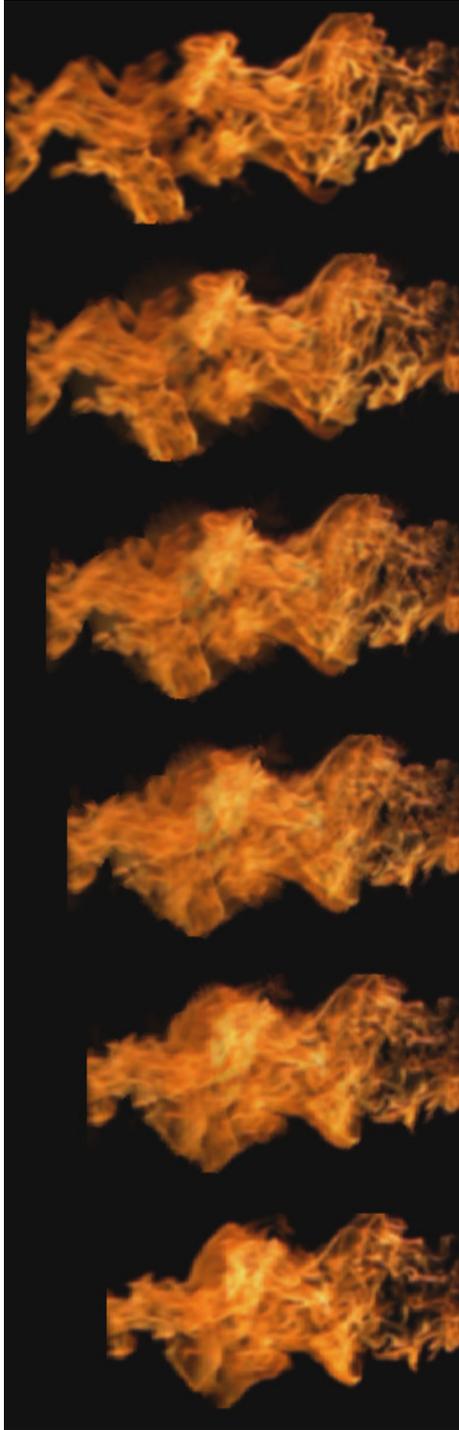
($t=0, 0.2, 0.4, 0.6, 0.8$ and 1)

Figure 47: Interpolation of *flame* images without comparison term.



($t=0, 0.2, 0.4, 0.6, 0.8$ and 1)

Figure 48: Interpolation of *flame* images using SSD as the comparison term.



($t=0, 0.2, 0.4, 0.6, 0.8$ and 1)

Figure 49: Interpolation of *flame* images using MI as the comparison term.

Figure 50 shows the comparison of three different methods at time $t = 0.5$. In pure L^2 MKP algorithm (Figure 50(a)), some *fade in* and *fade out* effects can be seen. These undesired effects are due to the mapping of some high intensity regions on low intensity regions or vice versa. By adding a SSD comparison term, these undesired effects have been reduced as seen in Figure 50(b), or in Figure 50(c), when MI is used as the comparison term. Basically, the effects of SSD and MI as the comparison terms are very similar, since both of the two images come from the same imaging modality. The strength of SSD is that it can reduce the *fade in* and *fade out* effects a little better than MI. However, it also causes some broken effects at some edges between the bright and dark regions, due to the high value of curl remaining in these regions. Hence, there is still a trade off between reducing fading effects and reducing curl. These effects are more obvious if seen in dynamic. Figures 51 to 53 show the deformed grid in these three cases.

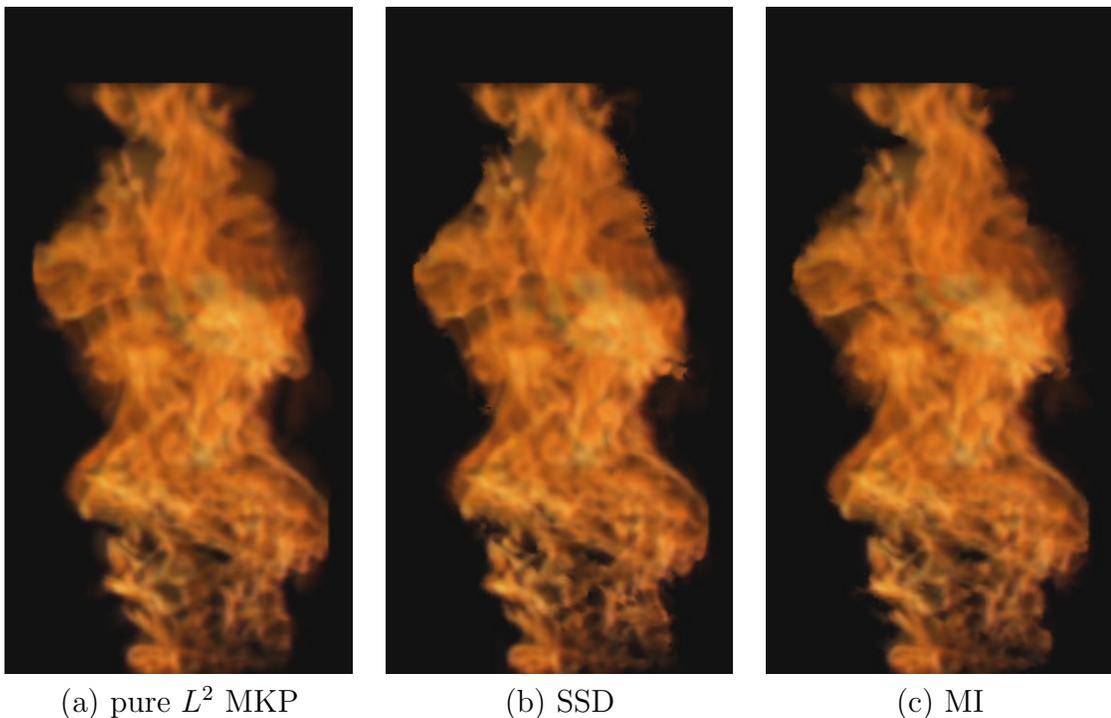


Figure 50: Comparison of pure L^2 MKP, SSD, and MI at $t = 0.5$.

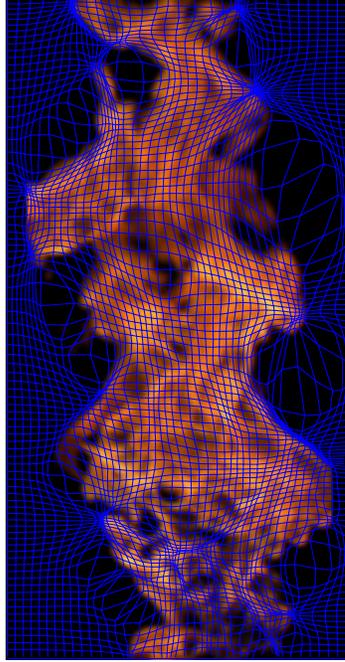


Figure 51: The deformed grid in the *flame* example without comparison term.

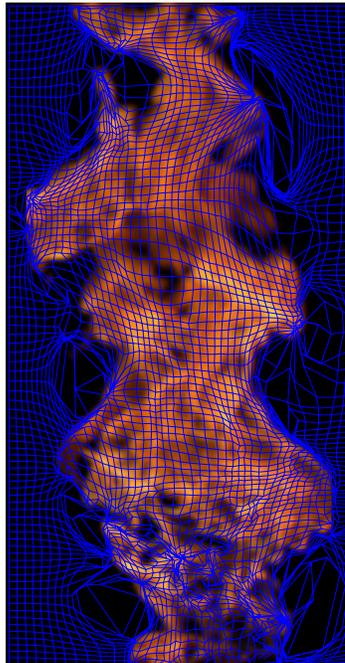


Figure 52: The deformed grid in the *flame* example with SSD similarity measure.

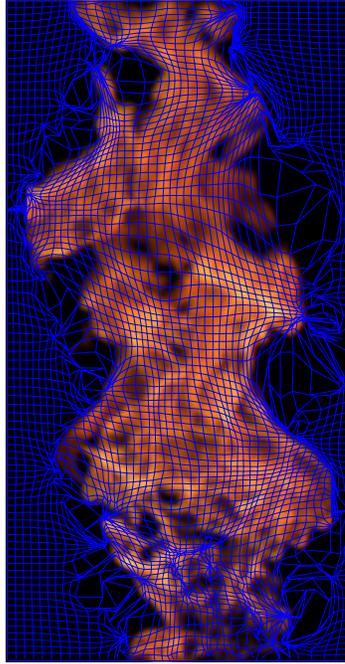


Figure 53: The deformed grid in the *flame* example with MI similarity measure.

Finally in this example, we will discuss how to choose the parameter α that controls the trade-off between preserving the intensity and minimizing mass moving. Generally speaking, there are no specific rules for choosing α , and the best way is to try different parameters and see which works the best. However, we find that it may be useful to plot the initial “forces” from the two terms separately and make them comparable before evolving u . For example, in the case of SSD as the comparison term, we may define P_1 as

$$P_1 = \frac{1}{\mu_0^2}(I_1 \circ u - I_0)^2 \nabla \mu_0 + \frac{2}{\mu_0}(I_1 \circ u - I_0) \nabla I_0,$$

and P_2 as

$$P_2 = 2\alpha(u - \underline{id}).$$

Using the method presented in this chapter, f_1 is solved as:

$$f_1 = -\frac{1}{\mu_0} \nabla^\perp \Delta^{-1} \operatorname{div}(P_1^\perp),$$

and f_2 is solved as:

$$f_2 = -\frac{1}{\mu_0} \nabla^\perp \Delta^{-1} \operatorname{div}(P_2^\perp).$$

$f_1 + f_2$ is the velocity vector field used to update u . The magnitude of f_1 is plotted as Figure 54(a), and its histogram is shown in Figure 54(b). We can see for most pixels $|f_1|$ is below 100. If α is set to be 15, similar graph can be plotted for $|f_2|$ as shown in Figure 54(c) and (d). From Figure 54(b) and (d), it is clear that by setting $\alpha = 15$, f_1 and f_2 has the same order of magnitude.

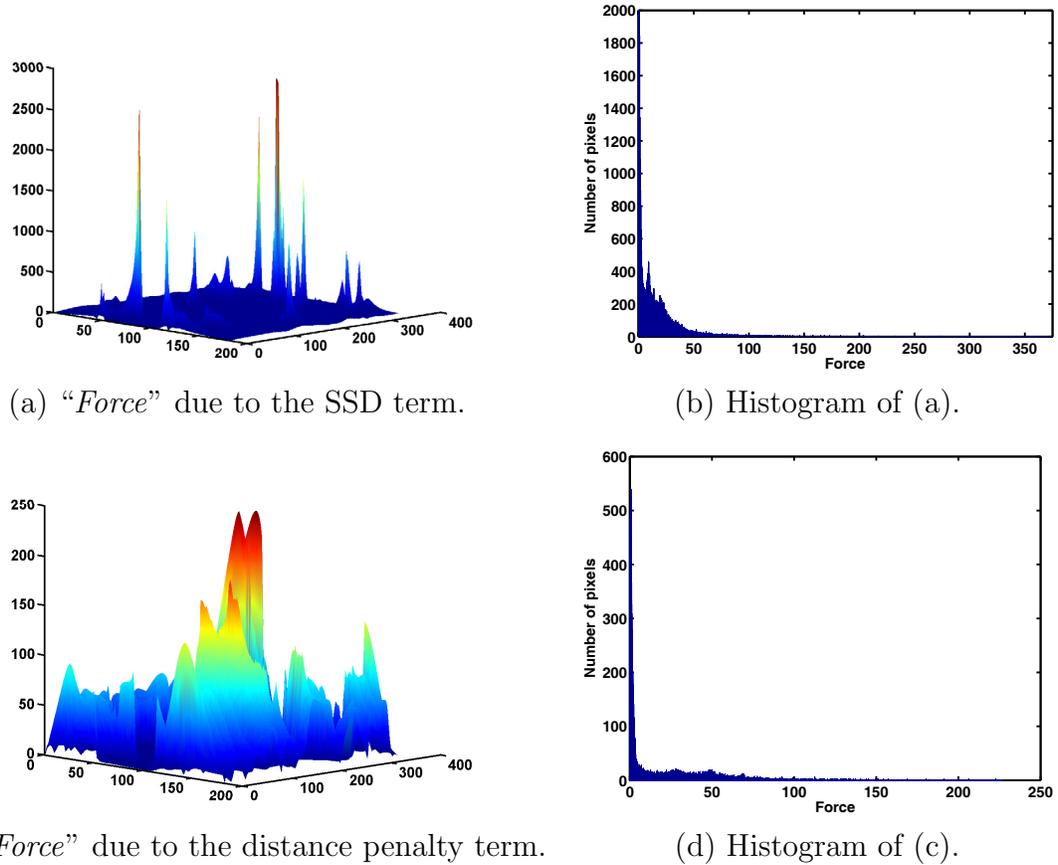


Figure 54: Decision of α .

4.3.4 The first brain example

This example is an image morphing between two T1 MRI brain images taken from a patient at the same position but at different times. Figure 55 is taken before the open-head surgery and Figure 60 is taken during the surgery after opening the head. Due to the pressure change, the ventricles in the brain are compressed. The two arrows in Figure 55 show the place where the brain has the largest deformation. Figures 56 to 59 show the synthetic process of the deformation. Figure 61 shows the deformed grid by using SSD as the similarity measure.

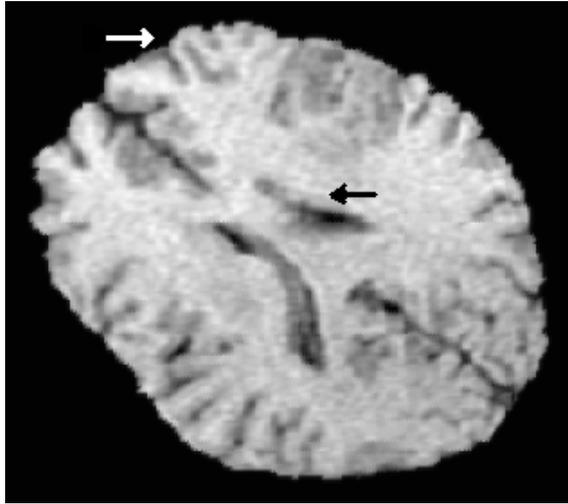


Figure 55: The starting T1 brain image ($t = 0$).

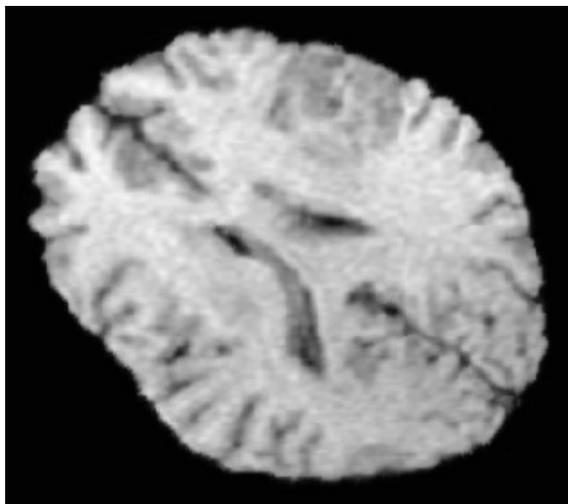


Figure 56: Interpolated T1 brain image ($t = 0.2$).

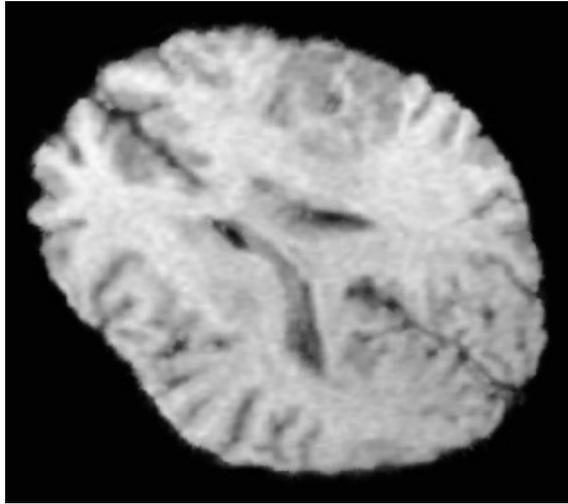


Figure 57: Interpolated T1 brain image ($t = 0.4$).

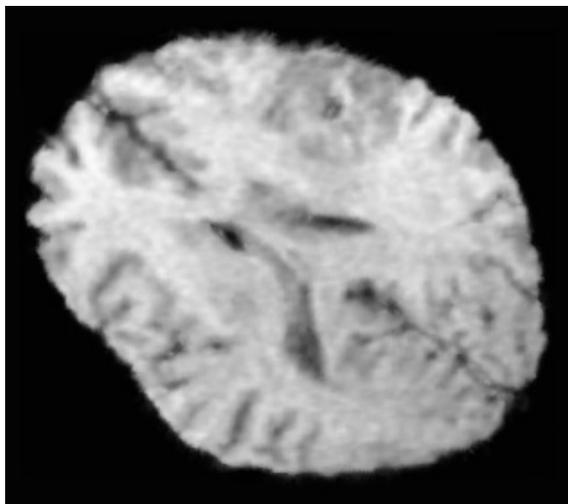


Figure 58: Interpolated T1 brain image ($t = 0.6$).

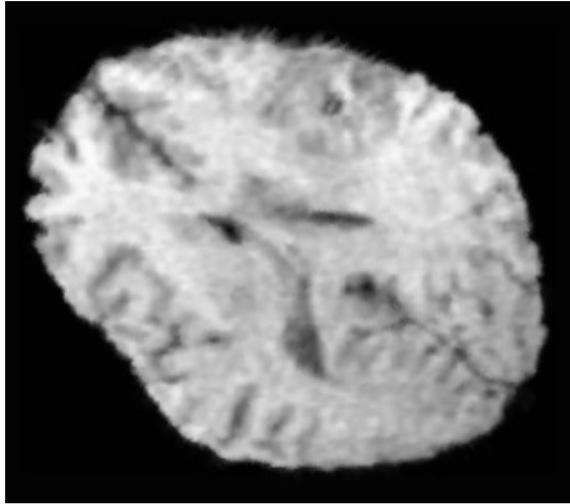


Figure 59: Interpolated T1 brain image ($t = 0.8$).

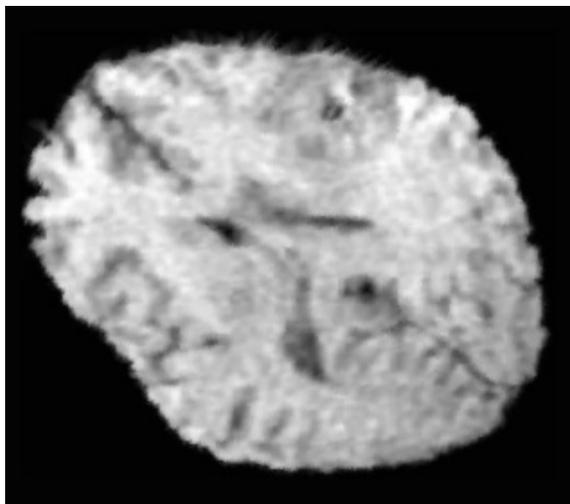


Figure 60: The ending T1 brain image ($t = 1$).

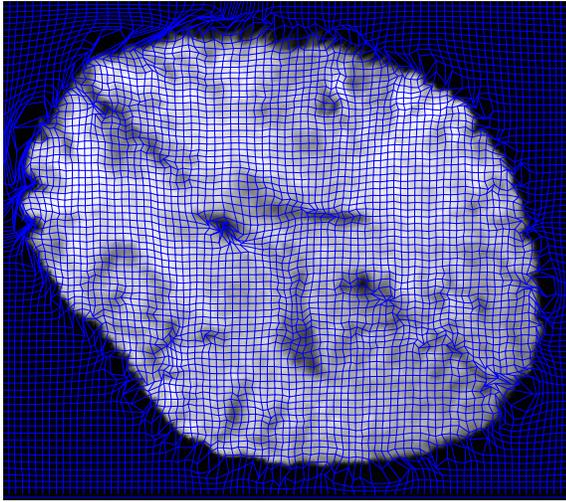


Figure 61: The deformed grid in the *brain* example.

4.3.5 The second brain example

This example is similar to the previous one. Figure 62 is a pre-operative T1 brain MRI image, and Figure 67 is another T1 brain MRI image from the same patient during surgery, after craniotomy and opening of the dura. Both of the two images are from <http://www.sop.inria.fr/epidaure/personnel/Olivier.Clatz>. This example is similar to the previous one. The only difference is that the skull is preserved in this example. We first did segmentation on the images to separate the brain tissue from the skull. Then we do image morphings on the brain and skull individually. Finally the two sequences are put together as shown in Figures 63 to 66.

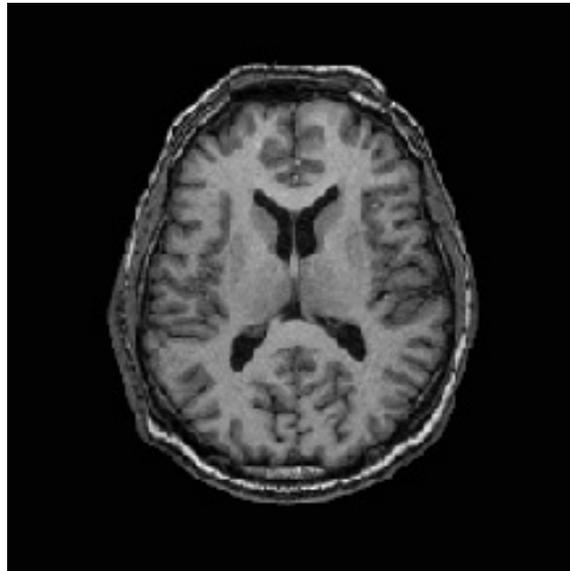


Figure 62: *Second brain example* ($t = 0$).

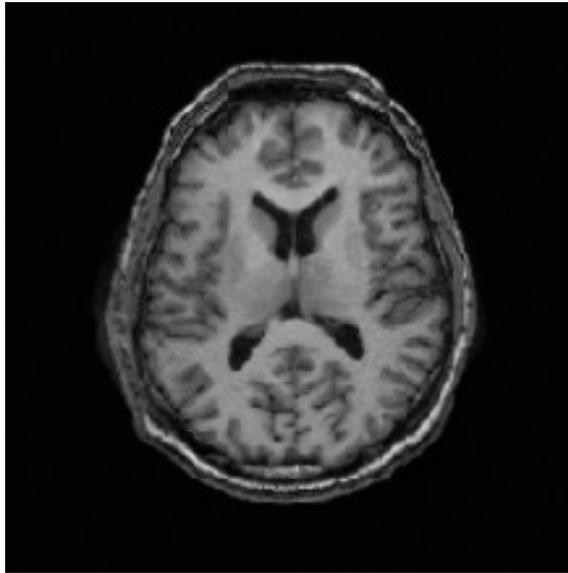


Figure 63: *Second brain example* ($t = 0.2$).

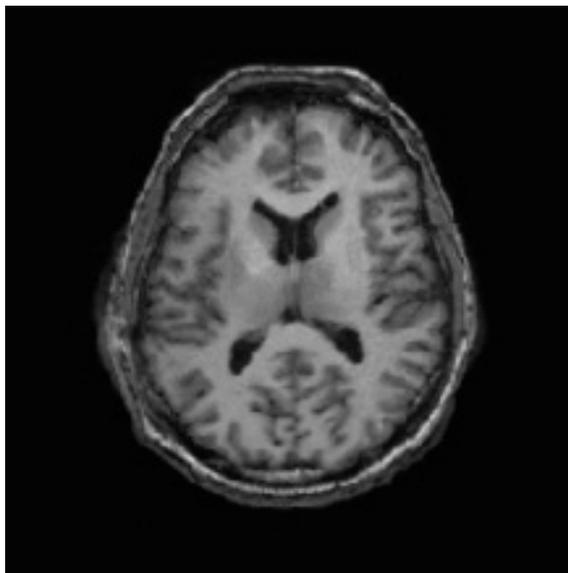


Figure 64: *Second brain example* ($t = 0.4$).

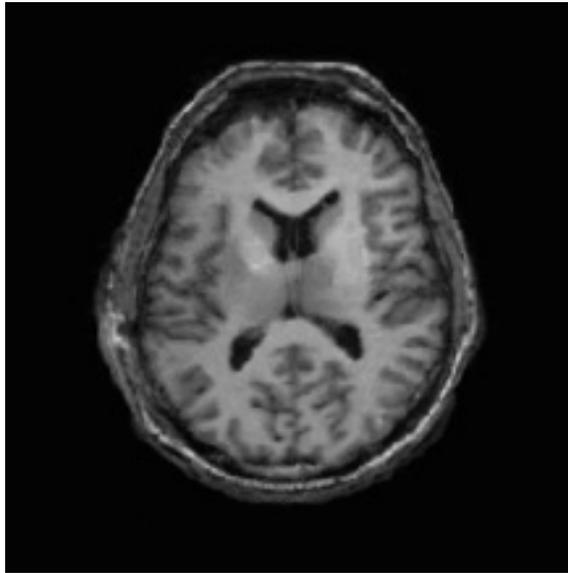


Figure 65: *Second brain example* ($t = 0.6$).

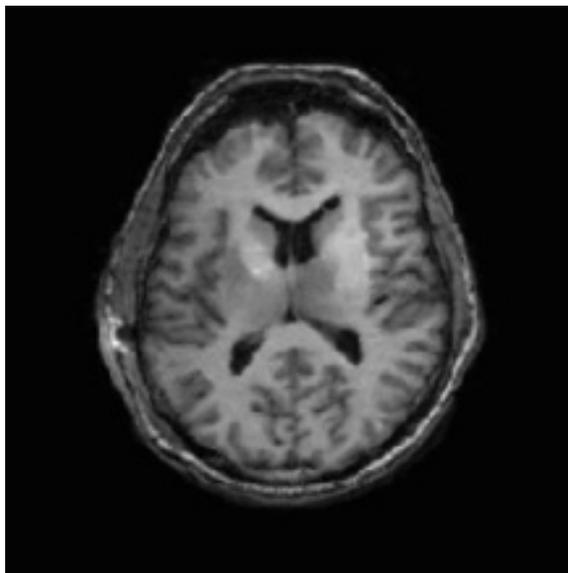


Figure 66: *Second brain example* ($t = 0.8$).

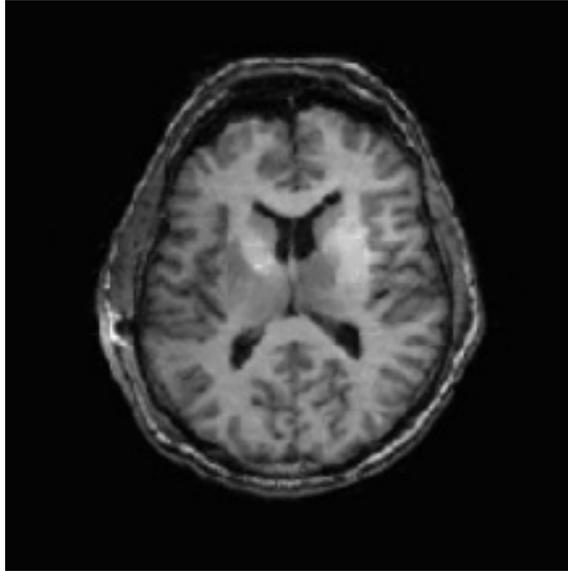


Figure 67: *Second brain example* ($t = 1.0$).

4.3.6 Heart example

This last example is for image morphing on a doubly-connected domain. Two CT images of a heart at the same position are given. Figure 71 is a diastolic image and Figure 76 is a systolic image. The black regions in Figures 68 and 69 define two multi-connected domains, corresponding to the heart muscle and other tissues that satisfy the mass-preserving assumption. Harmonic parametrization is done on each domain (as shown in Figure 36 for the diastolic image), and an FEM-based L^2 MKP is solved between the two domains to find the correspondence. Figure 70 shows the deformed grid. Figures 72 to 75 show the in-between images generated.

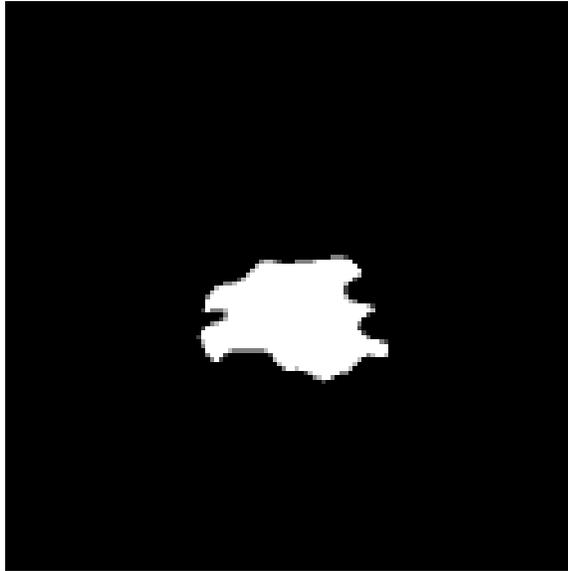


Figure 68: The mask for starting *heart* image

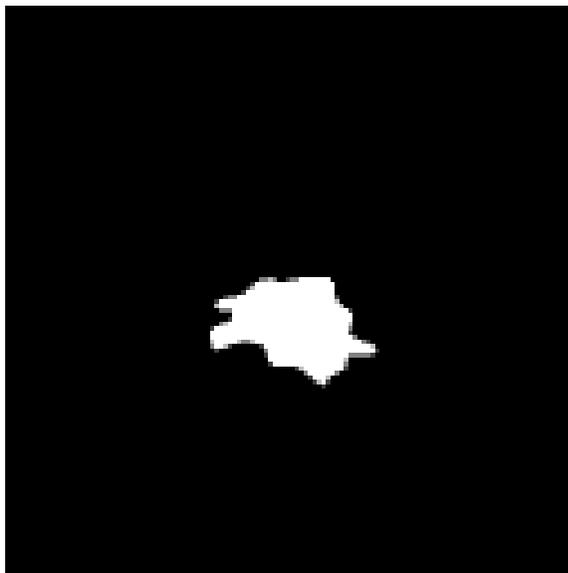


Figure 69: The mask for ending *heart* image

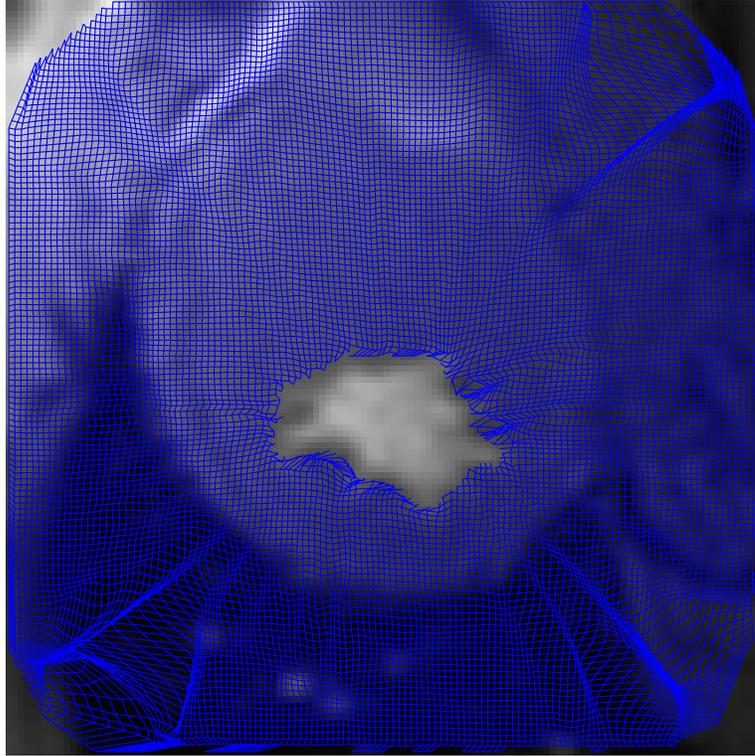


Figure 70: The deformed grid on the systolic heart image.

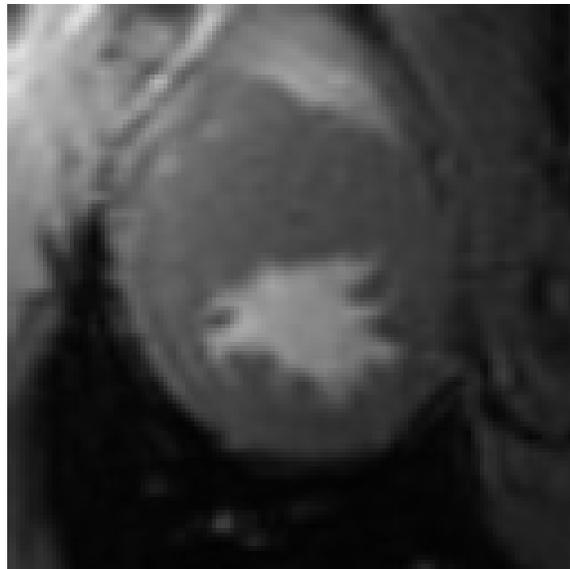


Figure 71: The starting heart image ($t = 0$)

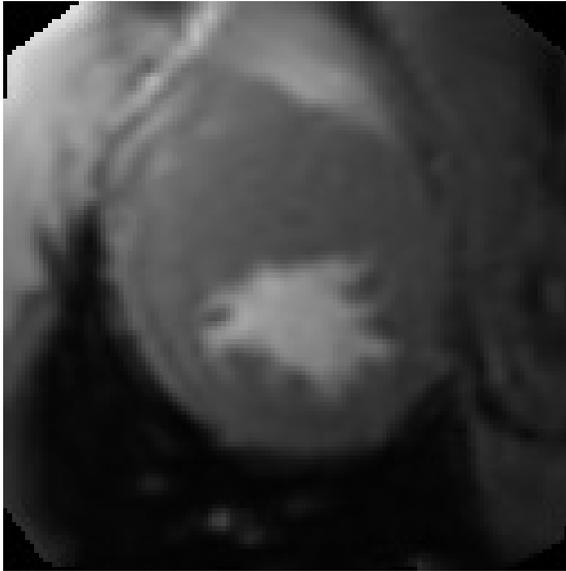


Figure 72: In-between *heart* image ($t = 0.2$)

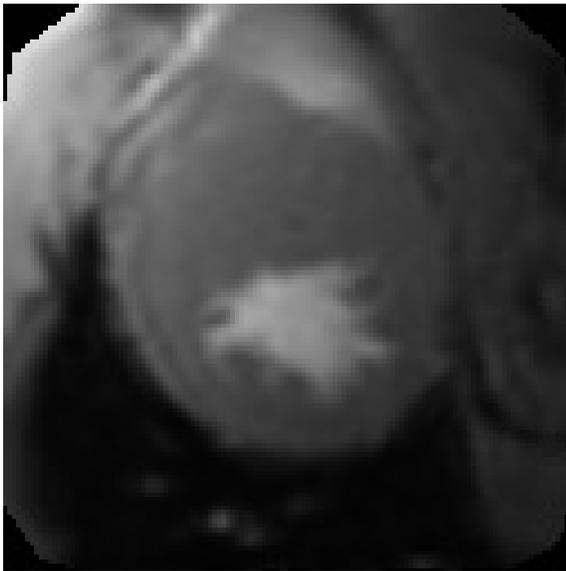


Figure 73: In-between *heart* image ($t = 0.4$)

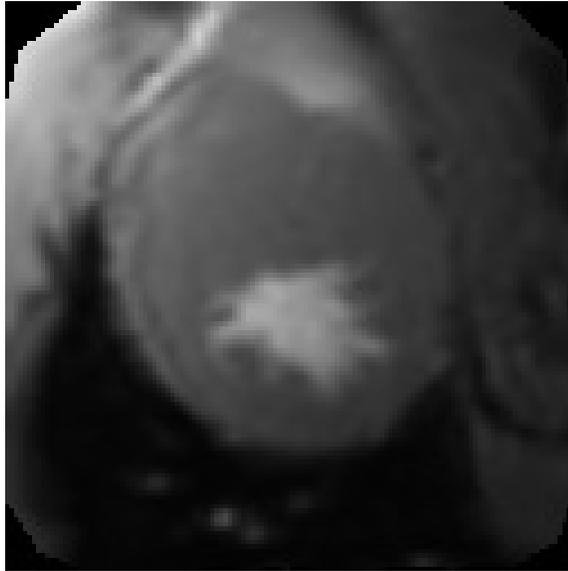


Figure 74: In-between *heart* image ($t = 0.6$)

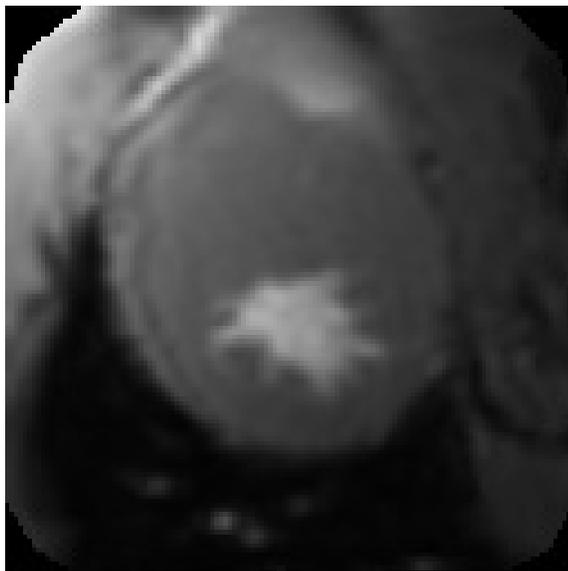


Figure 75: In-between *heart* image ($t = 0.8$)

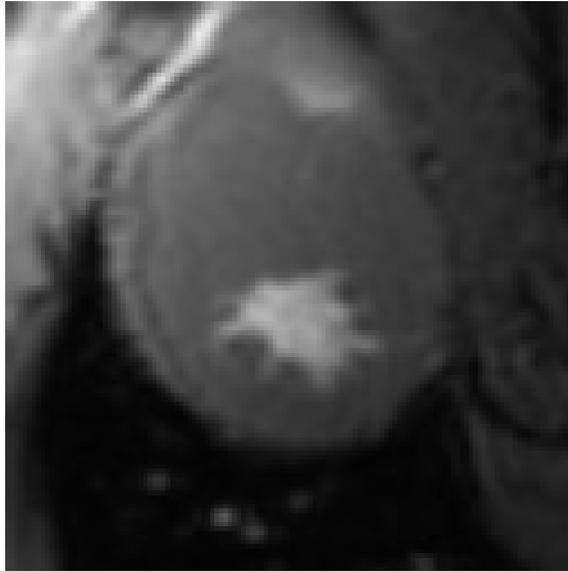


Figure 76: The ending heart image ($t = 1.0$)

CHAPTER 5

CONCLUSIONS AND FUTURE RESEARCH

Conformal mappings are a type of angle preserving mapping. They are often described as being similarities in the small, since the elements of the first fundamental form (E, F, G) are transformed as $(\rho E, \rho F, \rho G)$ with ρ depending on the point on the surface. When a conformal mapping is applied onto a brain surface, which topologically is a genus zero surface without any holes or self intersections, the brain surface can be mapped conformally onto a sphere and any local portion thereof onto a disc [3]. When a conformal mapping is applied onto a colon surface, which is a doubly connected domain or a half surface of a genus one surface, the colon surface can be mapped conformally onto a rectangle region [36]. In this thesis, a conformal mapping approach for flattening multi-branched medical surfaces has been presented. A multi-branched open-ended surface, for example a vessel, can be flattened onto a polygonal region while preserving angles on the surface except for a few *branch-points*. The flattening function is obtained as the solution of two second-order elliptic partial differential equations on the surface to be flattened. For triangulated surfaces, there exists a powerful and reliable finite element approach to numerically approximate the flattening function. We have applied this algorithm on several vessel datasets coming from different parts of the body. We have also discussed the construction of the *harmonic skeleton* and some useful applications of this method. The applications include the extraction of central line and camera directions for fly-throughs, and cross-sectional area measurement. Conformal mapping also provides a useful tool for parametrization as we demonstrated in Chapter 4 when finding the correspondence between two doubly-connected regions.

Optimal mass transportation is a well-studied theory in many fields and it has recently been employed within the context of imaging applications, such as content-based image retrieval [75, 76]. In this thesis, we applied it on the result of conformal mapping to restore the area with minimum distortion of the shape, in the L^2 mass moving sense. The algorithm defines a pseudo-mass density μ_0 according to the area distortion in the conformal mapping and then flattens it to unity. We have also applied the Monge-Kantorovich flow on the problem of image morphing by adding a comparison term. This technique can also be applied to the problem of medical image registration, if the underlining physics model satisfies the mass preserving assumption. It will be more accurate if we can specify corresponding landmarks. In Chapter 4 we have solved the L^2 MKP on doubly-connected domains, in which the inner boundary could be regarded as a single landmark if the radius of the boundary is small enough. This technique can also be extended into multi-connected domains (corresponding to multiple landmarks). As we have seen the main issue of extending the algorithm from a single-connected domain into a doubly-connected domain is how to define an initial MP mapping u^0 . The same problem arises here in finding an optimal MP mapping on a multi-connected domain.

For example, assume we have two multi-connected domains (Ω_0, μ_0) and (Ω_1, μ_1) with equal total mass. We may further assume there are two holes on each domain, without loss of generality. Ω_0 is divided into three regions, which are named P , Q and R , respectively. P and Q share only one common point x_0 . Similarly, Ω_1 is also divided into three regions named as P' , Q' and R' , respectively. And P' and Q' have only one common point x_1 . It is further required that region P and P' have the same total amount of mass, and the same rule applies to Q and Q' . Then we can define initial MP mappings u_P^0 from P to P' , u_Q^0 from Q to Q' , and u_R^0 from R to R' . u_P^0 , u_Q^0 and u_R^0 are all MP mappings from a double connected domain onto another, and can be constructed using the harmonic parametrization algorithm previously presented.

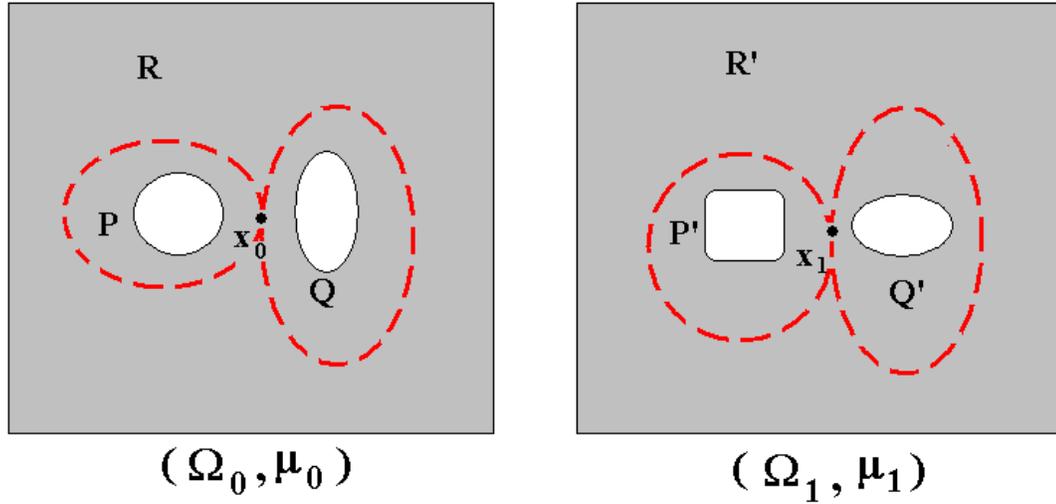


Figure 77: Two multi-connected domains

Then the combination of these three mappings is an initial MP mapping from Ω_0 to Ω_1 .

In solving the MKP, we use a gradient descent algorithm, which involves solving a Poisson equation in every iteration. This is a global approach, which is very time consuming. We may use some local approaches to replace it. For example, we may use a local flow in the form of

$$u_t = -\frac{1}{\mu_0} Du \nabla^\perp \operatorname{div}(u^\perp), \quad (130)$$

in the \mathbf{R}^2 case.

The L^2 mass moving penalty implemented in this thesis sometimes is too severe, in that it tends to favor change in density over moving mass around. In fact, this is the main reason for the fading effects we have seen in image morphing applications. L^1 type mass moving penalty is much less severe for large movement. However, the problem for L^1 MKP is that it can produce non-smooth results. In our future research, we may use a $L^{1+\epsilon}$ penalty instead, with ϵ being a value between 0 and 1. That is,

we can minimize the following energy functional,

$$M_{1+\epsilon} = \int |u - x|^{1+\epsilon} \mu_0 \tag{131}$$

with the MP constraint.

The MP constraint is general and is not necessarily constricted to MKP. For example, it can be combined with the concept of harmonic mapping to obtain a new approach to MP diffeomorphisms. Assume we have two domains (Ω_0, μ_0) and (Ω_1, μ_1) as previously defined, with μ_0 and μ_1 being positive densities. We can consider the minimization of the Dirichlet integral over all MP maps

$$\min_{u \in MP} \int_{\Omega_0} \|Du\|^2. \tag{132}$$

A minimizer (when existing) is called an MP mapping of minimal distortion [4].

Thus the Monge-Kantorovich flow and the MP constraint have a lot of possible research topics related to them. We intend to explore in this area in the next few years.

REFERENCES

- [1] AMBROSIO, L., “Lecture notes on optimal transport problems.” Available on <http://cvgmt.sns.it/papers/amb00a/>.
- [2] ANGENENT, S., HAKER, S., and TANNENBAUM, A., “Minimizing flows for the Monge-Kantorovich problem,” *SIAM J. Math. Analysis*, vol. 35, no. 1, pp. 61–97, 2003.
- [3] ANGENENT, S., HAKER, S., TANNENBAUM, A., and KIKINIS, R., “On the Laplace-Beltrami operator and brain surface flattening,” *IEEE Trans. on Medical Imaging*, vol. 18, pp. 700–711, 1999.
- [4] ANGENENT, S., HAKER, S., TANNENBAUM, A., and KIKINIS, R., *System Theory: Modeling, Analysis, and Control*, pp. 275–287. Kluwer, 1999.
- [5] ARULIAH, D., ASCHERY, U., HABERZ, E., and OLDENBURGX, D., “A method for the forward modelling of 3D electromagnetic quasi-static problems,” *Mathematical Models and Methods in Applied Sciences*, vol. 11, pp. 1–21, 2001.
- [6] BAJCSY, R. and KOVACIC, S., “Multiresolution elastic matching,” *Computer Vision, Graphics and Image Processing*, vol. 46, pp. 1–21, 1989.
- [7] BARRON, J., FLEET, D., and BEAUCHEMIN, S., “Performance of optical flow techniques,” *Int. Journal of Computer Vision*, vol. 12, no. 1, pp. 43–77, 1994.
- [8] BEIER, T. and NEELY, S., “Feature-based image metamorphosis,” in *Computer Graphics (Proc. SIGGRAPH '92)*, vol. 26, pp. 35–42, 1992.
- [9] BENAMOU, J. and BRENIER, Y., “A computational fluid mechanics solution to the Monge-Kantorovich mass transfer problem,” *Numerische Mathematik*, vol. 84, pp. 375–393, 2000.
- [10] BESL, P. and MCKAY, N., “A method for registration of 3D shapes,” *IEEE Trans. Pattern Anal. Machine Vision*, vol. 14, pp. 239–256, 1992.
- [11] BIMBO, A., “Analysis of optical flow constraints,” *IEEE Trans. on Image Processing*, vol. 4, no. 4, pp. 460–469, 1995.
- [12] BIMBO, A., “Optical flow computation using extended constraints,” *IEEE Trans. on Image Processing*, vol. 5, no. 5, pp. 720–739, 1996.
- [13] BLUM, H., “A transformation for extracting new descriptors of shape,” in *Proc. of Symposium Models Perception Speech Visual Form*, pp. 362–380, MIT Press, 1967.

- [14] BOOKSTEIN, F., “Principal warps: thin-plate splines and the decomposition of deformations,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 11, no. 6, pp. 567–585, 1989.
- [15] BRENIER, Y., “Polar factorization and monotone rearrangement of vector-valued functions,” *Com. Pure Appl. Math.*, vol. 64, pp. 375–417, 1991.
- [16] BRIGGS, W., HENSON, V., and MCCORMICK, S., *A Multigrid Tutorial*. Philadelphia: SIAM, 2000.
- [17] CARMAN, G., DRURY, H., and ESSEN, D. V., “Computational methods for reconstructing and unfolding the cerebral cortex,” *Cereb. Cortex*, vol. 5, no. 6, pp. 506–517, 1995.
- [18] CASELLES, V., KIMMEL, R., and SAPIRO, G., “Geodesic active contours,” *International Journal on Computer Vision*, vol. 22, no. 1, pp. 61–79, 1997.
- [19] CHRISTENSEN, G. E. and JOHNSON, H. J., “Consistent image registration,” *IEEE Trans. on Medical Imaging*, vol. 20, pp. 568–582, 2001.
- [20] CHRISTENSEN, G., BABBITT, R., and MILLER, M., “Deformable templates using large deformation kinematics,” *IEEE Trans. on Image Processing*, vol. 5, no. 10, pp. 1435–1447, 1996.
- [21] COURANT, R. and HILBERT, D., *Method of Mathematical Physics*. New York: Interscience, 1996.
- [22] CULLEN, M. and PURSER, R., “An extended lagrangian theory of semi-geostrophic frontogenesis,” *J. Atmos. Sci.*, vol. 41, pp. 1477–1497, 1984.
- [23] CURRY, C., *New Methods for the Computation of Optical Flow*. Ph.D. thesis, Georgia Institute of Technology, 2002.
- [24] DACOROGNA, B. and MOSER, J., “On a partial differential equation involving the Jacobian determinant,” *Ann. Inst. H. Poincaré Anal. Non Linéaire*, vol. 7, pp. 1–26, 1990.
- [25] DAVATZIKOS, C., PRINCE, J., and BRYAN, R., “Image registration based on boundary mapping,” *IEEE Trans. Med. Imaging*, vol. 15, no. 1, pp. 112–115, 1996.
- [26] DECLERCK, J., FELDMAR, J., GORIS, M., and BETTING, F., “Automatic registration and alignment on a template of cardiac stress and rest reoriented SPECT images,” *IEEE Trans. Med. Imaging*, vol. 6, pp. 727–737, 1997.
- [27] DUDA, R., HART, P., and STORK, D., *Pattern Classification*. Wiley-Interscience, 2000.
- [28] ERNESTOVICH, L., *Calculus of Variations*. London: Pergamon Press, 1962.

- [29] FRY, D., *Shape recognition using metrics on the space of shapes*. Ph.d. thesis, Harvard University, 1993.
- [30] GANGBO, W., “An elementary proof of the polar factorization of vector-valued functions,” *Arch. Rational Mechanics Anal.*, vol. 128, pp. 381–399, 1994.
- [31] GANGBO, W. and MCCANN, R., “The geometry of optimal transportation,” *Acta Math.*, vol. 177, pp. 113–161, 1996.
- [32] GEE, J., BARILLOT, C., BRIQUER, L. L., HAYNOR, D., and BAJCSY, R., “Matching structural images of the human brain using statistical and geometrical features,” in *Proc. 3rd International Conference Visualization in Biomedical Computing (VBC’94)*, pp. 191–204, 1994.
- [33] GOSHTASBY, A., “Registration of images with geometric distortions,” *IEEE Trans. Geosci. and Remote Sensing*, vol. 26, no. 1, pp. 60–64, 1988.
- [34] GU, X. and YAU, S., “Global conformal surface parameterization,” in *Proc. of the Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, pp. 127–137, 2003.
- [35] HAJNAL, J., HILL, D., and HAWKES, D., *Medical Image Registration*. Boca Raton: CRC Press, 2001.
- [36] HAKER, S., ANGENENT, S., TANNENBAUM, A., and KIKINIS, R., “Nondistorting flattening maps and the 3D visualization of colon CT images,” *IEEE Trans. on Medical Imaging*, vol. 19, pp. 665–670, 2000.
- [37] HAKER, S., ANGENENT, S., TANNENBAUM, A., KIKINIS, R., SAPIRO, G., and HALLE, M., “Conformal surface parameterization for texture mapping,” *IEEE Tran. on Visualization and Computer Graphics*, vol. 6, no. 2, pp. 181–189, 2000.
- [38] HAKER, S. and TANNENBAUM, A., “Mass-preserving maps for registration and visual tracking,” in *IEEE Proc. on Decision and Control*, pp. 4812–4817, 2001.
- [39] HAKER, S., ZHU, L., TANNENBAUM, A., and ANGENENT, S., “Optimal mass transport for registration and warping,” *International Journal on Computer Vision*, accepted.
- [40] HARALICK, R. and SHAPIRO, L., *Computer and Robot Vision*. New York: Addison-Wesley, 1992.
- [41] HASSANIEN, A. and NAKAJIMA, M., “Image morphing of facial images transformation based on navier elastic body splines,” in *Computer Animation Proceeding*, vol. 8, pp. 119–125, 1998.

- [42] HATA, N., NABAVI, A., WELLS, W., WARFIELD, S., KIKINIS, R., BLACK, P., and JOLESZ, F., “Three-Dimensional optical flow method for measurement of volumetric brain deformation from intraoperative mr images,” *J. Comput Assist Tomogr*, vol. 24, pp. 531–538, 2000.
- [43] HE, T., HONG, L., CHEN, D., and LIANG, Z., “Reliable path for virtual endoscopy: Ensuring complete examination of human organs,” *IEEE Tran. on Visualization and Computer Graphics*, vol. 7, no. 4, pp. 333–342, 2001.
- [44] HECKBERT, P., “Survey of texture mapping,” *IEEE Computer Graphics and Applications*, vol. 6, pp. 56–67, 1986.
- [45] HECKBERT, P., “Fundamentals of texture mapping and image warping,” master thesis, University of California, 1989.
- [46] HERMOSILLO, G., C.CHEFD’HOTEL, and FAUGERAS, O., “A variational approach to multi-modal image matching,” Tech. Rep. n°4117, INRIA, February 2001.
- [47] HERNANDEZ, M., FRANGI, A., and SAPIRO, G., “Three-dimensional segmentation of brain aneurysms in CTA using non-parametric region-based information and implicit deformable models: method and evaluation,” in *Proc. of Medical Image Computing and Computer-Assisted Intervention (MICCAI 2003)* (ELLIS, R. E. and PETERS, T. M., eds.), no. 2879, pp. 594–602, 2003.
- [48] HONG, L., KAUFMAN, A., WEI, Y., VISWAMBHARAN, A., WAX, M., and LIANG, Z., “3D virtual colonoscopy,” in *Proc. of Biomedical Visualization (BIOMEDVIS’95)*, pp. 26–32, IEEE CS Press, 1995.
- [49] HORN, B., *Robot Vision*. Cambridge, Massachusetts: The MIT Press, 1986.
- [50] HORNAK, J., *The Basics of MRI*. <http://www.cis.rit.edu/htbooks/mri/>.
- [51] HUGHES, T., *The Finite Element Method*. New Jersey: Prentice-Hall, 1987.
- [52] KAIJSER, T., “Computing the Kantorovich distance for images,” *Journal of Mathematical Imaging and Vision*, vol. 9, pp. 173–191, 1998.
- [53] KANATANI, K., “Analysis of 3-D rotation fitting,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 16, pp. 543–549, 1994.
- [54] KANTOROVICH, L., “On a problem of monge,” *Uspekhi Mat. Nauk.*, vol. 3, pp. 225–226, 1948.
- [55] KASSAM, A. and WOOD, M., “Fourier registration of three-dimensional brain mr images: exploiting the axis of rotation,” *J. Magn. Reson. Imaging*, vol. 6, pp. 894–902, 1996.

- [56] KICHENASAMY, S., OLVER, P., TANNENBAUM, A., and YEZZI, A., “Conformal curvature flows: from phase transitions to active contours,” *Archive Rational Mechanics and Analysis*, vol. 134, pp. 275–301, 1996.
- [57] KIM, B., BOES, J., FREY, K., and MEYER, C., “Mutual information for automated unwarping of rat brain autoradiographs,” *Neuroimage*, vol. 5, no. 1, pp. 31–40, 1997.
- [58] KNOTT, M. and SMITH, C., “On the optimal mapping of distributions,” *J. Optim. Theory*, vol. 43, no. 39-49, 1984.
- [59] KUMAR, A., HAKER, S., STILLMAN, A., CURRY, C., GIDDENS, D., TANNENBAUM, A., and YEZZI, A., “L1 based optical flow for cardiac wall motion tracking,” in *Proc. of SPIE Medical Imaging*, vol. 4322, 2001.
- [60] KUMAR, A., TANNENBAUM, A., and BALAS, G., “Optical flow: a curve evolution approach,” *IEEE Trans. Image Processing*, vol. 5, no. 4, pp. 598–610, 1996.
- [61] Lévy, B. and MALLET, J., “Non-distorted texture mapping for sheared triangulated meshes,” in *Proc. Computer Graphics (SIGGRAPH’98)*, vol. 32, pp. 343–352, 1998.
- [62] LEE, A., SWELDENS, W., Schröder, P., COWSAR, L., and DOBKIN, D., “Maps: Multiresolution adaptive parameterization of surfaces,” in *Proc. Computer Graphics (SIGGRAPH’98)*, vol. 32, pp. 95–104, 1998.
- [63] LEE, S.-Y., CHWA, K.-Y., HAHN, J., and SHIN, S., “Image morphing using deformation techniques,” *J. Visualization and Computer Animation*, vol. 7, no. 1, pp. 3–23, 1996.
- [64] LEVINA, E. and BICKEL, P., “The earth mover’s distance is the mallow’s distance: some insights from statistics,” in *Proc. IEEE Int. Conf. on Computer Vision*, 2001.
- [65] MAES, F., COLLIGNON, A., VANDERMEULEN, D., MARCHAL, G., and SUETENS, P., “Multimodality image registration by maximization of mutual information,” *IEEE Trans. on Medical imaging*, vol. 16, pp. 187–198, 1997.
- [66] MAINTZ, J. and VIERGEVER, M., “A survey of medical image registration,” *Medical Image Analysis*, vol. 2, no. 1, pp. 1–36, 1998.
- [67] MCCANN, R., “Polar factorization of maps on riemannian manifolds,” *Geom. Funct. Anal.*, vol. 11, pp. 589–608, 2001.
- [68] NEHARI, Z., *Conformal mapping*. New York: Dover Publications, 1975.
- [69] PAIK, D., BEAULIEU, C., JEFFREY, R., KARADI, C., and NAPEL, S., “Visualization modes for ct colonography using cylindrical and planar map projections,” *J. Comput. Assist. Tomogr.*, vol. 24, no. 2, pp. 179–188, 2000.

- [70] PELIZZARI, C., CHEN, G., SPELBRING, D., WEICHSELBRAUM, R., and CHEN, C., “Accurate three-dimensional registration of CT, PET and/or MR images of the brain,” *J. Comput. Assist. Tomogr.*, vol. 13, pp. 20–26, 1989.
- [71] PRESS, W., TEUKOLSKY, S., VETTERLING, W., and FLANNERY, B., *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge U.K.: Cambridge University Press, 2nd edition ed., 1992.
- [72] RACHEV, S. and RUSCHENDORF, L., *Mass transportation problems*, vol. Volumes I and II: Probability and Its Applications. New York: Springer, 1998.
- [73] RACHEV, S., “The Monge-Kantorovich mass transference problem and its stochastic applications (translated by B. Seckler,” *Theory of Probability and Its Applications*, vol. 29, pp. 647–676, 1984.
- [74] RAUCH, J., *Partial Differential Equations*. New York: Springer-Verlag, 1991.
- [75] RUBNER, Y., *Perceptual Metrics for Image Database Navigation*. Ph.d. thesis, Stanford University, May 1999.
- [76] RUBNER, Y., TOMASI, C., and GUIBAS, J., “The earth mover’s distance as a metric for image retrieval,” Technical Report STAN-CS-TN-98-86, Department of Computer Science, Stanford University, September 1998.
- [77] RUECKERT, D., SONODA, L., HAYES, C., HILL, D., LEACH, M., and HAWKES, D., “Non-rigid registration using free-form deformations: Application to breast mr images,” *IEEE Trans. Med. Imaging*, vol. 18, no. 8, pp. 712–721, 1999.
- [78] SCHROEDER, W., MARTIN, H., and LORENSEN, B., *The Visualization Toolkit*. Prentice-Hall, 1996.
- [79] SCHWARTZ, E., SHAW, A., and WOLFSON, E., “A numerical solution to the generalized mapmaker’s problem: flattening nonconvex polyhedral surfaces,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 11, no. 9, pp. 1005–1008, 1989.
- [80] SHANNON, C., “The mathematical theory of communication (parts 1 and 2),” *Bell Syst. Tech. J.*, vol. 27, pp. 379–423 and 623–656, 1948.
- [81] SIDDIQI, K., BOUIX, S., TANNENBAUM, A., and ZUCKER, S., “The hamilton jacobi skeleton,” in *Proc. of the Seventh IEEE International Conference on Computer Vision*, vol. 2, pp. 828–834, 1999.
- [82] SKRINJAR, O. and TAGARE, H., “Symmetric, transitive, geometric deformation and intensity variation invariant nonrigid image registration,” in *Proc. IEEE International Symposium on Biomedical Imaging (ISBI 2004)*, pp. 920–923, 2004.

- [83] STRANG, G., *Introduction to Applied Mathematics*. Wellesley, Mass.: Wellesley-Cambridge Press, 1986.
- [84] TAYLOR, M., *Partial Differential Equations III*. New York: Springer-Verlag, 1996.
- [85] THOMPSON, P. and TOGA, A., "A surface-based technique for warping three-dimensional images of the brain," *IEEE Trans. on Medical Imaging*, vol. 15, no. 4, pp. 1–16, 1996.
- [86] VIOLA, P., *Alignment by Maximization of Mutual Information*. Ph.d. thesis, Massachusetts Institute of Technology, 1995.
- [87] VIOLA, P. and WELLS, W., "Alignment by maximization of mutual information," *International Journal of Computer Vision*, vol. 24, no. 2, pp. 137–154, 1997.
- [88] WANG, G., DAVE, S., BROWN, B., ZHANG, Z., MCFARL, E., HALER, J., and M. VANNIER, "Colon unraveling based on electrical field – recent progress and further work," in *SPIE Proc. 3660*, 1999.
- [89] WANG, X., FENG, D., and JIN, J., "Elastic medical image registration based on image intensity," in *VIP2001*, (Sydney, Australia), 2001. Available online: <http://www.jrpit.flinders.edu.au/confpapers/CRPITV11Wang1.pdf>.
- [90] WANG, X. and SHAO, M., *Principles of finite element method and its numerical methods (in Chinese)*. Tsinghua Press, 2 ed., 1997.
- [91] WOLBERG, G., *Digital image warping*. Los Alamitos, CA: IEEE Computer Society Press, 1990.
- [92] WOLBERG, G., "Recent advances in image morphing," in *Computer Graphics Intl. '96*, (Pohang), pp. 64–71, 1996.
- [93] YANG, Y., TANNENBAUM, A., and GIDDENS, D., "Knowledge-based 3D segmentation and reconstruction of coronary arteries using CT images," in *Proc. IEEE EMBS04*, p. to appear, 2004.
- [94] YEZZI, A. and PRINCE, J., "An eulerian pde approach for computing tissue thickness," *IEEE Trans. on Medical Imaging*, vol. 22, no. 10, pp. 1332–1339, 2003.
- [95] ZARIN, C., GIDDENS, D., BHARADVAJ, B., SOTTIURAI, V., MABON, R., and GLAGOV, S., "Carotid bifurcation atherosclerosis: Quantitative correlation of plaque localization with flow velocity profiles and wall shear stress.," *Circulation Research*, vol. 53, no. 4, pp. 502–514, 1983.
- [96] ZHANG, C. and COHEN, F., "3-D face structure extraction and recognition from images using 3-D morphing and distance mapping," *IEEE Trans. on Image Processing*, vol. 11, no. 11, pp. 1249–1259, 2002.

- [97] ZHOU, Y. and TOGA, A., “Efficient skeletonization of volumetric objects,” *IEEE Trans. on Visualization and Computer Graphics*, vol. 3, no. 3, pp. 196–209, 1999.
- [98] ZHU, L., HAKER, S., and TANNENBAUM, A., “Area-preserving mappings for the visualization of medical structures,” in *Proc. of Medical Image Computing and Computer-Assisted Intervention (MICCAI 2003)* (ELLIS, R. E. and PETERS, T. M., eds.), no. 2879, pp. 277–284, 2003.
- [99] ZHU, L., HAKER, S., and TANNENBAUM, A., “Flattening maps for the visualization of multi-branched vessels,” *IEEE Trans. on Medical Imaging*, submitted, in revising.
- [100] ZHU, L., HAKER, S., TANNENBAUM, A., BOUX, S., and SIDDIQI, K., “Angle-preserving mappings for the visualization of multi-branched vessels,” in *Proc. of Int. Conf. on Image Processing 2002 (ICIP2002)*, vol. II, pp. 945–948, 2002.
- [101] ZHU, L. and TANNENBAUM, A., “Image interpolation based on optimal mass preserving mappings,” in *Proc. of International Symposium on Biomedical Imaging (ISBI’04)*, pp. 21–24, 2004.
- [102] ZHU, L., YANG, Y., TANNENBAUM, A., and HAKER, S., “Image morphing based on mutual information and optimal mass transport,” in *Proc. of Int. Conf. on Image Processing 2004 (ICIP2004)*, 2004. accepted.

VITA

Lei Zhu was born on September 4, 1976, in Beijing, China. He received his B.Eng. and M.Eng. degrees in Biomedical Engineering from Tsinghua University, Beijing, China, in 1998 and 2000, respectively. Since 2000, he has been working towards the Ph.D. degree in Biomedical Engineering at Georgia Institute of Technology, Atlanta, GA. At the same time, he was a research assistant in the group of Mathematical ImagINg, computER Vision and their Applications (MINERVA), Department of Biomedical Engineering, Georgia Tech. He also received his M.S. degree in Electrical and Computing Engineering from Georgia Institute of Technology, in 2003.

His research interests include surface mapping, image registration, and stereo vision.