Parallelizable Inpainting and Refinement of Diffeomorphisms using Beltrami Holomorphic Flow

Tsz Wai Wong
University of California, Los Angeles
Department of Mathematics,
520 Portola Plaza,
Math Science Building 6363, UCLA,
Los Angeles, CA 90095
alvinwong@math.ucla.edu

Xianfeng Gu
State University of New York at Stony Brook
Department of Computer Science,
Room 2425, Computer Science Building,
State University of New York at Stony Brook,
Stony Brook, NY 11794-4400
gu@cs.sunysb.edu

Tony F. Chan
Hong Kong University of Science and Technology
Office of the President, HKUST,
Clear Water Bay, Kowloon, Hong Kong
tonyfchan@ust.hk

Lok Ming Lui
The Chinese University of Hong Kong
Department of Mathematics,
Room 207, Lady Shaw Building, CUHK,
Shatin, Hong Kong
lmlui@math.cuhk.edu.hk

Abstract

In this paper, we propose novel algorithms for inpainting and refinement of diffeomorphisms. We first represent a diffeomorphism by its Beltrami coefficient. Then it is possible to refine and inpaint the diffeomorphism by processing this Beltrami coefficient. With the inpainted/refined Beltrami coefficient, we construct a new diffeomorphism using the exact Beltrami holomorphic flow algorithm proposed in this paper. We apply our algorithms on several practical applications, which include the inpainting of a highly distorted diffeomorphism, the inpainting of image sequences of deforming shapes, the super-resolution of diffeomorphisms and the global parameterization of cortical surfaces by combining local parameterizations. Experiments show that our algorithm can solve these problems with natural and smooth results. We demonstrate how our proposed method can be widely applied in areas from texture mapping to video processing, and from computer graphics to medical imaging.

1. Introduction

In computer graphics and medical imaging, a great deal of effort is spent on processing surface diffeomorophisms. For example, in computer-aided design, fine diffeomorphisms are important for high quality texture mapping of 3D models. In brain imaging, they are crucial for the registration of cortical surfaces. However, due to noise or highly convoluted surfaces like the cortical surface, certain regions of the surfaces may not able to get registered at all or result in highly distorted and/or overlapping regions. To fix this problem, we need to properly restore the missing region using existing data as much as possible.

For other situations such as video compression and computer games, the storage allowed is limited for practical reason. For example, in video compression, consecutive frames may be related by a smoothly varying diffeomorphisms, which allows further compression. In computer games, it is not practical to store the precise texture maps of thousands of in-game objects. To save storage, texture maps are often stored in a piecewise linear way with every triangle in simplified triangular meshes. This causes unnatural distortions when objects are zoomed in closely. To achieve better visualization, we need to restore the desired quality of the texture maps, like the degree of distortion and smoothness, from a compact representation as above.

Motivated by these problems, we are interested in developing effective algorithms to either 'inpaint' the missing or distorted regions of a diffeomorphism, or refine a diffeomorphism with low resolution to higher resolution. In this paper, we propose a novel approach to solve these problems. The basic idea is to represent a diffeomorphism by its Beltrami coefficient. Then we may inpaint or refine it by interpolating its Beltrami coefficients instead of its coordinate

978-1-4577-1102-2/11/$26.00

functions. A new diffeomorphism is then constructed from the inpainted/refined Beltrami coefficient using the exact Beltrami holomorphic flow (BHF) algorithm we propose. Compared with other methods, such as direct linear/cubic interpolation of coordinate functions, our method guarantees smoothness and diffeomorphic property. In the original missing region, the restored diffeomorphism follows the property of the original diffeomorphism on the non-occluded region. We apply our algorithms to three practical applications, including (i) super-resolution of texture maps to sharpen and smoothen surface textures, (ii) parallelizable landmark-matching surface mapping to parameterize complicated surfaces efficiently, and (iii) inpainting of image sequences of deforming shapes. Experimental results confirm the effectiveness and efficacy of our proposed algorithms.

2. Previous Work

Image inpainting and super-resolution have been extensively studied. Inpainting refers to the process of reconstructing lost or deteriorated parts of images. Image super-resolution aims to produce an aesthetically pleasing high resolution image from a low resolution image. Both problems are related to image interpolation. Recently, different approaches for this subject have been proposed. Belahmidi [2] proposed a PDE-based approach to zoom images by solving anisotropic heat diffusion equations. Bertalmío et al. [3] proposed to apply Navier-Stokes equations and fluid dynamics for image and video inpainting. Shen et al. [6] proposed local inpaintings of non-texture images based on the classical total variation (TV) denoising model. Later, Cha et al. [5] applied the PDE form of the TV energy for image zooming. Multiscale approach was also proposed. Carey et al. [4] proposed an image interpolation approach based on wavelets. Although image interpolation has been well-studied, the interpolation of surface diffeomorphisms preserving bijectivity has rarely been studied.

Surface registration has also been widely studied. Gu et al. [13] proposed conformal surface registration by minimizing some energy functionals. Lévy et al. [9] proposed a least square method to obtain conformal maps for texture mapping. To obtain a surface registration that matches important landmark features, Durrleman et al. [7] developed a framework using currents, a concept from differential geometry, to match landmarks within surfaces across subjects, for the purpose of inferring the variability of brain structure in an image database. Lui et al. [10] proposed to compute shape-based landmark matching registrations between brain surfaces using an integral flow method. Leow et al. [8] proposed a level-set-based approach for matching different types of features, including points, 2D and 3D curves represented as implicit functions.

This work is mainly based on the representation of diffeomorphisms by Beltrami coefficients. Studying diffeomorphisms by Beltrami coefficient was first proposed by Lui et al. [12] for medical shape analysis. They further proposed to compute geometric matching surface registration by adjusting Beltrami coefficients [11].

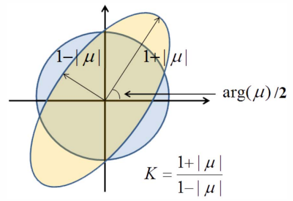


Figure 1. Illustration of how Beltrami coefficient μ measures the distortion of a quasiconformal mapping that maps a small circle to an ellipse with dilation K.

3. Theoretical Background

3.1. Quasiconformal Mappings and Beltrami Coefficients

A surface S with a conformal structure is called a *Riemann surface*. Given two Riemann surfaces M and N, a map $f : M \to N$ is *conformal* if it preserves the surface metric up to a multiplicative factor called the *conformal factor*. A generalization of conformal maps is *quasiconformal* maps, which are orientation-preserving diffeomorphisms between Riemann surfaces with bounded conformality distortion, in the sense that their first order approximation takes small circles to small ellipses of bounded eccentricity (see Figure 1).

Mathematically, $f : \mathbb{C} \to \mathbb{C}$ is quasiconformal if it satisfies the Beltrami equation $\frac{\partial f}{\partial \overline{z}} = \mu(z)\frac{\partial f}{\partial z}$, for some complex valued functions μ with $\|\mu\|_\infty < 1$ called the *Beltrami coefficient*. In particular, f is conformal around a small neighborhood of p if and only if $\mu(p) = 0$. f may be considered as a map composed of a translation to $f(p)$ together with a stretch map $S(z) = z + \mu(p)\overline{z}$, which is postcomposed by a multiplication of $f_z(p)$, which is conformal. All the conformal distortion of $S(z)$ is caused by $\mu(p)$. $S(z)$ is the map that causes f to map small circles to small ellipses. The angle of maximal magnification is $\arg(\mu(p))/2$ with magnifying factor $1 + |\mu(p)|$; the angle of maximal shrinkage is the orthogonal angle $(\arg(\mu(p)) - \pi)/2$ with shrinking factor $1 - |\mu(p)|$. The distortion or dilation is given by $K = (1 + |\mu(p)|)/(1 - |\mu(p)|)$. Thus, the Beltrami coefficient μ gives us important information about the properties of a map. An illustration of a quasiconformal mapping is given in Figure 1.

3.2. Adjusting Diffeomorphisms by Beltrami Holomorphic Flow

Let $f : \mathbb{C} \to \mathbb{C}$ be a diffeomorphism. We say that f fixes 0, 1 and ∞ if $f(0) = 0$, $f(1) = 1$ and $\lim_{z\to\infty} f(z) = \infty$.

Suppose f fixes 0, 1 and ∞ and satisfies the Beltrami equation $\frac{\partial f}{\partial \bar{z}} = \mu \frac{\partial f}{\partial s}$. If we set $\mu(t) = \mu + t\nu$, then $f^{\mu(t)}(w) = f(w) + t\dot{f}(w) + o(t^2)$, where

$$\dot{f}(w) = -\frac{1}{\pi} \iint \nu(z) R(f(z), f(w))(f_z(z))^2 \, dx \, dy, \quad (1)$$

with $R(z, w) := \frac{1}{z-w} - \frac{w}{z-1} + \frac{w-1}{z}$.

This formula gives the variation of f with respect to the variation of μ. A proof of it is given in [1]. Apparently, the integrand has singularities at $z = 0$, $z = 1$ and $z = w$. However, the integrand can be written as 3 terms, where each has just a simple pole. Integrating them on $\mathbb{R}^2$ gives a finte answer and does not create singularities. We call this formula the Beltrami holomorphic flow (BHF).

Using BHF, we may adjust any diffeomorphism $f^{\mu_\bullet}$ to any other diffeomorphism f^{μ}, with Beltrami coefficients μ_0 and μ respectively. When the initial $\mu = 0$, this amounts to computing f^{μ} from the identity map $f^0 = \mathrm{Id}$. Setting $\nu = \mu - \mu_0$, we compute the BHF for $f^{\mu(t)}$, where $\mu(t) = \mu_0 + t\nu$. Theoretically, the approximation of f^{μ} is given by setting $t = 1$, i.e., $f^{\mu}(w) \approx f(w) + \dot{f}(w, t)$. However, when ν is not small enough, we may face the problem of overlapping in $f^{\mu(1)}$. We will discuss how to choose an optimal t in Subsection 4.2.

3.3. TV Inpainting of of 2D Image Data

Inpainting can be regarded as a process of interpolating data on the occluded region from the known data on its neighborhood. To inpaint an occluded 2D image, we can fill in the missing region by solving the Perona-Malik diffusion model:

$$\begin{cases} \frac{\partial u}{\partial t} = div(g(|\nabla u|)\nabla u) & \text{on } D; \\ u^0 = v & \text{on } D^c, \end{cases}$$

where D is the occluded region, $v\colon D^c \to \mathbb{R}$ is the original image with occlusion, u is the inpainted image, and $g\colon \mathbb{R} \to \mathbb{R}$ is an increasing function with $g(0) = 0$ and $g(\infty) = \infty$. If we replace g by $\frac{1}{\nabla u}$, we get the familiar TV inpainting model, which is well-known to preserve edges.

Image inpainting has been extensively studied. However, as far as we know, no work has been done on the inpainting of 2D/3D diffeomorphisms. In this paper, we extend the TV inpainting algorithm to inpaint diffeomorphisms with occluded regions.

4. Our Proposed Algorithms

In this section, we propose several algorithms to deal with the inpainting and refinement of diffeomorphisms using BHF. Although BHF deals with diffeomorphisms of 2D domains, our algorithms can be easily extended to processing surface diffeomorphisms by reparameterization onto 2D domains, such as conformal parameterizations [13].

4.1. Exact Computation of the Beltrami Holomorphic Flow

Let $f\colon D \to D$ be a diffeomorphism on $D = [-1, 1]^2$ with $f(0) = 0$ and $f(1) = 1$. Denote the triangulation of D by $\mathrm{Tri}(D)$. For the discretized f, its value is known on every vertex of D. It is natural to assume that the actual f can be well approximated piecewise linearly on each face in $\mathrm{Tri}(D)$, where the Beltrami coefficient is constant. We may also assume that ν, the adjustment to μ, is also piecewise constant on each face of $\mathrm{Tri}(D)$. Then $\mu(t) = \mu + t\nu$ is a piecewise constant function on D. For every $T \in \mathrm{Tri}(D)$, denote the value of ν on T as ν_T, and the value of f_z on T as $f_{z,T}$. The direction of BHF in (1) becomes

$$\dot{f}(w) = \sum_{T \in \mathrm{Tri}(D)} \dot{f}(w, T), \quad (2)$$

where

$$\dot{f}(w, T) := -\frac{1}{\pi} \nu_T f_{z,T}^2 \iint_{z \in T} R(f(z), f(w)) \, dx \, dy. \quad (3)$$

Note that $R(f(z), f(w))$ can be written as a sum of 3 simple fraction terms. Since $f(w)$ is constant in the integral, we may pull the factors $f(w)$ and $f(w)-1$ in the last two terms out of the integral. Therefore to compute $\dot{f}(w, t, T)$, it suffices to compute integrals of the form $\iint_{z \in T} \frac{1}{f(z)-c} \, dx \, dy$ and sum. Note that $f(z) - c$ is a linear function in the integral. It turns out that all reciprocals of linear functions can be integrated exactly. This allows us to find the exact direction given by BHF.

With exact integration, our algorithm always give the exact derivative of f with respective to the adjustment ν in μ. The only source of error comes from the discretization of f, which is unavoidable for computations on triangular meshes. In the next subsection, we discuss the optimal step size to take after the direction given by BHF is computed.

4.2. Adjusting Diffeomorphisms using BHF with Adaptive Step Size

Given a diffeomorphism $f\colon \mathbb{C} \to \mathbb{C}$ with Beltrami coefficient μ_0 fixing 0, 1 and ∞. Suppose we want to adjust its Beltrami coefficient to μ on $D = [-1, 1]^2$. After setting $\nu = \mu - \mu_0$, $\mu(t) = \mu_0 + t\nu$ and computing $\dot{f}(w)$ using the exact BHF algorithm, it may be tempting to update f by setting $t = 1$ to get the required diffeomorphism. However, although exact integration of (1) gives the exact flowing direction of f with respect to the change in μ, the accuracy of this first order approximation depends on $\|\nu\|_\infty$. If t is too large, overlapping may occur and prevent the algorithm from converging. In this subsection, we propose a method which allows us to find the optimal step size t.

After computing $\dot{f}(w)$ for every vertex in $\mathrm{Tri}(D) = (\mathcal{V}, \mathcal{E}, \mathcal{F})$, the new mapping $h(w) = f(w) + t\dot{f}(w)$ satisfies the Beltrami equation with Beltrami coefficient $\sigma(t)$, which is piecewise constant on every $T \in \mathcal{F}$. When t is small, $\sigma(t)$ is approximately $\mu_0 + t\nu$. As t gets larger, the approximation gets worse. We propose to adjust f with a value of t that will not cause overlapping in h and such that $\sigma(t)$ is the best approximation to the target Beltrami coefficient μ. On every triangle $T = \{z_1, z_2, z_3\} \in \mathcal{F}$, we compute the smallest time $t_T > 0$ such that Jacobian of h on T will be zero. As $h(z_1)$, $h(z_2)$ and $h(z_3)$ move linearly as t increases, the Jacobian of h on T is quadratic in T. Therefore t_T is a root of this quadratic equation. If the Jacobian is positive for all $t > 0$, we set t_T to $+\infty$. The threashold value of t, $t_{\text{threshold}}$, is such at the Jacobian of h on at least one triangle reaches 0:

$$t_{\text{threshold}} := \min_{T\in\mathcal{F}} t_T \tag{4}$$

For the algorithm to work, t must be strictly less than $t_{\text{threshold}}$. We always choose t such that $t < t_{\text{threshold}}/2$ and $\sigma(t)$ best approximates μ. Using Newton's method, we find the optimal t that minimizes the L^2-norm of $\sigma(t) - \mu$:

$$t_{\text{optimal}} := \arg \min_{0<t<t_{\text{thresh}\bullet\text{ld}}/2} \|\sigma(t) - \mu\|_2 \tag{5}$$

One may also want to use other criteria for the closeness of approximation other than the L^2-norm. In our experiments, the above choice of $t_{optimal}$ gives rapid convergence within 25 iterations most of the time, and $t_{optimal}$ could be much larger than 1 towards the end of the algorithm.

In the next 2 subsections, we make use of the exact BHF algorithm discussed thus far on the problems of inpainting and super-resizing diffeomorphisms.

4.3. Beltrami Inpainting of Diffeomorophisms

In this subsection, we propose an algorithm to inpaint a surface diffeomorphism on any region defined by a user. Such algorithm is extremely useful in various situations. For example, in medical imaging, part of a biological surface (e.g. cortical surface) may not be registered properly with another biological surface and shows abnormal distortions, or in video compression, where one stores the most important correspondence between 2 frames and fills in the occluded parts during playback.

To inpaint a surface diffeomorphism, we parameterize it as a diffeomorphism on $\mathbb{C}$ fixing 0, 1 and ∞, where f restricted to $D = [-1, 1]^2$ corresponds to the surface diffeomorphism. We are interested to adjust the value f takes on D. Suppose we want to inpaint a diffeomorphism $f_0 : \mathbb{C} \to \mathbb{C}$ on a region $\Omega \subset D$, and only the value of f_0 on $D\backslash\Omega$ is known, as if a partial registration is obtained from the non-occluded region. Our target is to smoothly reconstruct the original f_0, given that $f = f_0$ on $D\backslash\Omega$.

We propose to restore f by smoothly interpolating the Beltrami coefficient μ in the occluded region, while ensuring $f = f_0$ on $D\backslash\Omega$. Let μ_0 be the Beltrami coefficient of f_0 on $D\backslash\Omega$. Using existing algorithms on vectorial TV inpainting, we define our target Beltrami coefficient μ as

$$\mu := \arg \min_{\mu=\mu_\bullet \text{on} D\backslash\Omega} \iint_\Omega \Big(|(\nabla \text{Re}(\mu))(x + \sqrt{-1}y)|^2 + |(\nabla \text{Im}(\mu))(x + \sqrt{-1}y)|^2 \Big)^{1/2} dx\,dy. \tag{6}$$

After computing the target μ, we iteratively use the exact BHF with adaptive time step algorithm to find a diffeomorphism f minimizing the following energy functional:

$$f = \arg \min_{f=f_0 \text{on} D\backslash\Omega} \iint_{D\backslash\Omega} |\mu(f) - \mu|^2\, dx\,dy \tag{7}$$

The resulting map is the diffeomorphism that preserves the value of f_0 on $D\backslash\Omega$ and smoothly interpolates the Beltrami coefficient μ_0 of f_0 into Ω using its value on $D\backslash\Omega$. This may also be considered as the diffeomorphism which extends f_0 into Ω in the least distorted way.

In the process of iterating with BHF, it may be possible that the condition $f = f_0$ on $D\backslash\Omega$ is violated because a change in Beltrami coefficient may affect the entire map. We solve this problem by setting f back to f_0 on $D\backslash\Omega$ and smoothly interpolating the adjustment of f on $\partial\Omega$ into Ω. We summarize the whole algorithm in Algorithm 1.

Algorithm 1 Inpaint a Diffeomorphism f_0 into Ω from its Beltrami Coefficient Using BHF

Require: A diffeomorphism $f_0 : D \to D \subset \mathbb{C}$ where its values on $\Omega \subset \mathbb{C}$ is unknown or need to be inpainted, represented piecewise linearly on D;
 Compute the target Beltrami coefficient μ according to (6);
 Initialize f by setting $f = f_0$ on $D\backslash\Omega$ and extending it into a piecewise linear diffeomorphism on D;
 repeat
  Compute the Beltrami coefficient $\mu(f)$ of f;
  Update f using exact BHF with adaptive step for $\nu = \mu - \mu(f)$;
  Adjust f such that $f = f_0$ on $D\backslash\Omega$;
 until f converges.

To obtain a faster convergence, we may adjust f to satisfy $f = f_0$ on $D\backslash\Omega$ once in a few iterations, or when $f(\partial\Omega)$ begins to deviate from $f_0(\partial\Omega)$. The results of this algorithm and its applications are shown in Section 5.

4.4. Super-Resolution of Diffeomorphisms

In this subsection, we propose a novel algorithm by modifying the inpainting algorithm in Subsection 4.3 to deal with the super-resolution of surface diffeomorphisms. In computer graphics, it is standard to map textures onto an object (in games or CAD programs) by specifying the position on the texture every vertex is mapped to. Then every face of the object mesh is colored by interpolating this vertex correspondence. This greatly limits the quality of texture maps by the resolution of triangular meshes. Even with bilinear or trilinear filtering, it is still unnatural if an object is zoomed in too closely. To solve this problem, our algorithm allows us to refine a diffeomorphism with high detail under the same limited vertex correspondence.

To start with, we assume that the diffeomorphism or texture mapping is reparameterized as f_0, a diffeomorphism on $\mathbb{C}$ fixing 0, 1 and ∞, where we are interested to refine the value it takes on $D = [-1,1]^2 \subset \mathbb{C}$. Define S_L to be the grid point set $\{s_{L,ij} = -1 - \sqrt{(-1)} + ih + \sqrt{(-1)}jh | i, j = 0, 1, \ldots, L\}$, where $h_L = 2/L$. Suppose only the value of f_0 on a low resolution grid point set S_L is available. We seek to refine this diffeomorphism by reconstructing f_0 with its Beltrami coefficient interpolated to a fine point set S_H, where H is divisible by L.

First we compute the Beltrami coefficient μ_0 of f_0 on every grid element of S_L. Then we construct the target Beltrami coefficient μ by refining μ_0 using cubic interpolation. Using BHF, we construct a diffeomorphism f identical to f_0 on S_0 and minimizes the L^2-norm of $\mu(f) - \mu$:

$$f = \arg \min_{f = f_\bullet \text{on} S_L} \iint_{D \setminus \Omega} |\mu(f) - \mu|^2 \, dx \, dy \tag{8}$$

This results in a diffeomorphism with smoothly varying distortions due to the refined Beltrami coefficient μ.

In the process of iterating with BHF, it may be possible that the condition $f = f_0$ on S_L is violated. Noting that each grid element of S_L is mapped by f onto areas like quadrilaterals, we may fix this condition by mapping values of f inside each quadrilateral $\{f(s_{L,ij}, f(s_{L,(i+1)j}), f(s_{L,(i+1)(j+1)}), f(s_{L,i(j+1)})\}$ back onto the quadrilateral $\{f_0(s_{L,ij}, f_0(s_{L,(i+1)j}), f_0(s_{L,(i+1)(j+1)}), f_0(s_{L,i(j+1)})\}$ using a bilinear map preserving the diffeomorphic property of f. We summarize the while algorithm in Algorithm 2.

To obtain a faster convergence, we may also adjust f to satisfy $f = f_0$ on S_L only once in a few iterations, or when $f(S_L)$ begins to deviate mildly from $f_0(S_L)$. In the next section, we show our results by applying algorithms in this section on the inpainting and super-resolution of texture mappings and diffeomorphisms, on both 2D and 3D examples and applications in brain imaging.

Algorithm 2 Refine a Diffeomorphism f_0 with Known Value on a Coarse Grid from its Beltrami Coefficient Using BHF

Require: A diffeomorphism $f_0 \colon D \to D \subset \mathbb{C}$ where only its value on a coarse grid S_L is known; A finer grid S_H where its refined values are to be computed, where H is divisible by L;
  Compute the Beltrami coefficient μ_0 of f_0 on S_L;
  Smoothly interpolate μ_0 into μ defined on grid elements of S_H using the cubic method;
  Initialize f as the identity function;
  repeat
    Compute the Beltrami coefficient $\mu(f)$ of f;
    Update f using exact BHF with adaptive step for $\nu = \mu - \mu(f)$;
    Adjust f such that $f = f_0$ on S_L;
  until f converges.

5. Results and Discussion

In this section, we present the results of our BHF inpainting and refinement algorithms on 2D and 3D examples and demonstrate their effectiveness of our proposed algorithms. In each application, the complete inpainting or refinement of diffeomorphisms took place on a laptop with Intel Core 2 Duo 1.86GHz CPU and 2GB of RAM in less than 2 minutes using Matlab, with grid size not exceeding 129 by 129.

5.1. BHF Inpainting of a Highly Distorted Diffeomorphism

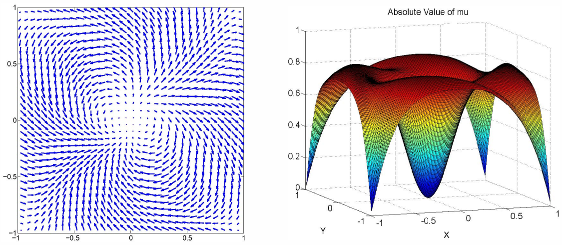
In this subsection, we apply the BHF inpainting algorithm on a diffeomorphism $f \colon [-1,1]^2 \to [-1,1]^2$, shown in Figure 2, with the inpainting region $\Omega \subset [-1,1]^2$ highlighted. From the plot of its Beltrami coefficient, f is highly distorted. The inpainting region lies on its most distorted area, making the inpainting problem challenging.

Applying the BHF inpainting algorithm on Ω, we restore the lost region of f by constructing a diffeomorphism with a smoothly inpainted Beltrami coefficient. As shown in Figure 3(a), the texture in the inpainting region smoothly blends into the surrounding texture, giving a natural diffeomorphism with continuous varying distortion. On the other hand, the result of inpainting using linear interpolation of coordinate functions of f shows only continuation on 3 sides of Ω due to the non-convexity of $f^{-1}(\Omega)$. The texture also shows a sudden jump near the boundary of Ω, giving a very unnatural mapping. Indeed, cubic interpolation shows an almost identical result. This shows that direct interpolation of coordinate functions cannot guarantee diffeomorphism. To reconstruct a diffeomorphism correctly, a careful consideration has to be given to the higher order changes of the diffeomorphism, which is achieved by smoothly inpainting the Beltrami coefficient using our algorithm.

5.2. BHF Inpainting of Image Sequences of Deforming Shapes

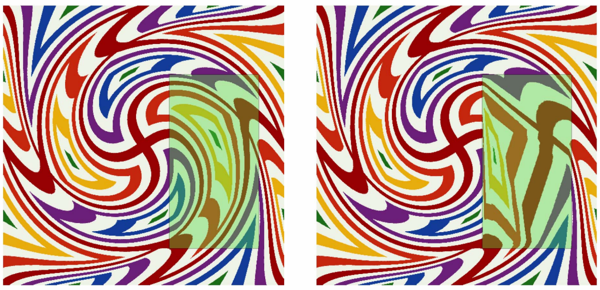
In this subsection, we demonstrate how the BHF inpainting algorithm can be applied to process image sequences

(a) (b)



(c) (d)

Figure 2. A highly distorted diffeomorphism f of $[-1,1]^2$. (a) shows the domain of f textured with a regular grid pattern. (b) shows how the texture is mapped under f onto $[-1,1]^2$, with the inpainting region highlighted. (c) shows a plot of its Beltrami coefficient μ as arrows. (d) shows a plot of the modulus of μ, indicating the high distortion of f.

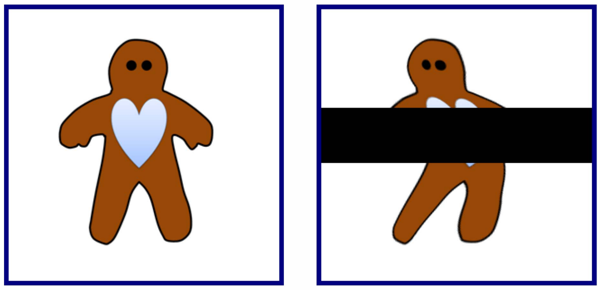


(a) (b)

Figure 3. A comparison of the result of the BHF inpainting algorithm (a) and the failed result using linear interpolation (b), with the inpainting region highlighted.

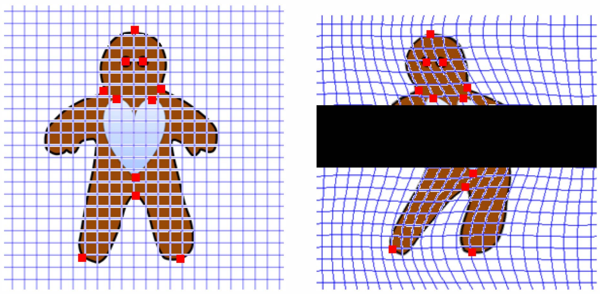
of deforming shapes, which has many applications in areas such as video processing and shape analysis of medical images over time. In this example, we aim at restoring the correspondence of 2 frames in an image sequence of a gingerbread man figure, where the second frame is distorted and occluded by unknown foreground object represented by a black region (see Figure 4). This is challenging for conventional image inpainting algorithms due to the size of the occluded region and the additional distortions.

First of all, we independently register the top and bottom non-occluded regions between the first and second frames. As illustrated in Figure 5, this can be done by marking a



(a) (b)

Figure 4. An image sequence of a gingerbread man showing (a) the initial frame, and (b) the next frame with distortion and occlusion.



(a) (b)

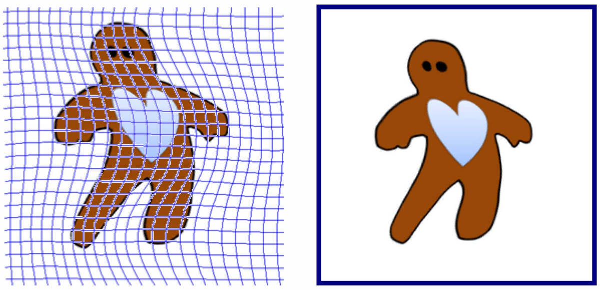
Figure 5. The highlighting of feature points and the registration between the top and bottom parts of frame 1 and 2.

number of correspondences between 2 frames in each region, and registering each region separately using existing algorithms [10]. After this, the Beltrami coefficient of the registration in the non-occluded regions can be computed. Using the BHF inpainting algorithm, we inpaint the Beltrami coefficient in the occluded region and construct the whole registration between the first and second frames, preserving the already registered top and bottom parts. The final diffeomorphism is shown in Figure 6(a). As can be seen, the complete diffeomorpism follows the pattern and geometry of the local registrations, and continues smoothly into the middle occluded region. Figure 6(b) shows the complete gingerbread man with the occluded region filled from the first frame using the complete diffeomorphism.

5.3. Super-Resolution of Diffeomorphisms Using the BHF Refinement Algorithm

In this subsection, we apply the BHF refinement algorithm on the super-resolution of diffeomorphisms. In our first test, a coarse version of the highly distorted diffeomorphism in Subsection 5.1 is used, which is represented with 17 by 17 points. As shown in Figure 7(a), the use of such sparse data causes unnatural jaggy visualization, which is similar to a texture mapping onto a coarse triangular mesh.

Using the BHF refinement algorithm, we refine the coarse mapping into a fine diffeomorphism of 129 by 129



(a) (b)

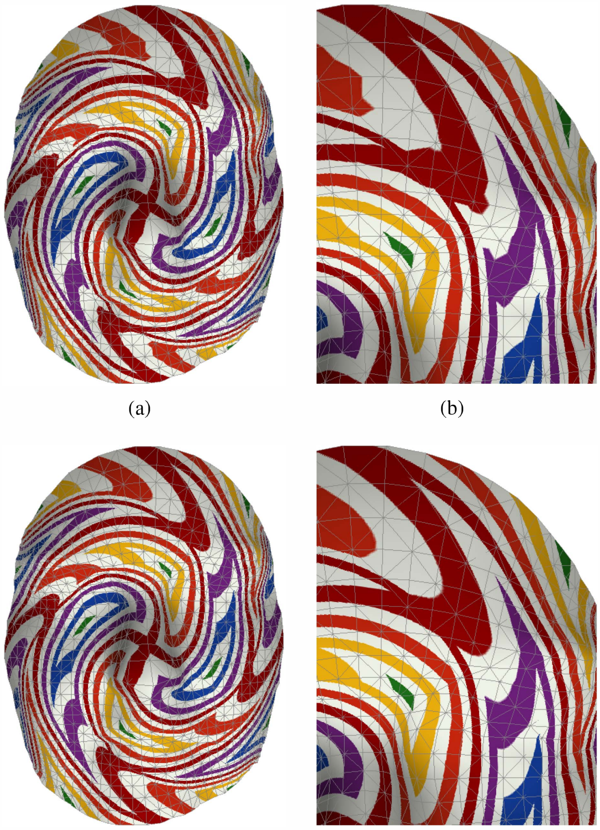
Figure 6. The result of registering frames 1 and 2 using the BHF inpainting algorithm. (a) shows the final registration. (b) shows the complete gingerbread man with the occluded region filled.

(a) (b)

Figure 7. Application of the BHF refinement algorithm on a 2D diffeomorphism. (a) shows a coarse diffeomorphism represented with 17 by 17 points. (a) shows the refinement result to 129 by 129 points using the BHF refinement algorithm.

points by interpolating the original 16 by 16 Beltrami coefficient on each face to a 128 by 128 version, and reconstructing a 129 by 129 diffeomorphism, fixing its values on the coarse 17 by 17 grid. The result of the refinement is shown in Figure 7(b), which is very smooth and almost looks identical to the original high resolution diffeomorphism. This shows that our algorithm can smoothly refine a diffeomorphism even only a tiny fraction of data is available.

Next we demonstrate the effectiveness of the algorithm in a real 3D texture mapping example. In this test, initially we have a face model represented by a 33 by 33 regular triangular mesh and textured with a highly convoluted texture mapping. As we can see in Figure 8(a) and 8(b), the coarse triangulation resulted in a poor visualization of the texture. Using the BHF refinement algorithm for surfaces, we refine this coarse texture map into a fine 129 by 129 texture map using only the initial coarse data. The result shown in Figure 8(c) and 8(d) is very smooth, as if textured using a much higher resolution mapping. This illustrates the effectiveness of our algorithm to represent texture maps in much higher details than the triangular mesh used.



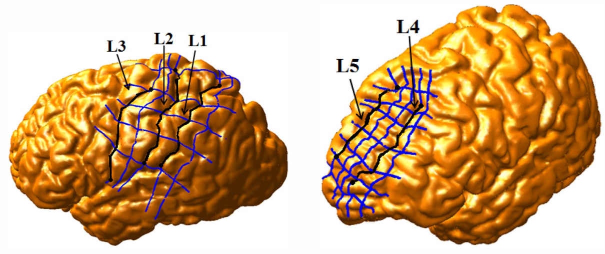
(a) (b) (c) (d)

Figure 8. Application of the BHF refinement algorithm on 3D texture mapping. (a) shows a normally visualized texture mapping on a coarse 33 by 33 mesh. (b) shows a zoom-in version to illustrate its low quality. (c) shows the refined texture mapping on the same mesh after BHF refinement. (d) shows a zoom-in version to illustrate its fine details.

5.4. Application in Cortical Surface Parameterization

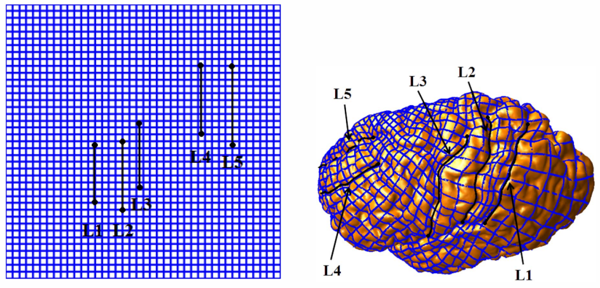
Finally, we apply the BHF inpainting algorithm to efficiently compute a landmark-matching surface parameterization of the cortical surface. In brain imaging, it is often necessary to map feature landmark lines, such as the sulcal and gyral lines highlighted by doctors, onto consistent locations of a parameter domain, where further analysis takes place. On convoluted surfaces such as the cortical surface, this involves mapping a large number of landmark curves onto consistent locations, making the solution infeasible.

We demonstrate the use of the BHF inpainting algorithm to solve this problem efficiently. Instead of solving the problem with all feature curves at once, we first compute consistent parameterizations of a few landmarks lines at a time. As shown in Figure 9, we divide a problem involving 5 feature curves on a cortical surface into 2 subproblems involving 2 and 3 feature curves respectively, and



(a) (b)

Figure 9. Breaking down a parameterization problem of the cortical surface into 2 subproblems and solve all problems simultaneously. (a) shows the first subproblem involving 3 feature curves. (b) shows the second subproblem involving 2 feature curves.



(a) (b)

Figure 10. The final result for the large parameterization problem. (a) shows a global parameter domain containing 2 local parameter domains. (b) shows the global parameterization computed using the BHF inpainting algorithm.

solve each subproblem with existing registration algorithms [10]. This breaks down the large problem into easier subproblems which can be solved simultaneously.

After each subproblem is solved, we place the local parameterizations into a larger parameter domain. Then a global parameterization that extends the two local parameterizations is computed using the BHF inpainting algorithm, as shown in Figure 10. The resulting global parameterization is a landmark-matching parameterization that smoothly extends the existing local parameterizations.

6. Conclusion and Future Work

In this paper, we derived an exact formula for the adjustment of diffeomorphisms using BHF, under the practical assumption that the diffeomorphism is piecewise linear on a triangular mesh. Using this algorithm, we further proposed 2 algorithms for the inpainting and refinement of diffeomorphisms. We applied these algorithms on the inpainting of a highly distorted diffeomorphism, the inpainting of image sequences of deforming shapes, the super-resolution of diffeomorphisms and the global parameterization of cortical surfaces by combining local parameterizations. Results show that our algorithms always reproduce fine details from diffeomorphisms with very low resolution or missing parts, where some other methods failed. This demonstrates the great versatility of our proposed algorithms on areas from texture mapping to video processing, and from computer graphics to medical imaging. In the future, we plan to improve the efficiency of algorithms by implementing them on GPUs and propose more application of them.

References

[1] L. Ahlfors. *Lectures on Quasiconformal Mappings*. American Mathematics Society, 1966.

[2] A. Belahmidi and F. Guichard. A partial differential equation approach to image zoom. *Proc. Int. Conf. on Image Processing*, 2004.

[3] M. Bertalmio, A. Bertozzi, and G. Sapiro. Navier-stokes, fluid dynamics, and image and video inpainting. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 355–362, 2001.

[4] W. Carey, D. Chuang, and S. Hemami. Regularity-preserving image interpolation. *IEEE Transactions on Image Processing*, 8:1293–1297, 1997.

[5] Y. Cha and S. Kim. Edge-forming methods for image zooming. *Journal of Mathematical Imaging and Vision*, 25:353–364, 2006.

[6] T. Chan and J. Shen. Mathematical models for local nontexture inpaintings. *SIAM J. Appl. Math*, 62:1019–1043, 2002.

[7] S. Durrleman, X. Pennec, A. Trouve, P. Thompson, and N. Ayache. Inferring brain variability from diffeomorphic deformations of currents: An integrative approach. *Medical Image Analysis*, 12:626–637, 2008.

[8] A. Leow, C. Yu, S. Lee, S. Huang, H. Protas, R. Nicolson, K. Hayashi, A. Toga, and P. Thompson. Brain structural mapping using a novel hybrid implicit/explicit framework based on the level-set method. *NeuroImage*, 24(3):910–927, 2005.

[9] B. Levy, S. Petitjean, N. Ray, and J. Maillot. Least squares conformal maps for automatic texture atlas generation. *ACM Transactions on Graphics (TOG)*, 21(3):362–371, 2002.

[10] L. Lui, S. Thiruvenkadam, Y. Wang, T. Chan, and P. Thompson. Optimized conformal parameterization of cortical surfaces using shape based matching of landmark curves. *SIAM Journal of Imaging Science*, 3(1):52–78, 2010.

[11] L. Lui, T. Wong, X. Gu, P. Thompson, T. Chan, and S. Yau. Shape-based diffeomorphic registration on hippocampal surfaces using beltrami holomorphic flow. In *Proceedings of the 13th MICCAI'10: Part II*, pages 323–330, 2010.

[12] L. Lui, T. Wong, W. Zeng, X. Gu, P. Thompson, T. Chan, and S. Yau. Detecting shape deformations using yamabe flow and beltrami coefficents. *Journal of Inverse Problem and Imaging (IPI)*, 4(2):311–333, 2010.

[13] Y. Wang, L. Lui, X. Gu, K. Hayashi, T. Chan, A. Toga, P. Thompson, and S. Yau. Brain surface conformal parameterization using riemann surface structure. *IEEE Transactions on Medical Imaging*, 26(6):853–865, 2007.