Solving PDEs on Manifolds with Global Conformal Parametrization

Lok Ming Lui, Yalin Wang, and Tony F. Chan

Mathematics Department, UCLA
{malmlui, ylwang, chan}@math.ucla.edu

Abstract. In this paper, we propose a method to solve PDEs on surfaces with arbitrary topologies by using the global conformal parametrization. The main idea of this method is to map the surface conformally to 2D rectangular areas and then transform the PDE on the 3D surface into a modified PDE on the 2D parameter domain. Consequently, we can solve the PDE on the parameter domain by using some well-known numerical schemes on $\mathbb{R}^2$. To do this, we have to define a new set of differential operators on the manifold such that they are coordinates invariant. Since the Jacobian of the conformal mapping is simply a multiplication of the conformal factor, the modified PDE on the parameter domain will be very simple and easy to solve. In our experiments, we demonstrated our idea by solving the Navier-Stoke's equation on the surface. We also applied our method to some image processing problems such as segmentation, image denoising and image inpainting on the surfaces.

1 Introduction

Image processing on the surface has become more and more important in medical imaging, computer graphics and computer vision. Many image processing techniques involve solving a partial differential equation (PDE) on the surface. In 2D image processing, variational approaches have been widely used. The minimization procedure can be reformulated as a partial differential equation, using the Euler-Lagrange equation. In order to extend the 2D image processing techniques to 3D, we therefore need to formulate a technique to solve PDEs on surfaces with arbitrary topologies.

In this paper, we propose to solve PDEs on surfaces by using the global conformal parametrization. The main idea is to map the surface conformally to the 2D rectangles with the minimum number of coordinates patches. The problem can then be solved by solving a modified PDE on the 2D parameter domain. To do this, we have to define a new set of differential operators on the manifold. Once a PDE on the 3D surface is reformulated to the corresponding PDE on the 2D domain, we can solve the PDE on 2D by using some well-known numerical schemes. Since the Jacobian of the conformal mapping is simply a multiplication of the conformal factor, the modified PDE on the parameter domain will be very simple and easy to solve.

N. Paragios et al. (Eds.): VLSM 2005, LNCS 3752, pp. 307–319, 2005.

Recently, some level set based PDE solving approaches have been proposed ([5,6]). Compared with the level set based approaches, we explicitly describe the manifold by the conformal parametrization, instead of the implicit representation of the level set function. We use a new set of differential operators on the manifold, without doing a projection of the Euclidean differential operators. Our method considers maps which are defined only on the manifold so we do not need to extend maps to a narrow band of the surface.

2 Previous Work

Several research groups have reported works on solving PDEs on the surface. Turk [1] proposed to generate textures on arbitrary surfaces using reaction-diffusion, which require to solve PDE on the surface. Dorsey et al. [2] propose to solve PDEs on the surface for virtual weathering. Both of them solved the PDE directly on the triangulated surface, which involve the discretization of the equations in general polygonal grid. Stam [3] proposed to simulate fluid flow on the surface via solving the Navier-Stokes equation. He achieved this by combining the two dimensional stable fluid solver with an atlas of parametrizations of a Catmull-Clark surface. Clarenz et al. [4] has proposed an algorithm for solving finite element based PDEs on point surfaces. They constructed a number of local FE matrices that represent the surface properties over small point neighborhoods. These matrices are next assembled in a single matrix that allows PDE discretization and solving on complete surface. Sapiro et al. [5] [6] implemented a framework for solving PDEs on the surface via the level set method. They represented the surface implicitly by the zero-level set of an embedding function and extend the data on the surface to the 3D volume. This allowed them to perform all the computation on the fixed Cartesian grid.

3 Mathematical Theory

3.1 Computation of Conformal Parameterization

A diffeomorphism $f : M \to N$ is a *conformal mapping* if it preserves the first fundamental form up to a scaling factor (the conformal factor). Mathematically, this means that $ds_M^2 = \lambda f^*(ds_N^2)$, where ds_M^2 and ds_N^2 are the first fundamental form on M and N respectively and λ is the conformal factor. (See [7]) For a diffeomorphism between two genus zero surfaces, a map is conformal if and only if it minimizes the harmonic energy,$E_{harmonic}$. However, this is not true for surfaces with genus equal to one or higher.

For high genus surfaces, Gu et. al [8] has proposed an efficient approach to parameterize surfaces conformally to the 2D rectangles. This approach is based on the homology group theory, the cohomology group theory and the Hodge theory. We can summarize the algorithm with the following five steps. For details, please refer to [8].

Step 1: Given a high genus surface, find the homology basis $\{\xi_1, ..., \xi_{2g}\}$ of its homology group.
Step 2: Given the homology basis $\{\xi_1, ..., \xi_{2g}\}$, compute its dual basis $\{w_1, ..., w_{2g}\}$ which is called the cohomology basis.
Step 3: Diffuse the cohomology basis elements to harmonic 1-forms. This can be done by solving the following simultaneous equations:
(1) $dw = 0$ (*closedness*) (2) $\Delta w = 0$ (*harmonity*) (3) $\int_{\xi_i} w_j = \delta_{ij}$ (*duality*)
The existence of solution is guaranteed by Hodge theory.
Step 4: Compute the Hodge star conjugate $\{^*w_1, ...,^* w_{2g}\}$ of $\{w_1, ..., w_{2g}\}$
Step 5: Integrate the holomorphic 1-form and get the conformal mapping: $f(x) = \int_\gamma w + i^*w$, where $w = \Sigma\lambda_i w_i$

The above five steps allow us to compute a conformal parametrization from the surface onto the 2D domain. (See Figure 1)

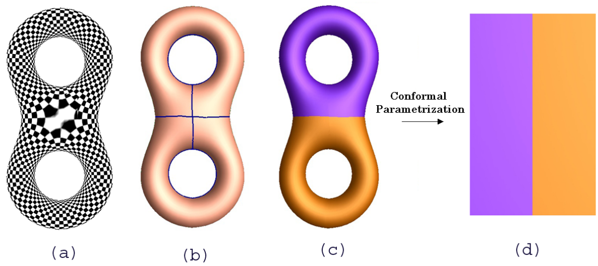

Fig. 1. Conformal parametrization of a high genus surface onto the 2D rectangles

3.2 Differential Operators on Manifolds

Many physical phenomenon can be explained via PDEs. In image processing, variational approaches are often used, which induces PDE solving. Therefore, it is important to define a set of partial differential operators on general manifolds. In this section, the partial differential operators on manifolds and the covariant differentiation on tensor fields will be discussed.

Let M be a manifold and $\phi : \mathbb{R}^2 \to M$ be the global conformal parametrization of M. With the conformal parametrization, we can do calculus on surfaces similar to what we do on $\mathbb{R}^2$. Suppose $f : M \to \mathbb{R}$ is a smooth map. We will firstly define partial derivative, $D_{x_i} f$, of f. On $\mathbb{R}^2$, we usually define the partial derivative, $\frac{\partial g}{\partial x_i}$, by taking limit. For example, $\frac{\partial g}{\partial x} = lim_{\triangle x \to 0} \frac{f(x+\triangle x, y) - f(x,y)}{\triangle x}$. With the conformal parametrization, we can define the partial derivative on scalar functions in the same manner. Because of the stretching effect, we have to modify the denominator in the limit a little bit. Specifically, we define (1):

$$D_x f = lim_{\triangle x \to 0} \frac{f\circ\phi(x+\triangle x, y) - f\circ\phi(x,y)}{dist(x+\triangle x, x)} = lim_{\triangle x \to 0} \frac{f\circ\phi(x+\triangle x, y) - f\circ\phi(x,y)}{\sqrt{\lambda}\triangle x} = \frac{1}{\sqrt{\lambda}} \frac{\partial f\circ\phi}{\partial x},$$

where λ is the conformal factor. $D_y f$ is defined similarly.

Now, the gradient of a function f, $\nabla_M f$, is characterized by: $df(Y) =< \nabla_M f, Y >$. Simple checking gives us: $\nabla_M f = \sum_{i,j} g^{ij} \partial_i f \partial_j$, where (g^{ij}) is the inverse of the Riemannian metric (g_{ij}).

With the conformal parametrization, we can define the *gradient* of f similar to the definition on $\mathbb{R}^2$. Namely, (2):

$$\begin{aligned} &\nabla_M f = D_x f \mathbf{i} + D_y f \mathbf{j} \text{ where} \\ &(\mathbf{i}, \mathbf{j}) = (\tfrac{\partial}{\partial x} / \sqrt{< \tfrac{\partial}{\partial x}, \tfrac{\partial}{\partial x} >} \ , \ \ \tfrac{\partial}{\partial y} / \sqrt{< \tfrac{\partial}{\partial y}, \tfrac{\partial}{\partial y} >}) \\ &\qquad = \tfrac{1}{\lambda} [\tfrac{f \circ \phi}{\partial x} \tfrac{\partial}{\partial x} + \tfrac{f \circ \phi}{\partial y} \tfrac{\partial}{\partial y}] \end{aligned}$$

Suppose $h : M \to \mathbb{R}$ is a smooth function. With this definition of gradient, we still have the following useful fact as in $\mathbb{R}^2$:

$$\begin{aligned} \text{Length of } h^{-1}(0) &= \textstyle\int_M \delta(h) \sqrt{< \nabla_M h, \nabla_M h >} dS \\ &= \textstyle\int_M \sqrt{< \nabla_M H(h), \nabla_M H(h) >} dS \\ &= \textstyle\int_{\mathbb{C}} \delta(h \circ \phi) \sqrt{\lambda}\ ||\nabla h \circ \phi|| dxdy \\ &= \textstyle\int_{\mathbb{C}} \sqrt{\lambda}\ ||\nabla H(h \circ \phi)|| dxdy \qquad (3) \end{aligned}$$

where H is the Heaviside function. (See Appendix)

Next, we need to give a well-defined definition of differential operator on vector field. This is based on the tensor calculus [9]. In Euclidean space, we conventionally differentiate the vector field $(x_1(t), ..., x_n(t))$ on a curve pointwisely to get $(x_1'(t), ..., x_n'(t))$. However, pointwise differentiation does not work for general manifolds because it is not coordinate invariant. For example, consider the parameterized circle in the plane given in Euclidean coordinate $(x(t), y(t)) = (cos\ t, sin\ t)$. Its acceleration at time t is $(-cos\ t, -sin\ t)$. However, in polar coordinates, the same curve is described as $(r(t), \theta(t)) = (1, t)$ and the acceleration is $(0, 0)$.

In order to differentiate a vector field $\overrightarrow{V}(t)$ along a curve, we have to write a difference quotient involving $\overrightarrow{V}(t)$ and $\overrightarrow{V}(t_0)$ which live on two different tangent spaces. Therefore, it is not appropriate to subtract. Secondly, even if we can differentiate the vector field pointwise, it is not guaranteed that the "derivative" is a tangent vector on the manifold.

We therefore need to define a differential operator on the vector field, which is coordinate invariant. This can be done by covariant differentiation $\nabla_X Y$, where X is called the direction of the differentiation. To do so, we need to develop a way to compare tangent vectors at different points. On $\mathbb{R}^2$, we usually parallelly translate the vectors and subtract. But on general manifolds, we do not have the concept of parallel translation. We say that a vector field $\overrightarrow{V}(\gamma(t))$ along a curve $\gamma(t)$ is *parallel* if: $D_t \overrightarrow{V}(\gamma(t))$ = orthogonal projection of $\frac{d}{dt} \overrightarrow{V}(\gamma(t))$ onto the tangent space $= 0$. We have the following important fact:

Parallel Translation : Given a curve $\gamma : I \to M$ and a vector $\overrightarrow{V}_0 \in T_{\gamma(t_0)} M$, there exists a unique parallel vector field $\overrightarrow{V}$ along γ with $\overrightarrow{V}(t_0) = V_0$.

With the parallel translation along a curve γ, we can define an operator: $P^{\gamma}_{t_0 t_1} : T_{\gamma(t_0)}M \to T_{\gamma(t_1)}M$ by setting $P^{\gamma}_{t_0 t_1}(\overrightarrow{V}_0) = V(t_1)$ where V is the parallel vector field along γ with $\overrightarrow{V}(0) = \overrightarrow{V}_0$. This is clearly an linear isomorphism.

Now, we can define $\nabla_X Y|_p$ as follows: let $\gamma : [0,1] \to M$ be a curve such that $\gamma(0) = p$ and $\gamma'(0) = Y|_p$. We define (4): $\nabla_X Y|_p = lim_{t\to 0} \frac{P^{\gamma}_{0t}{}^{-1} Y(\gamma(t)) - Y(p)}{t}$

The covariant derivative satisfies the following properties:

(P1) $\nabla_{fX_1+gX_2} Y = f\nabla_{X_1} Y + g\nabla_{X_2} Y$ for $f, g \in C^\infty(M)$
(P2) $\nabla_X(aY_1 + bY_2) = a\nabla_X Y_1 + b\nabla_X Y_2$, $a, b \in \mathbb{R}$
(P3) $\nabla_X(fY) = f\nabla_X Y + (Xf)Y$ for $f \in C^\infty(M)$.

The above properties will determine the expression of the covariant derivative. Given a Riemannian manifold (M, g) where $g = (g_{ij})$ is the Riemannian metric. Suppose $\{\partial_i\}$ is the coordinate basis of the vector field. A simple verification will tell us the covariant derivative can be calculated by the following formula:

$< \nabla_{\partial_i}\partial_j, \partial_l >= 1/2(\partial_i g_{jl} + \partial_j g_{li} - \partial_l g_{ij})$

Simple calculation gives (5):

$\nabla_{\partial_i}\partial_j = \Gamma^m_{ij}\partial_m \ where \quad \Gamma^m_{ij} = 1/2\ g^{ml}(\partial_i g_{jl} + \partial_j g_{li} - \partial_l g_{ij})$

Suppose now the parametrization is conformal. The Riemannian metric (g_{ij}) is simply $(g_{ij}) = \lambda I$, where λ, I are the conformal factor and the identity matrix respectively. We then have the following formula (6):

$\nabla_{\partial_x}\partial_x = \frac{1}{2\lambda}\ \frac{\partial\lambda}{\partial x}\ (\partial_x - \partial_y)$; $\nabla_{\partial_y}\partial_y = \frac{1}{2\lambda}\ \frac{\partial\lambda}{\partial y}\ (-\partial_x + \partial_y)$; $\nabla_{\partial_x}\partial_y = \frac{1}{2\lambda}\ (\frac{\partial\lambda}{\partial y}\partial_x + \frac{\partial\lambda}{\partial x}\partial_y)$

With this formula and the above properties (P1)-(P3), we can calculate $\nabla_X Y$ easily. Thus for example:

$$\begin{aligned}\nabla_{a\frac{\partial}{\partial x}+b\frac{\partial}{\partial y}}\frac{\partial}{\partial y} &= a\nabla_{\frac{\partial}{\partial x}}\frac{\partial}{\partial y} + b\nabla_{\frac{\partial}{\partial y}}\frac{\partial}{\partial y} = \frac{a}{2\lambda}(\frac{\partial\lambda}{\partial y}\partial_x + \frac{\partial\lambda}{\partial x}\partial_y) + \frac{b}{2\lambda}(-\frac{\partial\lambda}{\partial y}\partial_x + \frac{\partial\lambda}{\partial y}\partial_y)\\ &= \frac{1}{2\lambda}\frac{\partial\lambda}{\partial y}(a-b)\partial_x + \frac{1}{2\lambda}(a\frac{\partial\lambda}{\partial x} + b\frac{\partial\lambda}{\partial y})\partial_y\end{aligned}$$

With the definition of covariant derivative, we can define the divergence of a vector field $\sum_{i=1}^{2} X_i \frac{\partial}{\partial x_i}$. The idea is to take the covariant derivative of X_i with respect to x_i and sum them up, we then get a scalar which is called the *divergence* of the vector field. For conformal parametrization, we have (7):

$div_M(\Sigma^2_{i=1} X_i \frac{\partial}{\partial x_i}) = \sum_{i=1}^{2} \frac{1}{\lambda}\partial_i(X_i\lambda)$

If we calculate the divergence of $\nabla_M f$, we get the Laplacian of f:

$$\triangle_M f = \sum_{j=1}^{2}(1/\lambda)\ \partial_j\partial_j f \qquad (8)$$

Interestingly, with the above definitions, we still have the integration by part formula and the Green's formula:

$\int_M < \nabla_M u, X > dV = -\int_M u div_M X dV + \int_{\partial M} u < X, \overrightarrow{N} > d\widetilde{V}$, $\overrightarrow{N}$ is the unit normal vector. (Integration by part) (9)

$\int_M (u\triangle_M v - v\triangle_M u)dV = \int_{\partial M}(u\nabla_M v \cdot \overrightarrow{N} - v\nabla_M u \cdot \overrightarrow{N})d\widetilde{V}$
(Green's Theorem) (10)

Also, suppose C is a curve represented by the zero level set of $\phi : M \to \mathbb{R}$. We have the following useful property, similar to that on $\mathbb{R}^2$:

Geodesic curvature of $C = div_M(\frac{\nabla_M \phi}{||\nabla_M \phi||})$ (11) (See Appendix)

4 Navier-Stokes Equation on Surfaces

In this section, we will illustrate our idea by solving the Navier-Stokes equation on surfaces with arbitrary topologies. The idea is to parameterize the Riemann surface conformally onto the rectangular parameter domain based on the holomorphic differential one forms (Section 3.1). We then use the stable fluid solver [10] on the 2-D domain to solve the problem.

On $\mathbb{R}^2$, fluid flow is governed by the Navier-Stokes equation. For incompressible fluid flow, we have the following (*):

$\frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u} \cdot \bigtriangledown)\mathbf{u} + v\nabla^2\mathbf{u} + \mathbf{f}$ and $\nabla \cdot \mathbf{u} = 0$ (**imcompressibility**) (12)

where $\mathbf{u} = (u^1, u^2)$ is the fluid's velocity, v is the viscosity and $\mathbf{f} = (f^1, f^2)$ are external forces.

We can simulate the fluid flow as follow: we first use the stable fluid solver to solve (*). Then update the position of the fluid by $\mathbf{x}^{new} = \mathbf{x}^{old} + \mathbf{u}dt$, where $\mathbf{x}^{new}$ = updated position of the fluid particle and $\mathbf{x}^{old}$= previous position of the fluid particle.

To simulate fluid flow on the Riemann surface, we have to modify the 2D Navier-Stokes equation by the manifold version of gradient and lapacian. Replacing the gradient and laplacian by the manifold version of gradient and laplacian, we get the corresponding Navier-Stokes equation for the Riemann surface M:

$$\frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u} \cdot \nabla_M)\mathbf{u} + v\triangle_M \mathbf{u} + \mathbf{f} \qquad (13)$$

Let ϕ be the conformal parametrization of M and $\mathbf{w} = \mathbf{u} \circ \phi$. We have:

$$\frac{\partial \mathbf{w}}{\partial t} = -\frac{1}{\lambda}(\mathbf{w} \cdot \bigtriangledown)\mathbf{w} + \frac{1}{\lambda}v\triangle\mathbf{w} + \mathbf{f} \qquad (14)$$

Note that it is really the governing equation for fluids on the manifold — it is the same physics that we know. For detail, see Aris's book. [11]

We can next use the Stable Fluid Solver introduced by Stam to solve the Navier-Stokes equation. We describe the algorithm as follow:

Step 1: (Adding force) We solve: $\frac{\partial \mathbf{w_1}}{\partial t} = \mathbf{f}$. The iterative scheme is: $\mathbf{w}_1 = \mathbf{w}_0 + dt\mathbf{f}$
Step 2: (diffusion equation) We solve: $\frac{\partial \mathbf{w_2}}{\partial t} = \frac{1}{\lambda}v\triangle\mathbf{w}_1$. We use a simple implicit solver to get the iterative scheme: $(I - dt\frac{1}{\lambda}v\triangle)\mathbf{w}_2 = \mathbf{w}_1$.

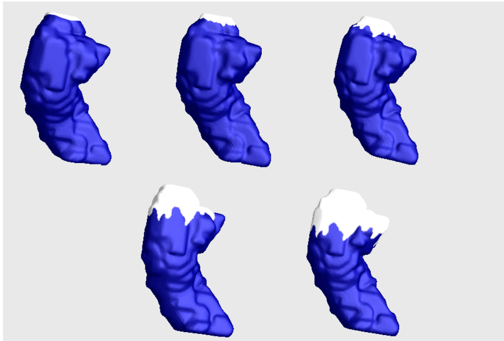

Fig. 2. Simulation of snow flowing down the surface

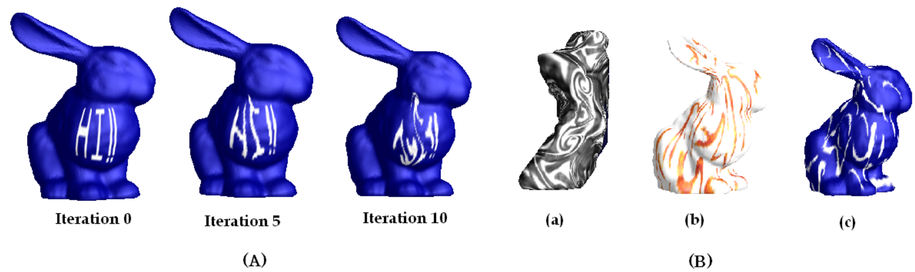

Fig. 3. Fluid flow on the surface in (A). Navier-Stoke's equation for surface decoration in (B).

Step 3: (advection equation) We solve: $\frac{\partial \mathbf{w}_3}{\partial t} = -\frac{1}{\lambda}(\mathbf{w}_2 \cdot \bigtriangledown)\mathbf{w}_3$. We use a semi-Lagrangian to get an iterative scheme: $\mathbf{w}_3 = \mathbf{w}_2(\mathbf{x} - dt\frac{1}{\lambda}\mathbf{w_2}(\mathbf{x}))$

Step 4: (projection) We project $\mathbf{w}$ onto its imcompressible (divergence free) component. For this, we first solve the Poisson equation: $\triangle\varphi = \nabla \cdot \mathbf{w}_3$
We then update: $\mathbf{w}_4 = \mathbf{w}_3 - \frac{1}{\lambda}\nabla\varphi$. Update $\mathbf{w} = \mathbf{w}_4$.

Step 5: (Update fluid position) Update $\mathbf{x}$ by $\mathbf{x}^{new} = \mathbf{x}^{old} + \mathbf{w}dt$

As an example, we simulate the snow flowing down the surface based on the Navier-Stokes equation in Figure 2. In Figure 3 (A), we simulate fluid flow on a bunny surface by adding a S-shaped force. In Figure 3 (B), we simulate fluid flow on surfaces for surface decoration.

5 Image Processing on Surfaces

5.1 Chan-Vese Segmentation on Surfaces

Segmentation is an important technique in image processing to extract useful region. One commonly used technique is the Chan-Vese (CV) segmentation technique, which is based on the level set method [12]. Here, we will extend the CV segmentation on $\mathbb{R}^2$ to arbitrary Riemann surface M.

Let $\phi : \mathbb{R}^2 \to M$ be the conformal parametrization of the surface M. We propose to minimize the following energy functional: (15)

$F(c_1, c_2, \psi) = \int_M (u_0 - c_1)^2 H(\psi) dS + \int_M (u_0 - c_2)^2 (1 - H(\psi)) dS + \nu length\ of$ $\psi^{-1}(\{0\})$ $= \int_M (u_0 - c_1)^2 H(\psi) dS + \int_M (u_0 - c_2)^2 (1 - H(\psi)) dS + \nu \int_M |\nabla_M H(\psi)|_M dS,$

where $\psi : M \to \mathbb{R}$ is the level set function and $| \cdot |_M = \sqrt{< \cdot, \cdot >}$.

With the conformal parametrization, we have:

$F(c_1, c_2, \psi) = \int_{\mathbb{R}^2} \lambda (u_0 \circ \phi - c_1)^2 H(\psi \circ \phi) dxdy + \int_{\mathbb{R}^2} \lambda (u_0 \circ \phi - c_2)^2 (1 - H(\psi \circ \phi)) dxdy$
$+\nu \int_{\mathbb{R}^2} \sqrt{\lambda} |\nabla H(\psi \circ \phi)| dxdy,$

For simplicity, we let $\zeta = \psi \circ \phi$ and $w_0 = u_0 \circ \phi$. Fixing ζ, we must have:

$c_1(t) = \frac{\int_\Omega w_0 H(\zeta(t,x,y)) \lambda dxdy}{\int_\Omega H(\zeta(t,x,y)) \lambda dxdy}$ (16)

$c_2(t) = \frac{\int_\Omega w_0 (1 - H(\zeta(t,x,y)) \lambda dxdy}{\int_\Omega (1 - H(\zeta(t,x,y))) \lambda dxdy}$ (17)

Fixing c_1, c_2, the Euler-Lagrange equation becomes:

$$\frac{\partial \zeta}{\partial t} = \lambda \delta(\zeta) [\ \nu \frac{1}{\lambda} \nabla \cdot (\sqrt{\lambda} \frac{\nabla \zeta}{||\nabla \zeta||}) - (w_0 - c_1)^2 + (w_0 - c_2)^2] \qquad (18)$$

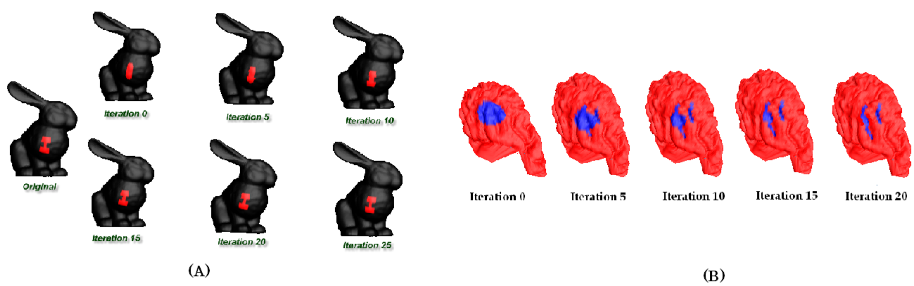

Fig. 4. CV segmentation on surface in (A). CV segmentation on surface for sulci extraction on the cortical surface in (B).

In Figure 4(A), we illustrate the CV segmentation on the bunny surface. As shown in the figure, the initial contour evolves to the original image in a few iterations. One application of CV segmentation is to extract the sulci position on the cortical surface. The sulci position is usually the high curvature region. We can consider the intensity as a function of curvatures, such as Mean curvatures and Gaussian curvatures. In Figure 4 (B), we illustrate how we can extract the sulci position on the cortical surface using CV segmentation. Here, we consider the mean curvature as the intensity.

5.2 Image Denoising on Surfaces

One important task of surface processing is the restoration or reconstruction of a true image u from an observed image f. In many applications, the measure image is polluted by noise or blur. The distorted image need to be denoised to understand the useful part of the image. On $\mathbb{R}^2$, Rudin, Osher and Fatemi (ROF) has proposed the following model [13]:

$$\mathbf{inf}_u F(u) = \int_{\Omega} |\nabla u| + \nu |f - u|^2 dxdy \qquad (19)$$

We proceed to extend the ROF on 2D to any surface M with arbitrary topologies. Let ϕ be the conformal parametrization of M and $\zeta = u \circ \phi$. Following the 2D ROF model, we propose to minimize the following energy:

$\mathbf{inf}_u F(u) = \int_M |\nabla_M u|_M + \nu |f - u|^2 dS \qquad (20)$ or

$\mathbf{inf}_u F(\zeta) = \int_{\mathbb{R}^2} \sqrt{\lambda} |\nabla \zeta| + \lambda \nu |f - \zeta|^2 dxdy \qquad (21)$

We can minimize the above energy by solving the Euler-Lagrange equation:

$\frac{\partial u}{\partial t} = 2\nu(f - u) + div_M(\frac{\nabla_M u}{|\nabla_M u|_M}) \qquad (22)$
or

$\frac{\partial \zeta}{\partial t} = 2\nu(f - \zeta) + \frac{1}{\lambda} div(\sqrt{\lambda} \frac{\nabla \zeta}{|\nabla \zeta|})$ on the rectangular parameter domain. (23)

and $\frac{\partial \zeta}{\partial \vec{n}} = 0$ on the boundary (24).

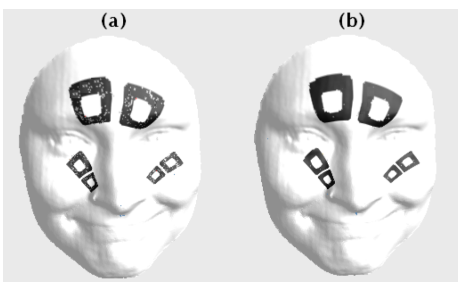

Fig. 5. ROF denoising on the human face

As an example, we use the ROF model to denoise the dirty scar on the human face in Figure 5. It is observed that the image can be significantly improved.

5.3 Image Inpainting on Surfaces

Inpainting, originally an artist's work, is the process of filling in the missing or desired image information where it is unavailable. (see Figure 6). Such "defect" domain may be introduced by the aging of the canvas and oil of an ancient painting, and the occlusion by undesired objects in front of a scene of interest. For 2D images, Chan & Shen has introduced an inpainting model via curvature driven diffusion (CDD) [14]. We are going to extend this model to 3D Riemann surfaces.

Suppose Ω is the domain of the image on $\mathbb{R}^2$. Let D be the inpainting domain (the occluded region). The CDD model reads:

$$\frac{\partial u}{\partial t} = \nabla \cdot [\frac{g(|\kappa|)}{|\nabla u|}\nabla u], \qquad x \in D \qquad (25)$$

and $u = u^0, \qquad x \in D^c \qquad (26)$

Here κ denotes the curvature, and g(s) is defined to be zero if $s = 0$ and equal to infinity if $s = \infty$.

The curvature κ at a pixel x is the scalar curvature of the isophote through it and is given by: $\kappa = \nabla \cdot (\frac{\nabla u}{|\nabla u|}) \qquad (27)$

Suppose now Ω is the image domain on a Riemann surface M. $D \subset M$ is the inpainting domain. Let ϕ be the conformal parametrization of the surface and let $\zeta = u \circ \phi$. Replacing the gradient and divergence by the manifold version of gradient and divergence, we get the CDD inpainting model for the Riemann surface M:

$$\frac{\partial u}{\partial t} = div_M \cdot [\frac{g(|\kappa|)}{|\nabla_M u|_M}\nabla_M u] = \frac{1}{\lambda}\nabla \cdot [\frac{\sqrt{\lambda}g(|\kappa|)}{|\nabla \zeta|}\nabla \zeta], \quad x \in \phi^{-1}(D) \quad (29)$$

and $\zeta = \zeta^0, \qquad x \in \phi^{-1}(D^c) \quad (30)$

The curvature κ at a pixel x is given by:

$$\kappa = div_M \cdot (\frac{\nabla_M u}{|\nabla_M u|_M}) = \frac{1}{\lambda}\nabla \cdot (\sqrt{\lambda}\frac{\nabla \zeta}{|\nabla \zeta|}) \qquad (31)$$

In Figure 6, we illustrate the image inpainting on the human face. In (a), some region of the image is occluded. In (b), the image is effectively restored using the curvature driven diffusion inpainting technique.

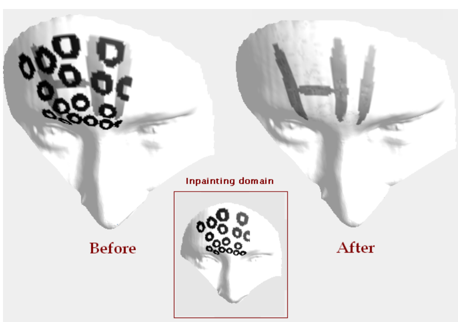

Fig. 6. Curvature driven diffusion inpainting on the human face

6 Conclusion and Future Work

In this paper, we propose a method to solve partial differential equations on surface with arbitrary topologies. The idea is to map the surface conformally onto a simple parameter domain, namely, the 2D rectangle. We can next transform the PDE on the surface into a modified PDE on the 2D domain. We can then solve the PDE with the well-developed numerical schemes on $\mathbb{R}^2$. With the conformal parametrization, the differential operators defined on the surface closely resemble to the usual Euclidean counterpart, except for a multiplication of the conformal factor. Also, the parametrization of the surface using holomorphic 1-form allows us to parametrize (high genus)surface with the minimum number of coordinate chart. Thus, less boundary adjustment are needed when solving the PDEs on the surface. Finally, unlike the conventional way that projects the differential operators on $\mathbb{R}^3$ onto the surface, we directly define differential operators on the parameter domain without the need of doing projection. We thus avoid the complicated projection operation in our algorithm. We have illustrated our method by solving the Navier-Stokes equation on the surface. We also tested our method by solving some PDE-based surface processing problems, such as surface segmentation and surface denoising. In the future, we will look for more applications of solving PDEs on the surface.

References

1. Turk, G.: Generating textures on arbitrary surfaces using reaction-diffusion. Computer Graphic **25** (1991) 289–298
2. Dorsey, J., Hanrahan, P.: Digital materials and virtual weathering. Scientific American **282** (2000) 46–53
3. Stam, J.: Flows on surfaces of arbitrary topology. Proceedings of ACM SIGGRAPH 2003 **22** (2003) 724–731
4. Clarenza, U., Rumpfa, M., Teleaa, A.: Surface processing methods for point sets using finite elements. Computers and Graphics **28** (2004) 851–868
5. Bertalmio, M., Cheng, L.T., Osher, S., Sapiro, G.: Variational problems and partial differential equations on implicit surfaces. Journal of Computational Physics **174** (2001) 759 – 780
6. Memoli, F., Sapiro, G., Thompson, P.: Implicit brain imaging. Neuroimage **23** (2004) 179–188
7. Schoen, R., Yau, S.T.: Lectures on Harmonic Maps. International Press (1997)
8. Gu, X., Yau, S.T.: Global conformal surface parameterization. ACM Symposium on Geometry Processing 2003 (2003)
9. Syngen, J., Schild, A.: Tensor Calculus. Dover Publication (1949)
10. Stam, J.: Stable fluids. Proceedings of ACM SIGGRAPH 1999 (1999) 121–128
11. Aris, R.: Vectors, Tensors and the Basic Equations of Fluid Mechanics. Dover Publication (1989)
12. Vese, L.A., Chan, T.F.: Multiphase level set framework for image segmentation using the mumford and shah model. International Journal of Computer Vision **50** (2002) 271–293
13. Rudin, L., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. Physica D: Nonlinear Phenomena **60** (1992) 259–268
14. Chan, T.F., Shen, J.: Non-texture inpainting by curvature-driven diffusions (cdd). J. Visual Comm. Image Rep. **12** (2001) 436–449

Appendix

Claim : Suppose $h : M \to \mathbb{R}$ is a smooth function. Then:
Length of $h^{-1}(0) = \int_M \sqrt{< \nabla_M H(h), \nabla_M H(h) >} dS = \int_{\mathbb{C}} \sqrt{\lambda}\ ||\nabla H(h \circ \phi)|| dxdy$

Proof :

Recall that the Co-area formula reads:
$\int_{\Omega \subset \mathbb{R}^2} f(x,y) |\nabla u| dxdy = \int_{\mathbb{R}} \int_{\{u(x,y)=r\}} f(x,y) d\mathcal{H} dr$
where $\mathcal{H}$ is the Hausdorff measure.

Let ϕ be the conformal parametrization of the surface M and $\zeta = u \circ \phi$.
Then,
$\int_M |\nabla_M H(u)|_M dS = \int_{\mathbb{R}^2} \delta(\zeta) |\nabla \zeta| \sqrt{\lambda} dxdy = \int_{\mathbb{R}} \int_{\{\zeta(x,y)=r\}} \sqrt{\lambda} \delta(\zeta) ds dr = \int_{\{\zeta(x,y)=0\}} ds$
$= \int_0^1 \sqrt{\lambda} |\mathbf{c}'(t)| dt = \int_0^1 \sqrt{\lambda} |\phi \circ \mathbf{c}'(t)| dt = length\ of \{u = 0\}$
where $\mathbf{c}(t)$ is the parametrization of $\zeta(x,y) = 0$ **Q.E.D.**

Claim : Let $\phi : M \to \mathbb{R}$. The geodesic curvature κ of $\phi^{-1}(\{0\}) = div_M\left(\frac{\nabla_M \phi}{\sqrt{<\nabla_M \phi, \nabla_M \phi>}}\right)$

Proof :

Recall that the geodesic curvature of of a curve $\overrightarrow{\gamma}$

$= \frac{\sqrt{<D_t \dot{\overrightarrow{\gamma}}, D_t \dot{\overrightarrow{\gamma}}>}}{<\dot{\overrightarrow{\gamma}}, \dot{\overrightarrow{\gamma}}>} - \frac{<D_t \dot{\overrightarrow{\gamma}}, \dot{\overrightarrow{\gamma}}>}{<\dot{\overrightarrow{\gamma}}, \dot{\overrightarrow{\gamma}}>^{3/2}} = \frac{<\dot{\overrightarrow{\gamma}}, D_t \dot{\overrightarrow{\gamma}}^{\perp}>}{<\dot{\overrightarrow{\gamma}}, \dot{\overrightarrow{\gamma}}>^{3/2}}$

Let the parametrization of the zero level set of ϕ be $\overrightarrow{\gamma} = (X(t), Y(t))$. Then $\phi(X(t), Y(t)) = 0$.

This implies (1): $< \nabla_M \phi, \dot{\overrightarrow{\gamma}} >= 0$

and (2): $< D_t(\nabla_M \phi), \dot{\overrightarrow{\gamma}} > + < D_t \dot{\overrightarrow{\gamma}}, \nabla_M \phi >= 0$

Now, $D_t \overrightarrow{V}(t) = \sum_{i=1}^2 \sum_{j=1}^2 \sum_{k=1}^2 (\dot{V}_k + \Gamma_{ij}^k \gamma_i V_j) \partial_k$

Thus, for conformal parametrization we have (3):

$D_t \dot{\overrightarrow{\gamma}} = (\ddot{X} + (\frac{1}{2\lambda}\frac{\partial \lambda}{\partial x})(\dot{X}^2 - \dot{Y}^2) - (\frac{1}{\lambda}\frac{\partial \lambda}{\partial y}\dot{X}\dot{Y})\ ,\ \ddot{Y} - (\frac{1}{2\lambda}\frac{\partial \lambda}{\partial y})(\dot{X}^2 - \dot{Y}^2) - (\frac{1}{\lambda}\frac{\partial \lambda}{\partial x})\dot{X}\dot{Y})$

and (4):

$D_t(\nabla_M \phi) = (\dot{\phi_x} + (\frac{1}{2\lambda}\frac{\partial \lambda}{\partial x})({\phi_x}^2 - {\phi_y}^2) - (\frac{1}{\lambda}\frac{\partial \lambda}{\partial y}\phi_x \phi_y)\ ,\ \dot{\phi_y} - (\frac{1}{2\lambda}\frac{\partial \lambda}{\partial y})({\phi_x}^2 - {\phi_y}^2) - (\frac{1}{\lambda}\frac{\partial \lambda}{\partial x})\phi_x \phi_y\ )$

Combining (1), (2), (3), (4), we have: $\dot{X}^2 + \dot{Y}^2 = (1 + (\phi_x/\phi_y)^2)\dot{X}^2$ and $\frac{<D_t \dot{\overrightarrow{\gamma}}^{\perp}, \dot{\overrightarrow{\gamma}}>}{<\dot{\overrightarrow{\gamma}}, \dot{\overrightarrow{\gamma}}>^{3/2}} = \lambda(\dot{X}\ddot{Y} - \dot{Y}\ddot{X})$

$$= -\frac{\lambda}{\phi_y}[\phi_{xx}\dot{X}^2 + 2\phi_{xy}\dot{X}\dot{Y} + \phi_{yy}\dot{Y}^2]\dot{X} - \dot{X}(\dot{X}^2 + \dot{Y}^2)(\frac{1}{2\lambda}\frac{\partial\lambda}{\partial y})$$
$$+\dot{Y}(\dot{X}^2 + \dot{Y}^2)(\frac{1}{2\lambda}\frac{\partial\lambda}{\partial x})$$

So, $\kappa = \frac{<\dot{\vec{\gamma}}, D_t\dot{\vec{\gamma}}^{\perp}>}{<\dot{\vec{\gamma}}, \dot{\vec{\gamma}}>^{3/2}} = \frac{\lambda(\dot{X}\ddot{Y} - \dot{Y}\ddot{X})}{\lambda^{3/2}(\dot{X}^2 + \dot{Y}^2)^{3/2}}$

$$= \frac{1}{\sqrt{\lambda}}(\frac{\phi_{xx}\phi_y^2 - 2\phi_{xy}\phi_x\phi_y + \phi_{yy}\phi_x^2}{(\phi_x^2 + \phi_y^2)^{3/2}}) + \frac{1}{2\lambda^{3/2}}(\phi_x\frac{\partial\lambda}{\partial x} + \phi_y\frac{\partial\lambda}{\partial x})$$
$$= \frac{1}{\sqrt{\lambda}}\nabla\cdot(\frac{\nabla\phi}{|\nabla\phi|}) + \frac{1}{\lambda^{3/2}}\nabla\phi\cdot\nabla\lambda = \frac{1}{\lambda}\nabla\cdot(\lambda(\frac{1/\lambda\nabla\phi}{\sqrt{\lambda|\nabla\phi|^2}}))$$
$$= \frac{1}{\lambda}\nabla\cdot(\lambda(\frac{\nabla_M\phi}{\sqrt{<\nabla_M\phi, \nabla_M\phi>}})) = div_M(\frac{\nabla_M\phi}{\sqrt{<\nabla_M\phi, \nabla_M\phi>}})$$

Q.E.D.