

# A Parameterization-based Numerical Method for Isotropic and Anisotropic Diffusion Smoothing on Non-Flat Surfaces

Anand A. Joshi, David W. Shattuck, Paul M. Thompson and Richard M. Leahy

**Abstract**—Neuroimaging data, such as 3D maps of cortical thickness or neural activation, can often be analyzed more informatively with respect to the cortical surface rather than the entire volume of the brain. Any cortical surface-based analysis should be carried out using computations in the intrinsic geometry of the surface rather than using the metric of the ambient 3D space. We present parameterization-based numerical methods for performing isotropic and anisotropic filtering on triangulated surface geometries. In contrast to existing FEM-based methods for triangulated geometries, our approach accounts for the metric of the surface. In order to discretize and numerically compute the isotropic and anisotropic geometric operators, we first parameterize the surface using a  $p$ -harmonic mapping. We then use this parameterization as our computational domain and account for the surface metric while carrying out isotropic and anisotropic filtering. To validate our method, we compare our numerical results to the analytical expression for isotropic diffusion on a spherical surface. We apply these methods to smoothing of mean curvature maps on the cortical surface, a step commonly required for analysis of gyrification or for registering surface-based maps across subjects.

**Index Terms**—Isotropic, anisotropic diffusion smoothings.  $p$ -harmonic parameterization.

## I. INTRODUCTION

GAUSSIAN kernel smoothing has been widely used in 3D medical imaging as a tool to increase the signal-to-noise ratio and to generate multiresolution or scale-space representations of images [1]. In many brain imaging applications, neuro-anatomical [2], [3], functional [4] or statistical [5], [6] data are defined with respect to the non-Euclidean cortical surface and ideally should be processed with respect to the geometry of the surface. The notion of Gaussian kernel smoothing in a Euclidean space can be generalized to curved surfaces using the heat equation. Thus filtering on curved surfaces can be formulated as the process of heat diffusion by explicitly solving an isotropic diffusion equation with the given data as an initial condition [7]. The drawback of this approach is the complexity of setting up a finite element method (FEM) formulation and difficulty in making a numerically stable scheme [8]. Here we describe an alternative approach to smoothing using the heat equation, based on a parameterized representation of the surface.

Another filtering method, anisotropic filtering or Perona-Malik flow [9], has been widely used in region selective smoothing and edge preserving filtering of 2D and 3D images. Anisotropic diffusion filtering on non-Euclidean surfaces has been applied to processing and modification of surface geometries [10], [11]. In contrast, here we focus on anisotropic filtering of anatomical or functional images which are scalar functions defined on these surfaces. In order to solve the isotropic and anisotropic diffusion equations on non-flat surfaces, the associated Laplace-Beltrami and anisotropic diffusion operators need to be discretized. The existing approaches to this discretization use FEM formulations [12], [13], [7], [14], [15]. FEM methods discretize the Laplace-Beltrami operator for isotropic smoothing in the neighborhood of each node in a triangulated mesh. Here we present an alternative numerical method which uses parameterization of the cortex. Additionally, we present a parameterization based anisotropic diffusion filtering method which computes Perona-Malik

flow on the surface. We generate a global parameterization of the surface, compute the metric tensor for the parameterization, and use this to compute the isotropic and anisotropic diffusion operators. We first parameterize the cortical surface using a  $p$ -harmonic mapping technique. We then resample the surface on a regular lattice with respect to the 2D parameterization and solve the associated PDEs using this discretization while accounting for the  $p$ -harmonic mapping transformation. In the Euclidean case, discretization of the time derivative in the diffusion equations can be carried out using the Crank-Nicolson method [16] due to its numerical accuracy and stability. Our approach allows us to generalize this method to non-Euclidean cortical surfaces thus making the method stable and accurate. We note that these smoothings, and other time-dependent PDEs on manifolds, can be implemented using implicit surface representations (i.e., level sets) as described in [17], [18], [19] but in many applications, data is available on triangulated surface meshes rather than on implicit surfaces and it might be advantageous to use explicit methods such as the one presented here to process this data.

## II. MATHEMATICAL FORMULATION

We assume as input a genus zero surface mesh that represents the cortex on which we define a scalar-valued field representing the anatomical or functional image of interest. We also assume that a 2D coordinate system is assigned to this surface through a parameterization process. Our approach to generating this parameterization is summarized in Section III. Our goal is to define filtering operations on the image that are computed with respect to the intrinsic geometry of the surface, which guarantees that within-surface distances are handled correctly during smoothing.

Throughout this paper we use Einstein's summation convention [20] to simplify the notation. Let  $I(s, t)$  be a scalar function which denotes the image given on the cortical surface  $S$  and  $t$  denotes time.  $I(s, 0)$  represents the original unsmoothed image. Let  $g, g_{ij} : i, j \in \{1, 2\}$  denote the metric tensor associated with  $S$  for a given coordinate system and  $g^{ij} : i, j \in \{1, 2\}$  denote the inverse of the metric  $g_{ij}$ .

### A. Isotropic filtering

The isotropic diffusion equation on surface  $S$ , with the original image  $I(s, 0), s \in S$  as the initial condition, is given by

$$\frac{\partial}{\partial t} I(s, t) = \Delta I(s, t), \quad (1)$$

where  $\Delta$  is the Laplace-Beltrami operator that generalizes the Laplacian in Euclidean space to a Riemannian space  $S$ :

$$\Delta I(s, t) = \frac{1}{\sqrt{g}} \frac{\partial}{\partial u^\nu} \left( \sqrt{g} g^{\mu\nu} \frac{\partial I(s, t)}{\partial u^\mu} \right). \quad (2)$$

We discretize this operator using discrete approximations of the metric tensor and thus explicitly model the geometry of the surface in our method as described in Sec. III.

The discretization of the time derivative on the left hand side of (1) can be carried out by explicit discretization methods for hyperbolic PDEs. In the explicit scheme for solving (1), time  $t$  is discretized

using a forward difference and  $I(s, n)$  is used for calculation of the left-hand side of (1), where  $I(s, n)$  denotes the image value at iteration  $n$ . Let  $L$  denote the discretization of  $\Delta$  and  $\delta$  the time step; the resulting discretized equation is given by

$$\frac{I(s, n+1) - I(s, n)}{\delta} = LI(s, n). \quad (3)$$

Rearranging terms we have the update equation:

$$I(s, n+1) = I(s, n) + \delta LI(s, n). \quad (4)$$

This is an explicit method for solving the heat equation. Since the diffusion equations are solved iteratively over time, discretization of the time derivative is an important consideration as there is a possibility of numerical instability. Accumulation and amplification of numerical errors at earlier time points results in large deviations or oscillations in the solution at later time points. Therefore choice of time step becomes critical in the explicit method, which is generally used for its speed. There is an upper bound on the choice time step  $\delta$  required for stable solution. Violating this upper bound results in divergence of the solution. A theoretical upper limit on the size of  $\delta$  depends on grid size and the metric tensor coefficients  $\{g_{i,j}\}$  and is hard to determine. Violating the upper limit on the value of  $\delta$  causes amplification of numerical errors which in turn results in divergence of the solution [16]. In order to overcome this difficulty, here we adapt the Crank-Nicolson scheme to suit our particular equation. For the CrankNicolson numerical scheme, a small time step is not required for stability, however it is required for numerical accuracy. While it is slower than the explicit method (4), it has the advantage of being stable as well as accurate [16]. We note that the Crank-Nicolson scheme is unconditionally stable, but it can result in decaying spurious oscillations if the time step is too large. We chose time steps of  $10^{-5}$  which did not result in such oscillations. In this case, (1) is discretized as

$$\frac{I(s, n+1) - I(s, n)}{\delta} = \frac{1}{2}L(I(s, n) + I(s, n+1)). \quad (5)$$

Rearranging terms gives:

$$I(s, n+1) - \frac{\delta}{2}LI(s, n+1) = I(s, n) + \frac{\delta}{2}LI(s, n) \\ L_1 I(s, n+1) = b, \quad (6)$$

where  $L_1 = I - \frac{\delta}{2}L$  and  $b = I(s, n) + \frac{\delta}{2}LI(s, n)$ . This linear system of equations is then solved at each iteration using the conjugate gradient method to compute  $I(s, n+1)$  from  $I(s, n)$ .

### B. Anisotropic filtering

Anisotropic diffusion filtering of planar images was first described by Perona and Malik [9]. Here we generalize this idea to non-Euclidean surfaces [21], which allows us to perform spatially variant and image dependent nonlinear filtering of surface constrained image data within the geometry of the surface itself. The anisotropic diffusion filter is formulated as a diffusion process that encourages smoothing within regions of slowly varying intensity while inhibiting smoothing across boundaries characterized by large image gradients. The anisotropic diffusion equation has the form:

$$\frac{\partial I(s, t)}{\partial t} = \nabla \cdot (D(s, t)) \nabla I(s, t), \quad (7)$$

where the diffusion coefficient  $D(s, t)$  is a monotonically decreasing function of image gradient magnitude:

$$D(s, t) = f(\|\nabla I(s, t)\|). \quad (8)$$

Varying the diffusion coefficient with image gradient allows for locally adaptive edge preserving smoothing. Two choices for  $f$  were suggested [9]:

$$f_1(s, t) = \exp\left(-\left(\frac{\|\nabla I(s, t)\|}{\chi}\right)^2\right) \quad (9)$$

$$f_2(s, t) = \frac{1}{\exp\left(-\left(\frac{\|\nabla I(s, t)\|}{\chi}\right)^{1+\alpha}\right)} \quad \alpha > 0, \quad (10)$$

where  $\chi$  is referred to as the flow constant. Since these filters are expressed using PDEs, they generalize to non-Euclidean spaces. For the cortical surface, we replace (7) by

$$\frac{\partial I(s, t)}{\partial t} = \frac{1}{\sqrt{g}} \frac{\partial}{\partial u^\nu} \left( \sqrt{g} D(s, t) g^{\mu\nu} \frac{\partial I(s, t)}{\partial u^\mu} \right). \quad (11)$$

To compute the diffusion constants we also need an estimator of the gradient. We replace  $\|\nabla I(s, t)\|^2$  by the *differentiator of the first order* [20] given by

$$\nabla(I(s, t), I(s, t)) = g^{\alpha\beta} \frac{\partial I(s, t)}{\partial u^\alpha} \frac{\partial I(s, t)}{\partial u^\beta}. \quad (12)$$

With these substitutions, the anisotropic heat equation is well-defined on the cortical surface independently of the particular parameterization used for its computation.

Similar to the isotropic case in (5), discretization of temporal and spatial derivatives in (7) lead to

$$\frac{I(s, n+1) - I(s, n)}{\delta} = \frac{1}{2}L^\alpha(I(s, n) + I(s, n+1)) \quad (13)$$

where  $L^\alpha$  denotes anisotropic diffusion operator. After simplification similar to (6), it becomes

$$L_1^\alpha I(s, n+1) = b_\alpha \quad (14)$$

where  $L_1^\alpha = I - \frac{\delta}{2}L^\alpha$  and  $b^\alpha = I(s, n) + \frac{\delta}{2}L^\alpha I(s, n)$ . Similar to the isotropic case, the linear system of equations is solved using the conjugate gradient method to compute  $I(s, n+1)$  from  $I(s, n)$ . Discretization of the diffusion operators  $L$  and  $L^\alpha$  is described in the next section.

## III. DISCRETIZATION AND NUMERICAL METHOD

### A. Cortical Surface Extraction and Parameterization

We first extract cortical surface mesh models from MRI for each subject using the BrainSuite software [22], [23], which includes a 6 stage cortical modeling sequence. First the brain is extracted from the surrounding skull and scalp tissues using a combination of edge detection and mathematical morphology. Next the intensities of the MRI are corrected for low-frequency ‘bias fields’ resulting from RF inhomogeneities. Each voxel in the corrected image is labeled according to tissue type using a statistical classifier. Coregistration of the segmented volume to a standard atlas is then used to automatically identify the white matter volume, fill ventricular spaces and remove the brain stem and cerebellum, leaving a binary volume whose boundary surface represents interface between the white matter and the grey matter of the cerebral cortex. It is likely that the tessellation of this volume will produce surfaces with topological handles. Prior to tessellation, these handles are identified and removed automatically using a graph-based approach [24]. The resulting binary volume is then tessellated to produce a genus-zero surface.

We use a  $p$ -harmonic map for parameterization in which we minimize a  $p$ -harmonic energy function [25], [4] while constraining a closed curve on the interhemispheric fissure, which divides the brain into two hemispheres, to map to the boundary of a unit square. This

procedure produces a one-to-one mapping from each hemisphere to a unit square. Details of this procedure can be found in the Appendix.

The metric coefficients or the coefficients of the first fundamental form contain information about the intrinsic geometry of the surface. Let  $\mathbf{x}$  denote the 3-D position vector of a point on the cortical surface. Let  $u^1, u^2$  denote the coordinates in the parameter space. The metric tensor coefficients required in the computation of the diffusion operators are given by:

$$g_{11} = \left\| \frac{\partial \mathbf{x}}{\partial u^1} \right\|^2, \quad g_{22} = \left\| \frac{\partial \mathbf{x}}{\partial u^2} \right\|^2, \quad (15)$$

$$g_{12} = g_{21} = \left\langle \frac{\partial \mathbf{x}}{\partial u^1}, \frac{\partial \mathbf{x}}{\partial u^2} \right\rangle, \quad (16)$$

$$g = g_{11}g_{22} - (g_{12})^2, \quad (17)$$

The inverse metric coefficients  $g^{ij}$  are given by:

$$g^{11} = \frac{g_{22}}{g}, g^{12} = g^{21} = -\frac{g_{12}}{g}, g^{22} = \frac{g_{11}}{g}. \quad (18)$$

We assume that they are arranged in vectors for numerical computations as described in the next section.

### B. Discretization Algorithm

In order to solve the diffusion equations numerically, we need to discretize the isotropic and anisotropic Laplace-Beltrami operators in (2) and (11). We use the unit-square  $p$ -harmonic maps of the triangulated tessellation of the cortical surface to define a 2D coordinate system [25]. The square maps for each hemisphere are resampled on a regular 256x256 grid. Note that this resampling can be avoided by discretization on the triangulated 2D mesh, but we prefer a regular grid based method for simplicity. The co-ordinate system we assign to the cortical surface is depicted in Fig. 1. The two squares in the  $(u^1, u^2)$  parameter space represent the two hemispheres in the  $x^1, x^2, x^3$  space. The boundaries of the squares correspond to the common interhemispheric fissure between the two cortical hemispheres. A closed curve is formed at the boundary formed by cutting the genus zero brain surface into two hemispheres. This curve is constrained to map as a unit speed curve to the boundaries of the unit squares for each of the two hemispheres. One can move continuously from one hemisphere to the other across this boundary. The neighborhood relations between the edges of the two squares is depicted by different arrows in Fig 1. Because the interhemispheric fissure is fixed on the boundary of the squares representing the two hemispheres, one can visualize the  $u^1 - u^2$  parameter space as two squares placed on top of each other and connected at the boundaries of the squares. We follow these neighborhood relations when discretizing the partial derivatives at the boundary of each squares. This allows us to compute partial derivatives across the two cortical hemispheres making the boundary separating them transparent to the numerical discretizations. This boundaryless parameter space is then used for discretizing the partial derivatives with respect to the  $u^1$  and  $u^2$  spatial coordinates in the solution of the differential equations. For instance, assume that  $f : M \rightarrow \mathbb{R}$  is a scalar valued function defined on the cortical surface  $M$ . We arrange its discretized representation at each vertex in the regular grid of the surface in a vector  $\vec{f}$ . In order to discretize  $\frac{\partial f}{\partial u^1}$  by central differences on the entire surface, we calculate the usual central differences at the points that are not on the boundary of the squares. On the boundary points of the squares, we use the connectivity relationship shown in Fig. 1 for the neighborhood in the central difference approximation. Using these relations, we compose a forward difference matrix  $D_{u^1}^f$  and obtain discretization of  $\frac{\partial f}{\partial u^1}$  as  $D_{u^1}^f \vec{f}$ . Similarly, we compose backward difference matrices  $D_{u^1}^b$

the forward and backward difference operators for the  $u^1$  coordinate, and  $D_{u^2}^f$  and  $D_{u^2}^b$ , the forward and backward difference operators for the  $u^2$  coordinate. Explicitly, these operators in matrix form are given by following equations:

$$D_{u^1}^f(i, j) = \begin{cases} -1 & \text{if } i = j \\ 1 & \text{if } j = i + 1 \text{ and } i \neq 0 \text{ mod } 256, i \leq 256^2 \\ 1 & \text{if } j = i - 1 \text{ and } i \neq 1 \text{ mod } 256, i > 256^2 \\ 1 & \text{if } j = i + 256^2 \text{ and } i = 0 \text{ mod } 256, i \leq 256^2 \\ 1 & \text{if } j = i - 256^2 \text{ and } i = 1 \text{ mod } 256, i > 256^2 \\ 0 & \text{otherwise,} \end{cases} \quad (19)$$

$$D_{u^1}^b(i, j) = \begin{cases} 1 & \text{if } i = j \\ -1 & \text{if } j = i - 1 \text{ and } i \neq 1 \text{ mod } 256, i \leq 256^2 \\ -1 & \text{if } j = i + 1 \text{ and } i \neq 0 \text{ mod } 256, i > 256^2 \\ -1 & \text{if } j = i + 256^2 \text{ and } i = 1 \text{ mod } 256, i \leq 256^2 \\ -1 & \text{if } j = i - 256^2 \text{ and } i = 0 \text{ mod } 256, i > 256^2 \\ 0 & \text{otherwise,} \end{cases} \quad (20)$$

$$D_{u^2}^f(i, j) = \begin{cases} -1 & \text{if } i = j \\ 1 & \text{if } j = i + 256 \text{ and } i \leq 256 * 255 \\ 1 & \text{if } j = i - 256 \text{ and } 256^2 < i \leq (256^2 + 256 * 255) \\ 1 & \text{if } j = i + 256^2 \text{ and } 256 * 255 < i \leq 256^2 \\ 1 & \text{if } j = i - 256^2 \text{ and } i > (256^2 + 256 * 255) \\ 0 & \text{otherwise,} \end{cases} \quad (21)$$

$$D_{u^2}^b(i, j) = \begin{cases} 1 & \text{if } i = j \\ -1 & \text{if } j = i + 256^2 \text{ and } i \leq 256 \\ -1 & \text{if } j = i - 256 \text{ and } 256 < i \leq 256^2 \\ -1 & \text{if } j = i + 256 \text{ and } 256^2 < i \leq 256 * 255 \\ -1 & \text{if } j = i + 256^2 \text{ and } i > 256 * 255 \\ 0 & \text{otherwise.} \end{cases} \quad (22)$$

For discretization of the anisotropic diffusion operator from (7), we compute the conduction function  $D(s, t)$  using (9) and arrange it in a column, denoted by  $D$ .

Then the isotropic and anisotropic diffusion operators are given by

$$L = G^I \begin{pmatrix} D_{u^1}^f & D_{u^2}^f \end{pmatrix} G^r G \begin{pmatrix} D_{u^1}^f \\ D_{u^2}^f \end{pmatrix}, \quad (23)$$

$$L^a = G^I \begin{pmatrix} D_{u^1}^f & D_{u^2}^f \end{pmatrix} D^r G \begin{pmatrix} D_{u^1}^f \\ D_{u^2}^f \end{pmatrix}, \quad (24)$$

where

$$G^I = \text{diag}(1/\sqrt{g}), \quad (25)$$

$$D^r = \text{diag}([D, D][\sqrt{g}, \sqrt{g}]) \quad (26)$$

$$G^r = \text{diag}([\sqrt{g}, \sqrt{g}]) \text{ and} \quad (27)$$

$$G = \begin{pmatrix} \text{diag}(g^{11}) & \text{diag}(g^{12}) \\ \text{diag}(g^{12}) & \text{diag}(g^{22}) \end{pmatrix}. \quad (28)$$

In the discretization, we use forward and backward difference operators alternately in order to keep the numerical template of derivatives small [16].

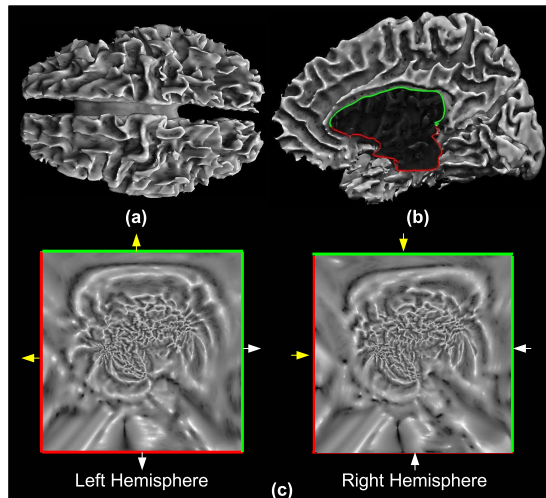


Fig. 1. (a) A cortical surface extracted from volumetric MR data using the BrainSuite software; (b) a closed curve in the fissure between the two hemispheres divides the brain in half; this curve is constrained to lie on the boundary of the unit square for both hemispheres; (c) the 2D coordinate system resulting from the  $p$ -harmonic mapping of the cortical surface. The mean curvature of the cortical surface is shown as a scalar image in this 2D space. The different sized arrows indicate the connectivity along the boundary of the square between the two hemispheres, which is used for computing numerical approximations of the partial derivatives along this boundary.

The discretization of the isotropic and anisotropic operators is carried out in the following steps:

- 1) Parameterize the cortical surface to map it in two squares and assign it the coordinate system described in the Appendix.
- 2) Form the forward and backward difference matrices  $D_{u^1}^f, D_{u^2}^f, D_{u^1}^b, D_{u^2}^b$  and for  $u^1$  and  $u^2$  coordinates respectively according to (19), (20), (21) and (22).
- 3) Compute the surface metric coefficients  $g, g_{11}, g_{12}, g_{21}$  and  $g_{22}$  and also the inverse metric coefficients  $g^{11}, g^{12}, g^{22}$ . This is done by replacing partial derivatives in (15), (16), (17), (18) by their discrete versions from step 2.
- 4) Compute the isotropic or anisotropic Laplace-Beltrami operators using (23) or (24), respectively. For the anisotropic diffusion operator (11), we use the conduction function given by (9), with coefficient  $\chi = 0.1$ . This value was chosen empirically.

In the case of isotropic diffusion, discretization of the diffusion operator needs to be carried out only once before starting the time iterations. On the other hand, for anisotropic diffusion the diffusion operator depends on  $I(s, t)$  and hence needs to be updated by carrying out the last step 4 repeatedly after every time step. In order to decrease this numerical cost, we update the operator after every 100 iterations assuming that the incremental change in  $I(s, n)$  is small.

The impulse response of the isotropic diffusion filter is shown in Fig. 2 both on the cortical surface and in the parameter space of one hemisphere. It can be seen that use of the surface metric results in an impulse response on the cortical surface with a greater degree of circular symmetry. Note that because of the non-linear nature of the anisotropic diffusion filter, its behavior cannot be fully characterized by its impulse response.

#### IV. RESULTS

We performed numerical studies on an Intel Pentium 4, 3.2 GHz computer with 2 GB of RAM using MATLAB. The cortical surface was extracted from a  $256 \times 256 \times 170$  voxel T1-weighted brain MR image of a volunteer subject. Processing time from the raw MR volume to extraction of the topologically corrected and tessellated cortex using BrainSuite took 7 mins. The tessellated cortex had a total of 1.4 million nodes. The  $p$  harmonic parameterization of

the 1.4 million node cortical surface with  $p = 4$  took 37 secs. Adding the parameterization step does not add significantly to the total computational cost compared to a direct FEM method [13], [12]. The number of iterations  $n$  along with the size of the time step  $\delta$  determines the amount of smoothing applied. Smaller values of  $\delta$  result in more numerically accurate solutions while the execution time is directly proportional to  $n$ . We chose  $\delta = 1 \times 10^{-5}$  and  $n = 40000$ . Isotropic diffusion on the resampled surface took 20 mins with this choice of  $n$  and  $\delta$  while anisotropic diffusion took 1.5 hours. The difference in execution times is mainly due to the non-linear nature of the anisotropic diffusion which requires recomputation of the diffusion operator  $L$  repeatedly during the iterations. The code through parameterization was implemented in C/C++ with substantial effort directed at optimizing run-times while the diffusions were computed in MATLAB and, based on our earlier experience, we can expect a several-fold speed up when these are reprogrammed directly in C/C++.

In order to validate the method for isotropic diffusion, we compute the impulse response  $K_t(p, q)$  of the diffusion operator (heat-kernel) on a unit sphere. The heat-kernel  $K_t(p, q)$  is defined as the solution to the initial value problem:

$$K_0(p, q) = \delta(p, q); \quad (29)$$

$$\frac{\partial K_t(p, q)}{\partial t} = \Delta K_t(p, q). \quad (30)$$

The heat-kernel on the unit sphere has a closed form analytical expression [26] given by:

$$K_t(p, q) = \sum_{l=0}^{\infty} \frac{2l+1}{4\pi} e^{-l(l+1)t} P_l^0(p \cdot q) \quad (31)$$

where  $p$  is the starting point of the impulse and  $q$  is the point where measurement of the impulse response is made. In this case, the amount of spread is governed by time  $t$  in the diffusion equation (1).  $P_l^0$  are the Legendre polynomials. The impulse response for different values of  $t$  is plotted on the unit sphere in Fig. 3. We also calculated the heat kernel using the diffusion filter defined in Sec. II-A and compared its accuracy with the analytically computed filter. We used a triangulated mesh with 300,000 vertices to represent the unit sphere. Each hemisphere was mapped to a unit square by

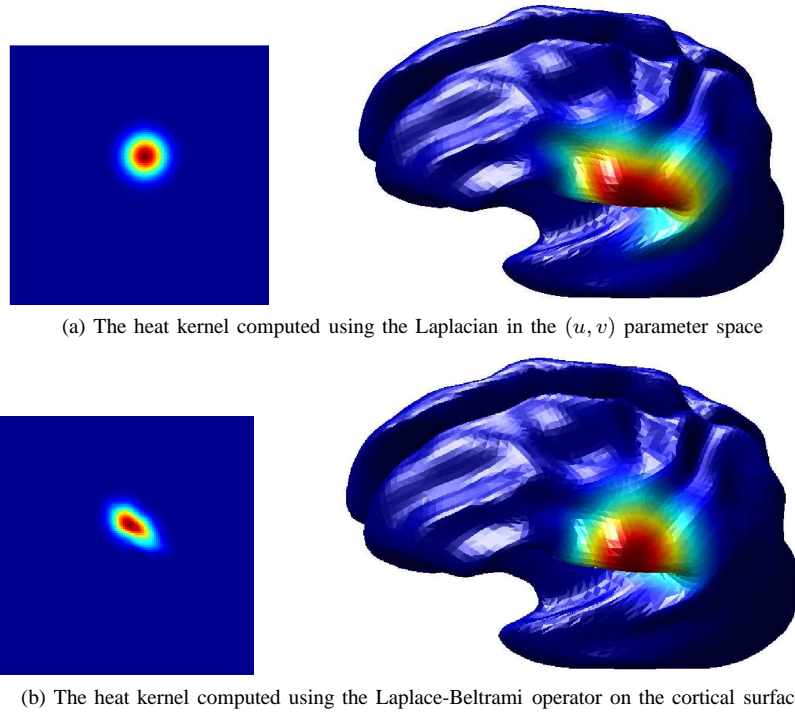


Fig. 2. The figure depicts the importance of using metric coefficients in (2). (a) The impulse response of the isotropic smoothing filter computed by solving the diffusion equation directly with respect to the 2D coordinates without using metric coefficients. Note that the kernel is radially symmetric in the flat space but not symmetric when mapped back to the cortical surface. (b) in this case we generate the isotropic smoothing kernel with respect to the intrinsic geometry of the cortical surface. In this case the kernel appears highly nonsymmetric in the flat space, but becomes symmetric when mapped back to the cortical surface.

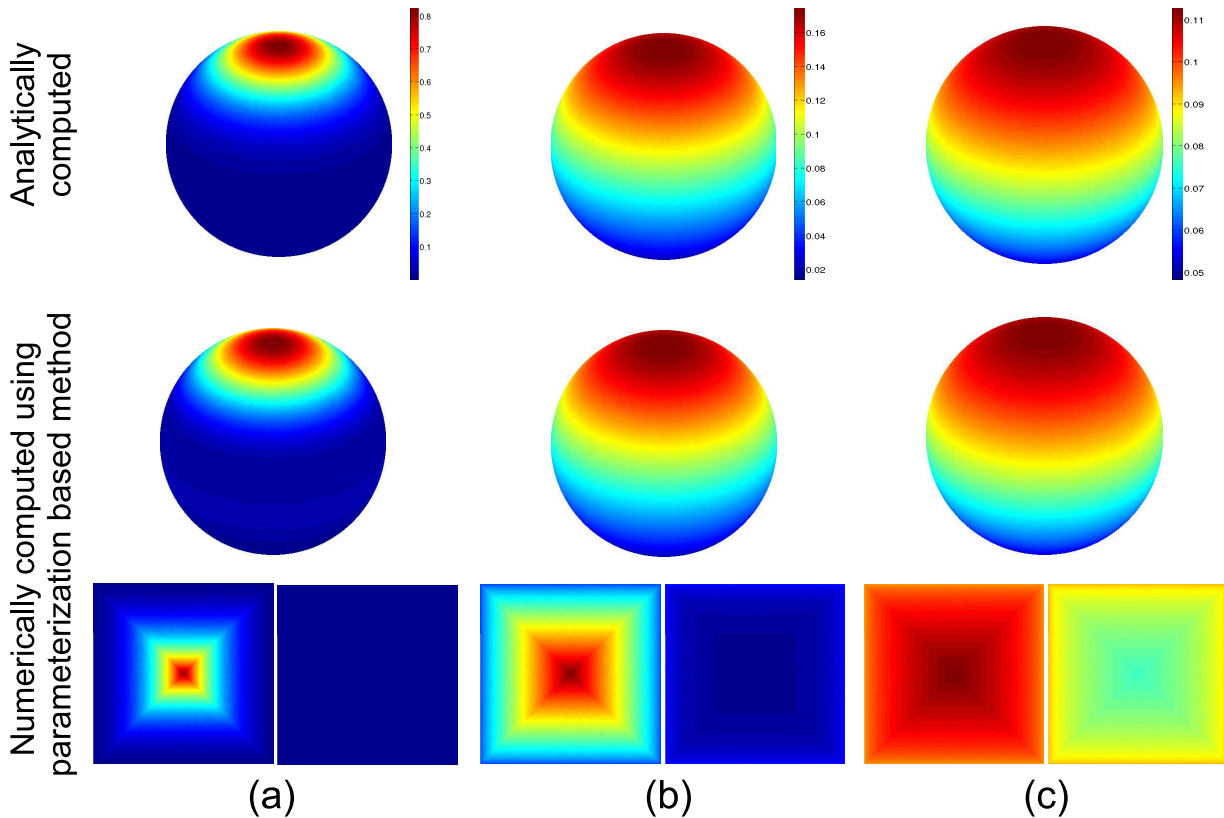


Fig. 3. Heat-Kernel on the unit sphere computed using the analytical expression as well as the parameterization based numerical method for  $t = 0.1$  (b)  $t = 0.5$  and (c)  $t = 1$ . The top row shows the heat-kernel computed analytically on the sphere. The second and third rows show numerically computed kernel on the sphere and in the flat parameter space. The two squares are maps of the upper and lower hemispheres, with the square boundary corresponding to a great circle around the equator of the sphere.

$(x, y, z) \mapsto (\text{sign}(x)x^2 + \text{sign}(y)y^2, \text{sign}(x)x^2 - \text{sign}(y)y^2)$ . This parameterization was used for the numerical discretization of the Laplace-Beltrami operator in (1).

The root mean square percentage error compared to the analytical expression for  $t = 0.1, 0.5$  and  $1$  was  $2.31, 2.1$  and  $1.32$  respectively. The errors are relatively small and do not appear to increase with time. Note that since the metric distortion is accounted for in the computation of the Laplace-Beltrami operator, the errors result from resampling and discretization of the parameterized surface and numerical approximation of derivatives. In order to study the behavior of the numerical errors in our method, we compared the percentage error in the numerically computed heat-kernel for different step sizes and grid sizes. As shown in Fig. 4 we can see that this error is a monotonically decreasing function of grid size and a monotonically increasing function of step size.

We also applied the diffusion operators to mean curvature maps computed on the cortical surface. We compute the mean curvature using the method described in [27] and resample it on the regular grid. However we note that an added advantage of our approach is that we can also compute the mean curvature by using our discretization of the metric tensor. In particular, the mean curvature  $H$  can be computed as:

$$H = \frac{1}{2} b_{\alpha\beta} g^{\alpha\beta}, \quad (32)$$

where the *second fundamental form*  $b_{\alpha\beta}$  is given by

$$b_{\alpha\beta} = \frac{1}{\sqrt{g}} \left| \frac{\partial \mathbf{x}}{\partial u^1} + \frac{\partial \mathbf{x}}{\partial u^2} + \frac{\partial^2 \mathbf{x}}{\partial u^\alpha \partial u^\beta} \right|. \quad (33)$$

The minima and saddle points of the mean curvature of the cortical surface are known to follow the sulcal patterns [27] and therefore are vital features for automatic labeling of the sulci [28]. However, as can be seen in Fig. 5(a), there is a considerable amount of noise in the mean curvature computed on the cortical surface. This is primarily due to the fact that the mean curvature is a local feature and is therefore prone to errors in extraction and discretization of the cortical surface. We see that the isotropic diffusion filtering smoothes out this noise, but since this filtering is not region selective, it also blurs the regions between sulci (positive mean curvature) and gyri (negative mean curvature) as seen in Fig. 5(b). On the other hand, anisotropic filtering removes noise while carrying out the smoothing only within regions and thus respects boundaries between sulci and gyri as seen in Fig. 5(c), thus illustrating the advantage of anisotropic filtering.

## V. DISCUSSION

We have described a computational procedure for diffusion filtering of scalar images defined on non-Euclidean surfaces. Using a parameterized representation of the surface, we compute a solution to the diffusion equation in which metric tensors are used to account for the curvature of the surface and intrinsic distances. The resulting diffusions are computed with respect to the intrinsic geometry of the surface and are therefore independent of the specific parameterization used, so long as the parameterization is fine enough. We also presented a Crank-Nicolson type scheme for discretization of the time derivative in order to make these filters stable. Our parameterization-based method can be used for both isotropic and anisotropic diffusion smoothing, as we demonstrated by performing these operations to mean curvature maps on models of the cerebral cortex. The resulting computational procedures are relatively efficient and stable, requiring approximately 30 mins of computation for isotropic smoothing on a single cortex, including the preprocessing time necessary to extract the parameterized representation of the cortical surface.

In the context of neuroimaging, isotropic forms of diffusion described here may be used for spatially invariant smoothing of

anatomical, functional or statistical data on the cortical surfaces in situations where selective smoothing is not required. Applications include filtering of functional data when smoothness is required for application of parametric random-field methods for control of false positives in multiple hypothesis testing [29].

Anisotropic diffusion filtering can be used for applications where region selective smoothing is advantageous, such as for studying gyrification of the cortex by mean curvature maps [30], [31], [32]. In functional studies involving different visual tasks, it may be advantageous to discourage smoothing across different visual areas [33]. These techniques can also be used for multiscale representations of functional activation [4], statistical data [5] and neuro-anatomical variability [3]. Surface matching across subjects may also be implemented using covariant PDEs, which essentially compute flows that are regular with respect to the surface metric, and are invariant to smooth changes in coordinates [4], [3]. In these flows, it is also necessary to compute the Laplace-Beltrami operators of a vector field, and it is also possible, by analogy, to regularize higher order tensor fields using a manifold-derived metric (e.g., for enhancing diffusion tensor images).

## REFERENCES

- [1] B. Romeny, *Geometry-Driven Diffusion in Computer Vision*. Kluwer Academic Publishers Norwell, MA, USA, 1994.
- [2] P. Thompson and A. Toga, "A surface-based technique for warping three-dimensional images of the brain," *Medical Imaging, IEEE Transactions on*, vol. 15, no. 4, pp. 402–417, 1996.
- [3] P. M. Thompson et al., "Dynamics of gray matter loss in Alzheimer's disease," *The Journal of Neuroscience*, vol. 23, no. 3, pp. 994–1005, 2003.
- [4] A. A. Joshi, D. W. Shattuck, P. M. Thompson, and R. M. Leahy, "A framework for registration, statistical characterization and classification of cortically constrained functional imaging data," in *Lecture Notes in Computer Science*, vol. 3565, July 2005, pp. 186–196.
- [5] M. K. Chung, S. M. Robbins, K. M. Dalton, R. J. Davidson, A. L. Alexander, and A. Evans, "Cortical thickness analysis in autism with heat kernel smoothing," *NeuroImage*, vol. 25, no. 4, pp. 1256–1265, May 2005.
- [6] D. J. Hagler Jr., A. P. Saygin, and M. I. Sereno, "Smoothing and cluster thresholding for cortical surface-based group analysis of fMRI data," *NeuroImage*, vol. 33, no. 4, pp. 1093–1103, Dec. 2006.
- [7] M. K. Chung et al., "Diffusion smoothing on the cortical surface," *NeuroImage*, vol. 13, no. 6S1, p. 95, Jun 2001.
- [8] M. K. Chung, "Introducing heat and geodesic kernel smoothing on cortical manifolds," *11th Annual Meeting of the OHBM, Toronto*, 2005.
- [9] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-12, no. 7, pp. 629–639, Jul. 1990.
- [10] K. Hildebrandt and K. Polthier, "Anisotropic filtering of non-linear surface features," *Computer Graphics Forum*, vol. 23, no. 3, pp. 391–400, Sept. 2004.
- [11] U. Clarenz, U. Diewald, and M. Rumpf, "Processing textured surfaces via anisotropic geometric diffusion," *IEEE Trans. on Image Processing*, vol. 13, no. 2, pp. 248–261, Feb. 2004.
- [12] L. Lopez-Perez, R. Deriche, and N. Sochen, "The beltrami flow over triangulated manifolds," in *ECCV Workshops CVAMIA and MMBIA*, 2004, pp. 135–144.
- [13] C. Bajaj and G. Xu, "Anisotropic diffusion of surfaces and functions on surfaces," *ACM Transactions on Graphics (TOG)*, vol. 22, no. 1, pp. 4–32, 2003.
- [14] A. Andrade, F. Kherif, J. Mangin, K. Worsley, A. Paradis, O. Simon, S. Dehaene, D. L. Bihan, and J. Poline, "Detection of fMRI activation using cortical surface mapping," *Human Brain Mapping*, vol. 12, pp. 79–93, 2001.
- [15] A. Qiu, D. Bitouk, and M. Miller, "Smooth functional and structural maps on the neocortex via orthonormal bases of the Laplace-Beltrami operator," *Medical Imaging, IEEE Transactions on*, vol. 25, no. 10, pp. 1296–1306, 2006.
- [16] G. D. Smith, *Numerical solution of partial differential equations: Finite difference methods*, 3rd ed., ser. Oxford Applied Mathematics and Computing Science Series. Oxford : Clarendon Press, 1985.



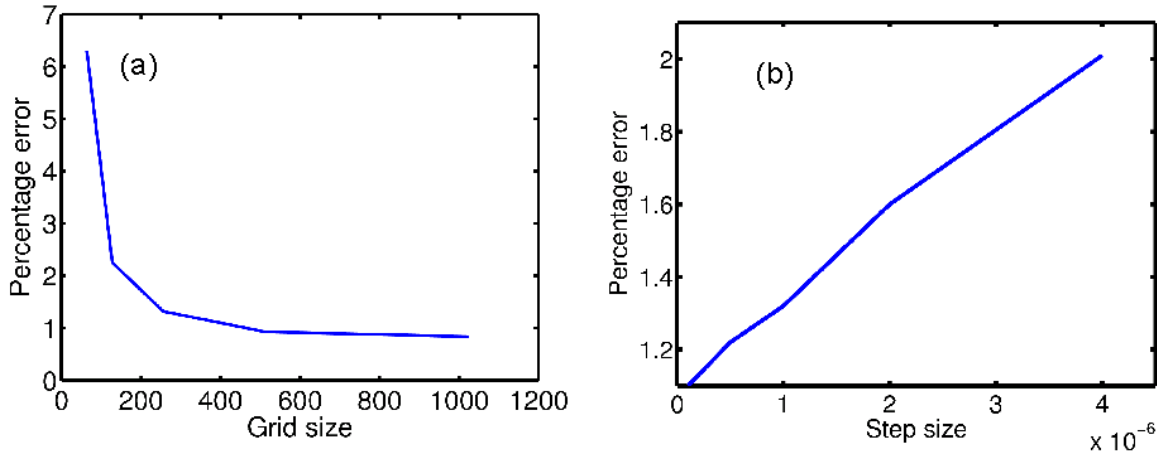


Fig. 4. Percentage error in the numerically computed heat kernel on the sphere as a function of (a) grid size with fixed step size  $\delta = 1 \times 10^{-5}$  and (b) step size for fixed grid of size  $256 \times 256$ .

- [17] M. Bertalmio, L.-T. Cheng, S. Osher, and G. Sapiro, "Variational problems and partial differential equations on implicit surfaces: The framework and examples in image processing and pattern formation," UCLA, Tech. Rep., June 2000.
- [18] F. Mémoli, G. Sapiro, and S. Osher, "Solving variational problems and partial differential equations mapping into general target manifolds," *J. Comput. Phys.*, vol. 195, no. 1, pp. 263–292, 2004.
- [19] F. Mémoli, G. Sapiro, and P. Thompson, "Implicit brain imaging," *Neuroimage : Mathematics in Brain Imaging*, vol. 23, no. Supp. 1, pp. S179–S188, 2004.
- [20] I. Kreyzig, *Differential Geometry*. Dover, 1999.
- [21] N. Sochen, R. Deriche, and L. Perez-Lopez, "Variational beltrami flow over implicit manifolds," in *Proceedings of the International Conference Image Processing, Barcelona, Spain*, 2003.
- [22] D. W. Shattuck and R. M. Leahy, "Brainsuite: An automated cortical surface identification tool," *Medical Image Analysis*, vol. 8, no. 2, pp. 129–142, 2002.
- [23] D. W. Shattuck, S. R. Sandor-Leahy, K. A. Schaper, and D. A. R. and R. M. Leahy, "Magnetic resonance image tissue classification using a partial volume model," *NeuroImage*, vol. 13, no. 5, pp. 856–876, May 2001.
- [24] D. W. Shattuck and R. M. Leahy, "Automated graph-based analysis and correction of cortical volume topology," *IEEE Trans. Med. Imaging*, vol. 20, no. 11, pp. 1167–1177, Nov. 2001.
- [25] A. A. Joshi, R. M. Leahy, P. M. Thompson, and D. W. Shattuck, "Cortical surface parameterization by p-harmonic energy minimization," in *IEEE International Symposium on Biomedical Imaging*, 2004, pp. 428–431.
- [26] M. K. Chung, "Heat kernel smoothing on unit sphere," in *IEEE International Symposium on Biomedical Imaging*, vol. 1430, 2006.
- [27] A. Cachia et al., "A primal sketch of the cortex mean curvature: a morphogenesis based approach to study the variability of the folding patterns," *IEEE Trans. Med. Imaging*, vol. 22, no. 6, pp. 754–765, Jun 2003.
- [28] M. Rettmann, X. Han, C. Xu, and J. Prince, "Automated sulcal segmentation using watersheds on the cortical surface," *NeuroImage*, vol. 15, no. 2, pp. 329–344, 2002.
- [29] K. Worsley, S. Marrett, P. Neelin, A. Vandal, K. Friston, and A. Evans, "A unified statistical approach for determining significant signals in images of cerebral activation," *Human Brain Mapping*, vol. 4, pp. 58–73, 1996.
- [30] E. Luders, P. Thompson, K. Narr, A. Toga, L. Jancke, and C. Gaser, "A curvature-based approach to estimate local gyrification on the cortical surface," *NeuroImage*, vol. 29, no. 4, pp. 1224–1230, Feb. 2006.
- [31] C. Gaser, E. Luders, P. M. Thompson, A. D. Lee, R. A. Dutton, J. A. Geaga, K. M. Hayashi, U. Bellugi, A. M. Galaburda, J. R. Korenberg, D. L. Mills, A. W. Toga, and A. L. Reiss, "Increased local gyrification mapped in williams syndrome," *NeuroImage*, vol. 33, no. 1, pp. 46–54, Oct. 2006.
- [32] L. M. Lui, Y. L. Wang, T. F. Chan, and P. M. Thompson, "Brain anatomical feature detection by solving partial differential equations on a general manifold," *Journal of Discrete and Continuous Dynamical Systems-Series B*, vol. 7, no. 3, pp. 605–618, May 2007.
- [33] V. D. Calhoun, T. Adali, V. B. McGinty, J. J. Pekar, T. D. Watson, and G. D. Pearlson, "fMRI activation in a visual-perception task: Network of areas detected using the general linear model and independent components analysis," *NeuroImage*, vol. 14, no. 5, pp. 1080–1088, 2001.
- [34] B. Tang, G. Sapiro, and V. Caselles, "Diffusion of general data on non-flat manifolds via harmonic maps theory: The direction diffusion case," *International Journal of Computer Vision*, vol. 36, no. 2, pp. 149–161, 2000.
- [35] M. K. Chung, A. Qiu, B. M. Nacewicz, K. M. Dalton, S. Pollak, and R. J. Davidson, "Tiling manifolds with orthonormal basis," in *2nd MICCAI Workshop on Mathematical Foundations of Computational Anatomy (MFCA)*, 2008.
- [36] Y. Xin, *Geometry of harmonic maps*. Birkhäuser, 1996.
- [37] A. Fardoun and R. Regbaoui, "Heat flow for p-harmonic maps between compact Riemannian manifolds," *Indiana Univ. Math. J.*, vol. 51, pp. 1305–1320, 2002.
- [38] R. Hamilton, "Harmonic maps of manifolds with boundary," in *Lecture Notes in Mathematics*, ser. 471. Springer, 1975.

## APPENDIX

In order to solve the smoothing equations on the cortical surface, a  $p$ -harmonic parameterization described in [25] is used. The 2D coordinate system generated by this parameterization is used for the discretizations as described in Sec. III. Note that since the metric associated with the coordinate system is included in the computations, any other flattening method can be used instead of these mappings. In this appendix, we briefly describe our  $p$ -harmonic parameterization scheme.

Let  $\vec{u} = (u^1, u^2)$  be a function defined on a surface  $S$  such that  $\vec{u} : S \mapsto \mathbb{R}^2$ . The  $p$ -harmonic energy functional is defined as the surface integral

$$\int_S |\nabla \vec{u}|^p dS. \quad (34)$$

The  $p$ -harmonic maps are the minimizers of this energy. The existence, uniqueness and other properties of these maps have been studied [36], [37], [38]. We use these maps to flatten the two brain hemispheres to two unit squares. The cortical surface model extracted using BrainSuite is represented as a high resolution triangulated mesh with around 1 million vertices. The corpus callosum which divides the two cortical hemispheres is identified as a closed contour in the cortical mesh. This contour is then constrained to lie on the unit square. With this as the mapping constraint, the  $p$ -harmonic energy minimization leads to a system of linear equations with a sparse system matrix as explained in [25]. Here we present the discretization

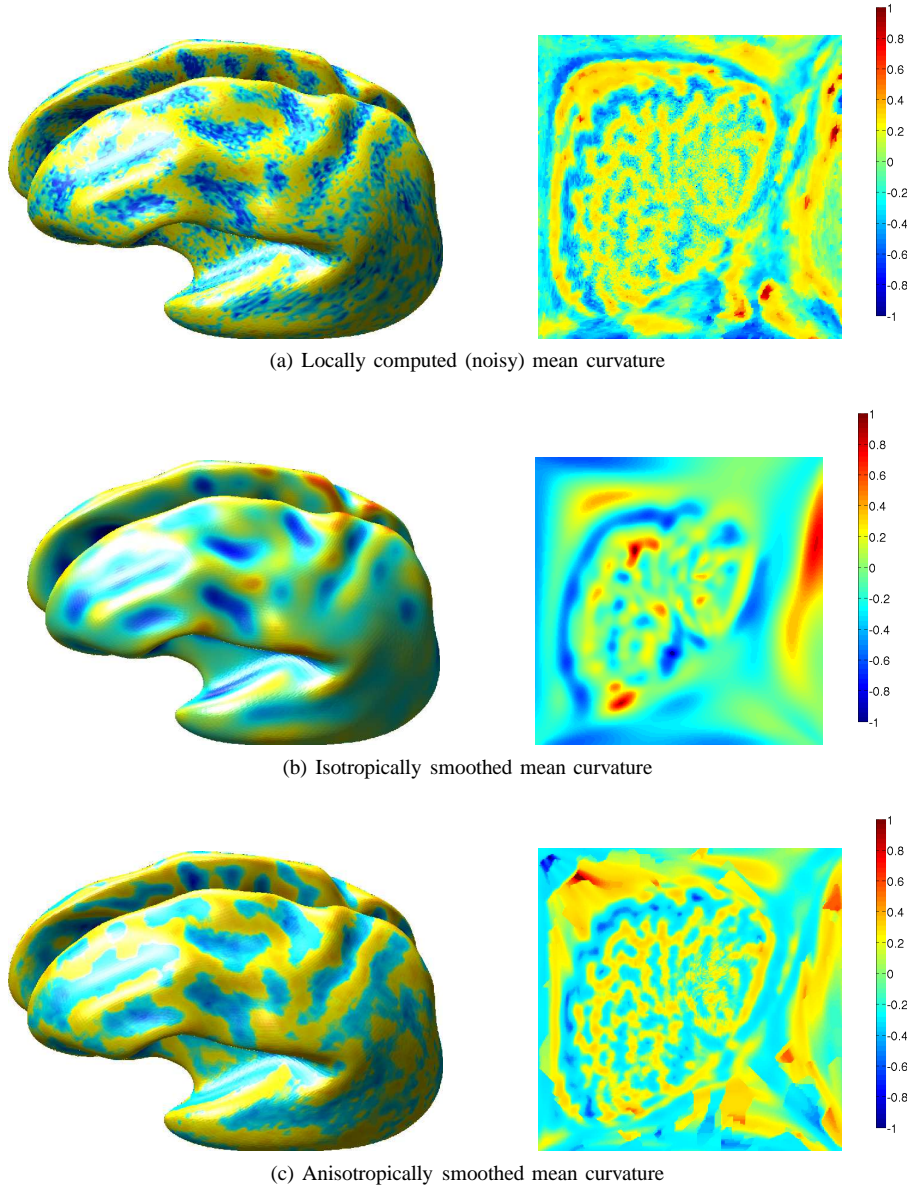


Fig. 5. left: The mean curvature of the cortical surface plotted on a smoothed representation for improved visualization of curvature in sulcal folds; right: The mean curvature plotted in 2D parameter space for a single cortical hemisphere. (a) shows the mean curvature maps on the surface and in the flat space for one hemisphere. (b) Isotropic diffusion filtering smooths uniformly on the cortical surface thus losing detail in the curvature map while (c) anisotropic diffusion filtering removes the fine grain noise structure, while retaining the larger scale details in the curvature map.

formulae for completeness and ease of implementation. The detailed derivation is presented in [25].

We assume that the mapping function  $\vec{u}$  is piecewise linear over the surface, i.e. it is linear over each triangle. For such functions, the gradient operator for  $i^{th}$  triangle is given by

$$\begin{aligned} \vec{\nabla} &= \frac{1}{|2A^i|} \underbrace{\begin{bmatrix} y_2^i - y_1^i & y_3^i - y_1^i & y_1^i - y_2^i \\ x_1^i - x_2^i & x_1^i - x_3^i & x_2^i - x_1^i \end{bmatrix}}_{B^i} \\ &= \frac{1}{2A^i} B^i, \end{aligned}$$

where  $A^i$  is the area of triangle  $i$  and  $x_k^i, y_k^i, z_k^i$  denote the  $x, y, z$  coordinates of the  $k^{th}$  vertex of  $i^{th}$  triangle. Since  $\vec{u}$  is piecewise

linear, its gradient is constant over each triangle  $i$ , so that:

$$\int \|\nabla \vec{u}\|^p dS = \sum_i \|\nabla \vec{u}\|^p A^i$$

where the sum is over all triangles. The energy minimization discretizes to

$$\arg \min_{\vec{u}} \int \|\nabla \vec{u}\|^p dS = \arg \min_{\vec{u}} \sum_i \left\| \frac{1}{2^p (A^i)^{p-1}} B^i \vec{u}^i \right\|^p,$$

where  $\vec{u}^i$  are the values of the map  $\vec{u}$  on the  $i^{th}$  triangle. This energy discretization with the unit square mapping constraint on the vertices corresponding to corpus callosum is solved using a preconditioned conjugate-gradient minimization method resulting in the p-harmonic maps shown in Fig. 1. We use a Jacobi preconditioner for the minimization. The mapping function is integrated into our BrainSuite package [22] and is available for download through <http://brainsuite.usc.edu>.