

# Using the Fast Marching Method to Extract Curves with Given Global Properties

Xiaodong Tao<sup>1,\*</sup>, Christos Davatzikos<sup>2</sup>, and Jerry L. Prince<sup>3</sup>

<sup>1</sup> Visualization and Computer Vision Lab, GE Global Research Center,  
One Research Circle, KW-C223A, Niskayuna, NY 12309

[taox@research.ge.com](mailto:taox@research.ge.com)

<sup>2</sup> SBIA, Department of Radiology,  
University of Pennsylvania, Philadelphia, PA 19104

[christos@rad.upenn.edu](mailto:christos@rad.upenn.edu)

<sup>3</sup> IACL, Department of Electrical and Computer Engineering,  
Johns Hopkins University, Baltimore, MD 21218

[prince@jhu.edu](mailto:prince@jhu.edu)

**Abstract.** Curves are often used as anatomical features to match surfaces that represent biological objects, such as the human brain. Automated and semi-automated methods for extracting these curves usually rely on local properties of the surfaces such as the mean surface curvature without considering the global appearance of the curves themselves. These methods may require additional human intervention, and sometimes produce erroneous results. In this paper, we present an algorithm that is based on the fast marching method (FMM) to extract weighted geodesic curves. Instead of directly using the local image properties as a weight function, we use the surface properties, together with the global properties of the curves, to compute a weight function. This weight function is then used by the FMM to extract curves between given points. The general framework can be used to extract curves with different global properties. The resulting curves are guaranteed to be weighted geodesic curves without cusps usually introduced by intermediate points through which the curves are forced to pass. We show some results on both a simulated image and a highly convoluted human brain cortical surface.

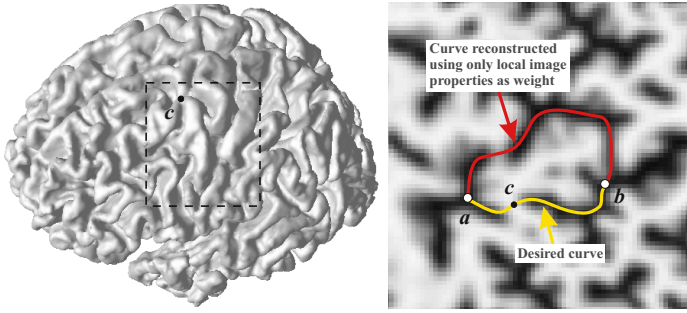
## 1 Introduction

The problem of finding curve features in an image arises in many computer vision and medical image analysis applications. In particular, curves representing anatomical features can be used to match brain surfaces [1, 2, 3]. The curves of sulcal fundi can be viewed as crestlines [5], sulcal roots [4], or as weighted geodesic curves [3, 6], where the weight is usually computed from the local properties (e.g., mean curvature) of the surface [3, 6, 7, 8].

In literature, there have been some methods on how such weighted geodesic curves can be obtained. In [8], finding sulcal fundal curves is formulated as an

---

\* This work was done at Johns Hopkins University.



**Fig. 1.** Left: A human cortical surface. Right: A flattened map [9] of the surface with mean surface curvature of the region around the pre-central sulcus (yellow curve).

optimization problem, which is solved by using a dynamic programming method. The resulting curve is restricted to travel along the edges of the surface, therefore depends on the parameterization of the surface and suffers from metric distortion. More recently, an algorithm based on the FMM has been reported to extract curves of sulcal fundi [3, 6]. In the algorithm, the weight is set to be a function of local “valley-ness”. A FMM is used to compute the weighted distance between any points on the surface to a given origin. A weighted geodesic curve can be constructed by following the negative gradient directions. One of the advantages of this algorithm is that it treats the surface as a continuum; therefore, the resulting curves are not restricted to triangle edges and are independent of the surface parameterization.

A problem with both of the above methods is that there is little control over the global properties of the curves. The method may fail in complicated geometries such as the human cortical surface. As an example, consider the situation shown in Fig. 1. When we use a weight function computed from only mean surface curvature to extract the pre-central sulcal curve between **a** and **b**, we will get the upper curve instead. Because the weight function is computed locally, points in the neighborhood of **c** introduce large cost to curves passing them because they have small mean curvature values. Previously, people circumvented this problem by manually picking intermediate points to force the curve to go over the interruption, such as point **c** shown in Fig. 1. This solution, however, is practical only with additional human intervention.

In order to address the above problem, we develop a novel algorithm based on the FMM to extract weighted geodesic curves that have desired global properties. The method uses the desired properties of the curve to compute a weight function, which is then used to extract the curve.

## 2 Weighted Geodesic Curves

*Background.* In this section, the problem of finding geodesic curves is stated in 2D Cartesian space, but generalization to surfaces is straightforward with

the FMM on triangulated meshes. For a 2D image domain  $\Omega = [0, 1] \times [0, 1]$  and a positive weight function  $w(\mathbf{x}) : \Omega \rightarrow \mathbb{R}^+$ , we define the cost for a curve  $\mathbf{c}(t) : [0, 1] \rightarrow \Omega$  as  $\Phi(\mathbf{c}) = \int_0^1 w(\mathbf{c}(t)) |\mathbf{c}'(t)| dt$ .

If two points  $\mathbf{a} = (a_x, a_y)$ ,  $\mathbf{b} = (b_x, b_y) \in \Omega$  are given, the problem of finding the curve connecting  $\mathbf{a}$  and  $\mathbf{b}$  that minimizes the above cost can be expressed as the following optimization problem:

$$\min \Phi(\mathbf{c}) = \int_0^1 w(\mathbf{c}(t)) |\mathbf{c}'(t)| dt, \quad \text{s.t.} \quad \mathbf{c}(0) = \mathbf{a}, \text{ and } \mathbf{c}(1) = \mathbf{b}. \quad (1)$$

By convention, we define  $F(\mathbf{x}) = 1/w(\mathbf{x})$  as a *speed function*. The solution to Equation (1) can be obtained by first solving the Eikonal equation for  $T(\mathbf{x})$ :

$$\|\nabla T(\mathbf{x})\| = F(\mathbf{x}) = 1/w(\mathbf{x}), \text{ with } T(\mathbf{a}) = 0, \text{ for } \mathbf{x} \in \Omega, \quad (2)$$

then following the negative gradient directions of the function  $T(\mathbf{x})$  starting from  $\mathbf{b}$  until reaching  $\mathbf{a}$ . The Eikonal equation can be solved efficiently using the fast marching method [10]. If the weight function  $w(\mathbf{x}) \equiv 1$ , the cost  $\Phi(\mathbf{c})$  is the length of the curve. The solution to Equation (1) is a geodesic curve on the surface.

*Curves with Global Properties.* For the purpose of clarity, the global property is set to be the intensity profile along the curve segment connecting two given points. Generalization to other properties is straightforward.

Given an image  $I : \Omega \rightarrow \mathcal{R}$  and two points,  $\mathbf{a}$  and  $\mathbf{b}$ , we want to find a curve  $\mathbf{c}(t) = [x(t), y(t)] \subset \Omega$  with  $\mathbf{c}(0) = \mathbf{a}$ ,  $\mathbf{c}(1) = \mathbf{b}$ , such that the intensity values of the image along the curve resemble a given function  $g(t)$  in the least square sense. This problem can be written as the following optimization problem:

$$\begin{aligned} \min \mathcal{E}[x(t), y(t)] &= \int_0^1 [I(x(t), y(t)) - g(t)]^2 \sqrt{x'^2(t) + y'^2(t)} dt, \\ \text{s.t.} \quad x(0) &= a_x, y(0) = a_y, x(1) = b_x \text{ and } y(1) = b_y. \end{aligned} \quad (3)$$

When  $g(t)$  is not a constant, the cost function also depends on the parameterization of the curve  $\mathbf{c}(t)$ , which means that the same curve with different parameterizations will have different costs. One way to circumvent this difficulty is to impose a constant parameterization constraint on the curve  $\mathbf{c}(t)$ , which can be stated as  $|\mathbf{c}'(t)| = L$  with  $L$  being the *unknown* length of the curve. Since  $L$  is unknown,  $|\mathbf{c}'(t)| = L$  cannot be used directly. Instead, we use an equivalent constraint in differential form:  $\frac{d}{dt} |\mathbf{c}'(t)| = 0 \Leftrightarrow x'(t)x''(t) + y'(t)y''(t) = 0$ .

By doing this, Equation (3) is modified to the following form:

$$\min \mathcal{E}[x(t), y(t)] = \int_0^1 [I(x(t), y(t)) - g(t)]^2 \sqrt{x'^2(t) + y'^2(t)} dt \quad (4)$$

and satisfy the initial and final conditions:  $x(0) = a_x$ ,  $y(0) = a_y$ ,  $x(1) = b_x$ , and  $y(1) = b_y$ , and the second order differential equation:  $x'(t)x''(t) + y'(t)y''(t) = 0$ . This is a Lagrange problem with optimality condition given by Mayer Equations [11], which is a system of nonlinear partial differential equations

of order four. Directly solving the problem is difficult. Later, we describe a two-step algorithm to first find a speed function using these global properties, and then extract the weighted geodesic curves using the speed function.

In the above discussion, we assume  $g(t)$  is determinant and known. Sometime, we can only estimate the statistical distribution of  $g(t)$  from a set of training data. For simplicity, we assume that for fixed  $t$ ,  $g(t)$  follows a Gaussian distribution and for  $t_1 \neq t_2$ ,  $g(t_1)$  and  $g(t_2)$  are independent. The point-wise mean and variation of  $g(t)$  are  $\mu_g(t)$  and  $\sigma_g^2(t)$ . If we want to find a curve  $\mathbf{c}(t) = (x(t), y(t))$ , such that  $I(x(t), y(t))$  has large probability according to the statistical properties of  $g(t)$ , the problem in Equation (3) can be written as:

$$\begin{aligned} \min \quad & \mathcal{E}[x(t), y(t)] = \int_0^1 \left[ \frac{I(x(t), y(t)) - \mu_g(t)}{\sigma_g(t)} \right]^2 \sqrt{x'^2(t) + y'^2(t)} dt, \\ \text{s.t. } & \mathbf{c}(0) = (a_x, a_y), \mathbf{c}(1) = (b_x, b_y), \text{ and } x'(t)x''(t) + y'(t)y''(t) = 0. \end{aligned} \quad (5)$$

Again, directly solving the above problem will involve a system of 4th order nonlinear partial differential equations. In the following, we develop an algorithm to find a solution to Equation (5), which we refer to as a *statistical geodesic curve*. We will use speed function  $F(\mathbf{x})$  instead of the weight function  $w(\mathbf{x})$ .

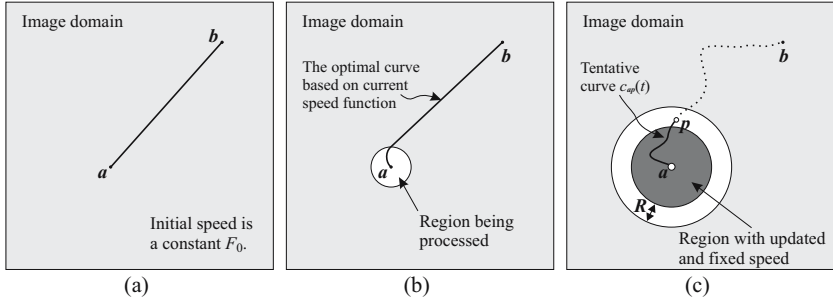
### 3 Finding Statistical Geodesic Curves

Suppose a curve  $\mathbf{c}_0(t)$  is the solution to the problem (5). It minimizes the cost function  $\mathcal{E}(\mathbf{c})$  and must be a weighted geodesic curve under a properly chosen speed function. *Since the speed function uniquely determines the curve when the end points are specified, problem (5) is equivalent to finding the proper speed function.* When the speed function is obtained, the curve can be extracted as a weighted geodesic curve using the method described before.

Now, suppose in addition to the point-wise statistics of  $g(t)$ , we also know that the length of the curve connecting  $\mathbf{a}$  and  $\mathbf{b}$ ,  $\ell$ , follows a Gaussian distribution that is independent of  $g(t)$ . The probability density function is  $f_L(\ell) = 1/\sqrt{2\pi}\sigma_L \exp\{-(\ell - \mu_L)^2/2\sigma_L^2\}$ . The probability density function of the intensity value along the curve is  $f_G(g; t) = \frac{1}{\sqrt{2\pi}\sigma_g^2(t)} \exp\left\{-\frac{(g - \mu_g(t))^2}{2\sigma_g^2(t)}\right\}$ .

We use these probability density functions to find the proper speed function (see Fig. 2 for an illustration). For an image  $I$ , on which the end points  $\mathbf{a}$  and  $\mathbf{b}$  are given, the speed function in the image domain is initialized to be an arbitrary constant. Then an iterative algorithm is applied to successively larger regions surrounding  $\mathbf{a}$ , until the regions cover the entire image domain. At each iteration, for every point  $\mathbf{p}$  [Fig. 2(c)] in the region being processed, a tentative curve,  $c_{\mathbf{ap}}(t)$ , is reconstructed connecting  $\mathbf{a}$  and  $\mathbf{p}$  using the current speed function. The length of  $c_{\mathbf{ap}}(t)$  and the intensity along  $c_{\mathbf{ap}}(t)$  are compared to those of the appropriate segment of the desired curve between  $\mathbf{a}$  and  $\mathbf{b}$ . This step results in a measure of how good  $c_{\mathbf{ap}}(t)$  resembles the corresponding part of the desired curve  $c_{\mathbf{ab}}(t)$ . We use this measure to update the speed value at  $\mathbf{p}$ .

When we update the weight at  $\mathbf{p}$ , we *assume* that  $\mathbf{p}$  is on the final optimal curve [dotted curve in Fig. 2(c)] and compute the ratio  $\tau = \ell_{\mathbf{ap}}/\ell$ , where  $\ell$  is the



**Fig. 2.** Demonstration of the iterative algorithm for finding the speed function

curve length of  $c_{ab}(t)$ . By assumption,  $\ell$  follows a Gaussian distribution with mean  $\mu_L$  and variance  $\sigma_L^2$ . The probability distribution of  $\tau$  is:

$$f_T(\tau | \ell \geq \ell_{ap}) = \frac{dF_T(\tau | \ell \geq \ell_{ap})}{d\tau} \Big|_{\tau=\tau} = \frac{1}{Z} \frac{\ell_{ap}}{\tau^2 \sqrt{2\pi\sigma_L^2}} \exp \left\{ -\frac{(\ell_{ap}/\tau - \mu_L)^2}{2\sigma_L^2} \right\}, \quad (6)$$

where  $Z = \int_{\ell_{ap}}^{\infty} f_L(s)ds$  is a normalization factor. With the distribution of  $\tau$ , we compute the distribution of the intensity at  $\mathbf{p}$ ,  $f(g) = \int_0^1 f_G(g; \tau) f_T(\tau) d\tau$ :

$$f(g) = \frac{\ell_{ap}}{2\pi Z \sigma_L} \int_0^1 \frac{1}{\tau^2 \sigma_g(\tau)} \exp \left\{ -\frac{(g - \mu_g(\tau))^2}{2\sigma_g^2(\tau)} - \frac{(\ell_{ap}/\tau - \mu_L)^2}{2\sigma_L^2} \right\} d\tau. \quad (7)$$

The speed value at  $\mathbf{p}$  is updated as:

$$F(\mathbf{p}) = \int_{I(\mathbf{p})-\delta}^{I(\mathbf{p})+\delta} f(g) dg, \quad (8)$$

where  $I(\mathbf{p})$  is the image intensity at  $\mathbf{p}$  and  $\delta$  is a parameter. To summarize, the algorithm proceeds as follows.

- Step 0.** Initialize the speed function to be an arbitrary positive constant  $F_0$ ;
- Step 1.** For each point  $\mathbf{p} \in \mathcal{N}_k^a = \{\mathbf{p} : \mathbf{p} \in \Omega, (k-1)R \leq \|\mathbf{p} - \mathbf{a}\| < kR\}$ , perform steps 2 and 3.
- Step 2.** Reconstruct a tentative curve  $c_{ap}(t)$  using speed function  $F^{(k-1)}(\mathbf{x})$ .
- Step 3.** Update the speed function at point  $\mathbf{p}$  using Equations (6)–(8).
- Step 4.** Increase  $k$  by 1 and go to **Step 1**.

The above process proceeds until the speed function at every point  $\mathbf{x} \in \Omega$  has been updated. We use the speed function at the final iteration to reconstruct a weighted geodesic curve between  $\mathbf{a}$  and  $\mathbf{b}$ . The image intensity at every point along the resulting curve resembles that presented in the training datasets.

There are two parameters in the algorithm. The first parameter,  $R$ , controls the size of the region whose speed function is updated at each iteration. With

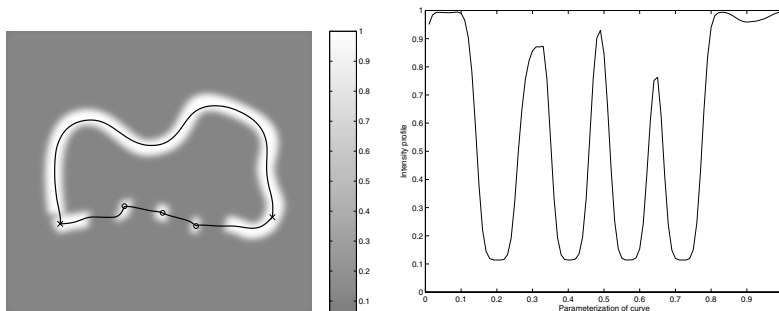
large  $R$ , we need fewer iterations to cover the entire image domain  $\Omega$ , and therefore need less computation time. But we will also get less accurate results, since at each iteration, the updating of the speed function for two points  $\mathbf{p}, \mathbf{q} \in \mathcal{N}_k^{\mathbf{a}}$  may depend not only on the speed function already updated for  $\mathcal{N}_j^{\mathbf{a}}$ ,  $j < k$ , but also on the speed values of each other. In practice, we choose  $R$  to be the pixel size of the image. The parameter  $\delta$  controls how restrict we are on the intensity at a given point. If  $\delta$  is large, we allow a point with a wide range of intensity to have relatively large speed value. An extreme is when  $\delta$  tends to  $\infty$ ,  $F(\mathbf{p}) = 1$ , no matter what  $I(\mathbf{p})$  is. The resulting curve is a true geodesic curve. If  $\delta$  is small, we require the resulting curve to have an intensity profile very close to the mean intensity profile presented in the training datasets.

## 4 Results

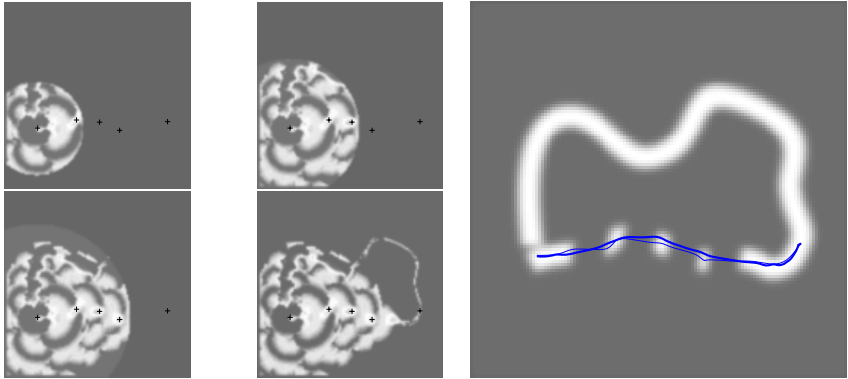
In this section, we describe two experimental results of finding the curves connecting two given points with global properties on the intensity profile of the curve. The first experiment is on a simulated image. And the second one is a segment of the pre-central sulcus on a highly convoluted cortical surface.

*A simulated example.* The simulated image is a gray level image with some bright structures (Fig. 3 left). In the image we specified the end points of the desired curve. If the speed function is set to be the intensity value, the resulting curve will run through regions with high intensity values as much as possible. This is the upper curve in the figure. If instead, the lower curve is desired, without constraints on the global appearance of the curve, we need to specify some intermediate points (circles in the figure). The intensity profile along the desired curve is shown in the right plot of Fig. 3. If we use this intensity as  $g(t)$  to solve problem (3), we get the speed function shown in the left image of Fig. 4. By using the FMM with this speed function and the usual curve tracking technique, we get the curve as shown in the right image of Fig. 4.

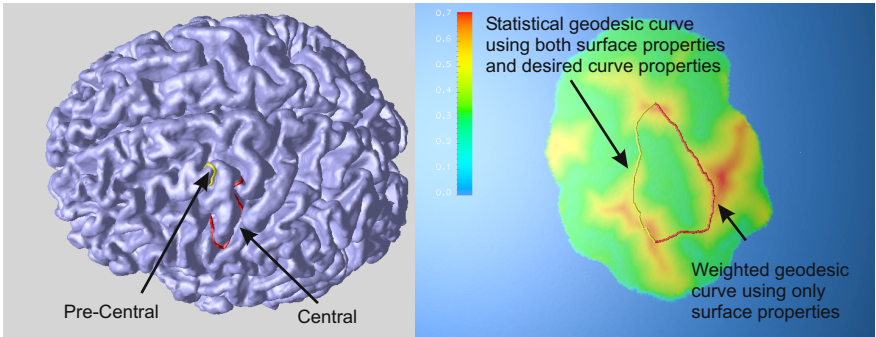
*A curve of sulcal fundi on a cortical surface.* In this experiment, we computed the point-wise statistics of the surface mean curvature along a segment of the curve



**Fig. 3.** Left: A simulated image. Right: The intensity profile along the lower curve in the left image.



**Fig. 4.** Left: Speed functions at different iterations. The small crosses are the intermediate points picked in order to get the desired curve. Right: Statistical geodesic curve (thick curve) extracted from two given end points. The thin curve is reconstructed with manually picked intermediate points and the intensity alone as the speed function.



**Fig. 5.** A statistical geodesic curve on a cortical surface (left), the curve mapped onto the unit sphere with the computed speed function (right, for visualization only)

of pre-central sulcus, and used this information to extract a statistical geodesic curve between two given points on a cortical surface. We used a training set of 30 subjects to estimate the distributions of the length of the sulcal curve and the mean curvature profile along the curve. In the training set, each sulcal curve is extracted using a semi-automatic method with point correspondence built by constant speed parameterization. The results are shown in Fig. 5. When only the mean curvature of the surface is used to extract the curve, we get a segment that connects the pre-central sulcus to the central sulcus (right curve in the figures) because of the structures connecting these two sulci have large local “valley-ness” measure. If we use the statistical information, we are able to get the desired curve representing the part of the pre-central sulcus.

## 5 Conclusions and Discussion

In this paper, we presented an algorithm to extract curves as weighted geodesics using the FMM. The speed function for FMM is computed using both the surface properties and the global properties of the curves. The algorithm can be easily extended to incorporate different global properties.

There are several advantages of the proposed method. The first advantage is automation. The proposed method allows us to extract the curves from only the most distinctive points, which are relatively easy to be automatically located, while manually picked intermediate points do not necessarily have clear features to facilitate automatic localization. The second advantage is that the entire curve extracted as a statistical geodesic curve is always a weighted geodesic curve under the speed function computed, while curve obtained using manually picked intermediate points consists of pieces of weighted geodesic curve segments, and any intermediate point is a cusp of the curve, where the tangent vector is not continuous.

Future work includes investigation on the optimality of the speed function with given constraints. Although the iterative algorithm presented in this paper solves the problem of finding weighted geodesic curves that satisfies given global constraints, it is not known what the optimal speed is for given properties of the curves. A theoretical understanding of the optimality condition of the speed function, as well as the optimality condition in problem (4) is also needed. Another topic of interest is to use other properties of partially reconstructed curve segments, such as curve shape, as the global properties for curve reconstruction.

## References

1. M. Bakircioglu and U. Grenander and N. Khaneja and M. I. Miller, "Curve Matching on Brain Surfaces Using Frenet Distances", *Human Brain Mapping*, 6, 1998.
2. M. Vaillant and A. Davatzikos, "Hierarchical Matching of Cortical Features for Deformable Brain Image Registration", *IPMI*, 1999.
3. X. Tao and X. Han and M. E. Rettmann and J. L. Prince and C. Davatzikos, "Statistical Study on Cortical Sulci of Human Brains", *IPMI*, 2001.
4. D. Riviere, J. F. Mangin, *et al*, "Automatic recognition of cortical sulci of the Human Brain using a congregation of neural networks". *MedIA*, 6(2), 2002.
5. J.P. Thirion and A. Gourdon, "The 3D Marching Lines Algorithm", *Graphical Models and Image Processing*, 58(6), 1996.
6. A. Bartesaghi and G. Sapiro, "A System for the Generation of Curves on 3D Brain Images", *Human Brain Mapping*, 14(1), 2001.
7. A. M. Lopez and F. Lumberras and J. Serrat and J. J. Villanueva, "Evaluation of Methods for Ridge and Valley Detection", *IEEE-PAMI*, vol. 21, no. 4, 1999.
8. N. Khaneja, M. I. Miller and U. Grenander, "Dynamic programming generation of curves on brain surfaces", *IEEE-PAMI*, vol. 20, no. 11, 1998.
9. D. Tosun and J. L. Prince, "A Hemispherical Map for the Human Brain Cortex", *Proc. of SPIE Conf. on Medical Imaging*, 2001.
10. J. A. Sethian, "A fast marching level-set method for monotonically advancing fronts", *Proceedings of the National Academy of Sciences*, vol. 93, 1996.
11. Hans Sagan, "Introduction to the Calculus of Variations", 1992.