The Beltrami Flow over Implicit Manifolds

Nir Sochen

Department of Applied Mathematics
University of Tel-Aviv
Tel-Aviv 69978, Israel

Rachid Deriche and Lucero Lopez Perez

Odyssée Lab.
INRIA, Sophia-Antipolis
Sophia-Antipolis 06902 CEDEX, France

Abstract

In many medical computer vision tasks the relevant data is attached to a specific tissue such as the colon or the cortex. This situation calls for regularization techniques which are defined over surfaces. We introduce in this paper the Beltrami flow over implicit manifolds. This new regularization technique overcomes the over-smoothing of the L_2 flow and the staircasing effects of the L_1 flow, that were recently suggested via the harmonic map methods. The key of our approach is first to clarify the link between the intrinsic Polyakov action and the implicit Harmonic energy functional and then use the geometrical understanding of the Beltrami Flow to generalize it to images on implicitly defined non flat surfaces. It is shown that once again the Beltrami flow interpolates between the L_2 and L_1 flows on non-flat surfaces. The implementation scheme of this flow is presented and various experimental results obtained on a set of various real images illustrate the performances of the approach as well as the differences with the harmonic map flows. This extension of the Beltrami flow to the case of non flat surfaces opens new perspectives in the regularization of noisy data defined on manifolds.

1. Introduction

We have seen in recent years an expansion in the use of differential geometry and calculus of variation for various problems in image processing, computer vision and computer graphics. In particular, one can notice that the problem of regularizing noisy data defined on non flat implicit or intrinsic surfaces has been tackled with two functionals which operate on the space of embedding of Riemannian manifolds (see [9] for non-variational approach)

- **The Polyakov action:** It was introduced in the Beltrami framework [18, 19, 21, 10, 17]. It uses a local and intrinsic-parametric description of the manifolds and an explicit form of the metric.
- **The Harmonic maps.** Recently used to denoise images on manifolds [2, 13, 5, 3, 16]. It uses an implicit description for the surface where the noisy data are constrained to live.

It is the aim of this paper to first clarify the relation between the intrinsic Polyakov action of the Beltrami framework [18, 19, 21, 10, 17] and the implicit harmonic energy functional [2, 13, 5, 3, 16] . It is found that although the functionals are basically the same, there are differences in the way various problems are formulated and consequently in the way the functionals are applied. Specifically for the case of denoising images on non-flat surfaces there are differences in the definition of the manifolds and of the embedding functions.

Usually we have, in various problems in vision, an underline manifold, flat or curved, and the features are defined over this manifold. A typical situation is when the image rectangle is the underlying flat manifold and at each pixel we assign values such as intensity, color, gradient value, gradient direction, motion vector, disparity vector, texture characteristics etc. This is easily described mathematically as a **fiber bundle** in which a space is attached to each point in the base manifold. The spaces at different points of the manifolds are isomorphic. A choice of one point in the attached space for every point in the base manifold is called a **section**. If the attached space is a vector space then this section is called a vector field.

In order to understand better the difference between the two formulations, we consider the case of a gray-value image defined on a surface (flat or not). In the harmonic energy approach it is usually assumed that the map is from the 2D surface (the base manifold) to the real axis (the fiber). This means that **the metric of the base manifold** is used for the derivatives and the fiber's 1D flat metric is used for the values of the scalar field. If the norm chosen is L_2 we get a linear diffusion process whether the base manifold is flat or not. In the case of L_1 norm or the Φ formulation [11, 15, 7] we get non-linear flows. These flows only depend on the absolute value of the gradient, defined over the base manifold [3, 22]

In the Beltrami framework [18, 19, 10], the basic object is **the section embedded in the fiber bundle**. For a flat gray-

Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV 2003) 2-Volume Set
0-7695-1950-4/03 $17.00 © 2003 **IEEE**

value image, thus, the graph of the intensity function is the section of this 3D fiber bundle and is the primary object of interest. The metric of the fiber bundle induces a metric on this section and both are used in the functional. The flow depends on the **geometry of the data** and not only on the geometry of the base manifold. This means that **the image metric, that is, the section metric is being changed along the flow**. It also means that the flow may depend on the direction of the gradients and not only on their amplitudes. The Beltrami flow which has been shown to interpolate, for gray-value images, between the L_2 norm and the L_1 norm [19, 20] is generalized in this paper to images on non-flat surfaces. It is shown, in this work, that the Beltrami flow interpolates between the L_2 and L_1 flows on non-flat surfaces that were derived recently [3, 22] and opens new perspectives in the regularization of noisy data defined on surfaces. A common feature of both frameworks is the ability to deal with non-flat feature spaces [21, 22, 17, 23, 4]. Due to space limitations we follow here a pure geometric approach in the derivation of the denoising algorithm (see [9] for similar approach). The variational point of view is detailed in [1].

This article is organized as follows: In Section 2 we briefly review the Beltrami and harmonic map frameworks. We take in Section 3 the Beltrami formulation of the problem and reformulate it in an implicit form. Section 4 presents examples and results and we summarize and conclude in Section 5.

2 The Beltrami and the Harmonic Map frameworks

There are two ways to write the geometric functional which is called harmonic energy or the Beltrami functional that weight embedding of Riemannian manifolds (for early non-variational use of the Laplace-Beltrami operator on flat images see [8]) . We focus our attention to the case of a gray-value image defined over a two- (or one-) dimensional manifold i.e. a surface (curve). The first way is to represent the surface implicitly i.e. as a zero section of a distance function, defined over $\mathbb{R}^3$. The other way will be to choose a local coordinate system and define the surface intrinsically in a parametric way. The difference with the latter is explained theoretically and demonstrated on real and synthetic images. developed in more details in the next two-subsections.

2.1 Implicit formulation

Let's suppose that we have a scalar data U defined on a known surface Ψ, that is we have a mapping $U : \Psi \rightarrow \mathbb{R}^1$ where Ψ is a known surface, given by its implicit representation $\Psi(X^1, X^2, X^3) = 0$. The Data function U is extended to $\mathbb{R}^3$. One possible way for this extension is to solve the initial value problem $U_t + sign(\Psi)(\nabla U \cdot \nabla \Psi) = 0$ as done in [2]. The *Harmonic maps* approach [2, 13, 5, 3] is to look for the minimizer of the following energy

$$S_{imp}[U] = \int_{\Psi} \| \nabla_{\nabla \Psi} U \|^2 \, d\Psi \tag{1}$$

where $d\Psi$ is a surface element $\delta(\Psi)||\nabla\Psi||dX^1dX^2dX^3$. We denote here by δ the Dirac function and $\nabla_{\nabla\Psi}U$ denotes the gradient intrinsic to the surface Ψ i.e the surface gradient applied to the restriction of U on Ψ. Noting the property that $\nabla_{\nabla\Psi}U$ is just the projection onto the surface of the 3D Cartesian classical gradient $\nabla_{\mathbb{R}^3}$ that is to say $\nabla_{\nabla\Psi}U = P_\eta \nabla_{\mathbb{R}^3} U$ where $P_{\vec{\eta}}$ is the orthogonal projection operator on the surface Ψ and η, the unit normal vector to the surface Ψ with

$$\eta = \frac{\nabla_{\mathbb{R}^3}\Psi}{||\nabla_{\mathbb{R}^3}\Psi||}$$

and

$$\nabla_{\mathbb{R}^3} U^t = \left(\frac{\partial U}{\partial X^1}, \frac{\partial U}{\partial X^2}, \frac{\partial U}{\partial X^3} \right)$$

We can rewrite this harmonic energy as follows :

$$\begin{aligned} S_{imp}[U] &= \int_{\Psi} \| P_\eta \nabla U \|^2 \, d\Psi \\ &= \int_{\mathbb{R}^3} \| P_{\nabla\psi} \nabla U \|^2 \, \delta(\Psi) \, \| \nabla \Psi \| \, dX^1 dX^2 dX^3 \end{aligned}$$

all the expressions above are considered in the sense of distributions. Noting that given a surface with a unit normal vector η, the orthogonal projection operator $P_{\vec{\eta}}$ on that surface is given by :

$$P_\eta = I - \eta\eta^t \tag{2}$$

where I is the 3×3 Identity matrix and η^t denotes the transposed vector η, one can rewrite this functional as follows :

$$S_{imp}[U] = \int_{\mathbb{R}^3} \nabla_{\mathbb{R}^3} U^t (I - \eta\eta^t) \nabla_{\mathbb{R}^3} U d\Psi \tag{3}$$

This functional is the generalization of the L_2 norm from flat to non-flat manifolds. The modified gradient descent equation reads:

$$u_t = \frac{1}{||\nabla\Psi||} \mathrm{Div}\left(||\nabla\Psi|| P_{\nabla\Psi} \nabla U \right) \tag{4}$$

where Div stands for the Divergence operator. This equation can be generalized via the Φ formulation [11, 15, 7] to

Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV 2003) 2-Volume Set
0-7695-1950-4/03 $17.00 © 2003 IEEE

$$S_{imp}[U] = \int_{\mathbb{R}^3} \Phi\left(||\nabla_{\nabla\Psi} U\ ||\right) d\Psi \tag{5}$$

and in particular, for the generalized L_1 norm

$$S_{imp}[U] = \int_{\mathbb{R}^3} ||\nabla_{\nabla\Psi} U\ ||\ d\Psi \tag{6}$$

we obtain the following minimization flow

$$u_t = \frac{1}{||\nabla\Psi||||P_{\nabla\Psi}||} \mathrm{Div}\left(\frac{||\nabla\Psi||P_{\nabla\Psi}\nabla U}{||P_{\nabla\Psi}\nabla U||}\right) \tag{7}$$

For the interested reader, we refer to [2, 13, 5, 3] for more details and results on this implicit formulation and its various applications.

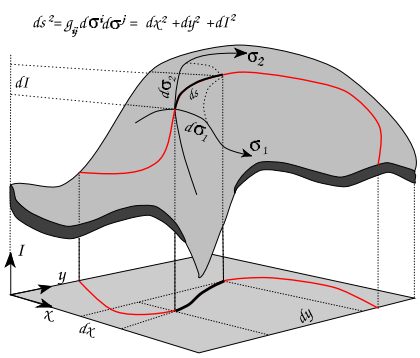


Figure 1: The graph of the intensity map as a surface embedded in $\mathbb{R}^3$

2.2 Intrinsic formulation

Suppose we have an 2-dimensional manifold Σ with local coordinates σ^1, σ^2 embedded in an 3-dimensional manifold M with coordinates X^1, X^2, X^3, the embedding map $X : \Sigma \to M$ is given explicitly by the 3 functions of 2 variables $X : (\sigma^1, \sigma^2) \longrightarrow (X^1(\sigma^1, \sigma^2), X^2(\sigma^1, \sigma^2), X^3(\sigma^1, \sigma^2))$
A gray-value image is represented, in this framework, as the embedding $(X^1 = \sigma^1, X^2 = \sigma^2, X^3 = U(\sigma^1, \sigma^2))$ (see Fig. 1).
Denote by (Σ, g) the image manifold and its metric and by (M, h) the space-feature manifold and its metric, then the map $X : \Sigma \to M$ has the following weight :

$$S_{int}[U, g] = \int \sqrt{g} g^{\mu\nu} \partial_\mu X^i \partial_\nu X^j h_{ij}(X) d\sigma^1 d\sigma^2 \tag{8}$$

where g is the determinant of the image metric, $g^{\mu\nu}$ is the inverse of the image metric, the range of indices is $\mu, \nu = 1, .., 2$ and $i, j = 1, ..3$ and h_{ij} is the metric of the embedding space. We use the summation convention: indices that appear twice are being summed over. This functional was first proposed by Polyakov in the context of high energy physics [14] .
The gradient descent equations, with respect to the embedding functions, are

$$X^i_t = \frac{1}{\sqrt{g}} \partial_\mu(\sqrt{g} g^{\mu\nu} \partial_\nu X^i) + \Gamma^i_{jk}(\partial_\mu X^j)(\partial_\nu X^k) g^{\mu\nu}. \tag{9}$$

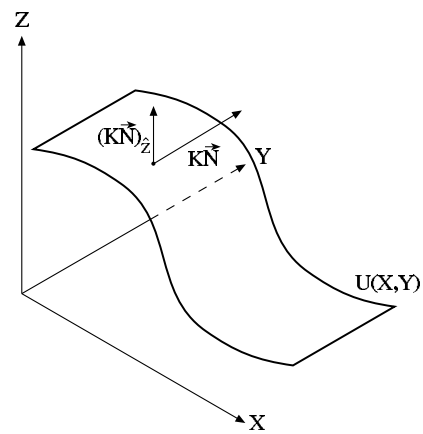


Figure 2: The Beltrami flow velocity as a projection of the mean curvature on the data = z axis

In our case X^1 and X^2 are fixed and we only change $X^3 = u$. The embedding space is Euclidean and therefor the Levi-Civita Γ^i_{jk} coefficients are zero.
We end up with the following flow:

$$U_t = \frac{1}{\sqrt{g}} \partial_\mu(\sqrt{g} g^{\mu\nu} \partial_\nu U) \tag{10}$$

Since our two-dimensional manifold is the section, i.e. the graph of U, the metric $(g_{\mu\nu})$ is the metric of the section. It can be calculated explicitly, as the induced metric, from $\mathbb{R}^3$. It is find to be [19] :

$$(g_{\mu\nu}) = \begin{pmatrix} 1 + U_x^2 & U_x U_y \\ U_x U_y & 1 + U_y^2 \end{pmatrix}. \tag{11}$$

Note that this metric is also **the minimizer of** $S_{int}[U, g]$ with respect to the metric [18].
Equation (10), with the induced metric Eq. (11), has a clear geometric meaning. It is the projection of the mean curvature vector in the $X^3 = U$ direction:

$$u_t = K N_{\hat{U}} \tag{12}$$

where K is the mean curvature magnitude. N is the normal to the surface and $\hat{U}$ is a unit vector in the $X^3 = U$ direction (see Fig. 2).

Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV 2003) 2-Volume Set
0-7695-1950-4/03 $17.00 © 2003 **IEEE**

It is this geometric understanding that we are using below in order to derive the Beltrami flow on non-flat implicit surfaces.

3 Data Regularization on an implicitly defined curves and surfaces

Although the derivation of the flow equations for scalar data defined over a curve and that over a surface is very similar it is easier to have a clear mental image of the situation for curves. We describe, therefore, in the first subsection, the derivation in details in this simpler situation. That derivation is then generalized, in the second subsection, in a straightforward manner to the case of the surface. We analyze in the last subsection the L_1 and L_2 limits.

3.1 Scalar field defined on a 2D Curve

The curve is given in the x-y plane and the data is pictured as the height in the z direction. The data from this point of view is a curve in $\mathbb{R}^3$. Note that this curve lies on the cylinder-like surface defined by $\Psi(x,y) = 0$. We want to produce a curvature flow of this data such that it is confined to the surface which means that it is defined on the base curve. The projected curvature on the surface is the geodesic curvature. Note however that the fact that the data curve is on the cylinder-like surface, generated by the base curve, is not enough in general to guaranty that the data curve can be represented as a function on the base curve since the evolution is not done in the z direction only (compare with [5]). It may happen, for a general flow, that at some time of the flow more than one point of the data curve has the same x-y values. In order to avoid that and in order to have a flow that respect discontinuities we confine the flow by projecting the geodesic curvature on the z direction. It is the analog of the Beltrami flow which is a projection of the mean curvature flow on the intensity direction.

Formulating this geometric understanding, we represent the curve as the intersection of two surfaces :

$$\begin{aligned} \Psi(x,y) &= 0 \\ \Phi(x,y,z) &= z - \beta U(x,y) = 0. \end{aligned} \tag{13}$$

The surface defined by Ψ is a cylinder-like surface defined by the base curve, which lies in the x-y plane. The second surface is the graph of the function U which is defined over the x-y plane (and by restriction over the base curve). The data function is modified along the flow in the z direction only. It remain, thus, a function along the flow (see Fig. 3). We denote by D the 3D gradient $D = (\partial_x, \partial_y, \partial_z)^t$, and by ∇ the 2D gradient $\nabla = (\partial_x, \partial_y)^t$. We define the tangent vector to the curve by

$$T = \frac{D\Psi \times D(z - \beta U)}{||D\Psi \times D(z - \beta U)||} = [T_1, T_2, T_3]^t$$

The curvature is given by the second derivative of the curve.

$$\begin{aligned} kN^t &= \partial_s T = x_s \partial_x T + y_s \partial_y T + z_s \partial_z T = T^t \nabla T \\ &= (T \cdot \nabla T_1, T \cdot \nabla T_2, T \cdot \nabla T_3)^t \end{aligned} \tag{14}$$

The projection of this vector on the surface defined by $\Psi(x,y) = 0$ is $P_{D\Psi} kN$ and keeping only the z component we get

$$u_t = (P_{D\Psi} kN)_z \tag{15}$$

which is the non-flat analogue of Eq. (12).

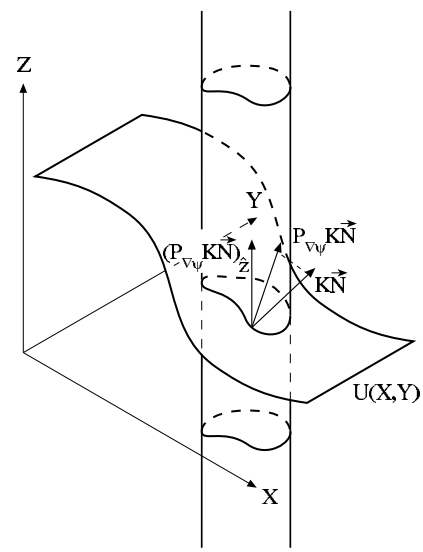


Figure 3: The data curve as the intersection of the cylinder-like surface induced by the base planar curve, extended to $\mathbb{R}^3$, and the graph of the function U.

Let us calculate it explicitly. We have the following:

$$\begin{aligned} (D\Psi \times D(z - \beta U))^t &= (\Psi_y, -\Psi_x, \beta(\Psi_y U_x - \Psi_x U_y)) \\ D(z - \beta U)^t &= (-\beta U_x, -\beta U_y, 1) \\ D\Psi^t &= (\Psi_x, \Psi_y, 0) \end{aligned} \tag{16}$$

Note that T does not depend on z! The projection operator is

$$\begin{aligned} P_{D\Psi} &= I - \frac{D\Psi D\Psi^t}{||D\Psi||^2} \\ &= \frac{1}{||D\Psi||^2} \begin{pmatrix} \Psi_y^2 & -\Psi_x\Psi_y & 0 \\ -\Psi_x\Psi_y & \Psi_x^2 & 0 \\ 0 & 0 & ||D\Psi||^2 \end{pmatrix} \end{aligned}$$

Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV 2003) 2-Volume Set
0-7695-1950-4/03 $17.00 © 2003 IEEE

The equation of motion now reads

$$u_t = \left(P_{D\Psi} \begin{pmatrix} T \cdot \nabla T_1 \\ T \cdot \nabla T_2 \\ T \cdot \nabla T_3 \end{pmatrix} \right)_z = T \cdot \nabla T_3$$

Since T does not depend on z, only the x and y components enter the calculation:

$$u_t = T_1 \partial_x T_3 + T_2 \partial_y T_3$$

The gradient notation ∇ and all other vector notations are from now on two-dimensional. We also denote the perpendicular gradient by $\nabla^p \Psi = (\Psi_y, -\Psi_x)^t$. Notice also that in the x-y plane we have the identities

$$\begin{aligned} ||\nabla\Psi||^2 P_{\nabla\Psi}\nabla U &= (\Psi_y U_x - \Psi_x U_y)\nabla^p \Psi \\ ||\nabla\Psi||^2 ||P_{\nabla\Psi}\nabla U||^2 &= (\Psi_y U_x - \Psi_x U_y)^2 \end{aligned} \quad (17)$$

Define

$$\begin{aligned} g &= ||\nabla\Psi||^2 + \beta^2 (\Psi_y U_x - \Psi_x U_y)^2 \\ &= ||\nabla\Psi||^2 \left(1 + \beta^2 ||P_{\nabla\Psi}\nabla U||^2\right). \end{aligned} \quad (18)$$

We can finally write the implicit Beltrami flow as follows

$$\begin{aligned} U_t &= \frac{\nabla^p \Psi}{\sqrt{g}} \cdot \nabla \left(\frac{\beta(\Psi_y U_x - \Psi_x U_y)}{\sqrt{g}} \right) \\ &= \frac{1}{\sqrt{g}} \mathrm{Div} \left(\frac{\beta ||\nabla\Psi||^2 P_{\nabla\Psi}\nabla U}{\sqrt{g}} \right) \end{aligned} \quad (19)$$

where we used the identities Eq. (17) and the fact that $\mathrm{Div}\nabla^p\Psi = \partial_x(\Psi_y) + \partial_y(-\Psi_x) = 0$.
In conclusion, this last equation allows us to denoise the scalar signal U lying on the implicitly defined curve $\Psi(x, y) = 0$. This Beltrami flow defined on implicit curve can also be generalized to the case of scalar field defined on implicit surface.

3.2 Scalar field defined on a 2D surface

The situation, in this case, is completely analogous to the one we described above. We are given an implicit surface $\Psi(X^1, X^2, X^3) = 0$ and a scalar image defined on the surface $U(X^1, X^2, X^3)$. The data surface is represented as a 2D surface embedded in 4D space. It is described by the intersection of two hyperplanes: $\Psi(X^1, X^2, X^3) = 0$ and a second one that emphasizes the role of the data direction : $\Phi(X^1, X^2, X^3, X^4) = X^4 - U(X^1, X^2, X^3) = 0$. The Beltrami flow is given by the geodesic curvature projected to the U direction. The resulting flow is the straightforward generalization of the curve case (see [1] for details) :

$$U_t = \frac{1}{\sqrt{g}} \mathrm{Div} \left(\frac{\beta ||\nabla\Psi||^2 P_{\nabla\Psi}\nabla U}{\sqrt{g}} \right). \quad (20)$$

3.3 The L_1 and L_2 limits

We show in this subsection that Eq. (20) interpolates between the non-flat L_1 flow Eq. (7) and the L_2 flow Eq. (4). Note that

$$\begin{aligned} \lim_{\beta\to 0} g &= ||\nabla\Psi||^2 \\ \lim_{\beta\to\infty} g &= \beta^2 ||\nabla\Psi||^2 ||P_{\nabla\Psi}\nabla U||^2 \end{aligned} \quad (21)$$

It is easily seen now that the implicit Beltrami flow Eq. (20), in the limit $\beta \to 0, \quad t \to \infty$ such that $\tau = t\beta$ is finite, goes to

$$u_\tau = \frac{1}{||\nabla\Psi||} \mathrm{Div}\left(||\nabla\Psi|| P_{\nabla\Psi}\nabla U \right),$$

which is the non-flat L_2 flow, which was derived differently by Bertalmio et al [2].
In the other limit, $\beta \to \infty, \quad t \to \infty$ such that $\tau = t/\beta$ is finite, we obtain another flow :

$$u_\tau = \frac{1}{||\nabla\Psi|| ||P_{\nabla\Psi}\nabla U||} \mathrm{Div}\left(\frac{||\nabla\Psi|| P_{\nabla\Psi}\nabla U}{||P_{\nabla\Psi}\nabla U||} \right)$$

which is the non-flat L_1 flow, used in [2].
In one limit, we find the L_2 norm which over-smooth the image, while in the other limit we find the L_1 flow with the notorious "staircasing" effect. Choosing for β intermediate value brings more degree of freedom in the regularization of noisy data defined on surfaces. It opens news perspectives with the advantage of smoothing anisotropically the image and conserving the edges while avoiding the disadvantages of the total variation type of flow. This is well illustrated and demonstrated on various synthetic and real images in the next section.

4 Examples and Results

In this section, we first give some implementation details and then illustrate the capabilities of the approach we have developed to regularize noisy signals and images defined on implicit curves and surfaces. Various experimental results have been carried out, but due to space limitations, we just give some figures for illustrations.

4.1 Implementing the regularization of scalar fields on surfaces

We compute the value of $u^n_{i,j,k}$, the value of u in the pixel (i, j, k) at the nth iteration, based on the values of u^{n-1} at the neighboring pixels. First, we compute the vector $\vec{N} \simeq \nabla\Psi$ (this is needed only once), by central differences. Then, for each iteration n, we visit all pixels to compute:

- The gradient, its projection, and g.

Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV 2003) 2-Volume Set
0-7695-1950-4/03 $17.00 © 2003 IEEE

- The divergence.
- The actualization of u.

For the gradient $\vec{v}^n_{i,j,k}$ we used backward differences,

$$\vec{v}^n_{i,j,k} = \nabla_+ u^n_{i,j,k} = \begin{pmatrix} u^n_{i+1,j,k} - u^n_{i,j,k} \\ u^n_{i,j+1,k} - u^n_{i,j,k} \\ u^n_{i,j,k+1} - u^n_{i,j,k} \end{pmatrix}$$

its projection on the surface,

$$\left(P_{\vec{N}}\vec{v}\right)^n_{i,j,k} = \vec{v}^n_{i,j,k} - \frac{\sum_{m=1}^{3} \vec{N}_{i,j,k}[m] \cdot \vec{v}^n_{i,j,k}[m]}{\left\|\vec{N}_{i,j,k}\right\|^2} \vec{N}_{i,j,k}$$

and g,

$$g^n_{i,j,k} = \|\vec{N}_{i,j,k}\|^2 \left(1 + \beta^2 \| \left(P_{\vec{N}}\vec{v}\right)^n_{i,j,k} \|^2\right),$$

where square brackets represent the component of the vector. To compute the divergence, we use backward differences,

$$\begin{aligned} \nabla_- \cdot \vec{w}_{i,j,k} \quad = \quad & \vec{w}_{i,j,k}[1] - \vec{w}_{i-1,j,k}[1] + \\ & \vec{w}_{i,j,k}[2] - \vec{w}_{i,j-1,k}[2] + \\ & \vec{w}_{i,j,k}[3] - \vec{w}_{i,j,k-1}[3] \end{aligned}$$

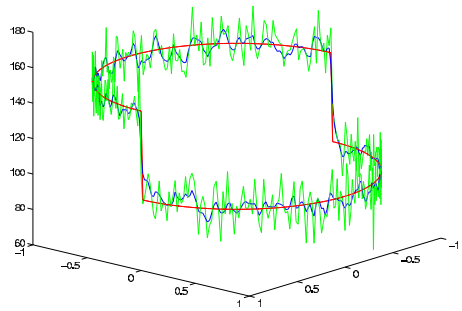
We switch forward differences and backward differences to avoid numerical problems. Finally, the flow implementation is:

$$u^{n+1}_{i,j,k} = u^n_{i,j,k} + \Delta t \frac{1}{\sqrt{g^n_{i,j,k}}} \nabla_- \cdot \left(\frac{\beta \left\|\vec{N}_{i,j,k}\right\| \left(P_{\vec{N}}\vec{v}\right)^n_{i,j,k}}{\sqrt{g^n_{i,j,k}}} \right)$$

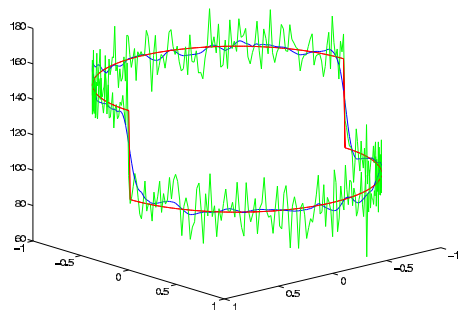
We use a time step $\frac{dt}{\beta}$, adjusted accordingly to section 3.3. The code is made in C++ using the libraries developed in our lab. For the visualization, we used the marching cubes algorithm [12] to obtain a triangulation from our implicit representation of the surface, and draw the data on this surface. This was made using VTK.

4.2 Examples

We present in this subsection few figures that illustrate the regularization of noisy data on various implicitly defined curves and surfaces. The results are given with various values of the parameter β. Note how the regularization of the data is done isotropically or anisotropically depending on the value of this parameter.
For the next set of images: the red curve is the original function, the green curve is obtained by adding noise, and the blue curve is after regularization. For images (a) and (b), we added a Gaussian noise with $\sigma = 15$.



(a) $\beta = 0.1$



(b) $\beta = 0$ (Isotropic diffusion)

Regarding the CPU time, for the tori (12500 pixels), it took less than 2 seconds for 20 iterations. Please note that we used a color map from red to blue in order to better see how the the discontinuities are treated. Regarding the noise level, we have added a Gaussian noise with ($\sigma = 40$) to the grey level images which scalar range is (0,256). [h] For the

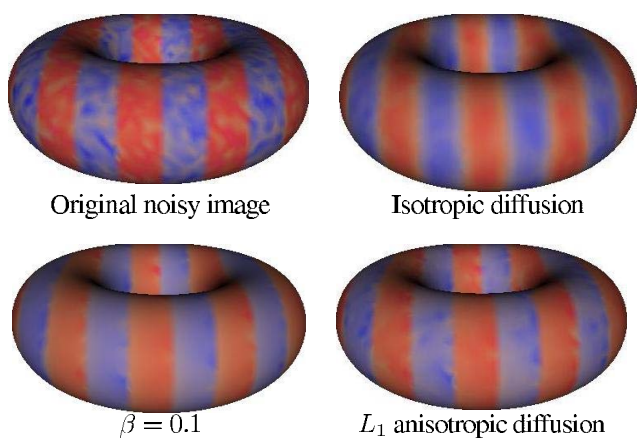


Figure 4: Synthetic image on the torus regularized with different β

Ella image, illustrated in Fig. (5) and defined in a solid that looks like a quadratic, the image is much bigger (2744000 pixels). The time to compute 20 iterations was almost 3.5 minutes in a 386 sun, 260 MB in RAM. We can see that in

Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV 2003) 2-Volume Set
0-7695-1950-4/03 $17.00 © 2003 IEEE

strongly noised images like these, where anisotropic regularization treat some noise as part of the image discontinuities (the points under the left eye, for example) the regularization with $\beta = 0.1$ performs better.

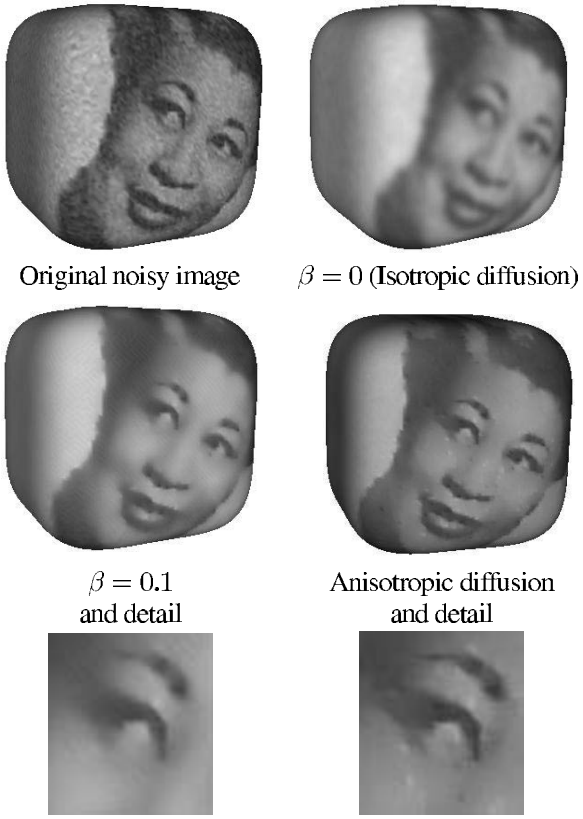


Figure 5: Ella image regularized with different β

Finally, for a slice from a cortex (97x222x143) acquired at IRMf Center of Marseille (N. Wotawa together with J.L. Anton and his colleagues are gratefully acknowledged for their support) it took less than 10 minutes to compute these results (fig 6). This is a retinotopic map i.e a neural representation within the visual cortex that preserves the spatial layout of the retina image. Notice the red holes in the noisy image. The small holes have to be filled in the map. Our approach allows to fill in these holes while preserving the important borders. While the isotropic smoothing also performs this task, it doesn't allow to preserve important information like the blue zone in the extreme right. With the L_1 anisotropic smoothing á la Bertalmio et al, the outer borders are thinner, but the inner holes rest. Our approach allows us to deal with more degree of freedom in the process to manage the weighting between the isotropy and anisotropy processes. Choosing beta in a range from 0.1 to 0.5 allows to produce a whole set of interesting and better results since the blue zone is kept and the holes are filled in.

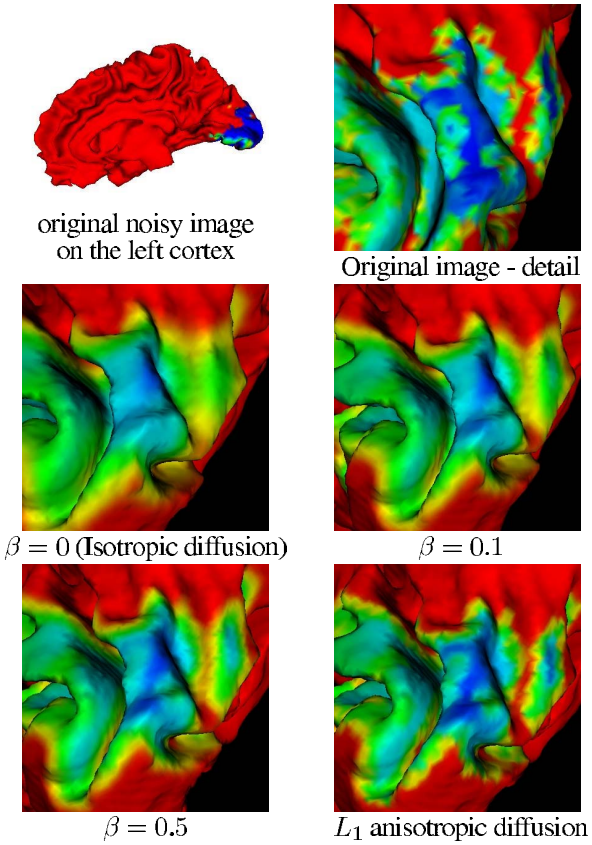


Figure 6: Retinotopic images regularized with different β

5 Summary and conclusions

In this paper, we have first clarified the link that exists between the the intrinsic Polyakov action of the Beltrami framework and the implicit harmonic energy functional . It is found that although the functionals are basically the same, there are differences in the way various problems are formulated and consequently in the way the functionals are applied.

We used the geometrical understanding of the flat Beltrami flow and generalized it to a denoising flow over implicitly defined curves and surfaces. It is shown that this flow depends on β which encodes the ratio between the data and spatial units. This parameter controls the edge-preserving characteristic of the flow. The Beltrami flow is shown to act as the linear diffusion (L_2-norm) in the limit $\beta \to 0$ and to the strongly edge preserving diffusion (L_1-norm) in the limit $\beta \to \infty$ (up to time scaling).

This work opens interesting perspectives in some important applications such as the inverse EEG-MEG problem. See for instance the work presented in [6] where the authors are interested by localizing cortex activity from EEG/MEG measurements. The PDE associated to the inverse problem

Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV 2003) 2-Volume Set
0-7695-1950-4/03 $17.00 © 2003 IEEE

includes a regularization term on the implicitly defined surface of the cortex. It could certainly be of interest to apply the implicit Beltrami flow, we developed, to regularize such data defined on implicit cortex surface. It will also be of great interest to compare our result to the methods developed to regularize data defined on triangulated surfaces. For more details, we refer the interested reader to our incoming research report where the intrinsic formulation and the derivation of the implicit Beltrami flow from a variational point of view are also presented (see [1] for details).

Acknowledgments

This research has been supported in part by the Israel Academy of Science, Tel-Aviv University fund, the Adams Super-Center for Brain Studies, the Israeli Ministry of Science, the European project MAPAWAMO, the INRIA Department of the European and International Relations (DREI) and the program CEFI-SFERE/CONACYT

References

[1] N. Sochen, R. Deriche and L. Lopez Perez, The Beltrami flow over Manifolds, Inria Research Report, July 2003

[2] M. Bertalmío, L. T. Cheng, S. Osher and G. Sapiro, "Variational Problems and Partial Differential Equations on Implicit Surfaces", Journal of Computational Physics 174 (2001) 759-780.

[3] T. Chan and J. Shen, "Variational restoration of non-flat image features: Models and algorithms", SIAM J. Appl. Math., 61 (2000) 1338-1361.

[4] C. Chefd'hotel, D. Tchumperle, R. Deriche, O. Faugeras, "Constrained Flows of Matrix-Valued Functions: Application to Diffusion Tensor Regularization", *Proceedings of 7th European Conference on Computer Vision*, Copenhagen, Denmark, May 2002

[5] L. Cheng, P. Burchard, B. Merriman and S. Osher, "Motion of Curves Constrained on Surfaces Using a Level Set Approach", September 2000 UCLA CAM Technical Report (00-32).

[6] M. Clerc, O. Faugeras, R. Keriven, J. Kybic, T. Papadopoulo, "A level set method for the inverse EEG/MEG problem", Poster No.: 10133, NeuroImage Human Brain Mapping 2002 Meeting

[7] R. Deriche and O. Faugeras, "Les EDP en Traitement des Images et Vision par Ordinateur", INRIA Research Report 2697, Nov. 1995. Also appeared in Traitement du Signal, 13 (6) 1996.

[8] L. Florack, A. Salden, B. ter Haar Romeny, J. J. Koenderink and M. A. Viergever, "Non-Linear Scale-Space", Image and Vision Computing 13: 279-294 1995.

[9] R. Kimmel, "Intrinsic Scale Space for Images on Surfaces: The Geodesic Curvature Flow", *Graphical Models and Image Processing* 59(5) 365-372 1997.

[10] R. Kimmel and R. Malladi and N. Sochen, "Images as Embedding Maps and Minimal Surfaces: Movies, Color, Texture, and Volumetric Medical Images", International Journal of Computer Vision 39(2) (2000) 111-129.

[11] P. Kornprobst, R. Deriche, and G. Aubert. "Nonlinear operators in image restoration". In Proceedings of the International Conference on Computer Vision and Pattern Recognition - pages 325-331.IEEE Computer Society. San Juan, Puerto Rico, June 1997.

[12] W. E. Lorensen, H. E. Cline, "Marching cubes: A high resolution 3D surface construction algorithm", Computer Graphics, 21(4), pages 163–169, 1987.

[13] F. Memoli and G. Sapiro and S. Osher, "Solving Variational Problems and Partial Differential Equations, Mapping into General Target Manifolds", January 2002 UCLA CAM Technical Report (02-04).

[14] A. M. Polyakov, "Quantum geometry of bosonic strings", *Physics Letters*, **103B** (1981) 207-210.

[15] L. Rudin, S. Osher and E. Fatemi, " Non Linear Total Variation Based Noise Removal Algorithms", *Physica D 60 (1992) 259-268.*

[16] G. Sapiro, "Geometric Partial Differential Equations and Image Analysis", Cambridge University Press, January 2001.

[17] N. Sochen and R. Kimmel, "Orientation Diffusion or How to comb a Porcupine", *Journal of Visual Communication and Image Representation* 13:238-248, 2001.

[18] N. Sochen and R. Kimmel and R. Malladi, "From high energy physics to low level vision", Report, LBNL, UC Berkeley, LBNL 39243, August, Presented in ONR workshop, UCLA, Sept. 5 1996.

[19] N. Sochen and R. Kimmel and R. Malladi, "A general framework for low level vision", *IEEE Trans. on Image Processing*, 7 (1998) 310-318.

[20] N. Sochen and Y. Y. Zeevi, "Representation of images by surfaces embedded in higher dimensional non-Euclidean space", 4th International Conference on Mathematical Methods for Curves and Surfaces, Lillehamer, Norway, July 1997.

[21] N. Sochen and Y. Y. Zeevi, "Representation of colored images by manifolds embedded in higher dimensional non-Euclidean space", Proc. IEEE ICIP'98, Chicago, 1998.

[22] B. Tang and G. Sapiro and V. Caselles, "Diffusion of General Data on Non-Flat Manifold via Harmonic Map Theory: The Direction Diffusion Case", *International Journal on Computer Vision* 36(2) 149-161, 2000.

[23] D. Tchumperlé and R. Deriche, "Orthonormal Vector Sets Regularization with PDEs and Applications", *International Journal on Computer Vision* 50(3) 237-252 (1992).

Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV 2003) 2-Volume Set
0-7695-1950-4/03 $17.00 © 2003 **IEEE**