

Geodesic Active Fields on the Sphere

Dominique Zosso, Jean-Philippe Thiran
Signal Processing Laboratory LTS5
École Polytechnique Fédérale de Lausanne (EPFL)
Station 11, 1015 Lausanne, Switzerland
dominique.zosso@epfl.ch, jp.thiran@epfl.ch

Abstract—In this paper, we propose a novel method to register images defined on spherical meshes. Instances of such spherical images include inflated cortical feature maps in brain medical imaging or images from omnidirectional cameras. We apply the Geodesic Active Fields (GAF) framework locally at each vertex of the mesh. Therefore we define a dense deformation field, which is embedded in a higher dimensional manifold, and minimize the weighted Polyakov energy. While the Polyakov energy itself measures the hyperarea of the embedded deformation field, its weighting allows to account for the quality of the current image alignment. Iteratively minimizing the energy drives the deformation field towards a smooth solution of the registration problem. Although the proposed approach does not necessarily outperform state-of-the-art methods that are tightly tailored to specific applications, it is of methodological interest due to its high degree of flexibility and versatility.

Index Terms—Biomedical image processing, Computational geometry, Differential geometry, Diffusion equations, Image registration, Scale-spaces, Spheres, Surfaces.

I. INTRODUCTION

In this work, we are interested in a method to register images that are sampled at the vertices of a spherical mesh. Spherical image registration has several applications, in particular in brain imaging, for functional and anatomical analysis of cortical feature maps, obtained through inflation of the original cortical sheet [1], [2]. Other potential applications include stereo vision and hyperresolution from omnidirectional images, for robot navigation or surveillance [3].

Strictly speaking, image registration is the concept of mapping homologous points of different images, representing a same object. In practice, however, it is often difficult to establish homology in images based on this definition. For automatic image registration, it is commonplace to substitute homology by a measurable criterion of image dissimilarity, which is to be minimized by an unknown deformation field \vec{u} . The determination of this deformation field is an ill-posed inverse problem.

In previous work, we propose to embed the deformation field in a higher dimensional manifold, inspired by the Beltrami-framework [4], [5]. There, the deformation field is driven by a minimization flow towards a harmonic map corresponding to the solution of the registration problem, much like geodesic active contours in image segmentation. The energy of the deformation field is measured with the Polyakov energy weighted by a suitable image distance, borrowed from standard registration models.

Here, we want to apply this framework, called geodesic active fields (GAF), to images defined on triangulations of the sphere. First, we establish the necessary tools to define the local embedding and to calculate the minimizing flow. Then, we characterize the impulse response and the denoising properties of the regularizer, and illustrate the framework on synthetic images.

II. GEODESIC ACTIVE FIELDS ON THE SPHERE

Geodesic active fields embed the deformation field in a higher dimensional space and define a variational model using the weighted Polyakov energy. While the Polyakov energy itself only provides a regularity constraint, the weighting drives the deformation field towards low image dissimilarity. This approach directly generalizes to non-Euclidean images, and thus automatically allows to work, e.g., with non-flat or multiscale images, that are smooth and parametrizable.

Due to the “hairy ball”-theorem, however, it is known that no artifact-free, global parameterization of the whole sphere exists. Therefore the GAF framework can not be applied directly to the whole spherical image. Instead, we decide to work in local coordinates, by defining a local coordinate chart for each vertex of the mesh. Using those local coordinate charts, the global deformation field can be embedded locally.

A. Deformation Field

Before we can set up the embedding, we need to define the deformation field and the local coordinate charts. We encode the individual displacement of each mesh vertex \vec{x}_i into \vec{x}'_i as a local tangent vector \vec{u}_i . The actual displacement of the mesh node is on the great circle along this tangent vector, and the sine of the angle between \vec{x}_i and \vec{x}'_i is equal to the length of the tangent vector \vec{u}_i , as shown in figure 1(a). This choice, and the fact that all vertices reside on the unit sphere, provides a simple expression for \vec{u}_i using vector products [6]:

$$\vec{u}_i = -\vec{x}_i \wedge (\vec{x}_i \wedge \vec{x}'_i). \quad (1)$$

Conversely, the displaced vertex is given as follows:

$$\vec{x}'_i = \sqrt{1 - \|\vec{u}_i\|^2} \cdot \vec{x}_i + \vec{u}_i \quad (2)$$

B. Local Coordinates and Parallel Transport

Now, we define a local coordinate chart for each vertex. First, a pair of linearly independent tangent vectors need to be found to define the tangential plane at \vec{x}_i . The cross product between the surface normal and another, linearly

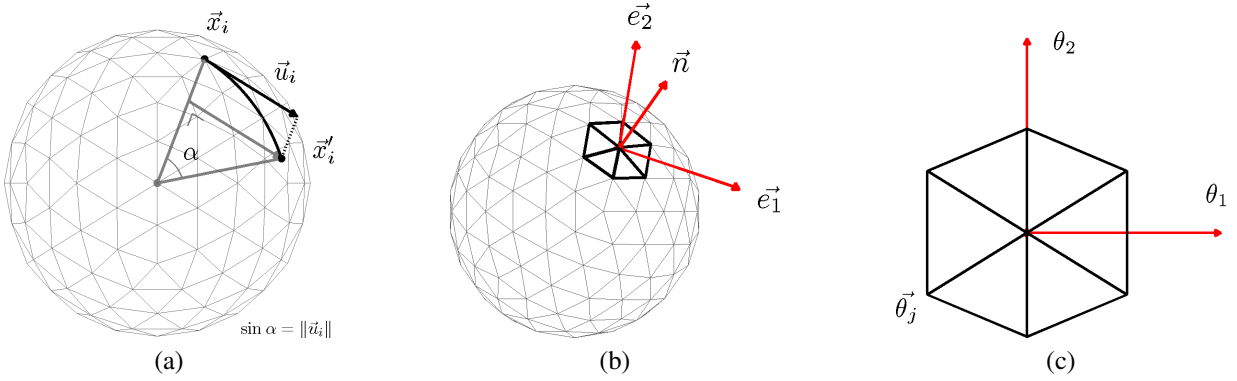


Fig. 1. (a) Deformation model. (b) One-ring patch on a spherical mesh and its local basis vectors. (c) Local coordinates of the one-ring neighbors within the tangential plane.

independent vector yields a first tangent, a subsequent cross product between the normal and this first tangent yields a second, orthogonal tangent:

$$\begin{aligned} \vec{e}_1 &= \frac{\vec{x}_i - \vec{x}_j}{\|\vec{x}_i - \vec{x}_j\|} \wedge \vec{n} \\ \vec{e}_2 &= \vec{n} \wedge \vec{e}_1, \end{aligned} \quad (3)$$

where \vec{x}_j is one of the 1-ring neighbors, $j \in N_1(i)$, as illustrated in figure 1(b). The projection of the coordinates of the neighbors into the newly defined local coordinate system of the tangent plane is then defined as

$$\vec{\theta}_j = (\theta_j^1, \theta_j^2) = (\langle \vec{x}_j - \vec{x}_i, \vec{e}_1 \rangle, \langle \vec{x}_j - \vec{x}_i, \vec{e}_2 \rangle). \quad (4)$$

Note that the node \vec{x}_i has coordinates $(0, 0)$ in its own local coordinate chart, see figure 1(c).

To work with the tangent vectors of neighboring vertices, we take them to \vec{x}_i using parallel transport. The tangent vector \vec{u}_j is parallelly transported along the great circle defined by \vec{x}_i and \vec{x}_j . Let the axis $\vec{\omega}$ of this great circle be

$$\vec{\omega} = \frac{\vec{x}_j \wedge \vec{x}_i}{\|\vec{x}_j \wedge \vec{x}_i\|}. \quad (5)$$

The great circle is a geodesic, and therefore parallel transport has two main properties: components perpendicular to the geodesic remain perpendicular, and components along it remain parallel. Therefore we have for the parallelly transported \vec{u}_j^i :

$$\vec{u}_j^i = \langle \vec{u}_j, \vec{\omega} \rangle \vec{\omega} + \langle \vec{u}_j, (\vec{\omega} \wedge \vec{x}_j) \rangle (\vec{\omega} \wedge \vec{x}_i). \quad (6)$$

Once the neighboring deformation vectors have been transported to the central vertex of the patch, they can be represented in the local coordinate chart as 2-dimensional vectors \vec{r} :

$$(u_j^i, v_j^i) = [\vec{e}_1 \vec{e}_2]^T \vec{u}_j^i. \quad (7)$$

C. Deformation Field Embedding and Flow

Using the local coordinates, the deformation field can be embedded locally around each vertex. To simplify the notation, the deformation field tangent vectors are written (u, v) . The embedding map $X : \Sigma \mapsto M$ is given by 4 functions of 2 variables. Further, a Riemannian structure is introduced: the

metric h_{ij} locally measures the distances on M , whereas on Σ distances are measured using the induced metric $g_{\mu\nu} = h_{ij} \partial_\mu X^i \partial_\nu X^j$. Thus:

$$\begin{cases} X : (\theta^1, \theta^2) \rightarrow (\theta^1, \theta^2, u, v) \\ h_{ij} = \text{diag}(1, 1, \beta^2, \beta^2) \\ g_{\mu\nu} = \begin{bmatrix} 1 + \beta^2(u_1^2 + v_1^2) & \beta^2(u_1 u_2 + v_1 v_2) \\ \beta^2(u_1 u_2 + v_1 v_2) & 1 + \beta^2(u_2^2 + v_2^2) \end{bmatrix} \\ g = 1 + \beta^2(|\nabla u|^2 + |\nabla v|^2) + \beta^4(\nabla u, \nabla v)^2, \end{cases} \quad (8)$$

where $(\nabla u, \nabla v) = u_x v_y - u_y v_x$ is defined as the magnitude of the cross product of the gradient vectors ∇u and ∇v and measures the misalignment of the gradients between different deformation field components [7]. All these settings, put into the general equations, produce the following energy functional and its minimizing flow:

$$\begin{cases} E_{GAF} = \int f \sqrt{g} d\theta^1 d\theta^2 \\ \partial_t u = f H^u + \partial_k f g^{\mu\nu} \partial_\mu X^k \partial_\nu u - \frac{4}{\beta^2} f u \\ \partial_t v = f H^v + \partial_k f g^{\mu\nu} \partial_\mu X^k \partial_\nu v - \frac{4}{\beta^2} f v. \end{cases} \quad (9)$$

In the following paragraphs, we will specify the weighting function f and its partial derivatives $\partial_k f$, as well as the method to estimate the curvature vector H , image gradients, point localisation and mesh interpolation.

D. Weighting Function

An intuitive primer for monomodal image registration is the squared error metric [8], leading to:

$$f(\vec{x}, \vec{x}') = 1 + \alpha \cdot (M(\vec{x}') - F(\vec{x}))^2, \quad (10)$$

where F and M refer to the fix and moving images, respectively. The 1 provides a lower bound of local weight in the Polyakov energy, and α allows to balance the impact of the data-term with respect to the regularization. The evolution equation includes the partial derivatives of the weighting function with respect to all components of the embedding. The spatial gradient is easy to be estimated numerically on the sampled mesh, whereas the derivatives with respect to the deformation are obtained using the chain rule:

$$\vec{f}_u = 2\alpha \cdot (M(\vec{x}') - F(\vec{x})) \cdot \frac{\partial \vec{x}'}{\partial \vec{u}} \cdot \vec{\nabla} M(\vec{x}'), \quad (11)$$

where $\vec{\nabla}M$ refers to the gradients of the fix and moving images, respectively. In local coordinates this writes $(f_u, f_v) = [\vec{e}_1 \vec{e}_2]^T \vec{f}_{\vec{u}}$. The second order tensor $\frac{\partial \vec{x}'}{\partial \vec{u}}$ is calculated as:

$$\frac{\partial \vec{x}'_b}{\partial \vec{u}_a} = \frac{\vec{x}_b \vec{u}_a}{\sqrt{1 - \|\vec{u}\|^2}} + \delta_{ab}, \quad (12)$$

where δ_{ab} refers to the Kronecker delta.

E. Curvature Vector Estimation

We use the Desbrun estimator to compute the mean curvature vector [9], [10]. It is of the following form:

$$\vec{H}_i = \sum_{j \in N_1(i)} w_{ij} (X_j - X_i), \quad (13)$$

where w_{ij} are the weights assigned to the edge connecting X_i to X_j . Several estimators of this class have been discussed in [11], [12]. The weighting proposed by Desbrun was derived using the first variation of area and reads as follows:

$$w_{ij} = \frac{\cot \alpha_j + \cot \beta_j}{4A}, \quad (14)$$

where α_j and β_j are the two angles opposite to the edge $X_i - X_j$ in the two triangles sharing this edge, and A is the sum of the areas of the triangles having X_i as a common vertex, i.e. the area of the one-ring patch. This estimator was already used with success in a similar context to build a Beltrami-flow scale-space on spherical meshes [10].

F. Gradient Estimation

For functions defined at the vertices of triangular meshes, the gradient can be fitted using least squares approximation [13]. As one is interested in the gradient *within* the tangent plane of the manifold, it is useful to enforce this by design, as happens when using local coordinate charts restricted to the tangential plane. Indeed, a first order Taylor expansion of some feature map c around vertex i , using the local coordinates $\vec{\theta}_i$ yields:

$$c(\vec{\theta}_j) = c_i + \langle \vec{\theta}_j, \vec{\nabla}c \rangle + \epsilon_j. \quad (15)$$

To simplify the notation, we define the residuals $y_j = c(\vec{\theta}_j) - c_i$ and rewrite equation (15) accordingly:

$$y_j = \langle \vec{\theta}_j, \vec{\nabla}c \rangle + \epsilon_j. \quad (16)$$

Now we look for an optimal $\vec{\nabla}c$, which minimizes $\vec{\epsilon}$ in the least square sense. We define the matrix $A_{kj} = [\vec{\theta}_j]_k$, where $k = 1, 2$ denotes the components of the local coordinates vector $\vec{\theta}_j$. Note that this matrix only depends on the geometry of the local patch. It can therefore be reused for the derivation of any number of features in that vertex. The minimization problem and its direct solution are thus given by

$$\widehat{\vec{\nabla}c} = \operatorname{argmin}(\|A\vec{\nabla}c - \vec{y}\|) = (A^T A)^{-1} A^T \vec{y}. \quad (17)$$

G. Point Location and Mesh Interpolation

The weighting function f is evaluated at each vertex of the fix image. Most of the time, this vertex, called the query point, will not coincide with a vertex of the warped moving image, and interpolation will be required.

To do so, one first has to find the face of the moving mesh, that contains the query point. Here, we use spherical triangle walk to locate the point [14]. Once the surrounding triangle has been found, the image value can be bilinearly interpolated from its three vertices using the barycentric coordinates of the query point.

III. RESULTS

The regularization of the GAF framework is characterized by its impulse response and denoising properties. The impulse response is obtained by disconnecting the data-term, i.e., by setting $alpha = 0$ and thus the weighting function to $f = 1$. The diffusion of an initial single peak is illustrated in figure 2(a)-(b). Regularization is largely independent of the sampling density, as the field looks almost the same after two runs on a fine and a coarse sphere mesh. Denoising for two different β is illustrated in figure 2(c)-(d). Smaller β produces Gaussian-like diffusion, whereas higher β corresponds to feature-preserving TV-norm minimization [4].

To illustrate the complete registration framework, a pair of synthetic images has been registered as shown in figure 3. Registration is successful, as the resampled moving image $M(\vec{x}')$ nicely matches the fixed image $F(\vec{x})$, under reasonable deformation $\vec{x} \rightarrow \vec{x}'$. Over the iterations, the decrease in the data-term (red) leads to a increased hyperarea of the embedding map (black), but the global energy still decreases monotonically (blue).

IV. CONCLUSIONS AND OUTLOOK

In this paper, we have introduced the geodesic active fields framework for image registration to applications, where images are defined on spherical meshes. We have shown that the proposed method works as expected under different aspects. The regularizer has the desired impulse response irrespective of the image sampling density, the anisotropy of the diffusion is tunable by the parameter β , and the framework performs correctly on a set of synthetic test images. Note that, in contrast to planar image parameterizations, where all constant-valued deformation fields are stable, the boundary-free mesh of the sphere allows only the trivial solution as steady state in absence of a data-term.

We do not claim to outperform highly specialized state-of-the-art methods, but like to emphasize the methodological aspects of our framework, in particular geodesy and sampling independence. Further research will focus on the application of this novel method to actual cases of spherical image registration, e.g. on cortical feature maps. In this context, the intrinsic capability of the GAF framework, to generalize to multi-scale images, will be of particular value.

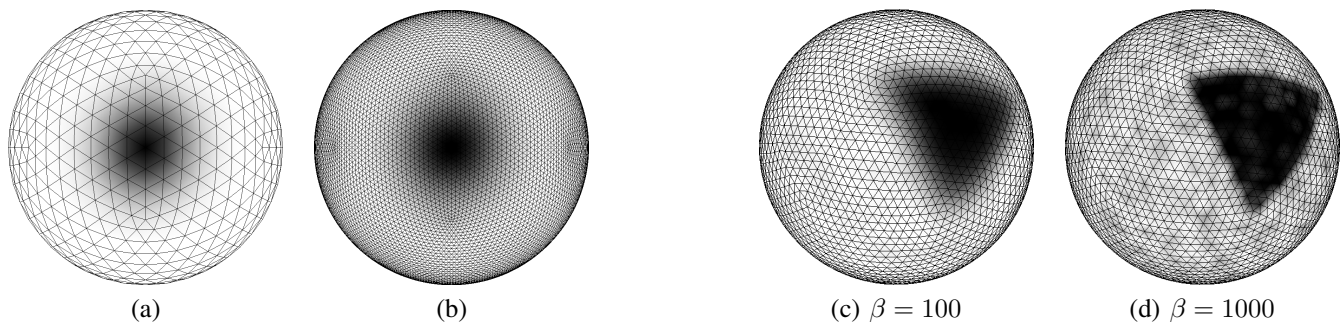


Fig. 2. (a)-(b) Impulse response on a coarse and fine sphere mesh. (c)-(d) Denoising.

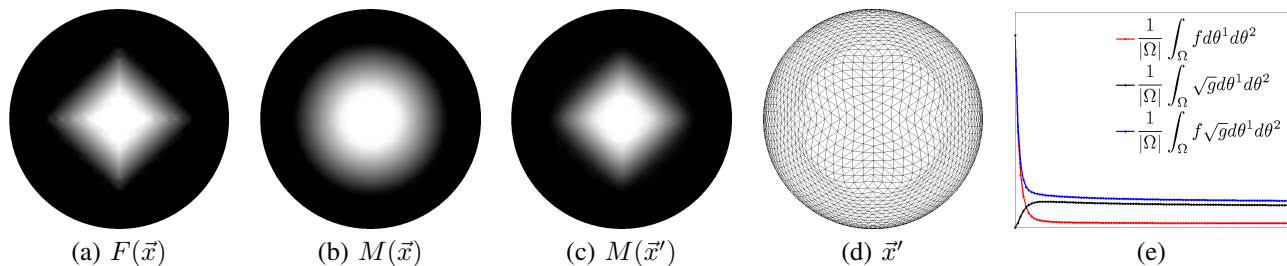


Fig. 3. Registration of a synthetic image pair. (a)-(b) Fix and moving image. (c) Warped moving image after registration. (d) Deformed mesh. (e) Evolution of complete GAF energy (blue), data term only (red), and regularizer only (black).

ACKNOWLEDGMENT

This work was supported by the National Competence Centre in Biomedical Imaging (NCCBI).

REFERENCES

- [1] M. Hebert, K. Ikeuchi, and H. Delingette, "A spherical representation for recognition of free-form surfaces," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, no. 7, pp. 681–690, Jul. 1995.
- [2] B. Fischl, M. I. Sereno, and A. M. Dale, "Cortical surface-based analysis II: Inflation, flattening, and a surface-based coordinate system," *Neuroimage*, vol. 9, no. 2, pp. 195–207, Feb. 1999.
- [3] Z. Arican and P. Frossard, "Dense disparity estimation from omnidirectional images," in *AVSS 2007*, Sep. 2007, pp. 399–404.
- [4] D. Zosso, X. Bresson, and J.-P. Thiran, "Geodesic active fields - a geometric framework for image registration," Ecole Polytechnique Fédérale de Lausanne (EPFL), Tech. Rep. LTS-REPORT-2010-001, 2010.
- [5] N. Sochen, R. Kimmel, and R. Malladi, "A general framework for low level vision," *IEEE Trans. Image Process.*, vol. 7, no. 3, pp. 310–318, Mar. 1998.
- [6] B. Yeo, M. Sabuncu, T. Vercauteren, N. Ayache, B. Fischl, and P. Goland, "Spherical demons: Fast surface registration," in *MICCAI 2008*, 2008, pp. 745–753.
- [7] R. Kimmel, R. Malladi, and N. Sochen, "Images as embedded maps and minimal surfaces: Movies, color, texture, and volumetric medical images," *Int. J. Comput. Vis.*, vol. 39, no. 2, pp. 111–129, 2000.
- [8] A. W. Toga, *Brain Warping*. Academic Press, 1999.
- [9] M. Desbrun, M. Meyer, P. Schröder, and A. H. Barr, "Implicit fairing of irregular meshes using diffusion and curvature flow," in *Proc. ACM SIGGRAPH '99*, 1999, pp. 317–324.
- [10] D. Zosso and J.-P. Thiran, "A scale-space of cortical feature maps," *IEEE Signal Process. Lett.*, vol. 16, no. 10, pp. 873–876, Oct. 2009.
- [11] G. Xu, "Convergent discrete laplace-beltrami operators over triangular surfaces," in *Proceedings of the Geometric Modeling and Processing 2004 (GMP'04)*, 2004.
- [12] —, "Discrete Laplace-Beltrami operator on sphere and optimal spherical triangulations," *International Journal of Computational Geometry and Applications*, vol. 16, no. 1, pp. 75–93, 2006.
- [13] K. Frank and U. Lang, "Gradient and curvature approximation in data-dependent surface simplification," *Comput. Visual Sci.*, vol. 2, pp. 221–228, 2000.
- [14] R. Sundareswara and P. R. Schrater, "Extensible point location algorithm," in *GMAG*, 2003, pp. 84–89.