N-DIMENSIONAL SURFACE RECONSTRUCTION FROM A NOISY NORMAL VECTOR FIELD

Guillaume Zinck, Marc Donias and Olivier Lavialle

Université de Bordeaux, IPB, Laboratoire IMS CNRS UMR 5218,
351 cours de la Libération, 33405 Talence cedex, France.
{guillaume.zinck, marc.donias, olivier.lavialle}@ims-bordeaux.fr

ABSTRACT

We present a method to reconstruct an implicit hypersurface of a N-dimensional vector space from a normal vector field supposed to be unreliable and noisy. Either the surface boundary or a point belonging to the surface is required. Assuming that a basis is known in which the surface is explicit, our approach consists in an accurate and noise robust global optimization technique based on a non linear partial derivative equation relied on local dip. The key point is the expression of the local dip in the new basis.

Index Terms— Surface reconstruction, partial derivative equation, Poisson equation, local dip transformation, normal vector field.

1. INTRODUCTION

This paper treats of a method dedicated to the reconstruction of a N-dimensional vector space hypersurface. Either the surface boundary or a point belonging to the surface is required. We deal with a known dense vector field of normals orthogonal to the surface, usually estimated in an orthonormal basis over the entire N-dimensional data and supposed to be unreliable and noisy (see figure 1). Application scopes and normal vector field achievement methods are various. In a two-dimensional (2-D) space, fingerprints can be reconstructed from a set of known minutiae points [1]. In this case, the normal vector field stems from the extrapolation of orientation vectors estimated at the minutiae points. In meteorology, the method can be applied to sea surface temperature images and satellite images to track eddies [2] and cyclones [3] or to characterize their location more precisely than by an ellipse [4] through a normal vector field resulting from a velocity field. In a three-dimensional (3-D) space, seismic horizons can be computed from a normal vector field estimated by a gradient field principal component analysis [5].

Methods to reconstruct a surface from a normal vector field are developed in literature. If a point belonging to the surface is known, the more intuitive reconstruction way is to track the normal vector field. In the 2-D case, a unique surface can be computed by integrating the tangent vector obtained from the normal one. This method is carried out in [1], combined with line integral convolution. If N is higher than two, an infinite number of surfaces can be computed from the (N-1)-dimensional tangent hyperplane. Nevertheless, the surfaces derive in case of noisy data. Active contour models, also called snakes, are an other framework to delineate a surface [6]. A model is initialized by a set of points which converge to the surface by an energy minimization. No constraint such as a point belonging to the surface or boundary conditions are imposed in standard algorithms and noise robustness of the method is extremely weak. In case of noisy data, level set methods [7] can be efficient, but are complex to implement and are limited to closed surfaces.

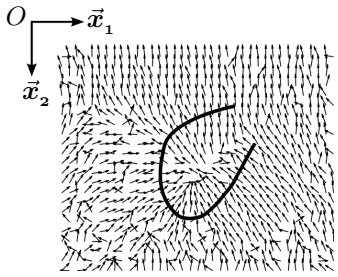


Fig. 1: A two-dimensional estimated normal vector field and an example of an implicit hypersurface in basis $(\vec{x}_1, \vec{x}_2)$.

In this paper, we present a global approach to reconstruct an implicit hypersurface of a N-dimensional vector space. We consider the local dip resulting from the normal vector field. Assuming that a basis is known in which the surface is explicit, the surface is obtained by solving a (N-1)-dimensional non linear partial derivative equation (PDE) relied on the local dip expressed in the new basis. The PDE approach was first introduced in a 3-D space by Lomask *et al.* [5]. A Gauss-Newton method is carried out by an iterative algorithm including the resolution of a Poisson equation.

This article is organized as follows: section 2 introduces the explicit surface reconstruction algorithm, section 3 deals with implicit surface reconstruction while the two last sections exhibit results on synthetic and real data in two and three-dimensional spaces.

2. EXPLICIT SURFACE RECONSTRUCTION

In a N-dimensional vector space V_N associated to an orthonormal basis $B = (\vec{\boldsymbol{x}}_1, ..., \vec{\boldsymbol{x}}_N)$ with center O, let a hypersurface be represented by an explicit function f defined on a domain Ω by $x_N = f(\mathrm{x})$, where $\mathrm{x} = (x_1, ..., x_{N-1})$. The function f is connected to the tangent $\mathbf{p}$ of the local dip in basis B by a PDE:

$$\forall\, \mathrm{x} \in \Omega, \quad \nabla f(\mathrm{x}) = \mathbf{p}\big(\mathrm{x}, f(\mathrm{x})\big), \tag{1}$$

where ∇ denotes the gradient operator [5]. The local dip in basis B gives the slope of the surface tangent hyperplane $\mathcal{P}$ compared to the N-1 first vectors of B. For a point X of V_N, the components p_i of $\mathbf{p}$ are given by:

$$\forall\, i \in [\![1, N-1]\!], \quad p_i(\mathrm{X}) = -\frac{n_i(\mathrm{X})}{n_N(\mathrm{X})}, \tag{2}$$

where n_i is the ith component of the normal vector $\mathbf{n}$ (see figure 2). The functions f and $\mathbf{p}$ are respectively considered of class C^2 and C^1.

The surface is obtained by solving a constrained optimization problem:

$$f = \arg\min_{g \in C^2} \int_\Omega \big\|\nabla g(\mathrm{x}) - \mathbf{p}\big(\mathrm{x}, g(\mathrm{x})\big)\big\|^2 \mathrm{dx}. \tag{3}$$

assuming that either the surface boundary or a point belonging to the surface is known. Equation (3) is non linear because $\mathbf{p}$ depends on f, thus an iterative algorithm is used to solve it [5]. The surface is initialized with a function f_0 and the iterative step is made of three parts : residual computation, update term computation and updating.

- *Residual computation:*

$$\forall\, \mathrm{x} \in \Omega, \quad \mathbf{r}_k(\mathrm{x}) = \nabla f_k(\mathrm{x}) - \mathbf{p}\big(\mathrm{x}, f_k(\mathrm{x})\big), \tag{4}$$

where k denotes the iteration number.

- *Update term computation:*

$$\delta f_k = \arg\min_{g \in C^2} \int_\Omega \big\|\nabla g(\mathrm{x}) + \mathbf{r}_k(\mathrm{x})\big\|^2 \mathrm{dx}. \tag{5}$$

The solution of equation (5) is obtained by solving a Poisson equation:

$$\Delta(\delta f_k) = -\operatorname{div}(\mathbf{r}_k), \tag{6}$$

where Δ denotes the Laplace operator and div is the divergence vector operator.

If the surface boundary is known, the Poisson equation is associated with boundary conditions:

$$\begin{array}{rrcl} \forall\, \mathrm{x} \in \partial\Omega, & \delta f_0(\mathrm{x}) & = & f(\mathrm{x}) - f_0(\mathrm{x}) \\ \text{and} & \delta f_k(\mathrm{x}) & = & 0 \quad \forall\, k > 0, \end{array} \tag{7}$$

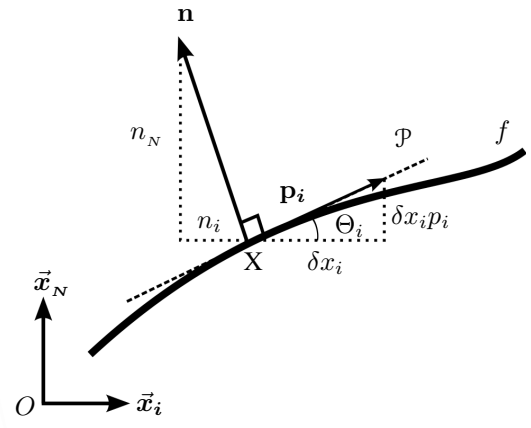


Fig. 2: Intersection of the surface f with the plane $(\vec{\boldsymbol{x}}_i, \vec{\boldsymbol{x}}_N)$. For a point X, the component p_i of the tangent $\mathbf{p}$ of the local dip is given by : $p_i(\mathrm{X}) = \tan\Theta_i(\mathrm{X}) = -\frac{n_i(\mathrm{X})}{n_N(\mathrm{X})}$.

where $\partial\Omega$ denotes the boundary of the domain Ω.

If a point P belonging to the surface is known, the Poisson equation is associated with an "inner" condition:

$$\begin{array}{rrcl} & \delta f_0(\mathrm{x}^P) & = & f(\mathrm{x}^P) - f_0(\mathrm{x}^P) \\ \text{and} & \delta f_k(\mathrm{x}^P) & = & 0 \quad \forall\, k > 0, \end{array} \tag{8}$$

where $\big(\mathrm{x}^P, f(\mathrm{x}^P)\big)$ are the coordinates of P.

- *Updating:*

$$\forall\, \mathrm{x} \in \Omega, \quad f_{k+1}(\mathrm{x}) = f_k(\mathrm{x}) + \delta f_k(\mathrm{x}). \tag{9}$$

Convergence is assumed to be reached after a number K of iterations.

3. IMPLICIT SURFACE RECONSTRUCTION

In this section, we deal with a hypersurface represented by an implicit function f defined by $f(\mathrm{x}, x_N) = 0$ in basis B. To reconstruct such a surface by a PDE approach, we assume that there is an orthonormal basis $B_y = (\vec{\boldsymbol{y}}_1, ..., \vec{\boldsymbol{y}}_N)$ with center O_y in which the surface can be represented by an explicit function f_y defined on a domain Ω_y. By adding a subscript x to the elements of the implicit basis B, each point of V_N is defined in basis B_x and B_y respectively by its coordinates $\mathrm{X} = (x_1, ..., x_N)$ and $\mathrm{Y} = (y_1, ..., y_N)$. The change of basis from B_x to B_y is defined by a bijective N-dimensional linear transformation $\mathcal{F}$ characterized by its invertible Jacobian matrix $J_\mathcal{F}$ whose inverse at a point Y is given by:

$${J_\mathcal{F}}^{-1}(\mathrm{Y}) = \begin{bmatrix} \frac{\partial x_1}{\partial y_1}(\mathrm{Y}) & \cdots & \frac{\partial x_1}{\partial y_N}(\mathrm{Y}) \\ \vdots & \ddots & \vdots \\ \frac{\partial x_N}{\partial y_1}(\mathrm{Y}) & \cdots & \frac{\partial x_N}{\partial y_N}(\mathrm{Y}) \end{bmatrix}. \tag{10}$$

In basis B_y, the surface is obtained by solving a PDE that connects f_y to the tangent $\mathbf{p_y}$ of the local dip:

$$\forall\, \mathrm{y} \in \Omega_y, \quad \nabla f_y(\mathrm{y}) = \mathbf{p_y}\big(\mathrm{y}, f_y(\mathrm{y})\big), \tag{11}$$

where $\mathrm{y} = (y_1, ..., y_{N-1})$. Equation (11) is solved by the algorithm presented in section 2. The local dip in basis B_y gives the slope of the surface tangent hyperplane $\mathcal{P}_y$ compared to the N-1 first vectors of B_y. For a point Y, the components $p_{y,i}$ of $\mathbf{p_y}$ are relied to the surface normal vector $\mathbf{n}$ and to the inverse transformation $\mathcal{F}^{-1}$:

$$\forall\, i \in [\![1, N-1]\!],$$

$$p_{y,i}(\mathrm{Y}) = -\frac{\sum_{j=1}^{N} \frac{\partial x_j}{\partial y_i}(\mathrm{Y}).n_j\big(\mathcal{F}^{-1}(\mathrm{Y})\big)}{\sum_{j=1}^{N} \frac{\partial x_j}{\partial y_N}(\mathrm{Y}).n_j\big(\mathcal{F}^{-1}(\mathrm{Y})\big)}. \tag{12}$$

Proof. Let us consider a point Y of V_N and the vector $\mathbf{p_{y,i}}$ which belongs to the intersection between the surface tangent hyperplane $\mathcal{P}_y$ and the plane $(\vec{\boldsymbol{y}}_{\boldsymbol{i}}, \vec{\boldsymbol{y}}_{\boldsymbol{N}})$ (see figure 2 by replacing basis B by basis B_y). Obviously, the inverse transformation of this vector $\mathcal{F}^{-1}(\mathbf{p_{y,i}})$ belongs to the surface tangent hyperplane $\mathcal{P}_x$. Knowing the surface normal vector $\mathbf{n}$, the value $p_{y,i}$ is therefore defined by the vector $\mathcal{F}^{-1}(\mathbf{p_{y,i}})$ which is orthogonal to $\mathbf{n}$, i.e. the vector for which the dot product between itself and $\mathbf{n}$ is zero.

For a displacement δy_i along $\vec{\boldsymbol{y}}_{\boldsymbol{i}}$, the vector $\mathbf{p_{y,i}}$ and its inverse transformation are respectively expressed in basis B_y and B_x by:

$$\mathbf{p_{y,i}} = \delta y_i \vec{\boldsymbol{y}}_{\boldsymbol{i}} + \delta y_i p_{y,i} \vec{\boldsymbol{y}}_{\boldsymbol{N}} \tag{13}$$

$$\mathcal{F}^{-1}(\mathbf{p_{y,i}}) = \sum_{j=1}^{N} \delta x_j \vec{\boldsymbol{x}}_{\boldsymbol{j}}. \tag{14}$$

The components of $\mathcal{F}^{-1}(\mathbf{p_{y,i}})$ are given by a linear approximation of the function $\mathcal{F}^{-1}$:

$$\begin{aligned} \delta x_j &\approx {\mathcal{F}_j}^{-1}(\mathrm{Y} + \mathbf{p_{y,i}}) - {\mathcal{F}_j}^{-1}(\mathrm{Y}) \\ &\approx \delta y_i \frac{\partial x_j}{\partial y_i}(\mathrm{Y}) + \delta y_i p_{y,i} \frac{\partial x_j}{\partial y_N}(\mathrm{Y}), \end{aligned} \tag{15}$$

where ${\mathcal{F}_j}^{-1}$ is the jth component of $\mathcal{F}^{-1}$.

Taking the dot product between $\mathcal{F}^{-1}(\mathbf{p_{y,i}})$ and $\mathbf{n}$ equal to zero gives then:

$$\sum_{j=1}^{N} \left(\frac{\partial x_j}{\partial y_i}(\mathrm{Y}) + p_{y,i} \frac{\partial x_j}{\partial y_N}(\mathrm{Y}) \right) n_j\left(\mathcal{F}^{-1}(\mathrm{Y})\right) = 0, \tag{16}$$

which corresponds to equation (12). □

4. TWO-DIMENSIONAL APPLICATIONS

4.1. Synthetic data

The approach is firstly evaluated on a 256×256 synthetic 2-D image depicting a positive logarithmic spiral (see figure 3). The surface to be reconstructed is an isoline passing through a point P assumed to be known. In a polar basis whose center corresponds to the spiral center O_y, the surface equation is:

$$\rho_e(\theta) = \rho^P \mathrm{e}^{b(\theta - \theta^P)}, \tag{17}$$

where ρ and θ denote respectively the positive radial coordinate and the angular coordinate while (θ^P, ρ^P) are the coordinates of P, e is the exponential function and b is a positive constant. The image is corrupted by an additive Gaussian noise with zero mean to reach a signal-to-noise ratio of 25 dB before the normal vector field is estimated by a standard gradient field principal component analysis (PCA). Several basis B_y in which the surface is explicit can be chosen for the reconstruction, like the polar basis, the positive logarithmic spiral basis or the positive arithmetic spiral basis with center O_y carried out in the example presented below. An arithmetic spiral is given in a polar basis by a linear function defined by a positive slope a:

$$\rho_s(\theta) = a\theta, \tag{18}$$

where θ is positive. In the arithmetic spiral basis, each point is determined by an angle θ and a value a which corresponds to the slope of the linear function passing through the point. The coordinates (θ, a) are relied on the Cartesian coordinates (x_1, x_2) by:

$$\left\{ \begin{array}{rcl} \theta &=& \arctan(x_1 - x_1^O, x_2 - x_2^O) + 2\pi L_1 \\ a &=& \frac{\sqrt{(x_1 - x_1^O)^2 + (x_2 - x_2^O)^2}}{\theta} \end{array} \right., \tag{19}$$

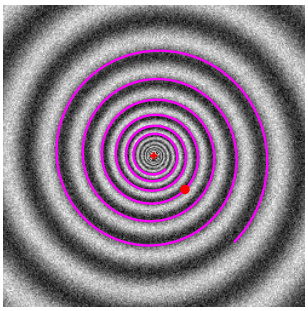
where (x_1^O, x_2^O) are the Cartesian coordinates of O_y and L_1 is a positive constant. The function $\arctan(x_1, x_2)$ gives the arc tangent of $\frac{x_2}{x_1}$ taking into account which quadrant the point (x_1, x_2) is in while the value of L_1 has to be chosen such as the domain Ω_y is included in $[0, +\infty[$. The inverse Jacobian matrix of the Cartesian to arithmetic spiral basis transformation at point (θ, a) is:

$$\left[\begin{array}{cc} a(\cos\theta - \theta\sin\theta) & \theta\cos\theta \\ a(\sin\theta + \theta\cos\theta) & \theta\sin\theta \end{array} \right]. \tag{20}$$

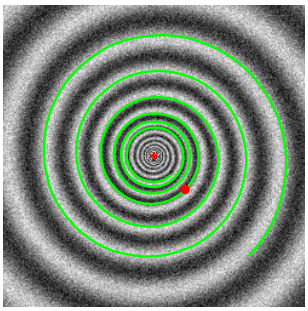
In the arithmetic spiral basis, the mathematic model of the surface passing through P whose coordinates are (θ^P, a^P) is given by:

$$f_{y,th}(\theta) = a^P \frac{\theta^P}{\theta} \mathrm{e}^{b(\theta - \theta^P)}. \tag{21}$$

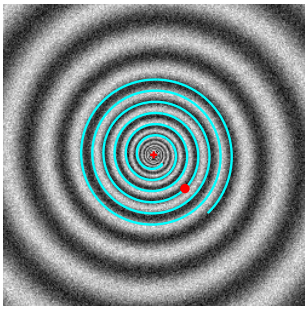
In the example shown in figure 3, the domain Ω_y is an interval whose center is θ^P and whose length is $2\pi L_2$ with L_2 = 6 while L_1 = 4 and b = 0.05. The reconstruction algorithm is initialized by a constant function $f_{y,0}$ equal to a^P,



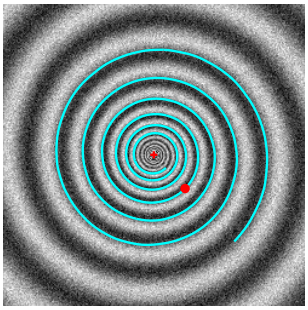
(a) Model $\mathcal{F}^{-1}(f_{y,th})$



(b) Integration $\mathcal{F}^{-1}(f_{y,int})$



(c) Initialization $\mathcal{F}^{-1}(f_{y,0})$



(d) Surface $\mathcal{F}^{-1}(f_y)$

Fig. 3: Surface reconstruction from a known point (represented by a red disk) on a 2-D synthetic image. The red cross corresponds to the arithmetic spiral basis center O_y.

represented in the Cartesian basis by figure 3c. Thirty iterations are required to observe convergence with 2,001 discrete points. The reconstructed surface f_y (see figure 3d) has to be compared with the mathematic model $f_{y,th}$ (see figure 3a) and the surface obtained by the orientation vector field integration $f_{y,int}$ (see figure 3b).

In the arithmetic spiral basis, the mathematic model and the reconstructed surface are almost superimposed while the orientation vector field integration diverges when moving away from P in both directions (see figure 4). In the Cartesian basis, the Euclidean distance $D(f_y)$ computed between the mathematic model and the reconstructed surface for each value of θ is extremely weak (< 0.5 pixel, see figure 5), which proves precision and noise robustness of our method. On the contrary, the distance $D(f_{y,int})$ between the mathematic model and the orientation vector field integration increases extremely quickly when moving away from P.

4.2. Real cyclone data

The approach is secondly carried out on a real cyclone satellite image. A cyclone is an area of closed circular fluid motion rotating in the same direction as the Earth and appearing in low-pressure area. In climate change prediction, recent works have been dedicated to cyclone detection and tracking [3] [4]. Here, we propose to reconstruct a surface corresponding to the cyclone location from an estimation of the cyclone center

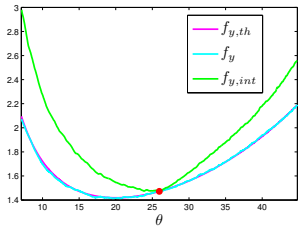


Fig. 4: Surface comparisons in the arithmetic spiral basis.

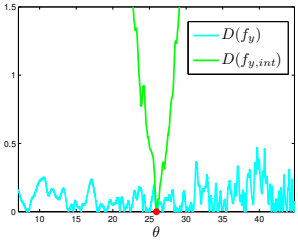


Fig. 5: Euclidean distances in the Cartesian basis.

O_y and the knowledge of the two boundary points P_1 and P_2. As previously, the normal vector field is estimated by a PCA. The cyclone represented in figure 6 shows an approximately logarithmic spiral pattern, so a positive arithmetic spiral basis with center O_y can be used in the reconstruction method. Denoting (θ^{P_i}, a^{P_i}) the coordinates of the point P_i, the surface domain is delineated by θ^{P_1} and θ^{P_2} and its length is taken between $2\pi L_2$ and $2\pi(L_2+1)$, where L_2 is a positive constant. The algorithm is initialized by a constant function $f_{y,0}$ equal to the mean value of a^{P_1} and a^{P_2}. By fixing $L_1 = 4$ and $L_2 = 2$, thirty iterations are required to observe convergence with 1,001 discrete points.

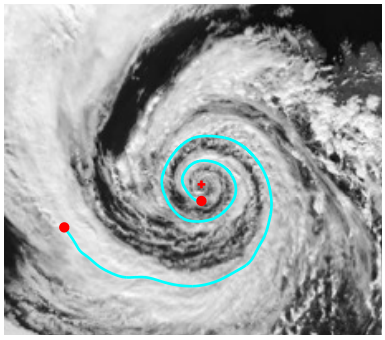


Fig. 6: Surface reconstruction from the boundary points knowledge (represented by the red disks) on a 2-D cyclone data. The red cross corresponds to the arithmetic spiral basis center O_y.

The reconstructed surface depicted in figure 6 corresponds to the observed cyclone pattern. The integration of the orientation vector field method can not take into account two points and is obviously not suitable. Furthermore, the integration starting from P_1 or P_2 leads to innacurate results.

5. THREE-DIMENSIONAL APPLICATION

The approach is also tested on a $128\times128\times128$ synthetic 3-D image depicting a tri-axial ellipsoid (see figure 7). The surface to be reconstructed is an entire isoline $\mathcal{E}$ or a part of it passing through a point P assumed to be known. The image is corrupted as described in section 4.1. An intuitive basis B_y in which the surface is explicit is the spherical basis whose

center O_y has to be chosen inside $\mathcal{E}$. Each point is determined in B_y by two angles θ and ϕ included respectively in $[-\pi, \pi]$ and $[0, \pi]$ and by a positive value ρ relied on the Cartesian coordinates (x_1, x_2, x_3) by:

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = R. \begin{bmatrix} x_1 - x_1^O \\ x_2 - x_2^O \\ x_3 - x_3^O \end{bmatrix}$$

$$\text{and} \begin{cases} \theta &= \arctan(y_1, y_2) \\ \phi &= \arccos(\frac{y_3}{\sqrt{y_1^2+y_2^2+y_3^2}}) \\ \rho &= \sqrt{y_1^2 + y_2^2 + y_3^2} \end{cases}, \quad (22)$$

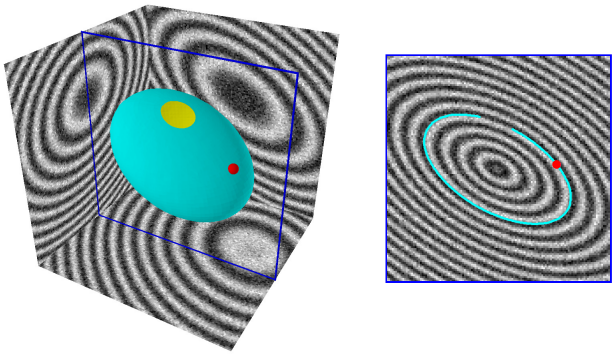
where (x_1^O, x_2^O, x_3^O) are the Cartesian coordinates of O_y and R is a 3×3 matrix characterizing a 3-D rotation of the Cartesian basis. The inverse Jacobian matrix of the Cartesian to B_y basis transformation at point (θ, ϕ, ρ) is then:

$$R^{-1}. \begin{bmatrix} -\rho \sin\theta \sin\phi & \rho \cos\theta \cos\phi & \cos\theta \sin\phi \\ \rho \cos\theta \sin\phi & \rho \sin\theta \cos\phi & \sin\theta \sin\phi \\ 0 & -\rho \sin\phi & \cos\phi \end{bmatrix}, \quad (23)$$

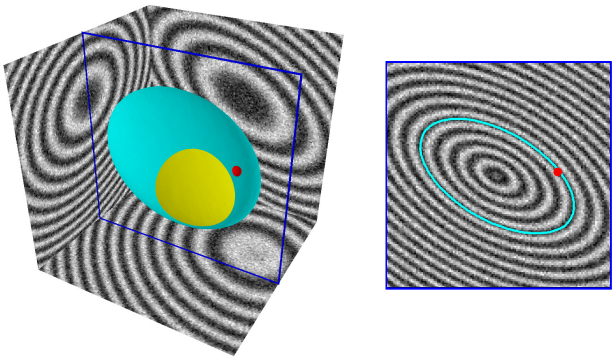
where R^{-1} is the inverse matrix of R. In the example presented below, we reconstruct a part of $\mathcal{E}$ defined on a domain Ω' corresponding to the rectangle $[-\pi, \pi] \times [0, \pi - \phi_0]$, where ϕ_0 takes its value in $]0, \pi[$. By choosing O_y as the center of the ellipsoid, the location of the non-reconstructed part of $\mathcal{E}$ depends exclusively on the rotation R. The reconstruction algorithm is initialized by a constant function $f_{y,0}$ passing through P, which corresponds to a part of a sphere in the Cartesian basis. Ten iterations are required to observe convergence with 101 discrete points for θ and 51 discrete points for ϕ. The reconstructed surfaces for two different rotations and values of ϕ_0 are depicted in figure 7. They are almost superimposed with the mathematic model. In the Cartesian basis, the Euclidean distance computed between each reconstructed surface and the mathematic model for each value of pair (θ, ϕ) is extremely weak (< 0.3 pixel).

6. CONCLUSION

We have developed a method to reconstruct an implicit hypersurface of a N-dimensional vector space from a normal vector field generally supposed to be unreliable and noisy. Either the surface boundary or a point belonging to the surface is required. Our approach consists in a global optimization technique whose key point is the expression of the local dip in a basis in which the surface is explicit. The obtained results on both synthetic and real data in 2-D and 3-D prove accuracy and noise robustness of the method, unlike to those obtained by an orientation vector field integration method.



(a) Rotation R_1, $\phi_0 = \frac{\pi}{9}$.



(b) Rotation R_2, $\phi_0 = \frac{\pi}{5}$.

Fig. 7: Surface reconstruction from a known point (represented by a red sphere) on a 3-D synthetic image for two different rotations R and values of ϕ_0. The 2-D images on the right show the cross-sections depicted by the blue frames in the 3-D images.

7. REFERENCES

[1] A. Ross, J. Shah, and A.K. Jain, "From Template to Image: Reconstructing Fingerprints From Minutiae Points," *IEEE Transactions on PAMI*, vol. 29, pp. 544–560, April 2007.

[2] V. Sood, B. John, R. Balasubramanian, and A. Tandon, "Segmentation and tracking of mesoscale eddies in numeric ocean models," in *International Conference on Image Processing (ICIP)*. IEEE, 2005, vol. 3, pp. 469–472.

[3] S-S. Ho and A. Talukder, "Automated Cyclone Identification From Remote QuikSCAT Satellite Data," in *Aerospace Conference*. IEEE, 2008, pp. 1–9.

[4] A.M. Fernandes, "Study on the Automatic Recognition of Oceanic Eddies in Satellite Images by Ellipse Center Detection - The Iberian Coast Case," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, pp. 2478–2491, August 2009.

[5] J. Lomask, A. Guitton, S. Fomel, J. Claerbout, and A. Valenciano, "Flattening without picking," *Geophysics*, vol. 71, pp. 13–20, July-August 2006.

[6] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision*, vol. 1, pp. 321–331, January 1988.

[7] S. Osher and J.A. Sethian, "Fronts Propagating with Curvature Dependent speed: Algorithms Based on Hamilton-Jacobi Formulation," *Journal of Computational Physics*, vol. 79, pp. 12–49, December 1988.