Proceedings of the 28th IEEE
EMBS Annual International Conference
New York City, USA, Aug 30-Sept 3, 2006

Optical flow computation and visualization in spherical context. Application on 3D+t bio-cellular sequences.

Wafa Rekik, Dominique Béréziat and Séverine Dubuisson
Université Pierre et Marie Curie, Laboratoire d'informatique de Paris 6
8, rue du Capitaine Scott,75015 Paris, France.
Wafa.Rekik@lip6.fr

Abstract— **The main focus of this paper is the optical flow computation for 2D movements of objects embedded on 3D surfaces. For spheric-shaped supports, object motion has only two degrees of liberty, thus the 3D optical flow constraint is not relevant. Constancy assumption is formulated using a suitable parametrization of the 3D surface, leading to a 2D equation. Input temporal sequence is also transformed according to the 3D surface parametrization. We build a complete 2D model, taking into account the underlying spherical surface. It has the merit to estimate at a lower cost velocity field in the temporal input sequence. In order to analyze motion computation results, we design an adapted visualization tool, instead of carrying out an inverse transformation for the velocity field. Adapted to the selected parametrization, it displays rapidly moving objects and velocity field and improves the understanding of the displayed information. We display optical flow computation results for 3D+t cell wall simulation sequences.**

I. INTRODUCTION

Motion computation is of growing interest in various computer vision applications. A fundamental widely investigated issue is the computation of optical flow. It aims at providing an approximation of the local image motion in a given sequence of images. We are mainly interested in estimating the velocity vector field of objects embedded on spheroid surfaces. We have therefore, a $3\text{D} + t$ structure of data organized on a set of 3D frames at each sample of the acquisition temporal process. Each 3D image is a ball represented by a given number of equal-sized plans. Since inside objects are meaningless, movement of structures evolving on the surface has only two degrees of liberty. As we explain in the next section, 3D optical flow constraint is not relevant for 2D object movement. It is then useful to investigate motion on the surface domain that profit from a 2D parametrization. We switch then to a system of spherical coordinates, with strong resemblance to the geographical one. We evaluate angular displacements, as radial temporal variations are negligible. The main advantage of this way of proceeding is to decrease the dimension of the input data set. We process then $2\text{D}+t$ instead of $3\text{D}+t$ sequences. For this sake, we need to map the 3D cartesian grid into an angular one. Due to angular distortion, a robust interpolation method is needed to recover missing points, looking like furrows. After the angular temporal variation estimation, it is possible to come back to the 3D structure of data. However, we are interested in visualizing, both texture and vector fields in the same reference. Since the geometry of the surface is not relevant, standard volume visualization methods do not seem to be well adapted to our context of study. We developed a visualization tool based on a suitable map projection, called **Mapvis**, that maps the data set sampled on a 2D angular grid into a cartographic 2D reference. **Mapvis** has hence the merit to provide relevant visualization properties for scalar and vectorial information for 3D surfaces.

II. OPTICAL FLOW COMPUTATION

The initial hypothesis in measuring image motion is that the intensity structures of local time-varying image regions are approximately constant under motion for a short duration. The optical flow constraint, introduced by Horn and Shunck [1], expresses the assumption that a moving pixel keeps the same gray value over time. Formally, if $I(x, y, t)$ is the image intensity function, then:

$$I(x, y, t) = I(x + \delta x, y + \delta y, t + \delta t),$$

where δx and δy are the displacements of the local image region at (x, y, t) after time δt. Expanding the left-hand side of this equation in a first order Taylor series leads to the well known Optical Flow Constraint:

$$\overrightarrow{\nabla} I.\overrightarrow{\omega} + I_t = 0, \tag{1}$$

where $\overrightarrow{\nabla I} = (I_x, I_y)$ is the spatial intensity gradient and $\overrightarrow{\omega} = (\frac{\partial x}{\partial t}, \frac{\partial y}{\partial t})$ is the velocity vector. Many techniques were developed to solve equation 1. We distinguish mainly intensity-based differential methods, correlation-based methods, frequency-based ones, multiple motion and temporal refinement ones. Reader is referred to [2] for details. Differential class of methods includes mainly local and global approaches. Local methods use normal velocity information in local neighborhoods to perform a least square minimization in order to find out the best fit for $\overrightarrow{w}$. That was first described by Lucas and Kanade [3]. Global methods, for instance [1], use an additional global constraint, usually a smoothness regularization term, to compute dense optical flow over large image regions. Computation in the 3D domain is most frequently performed by extension of the last two techniques to a higher dimension. However, this solution is not adapted to our context of study. Each 3D image is an empty ball organized on a sequence of equal sized plans. That means that our 3D data set is merely a sequence of variable diameter

1-4244-0033-3/06/$20.00 ©2006 IEEE.

rings of low thickness. Therefore, the optical flow constraint becomes not relevant for such structure of data. Moreover, computation of spatial derivatives is incoherent and noisy, mainly with reference to the normal to the surface. An attempt to improve the 3D variational model is to penalize the radial component of the vector field. However, this solution increases the complexity of the model, already extended to a higher dimension. Therefore, we thought of an alternative approach consisting in carrying out the estimation of the global and local spherical vector field.

III. Angular velocity computation

A. 3D dataset transformation

Since the movement of structures embedded on spherical surfaces has only two degrees of liberty, a system of 2D coordinates is sufficient to parametrize the displacement vector field. Spherical angular coordinates seem well adapted to the geometry of the input dataset. The idea is to map the 3D+t texture into the 2D+t angular grid, *i.e* indexed by θ, φ and t. For an entire globe, with reference to the geographical north pole and the equator, θ has range from 0 to π versus a double-sized range, for φ, *i.e* from $-\frac{\pi}{2}$ to $\frac{3\pi}{2}$. The main limit of this coordinate system is to the singularity of geographical poles. A solution to avoid computation of velocity vectors close the geographical polar zones when we ought to, could be a globe rotation, see [4] for mathematical formulates. After localizing a viewpoint, we carry out a rotation of angular coordinates (θ, φ), with reference to this viewpoint and a great circle oblique to the equator (see Figure 1).

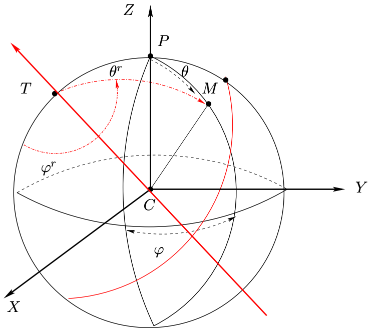

Fig. 1. Rotation of angular coordinates (θ, φ) with reference to the origin T.

Let $I(i,j,k,t)$, $I^s(\theta,\varphi,t)$ and $I^s(\theta_k,\varphi_l,t)$ be, respectively, the input $3\text{D}+t$ intensity sequence, the intermediate list of scattered points and the final 2D+t sequence. In order to compute the final sequence, we proceed as follows:

- First we convert the discrete 3D+t sequence, considered from a selected point of view, to a set of scattered values of dimension 3. First couple of variables are spherical angles (θ, φ) deduced from cartesian ones by trigonometrical conversions, and third one is time t. θ and φ, computed from discrete 3D, do not correspond to grid-positions in the output sampled image. Therefore, we obtain an intermediate list of intensity values $I^s(\theta,\varphi,t)$.
- Secondly, we attempt to fit a smooth $2\text{D} + t$ surface through the non-uniform distribution of $I^s(\theta,\varphi,t)$ data samples. For this purpose, we manage to represent our surface as a sum of weighted and shifted synthesis functions. B-splines stand apart for a good compromise between quality (namely high fidelity reconstruction and regularity) and computational issues. A framework introduced in [5] is dedicated to recover such samples, in the 2D case. The implemented algorithm makes use of a coarse-to-fine hierarchy of control lattices in order to generate a sequence of bi-cubic B-spline functions whose sum approaches the desired interpolation function. Since, we aim at estimating a 2D+t sequence, we carry out this method extended to a higher dimension.

B. Angular velocity estimation

The position of a moving pixel on the spheric surface is determined by the couple of angles (θ, φ), defined in figure 2 and the following equality : $\overrightarrow{OM} = R\overrightarrow{e_r}$.

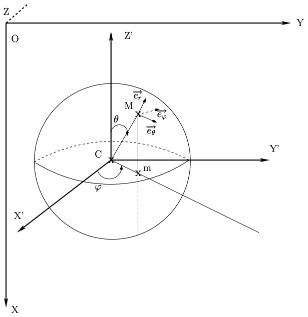

Fig. 2. Spherical system of coordinates.

The optical flow constancy constraint leads to:

$$I(R\overrightarrow{e_r},t) \approx I(R\overrightarrow{e_r} + \partial R\overrightarrow{e_r}, t+\partial t), \qquad (2)$$

where $R\overrightarrow{e_r} + \partial R\overrightarrow{e_r}$ is the displacement of the local image region at $(R\overrightarrow{e_r},t)$ after time ∂t. $\partial\overrightarrow{e_r}$ is:

$$\partial\overrightarrow{e_r} = \partial\theta\partial\overrightarrow{e_\theta} + \sin(\theta)\partial\varphi\partial\overrightarrow{e_\varphi}$$

Hence, equation 2 becomes, for a unity sphere ($R = 1$):

$$I(\overrightarrow{e_r},t) = I(\overrightarrow{e_r} + \partial\theta\partial\overrightarrow{e_\theta} + \sin(\theta)\partial\varphi\partial\overrightarrow{e_\varphi}, t+\partial t)$$

Expanding the right-hand side of this equation in a Taylor series, subtracting $I(\overrightarrow{e_r},t)$ on both sides and dividing by ∂t yields:

$$\frac{\partial I}{\partial\theta}\frac{\partial\theta}{\partial t} + \sin(\theta)\frac{\partial I}{\partial\varphi}\frac{\partial\varphi}{\partial t} + \frac{\partial I}{\partial t} = 0,$$

Let $\dot\theta$ and $\dot\varphi$ be temporal derivative of respectively θ and φ, I_θ and I_φ spatial derivatives with reference to respectively θ and φ. The optical flow constraint on a sphere is:

$$I_\theta\dot\theta + \sin(\theta)I_\varphi\dot\varphi + I_t = 0 \qquad (3)$$

C. Resolution

In order to estimate a dense motion field we use a variational formulation. We build a functional whose minimum, with respect to $\overrightarrow{v} = (\dot{\theta}, \dot{\varphi})$, corresponds to the equation 3 solution.

$$E_1(\overrightarrow{v}) = \int_\theta \int_\varphi |I_\theta \dot{\theta} + \sin(\theta) I_\varphi \dot{\varphi} + I_t|^2 d\theta d\varphi$$

Since equation 3 is under-determined, it is necessary to use an additional constraint to solve it. We then add a second term to our functional that penalizes high spatial deformations of v (it is not the only possible expression) :

$$E_2(\overrightarrow{v}) = \int_\theta \int_\varphi |\overrightarrow{\nabla} \dot{\theta}|^2 + |\overrightarrow{\nabla} \dot{\varphi}|^2 d\theta d\varphi$$

Finally, our functional E, is, with respect to $\overrightarrow{v}$:

$$E = E_1 + \alpha E_2,$$

where α is a tuning parameter between the two terms that weights the importance of the regularization term E_2. Differentiation of E, with reference to $\dot{\theta}$ and $\dot{\varphi}$ yields a set of Euler-Lagrange equations (reader is referred to [6]). Discretization of the latter with finite differences leads to a great linear system. We solve the minimization of the linear system within a Gauss-Seidel iterative scheme.

Moreover, we can estimate a piecewise continuous motion field by adding the extra assumption that the optical flow is locally constant. To find the flow vector at any point $M(\theta_M, \varphi_M)$, we minimize the function:

$$\sum_{m \in N(M)} W(m)(I_\theta(m)\dot{\theta}_M + \sin(\theta(m)) I_\varphi \dot{\varphi}_M + I_t(m))^2, \tag{4}$$

Where $W(m)$ is a normalized neighborhood weighting function and $N(M)$ is a neighborhood region surrounding M. The minimization problem is solved using singular value decomposition. As already outlined, both global and local resolution methods are similar to classical $2D$ ones, except that in our case, all assumptions are available in the spherical system of coordinates. Implementation has hence the merit to be rapid and stable, with few parameters to adjust.

IV. Visualization

We are interested in visualizing our computed optical flow field. Recall that in previous section, we performed a structural modification in our input $3\text{D} + t$ data set. We have a $2\text{D}+t$ sequence with different spatial resolutions and structure organization. Computed velocity vector field has the same spatial resolution than the $2\text{D} + t$ texture scalar sequence, as well as a coherent structure. Consequently, instead of carrying out an inverse transformation to convert our vector field to the original configuration, we select a suitable tool to visualize our spherical texture as well as the velocity vector in each couple of angular coordinates (θ, φ). Since the underlying enclosing surface has a spherical geometry, map projections are good candidates to project the 3D input surface into a planar cartographic reference. We opt for the *azimuthal equal area projection* presenting relevant visualization properties. We have then built a novel map-projection based tool for visualizing scalar and vectorial information lying on spherical surfaces, called **Mapvis**. It displays 3D scalar and vectorial data sets, see [7] for a whole description. We may sum up map-projection stages by the following scheme:

$$(\lambda, \varphi) \longmapsto (r, \theta) \longmapsto (x, y)$$

Geographical coordinates of each point of the input sphere are transformed into polar, then, cartesian ones in the projection plan. Since, our transformed $3\text{D}+t$ datasets are indexed with spherical coordinates (θ, φ), and conversion from spherical to geographical coordinates is trivial, displaying results with **Mapvis** becomes a natural simple task.

V. Results

A. Biological context

In bio-cellular imaging, microscopic cell wall models are designed in order to analyze the behavior of some biological structures and find out their functional role. Due to some biological constraints, these models, called vesicles, have spheroid shapes. Biologists inject fluorescent markers on the spheroid surface, in order to segregate between objects interacting with the input light spot. These markers target at viewing some lipid exterior phases generally of weak thickness called *rafts*, moving on vesicle surfaces. Biologists are interested on analyzing behavior of rafts (shape and mainly movement, *etc.*) in order to find out their functional role (see [8] for details). We estimate *raft* motion on the spheroid surface by computing optical flow velocity field on the simulation temporal sequences.

B. Validation

In order to validate optical flow computation on a sphere, we first consider the well known *urban taxi* sequence usually used to test 2D optical flow computation. In order to display results for $3\text{D}+t$ sequence, we map each frame of the input $2\text{D}+t$ series into a 3D sphere with suitable parameters. We then compute optical flow by both local and global algorithms on the designed spheres and project both textures and velocity fields into 2D cartographic references. In figure 3, we visualize results of movement of the white taxi toward the right road, on the scene mapped onto a spherical support. The projection origin is set as the taxi center for texture as well as velocity field on the showed frame.

We notice, in this example, that global regularization generates a smoother motion field. Local method, tested on a 20 pixel neighborhood, give similar results on moving structures *i.e* the white taxi and the black car on the left bottom side of the hemisphere. However it also yields some scattered vectors mainly in polar zone, where the deformations introduced by our cartographic projection are most important. In both cases, optical flow computation produces a set of correct vector fields (from amplitude and

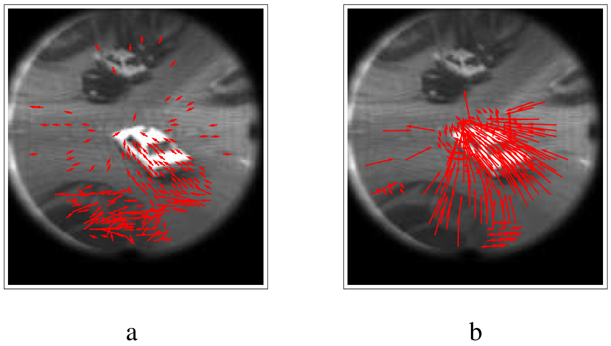

Fig. 3. Visualization of taxi sequence texture and velocity field computed, from right to the left, by global then local optical flow method on a 20 pixel neighborhood

direction point of views), perfectly tangent to the enclosing spherical surface. Besides, 3D cartesian displacements can be estimated with trivial trigonometrical operations.

C. biological data

We have tested both methods on several 3D+t cell wall simulation sequences. Due to the experimental context, images are blurred and have a very low contrast resolution, which doesn't ease optical flow computation. We display in figure 4, spherical optical flow computation results on a given sequence showing a *raft* moving on the vesicle surface, from the middle-left side to the right one on the front hemisphere. Quarter (a) shows a prospective projection of the vesicle holding the *raft* on the middle left border. In (b) we present a cartographic reference of the same frame by projecting the spheroid vesicle texture taking as origin the *raft* center. We can see that the *raft* shape is better recovered. In (c) we add to the cartographic reference the projected motion vector field computed by the global method and in (d) the projected one estimated by the local approach, in a 20 pixel neighborhood. Again, we can see that global method seem to give better results. Moreover, both methods capture correctly the *raft* movement with a minimum cost.

VI. Conclusion

In this paper, we have presented a new approach for optical flow computation and visualization in spherical context. The central idea is to take advantage from the geometry of the support on which structures are evolving to build an adapted model, involving a relevant data transformation. Since movement of structures of interest is embedded on a spherical surface, it has only two degrees of liberty. Consequently, we transform the $3D + t$ input temporal sequence to a $2D + t$ video sequence, indexed with angular coordinates. Local and global optical flow and smoothness regularization constraints are formulated on angular coordinates and they are solved with a suitable variational method. Compared to an extension of a classical method to a higher dimension, our model is more adapted to the processed data set, where the classical 3D optical constraint is not relevant, and obviously simpler and faster. Moreover, it is possible to visualize directly, the transformed sequence and the computed angular vector field with an adapted tool, **Mapvis**, with relevant visualization properties. The inverse data transformation is not necessary to highlight motion computation results. However, it is possible to deduce 3D cartesian displacements with trivial trigonometrical operations. In a future work, we plan to improve the accuracy of optical flow computation, mainly where first order constancy hypothesis doesn't hold, by building hierarchical models taking also advantage from the 2D parametrization. This framework is dedicated mainly to a spheroid geometry data context, but it could be interesting to generalize this concept to any regular surface. A solution could consist in estimating a global curvature of the enclosing surface or a local one near the object of interest.

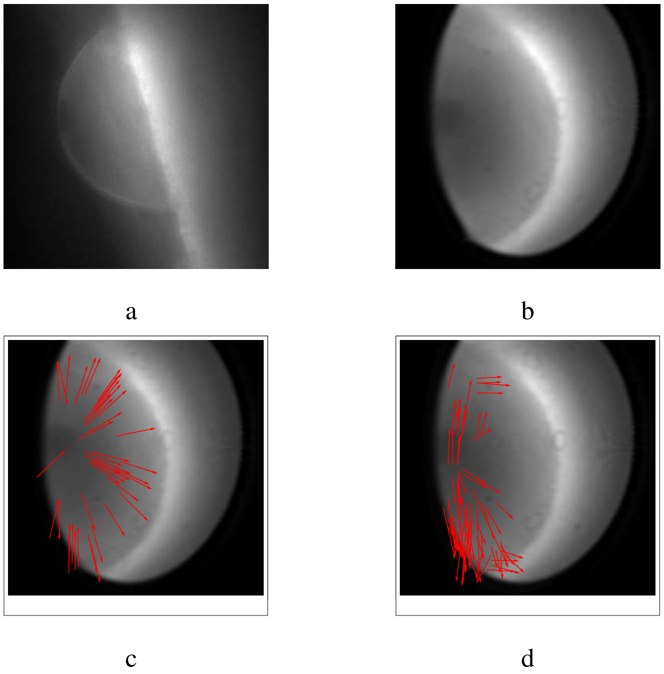

Fig. 4. Visualization of spherical optical flow computation for a 3D+t cell wall simulation sequence.

References

[1] B. Horn and Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, pp. 185–203, 1981.

[2] S. Beauchemin and J. Barron, "The computation of optical flow," *ACM Computing Surveys*, vol. 27, no. 3, pp. 433–467, 1995.

[3] B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proceedings of the 7th International Joint Conference on Artificial Intelligence*, Canada, 1981, pp. 674–697.

[4] I. Jonathan, *Datums and map projections for remote sensing, GIS, and surveying*. Scotland: Whittles Publishing: Caithness, 2000.

[5] S. Lee, G. Wolberg, and S. Shin, "Scattered data interpolation with multilevel b-splines," *IEEE Transactions on Visualization and Computer Graphics*, vol. 3, no. 3, pp. 228–244, July-September 1997.

[6] R. Glowinski, *Numerical Methods for Nonlinear Variational Problems*, springer ed. New York: Series in computational physics, 1984.

[7] W. Rekik, D. Béréziat, and S. Dubuisson, "Mapvis: a map-projection based tool for visualizing scalar and vectorial information lying on spheroidal surfaces," in *Proceedings of international conference of information visualization, IV05*, London, the United Kingdom, July 2005.

[8] K. Simons and E. Ikonen, "Functional rafts in cell membranes," *Nature*, vol. 387, pp. 569–572, 1998.