A NEW APPROACH TO THE TEMPORAL EVOLUTION OF A FAMILY OF CURVES

Vicenz Parisi Baradad and Hussein Yahia and Isabelle Herlin

Inria-Rocquencourt BP 105, 78153 Le Chesnay Cedex, France
parisi@eel.upc.es, Hussein.Yahia@inria.fr, Isabelle.Herlin@inria.fr

ABSTRACT

In this study the problem of modeling a family of curves is addressed. The need of such modeling appears frequently in many aspects of image processing where many linear structures keep spatial relationships during their evolution. We come up with a modeling tool well suited to the spatial modeling of a family of curves, and which can be very useful for motion tracking and curve evolution as well. The family of curves is represented as the line paths (orbits) of a " spline vector field ", i.e. a vector field interpolating data using a framework similar to the theory of spline curves. The model is exemplified with oceanic satellite data. Its usefullness for curve evolution modeling is also presented.

1. INTRODUCTION

Curve modeling and temporal evolution play a central role in many theoretical and applied problems [9, 8, 7, 3]. Although many researchers focused on the problem of curve evolution, few studies have been devoted to the problem of dealing simultaneously with structures interacting each other. The problem addressed in this paper is the following: given a sequence displaying different structures evolving in time, find a mathematical structure that permits:

- the modeling of each linear structure as a mathematical curve,
- a modeling of the natural relationships between these curves[1],
- the study of temporal evolution of these curves.

To take a specific example, let us consider the image displayed in picture 1. It is a NOAA-AVHRR Sea Surface Temperature (SST) of the Mediterranean sea. On this image, one sees the boundaries of different structures: a vortex and temperature fronts. These structures interact with each other, and maintain spatial relationships during their temporal evolution. Our goal is to find a single mathematical model that permit the spatial modeling of these different kinds of structures, and their temporal evolution.

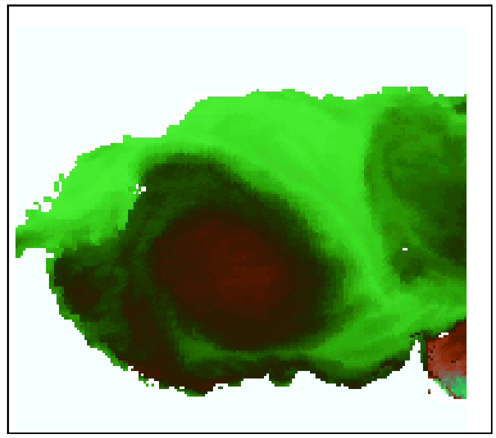

Figure 1: NOAA-AVHRR Sea Surface Temperature over the Alboran Sea. Image acquired on 24 august 1996. False colors.

In various areas of image processing and computer vision, specific modeling tools have been designed to properly handle similar phenomena. For instance in the area of face recognition and analysis, physical models have been devised [2] that can take into account salient features in a face. When the linear structures are not too spatially extended, and the deformations not too important, as it is the case in face recognition, it seems satisfactory to use standard templates and surface deformation methods. But in the general case, such methods cannot be used because the linear structures undergo too much deformation, making it untractable the use of well-known surface modeling techniques. In this study, we set up a modeling tool that can serve as a basis for modeling general disconnect sets of linear aggregations of pixels. It consists in building a vector field whose integral paths (also known as orbits of the field) are precisely the linear structures to be modeled. We argue that this

[1] For instance, in the domain of facial animation, a set of curves (the wrinkles) can be modelled as the parametrized isolines of a surface [10], and the mutual relationships between the curves are kept in the geometry of the surface.

tool is well suited for the analysis of motion. This paper is organized as follows. In section 2 we introduce the spline vector field modeling tool. The model is applied on satellite data in section 3. Then we end the paper with conclusion and perspectives.

2. A SPLINE VECTOR FIELD MODELING

Spline vector fields have been used in a similar framework for image reconstruction in [4] and in numerical analysis in [1]. In this study, we want to produce a vector field whose orbits are (among others) the linear structures (with a well chosen initial point) discussed in section 1. The vector field will be a " spline vector field " approximating a given vector data set. Moreover, since we are interested in a dense vector field, its knowledge gives the possibility of computing the path of any particle in the image. A linear structure is modeled from the vector field by giving an initial position, and computing the parametrized curve passing through that initial position using simple numerical schemes. Hence the main problem is how do we compute the vector field, and from what kind of data? These topics are discussed in the following two subsections.

2.1. The spline vector field

Given N vectors $\overrightarrow{V_i}$, we seek a C^1 vector field

$$V \; : \; I\!R^2 \longrightarrow I\!R^2$$

such that V approximates the N vectors $\overrightarrow{V_i}$. To achieve this, one could simply use standard spline interpolation techniques to compute the uncoupled x and y components of the field. But such a method looses control on the rotational and divergence of the resulting field. In fact, one would want to control the number of " curls " between the given vector data. L. Amodei and M.N. Benbourhim [1] minimize the following energy :

$$\begin{aligned} J_\alpha(V) = \quad & \alpha \int_{I\!R^2} \|\nabla \ \text{div} \ V\|^2 dxdy \; + \\ & (1-\alpha) \int_{I\!R^2} \|\nabla \ \text{rot} \ V\|^2 dxdy \end{aligned}$$

Using this energy functionnal, one tries to minimize the variations of the rotational and the divergence, and control the relative importance of each term w.r.t. the other term. The parameter α is a tuning coefficient that can be adjusted to control the relation between the rotationnal and the divergent part of the field. Given the N vectors $\overrightarrow{V_i}$, located at points X_i ($V(X_i) = \overrightarrow{V_i}$), an explicit solution is found for V. The main result is written in the following theorem [1].

Theorem 2.1 *Let*

$$\begin{aligned} g(X) = \quad & \sum_{i=1}^{N} a_i \Big(\frac{1}{\alpha}\frac{\partial^2}{\partial x^2} K(X - X_i) \; + \\ & \frac{1}{(1-\alpha)}\frac{\partial^2}{\partial y^2} K(X - X_i)\Big) \; + \\ & \sum_{i=1}^{N} b_i \Big(\frac{1}{\alpha} - \frac{1}{(1-\alpha)}\Big)\frac{\partial^2}{\partial x \partial y} K(X - X_i) \end{aligned}$$

with a_i and $b_i \in I\!R$ and

$$K(X) = -\frac{1}{2^7 \pi}\|X\|^4 \log \|X\| \; .$$

The solution

$$V_\alpha = (u_\alpha, v_\alpha)$$

of the minimization problem admits a unique expression depending of $X = (x, y)$:

$$\begin{aligned} u_\alpha(X) = \quad & g(X) + p(X) \\ v_\alpha(X) = \quad & g(X) + q(X) \; , \end{aligned}$$

where $p(X)$ and $q(X)$ are degree 1 polynomials. The coefficients are obtained by solving a linear system.

In the next subsection, we describe the preprocessing step.

2.2. Preprocessing

The preprocessing step consists in computing the initial vector data (denoted $\overrightarrow{V_i}$ in the previous subsection). We first use contour extraction (local extrema of the gradient norm and hysteresis thresholding) [5] to generate a contour image. We show, in figure 2, the result of contour extraction on image 1. It is a set of linear structures corresponding to local extrema of the gradient norm. From this data we want to generate an initial set of vectors on which the vector spline approximation is computed, in such a way that the linear structures will become the orbits of this spline vector field.

Let us denote by $\mathcal{S}$ the set of pixels belonging to the contours. To compute the direction of a tangent vector at a pixel $p \in \mathcal{S}$, we take an averaged sum of the directions of the segments linking pairs of connected pixels in the neighborhood of p. Hence the tangent direction at pixel p using the set $\mathcal{M}$ of N neighbours is given by θ_p^N:

$$\theta_p^N = \frac{1}{2N} \sum_{j \in \mathcal{M}} atan\Big(\frac{y_{j+1} - y_j}{x_{j+1} - x_j}\Big) \; .$$

In this equation $j+1$ refers to the pixel next to j. In that way, each vector $\overrightarrow{V_i}$ based at p is given the orientation of the computed angle θ_p^N.

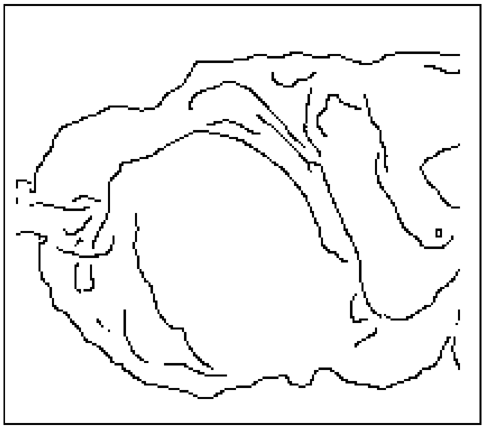

Figure 2: Result of contour extraction.

But there is a problem at this point: these sets of pixels belonging to $\mathcal{S}$ are not oriented, and the orientation of tangent vectors plays of course a crucial role in the geometry of the approximating field. To overcome this problem, we note that the optical flow field gives an instantaneous direction of motion for each pixel. Each one of the two possible tangent vectors is projected over the motion vector at that pixel and we keep the vector that gives the highest result (in norm). Each vector $\overrightarrow{V_i}$ is divided by its norm, so we take unitary initial vectors.

2.3. A synthetic example

Let us begin to investigate the results of the minimization process on synthetic data. In figure 3 is shown an instance of the N vectors $\overrightarrow{V_i}$, disposed along a rectangular vortex-like structure. In figure 4 the result of the approximation process using $\alpha = 0.1$ is shown. The result seems satisfactory, as it correctly interpolates the original data and generates a singular point (a zero of the vector field) at the center of the picture.

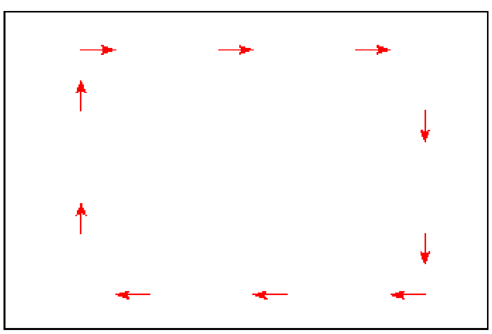

Figure 3: Initial value of N vectors data, disposed along a closed rectangular shape.

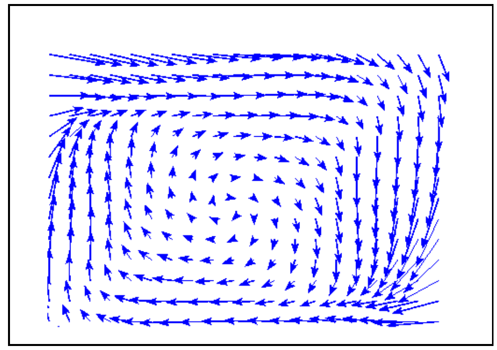

Figure 4: Computation of the interpolating vector field using the initial data of figure 3. Parameter $\alpha = 0.1$.

3. RESULTS

We use a SST (Sea Surface Temperature) NOAA AVHRR image sequence of the sea. The sequence displays, as shown in figure 1 a vortex, or gyre, turning clockwise, and also temperature fronts. Our goal is to be able to model the boundary of the gyre, and follow a particule of water along its trajectory path. As mentionned in the previous section, we need the optical flow for preprocessing. To compute the optical flow, we follow the method described in [6]. On figure 5 is shown the result of the optical flow computation. Figure 6 displays the result of the spatial flow field approximation (computation time is about 3 seconds on a 233 Mhz Dec alpha station). Note that the geometry of the field is exactly that of the gyre, with the correct sense of orientation. In figure 7 we draw, using the standard Runge-Kuntta numerical scheme, the path of a pixel entering the Strait of Gibraltar.

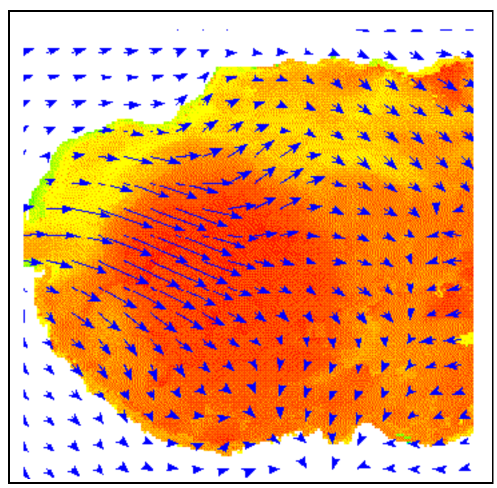

Figure 5: Result of the optical flow computation.

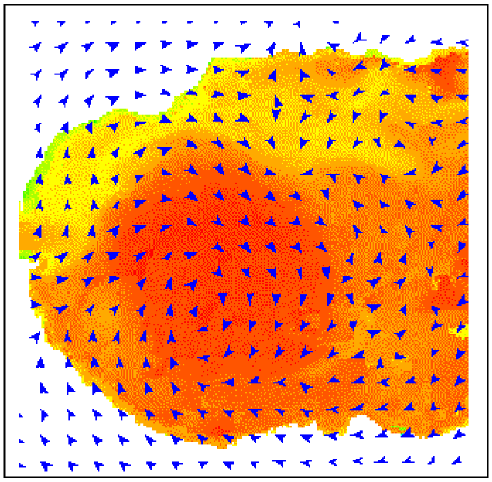

Figure 6: Result of the spatial vector field computation.

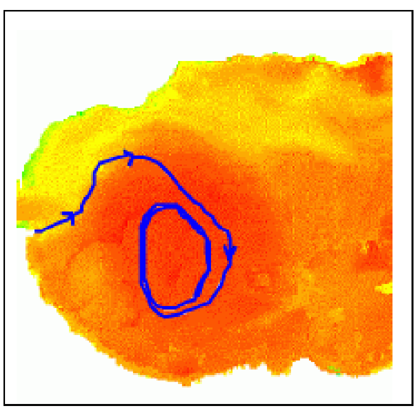

Figure 7: Computation of the orbit of a pixel.

4. CONCLUSION

A spline vector field model is used to produce a representation of sets of disconnected structures interacting each other in an image. The vector field is a " first order differential " representation of the structures, in the sense that the curves are obtained from the field using numerical integration. The field is computed by minimizing an energy controlling the rotational and divergence parts of the field. The method is applied on satellite data, showing its robustness and effectiveness. It encodes in a single representation the different curves that make the boundaries of different objects present in an image. The preprocessing uses the optical flow computation to solve the problem of orienting the initial vectors on which the approximation is performed. In a work in preparation, we use one-parameter set of diffeomorphisms generated by the optical flow to compute the image of the spline vector field by these diffeomorphisms. This is needed in the modeling of temporal evolution of the structures, which is our next target.

5. REFERENCES

[1] L. Amodei and M.N. Benbourhim. A vector spline approximation. *Journal of Approximation Theory*, 67:51–79, 1991.

[2] S. Basu, N. Olivier, and A. Pentland. 3d modeling and tracking of human lip motions. In *ICCV'98*, pages 337–343, Bombay, January 1998.

[3] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. In *ICCV'95*, pages 694–699, Boston, MA, June 1995.

[4] F. Chen and D. Suter. Image coordinate transformation based on div-curl vector splines. In *ICPR'98*, pages 518–520, Brisbane, August 1998.

[5] R. Deriche. Optimal edge detection using recursive filtering. *International Journal of Computer Vision*, pages 167–187, 1987.

[6] I. H. Herlin, I. Cohen, and S. Bouzidi. Image processing for sequences of oceanographic images. *The Journal of Visualization and Computer Animation*, 7:169–176, 1996.

[7] G. Sapins and A. Tannenbaum. Affine invariant scale-space. *International Journal of Computer Vision*, 11(1):25–44, 1993.

[8] J. A. Sethian. *Level Set Methods. Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision and Materials Science*. Cambridge Monographs on Applied and Computational Mathematics, 1996.

[9] K. Siddiqi, A. Tannenbaum, and S. W. Zucker. Hyperbolic smoothing of shapes. In *ICCV'98*, pages 215–221, Bombay, January 1998.

[10] M. L. Viaud and H. Yahia. Facial animation with wrinkles. In *Third Eurographics Workshop on Animation and Simulation*, Cambridge, September 1992.