

Using Crest Lines to guide Surface Reconstruction from Stereo

Richard Lengagne¹, Pascal Fua², Olivier Monga¹

¹INRIA, Domaine de Voluceau - Rocquencourt

BP. 105, 78153 Le Chesnay Cedex, France

²SRI International, 333 Ravenswood Avenue

CA 94025 Menlo Park, USA

e-mail: Richard.Lengagne@inria.fr, fua@ai.sri.com, Olivier.Monga@inria.fr

Abstract

We propose an approach to interleave surface reconstruction from multiple images and feature extraction. It uses an object-centered representation that is optimized to conform to the surface shape. It also extracts typical features such as crest lines and uses them to guide the optimization.

We present results using aerial images and terrain data.

1 Introduction

In previous work, we have developed surface reconstruction methods that use an object-centered representation (a triangulated mesh) to recover geometry and reflectance properties from multiple images [4]. These methods use a snake-like optimization process [5] that iteratively modifies the mesh in order to minimize an objective function. Feature extraction can then be treated as a separate process. For example, features can be extracted manually and optimized as linear snakes. The features, in turn, can be used to further refine the mesh by retriangulating and reoptimizing. Manual extraction, however, can be strenuous and inaccurate. We thus need an automated tool to extract relevant features from the mesh. To enforce the consistency of terrain and features, we propose to incorporate the feature extraction in the whole iterative optimization process. In this way, the terrain model and the feature data are more likely to be in agreement.

In this paper, we focus on crest lines: they provide us with a satisfactory geometrical representation of important physical properties such as ridge lines and valleys in the case of aerial images, or orbits and other typical characteristics in the case of face images.

Crest line extraction has been thoroughly studied [9, 6] and carried out recently on various kinds of data, especially medical 3D data [8, 7]. One of the main characteristics of those

features is that they use local information to yield a global description of the surface which is very stable and can be used for registration, surface modeling or recognition purposes [1].

Since crest lines are defined using directional derivatives of the maximum curvature of the surface at each point, it is crucial to be able to compute reliably the differential properties of a surface represented by a triangulation. Extracting such features, however, is difficult because triangulations are discrete approximations of the surface that do not naturally lend themselves to the computation of differential properties, especially not away from the vertices. This is a problem because features such as crest lines do not necessarily go through the vertices. In general, they traverse the triangulation facets between vertices.

To overcome this problem, our algorithm:

- fits a quadric to the neighborhood of each vertex of the mesh.
- computes the principal curvatures and the principal curvature directions of the surface at each vertex, as well as the derivative of the maximum curvature in the maximum curvature direction.
- extracts the zero-crossings of this derivative and tracks them over the whole mesh.

Once these features have been reliably detected, they can be used to improve the representation of the underlying surface, by either deforming the mesh so that its edges coincide with the crest lines, or by refining the triangulation in their vicinity. The mesh can then be reoptimized and these steps iterated.

This article is thus organized as follows.

In Section 2, we briefly describe our surface reconstruction method from multiple images.

Section 3 explains the computation of the differential properties of a surface represented by a triangulated mesh and focuses on the crest line extraction algorithm.

Section 4 shows some results obtained by this algorithm using real imagery.

Section 5 discusses the ways of combining feature extraction and surface reconstruction presented above.

2 Surface optimization with a snake-like method

We recover a model shape by minimizing an objective function $\mathcal{E}(S)$ that embodies the image-based information. It is the sum of a stereo term and a shape-from-shading term [4]. In this paper, however, we only consider the stereo term, which is very appropriate to highly-textured images like terrain images. On the other hand, the shape-from-shading term is most useful when dealing with areas with constant or slowly varying albedo.

Since we are dealing with calibrated stereo pairs, we can compute the stereo term by comparing the intensities of the projections in each image of some points regularly sampled on a facet of the mesh. Thus, Optimizing the mesh with respect to the stereo energy tends to minimize this term.

In all cases, $\mathcal{E}(S)$ typically is a highly nonconvex function, and therefore difficult to optimize. However, it can effectively be minimized [5] by :

- introducing a quadratic regularization term $\mathcal{E}_D = 1/2S^t K_S S$ where K_S is a sparse stiffness matrix,
- defining the total energy $\mathcal{E}_T = \mathcal{E}_D(S) + \mathcal{E}(S) = 1/2S^t K_S S + \mathcal{E}(S)$,
- embedding the curve in a viscous medium and iteratively solving the dynamics equation $\frac{\partial \mathcal{E}_T}{\partial S} + \alpha \frac{dS}{dt} = 0$, where α is the viscosity of the medium.

Because \mathcal{E}_D is quadratic, the dynamics equation can be rewritten as

$$\begin{aligned} K_S S_t + \alpha(S_t - S_{t-1}) &= - \left. \frac{\partial \mathcal{E}}{\partial S} \right|_{S_{t-1}} \\ \Rightarrow (K_S + \alpha I)S_t &= \alpha S_{t-1} - \left. \frac{\partial \mathcal{E}}{\partial S} \right|_{S_{t-1}} \end{aligned} \quad (1)$$

In practice, α is computed automatically at the start of the optimization procedure so that a prespecified average vertex motion amplitude is achieved. The optimization proceeds as long as the total energy decreases. When it increases, the algorithm backtracks and increases α , thereby decreasing the step size. In effect, this optimization method performs implicit Euler steps with respect to the regularization term [5] and is therefore more effective at propagating smoothness constraints across the surface than an explicit method such as conjugate gradient.

3 Crest line extraction on a triangulated mesh

3.1 Definition

We want to find a geometrical representation of some physical features such as ridges, river beds, valleys and design an algorithm which can extract them automatically. A natural idea would be to calculate the local curvatures of the surface at each vertex of the mesh and select the points where the maximum curvature is either high or locally maximum. However, since those features may cross the facets between vertices, simply extracting the vertices that are maxima of curvature would not yield the appropriate results.

To overcome this problem, we have defined a crest point as a zero-crossing of the derivative of the maximum curvature in the maximum curvature direction [7]. We can attach to each point of the surface two principal curvatures and two principal curvature directions. If k_1 and t_1 denote respectively the maximum curvature and the maximum curvature direction, a crest point is thus defined by the equation :

$$\langle \vec{\nabla} k_1, t_1 \rangle = 0$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product and $\vec{\nabla}$ is the gradient operator.

A crest line is the locus of these zero-crossings. The notion of crest point uses a third order derivative of the surface, and is therefore very sensitive to noise. We thus need to smooth the surface before starting any computation.

3.2 Curvature estimation

We compute the curvatures at each vertex of the mesh by fitting a quadric to the neighborhood of this vertex with a least-square method using the points of the neighborhood and the normals to the surface at these points. The size of the neighborhood used for quadric-fitting is an important parameter of the crest line extraction program. Increasing the neighborhood is equivalent to further smoothing the surface. We compute the first and the second fundamental forms attached to that quadric.

In the quadric-fitting approximation, the altitude z of vertex $V(x, y, z)$ is expressed as a function $z(x, y)$ of the x and y coordinates such that

$$z(x, y) = ax^2 + bxy + cy^2 + dx + ey + f$$

The tangent plane to the surface at point $V = (x, y, z(x, y))$ is defined by the two vectors

$$\vec{v}_1 = \frac{\partial V}{\partial x} = (1, 0, 2ax + by + d)$$

and

$$\vec{v}_2 = \frac{\partial V}{\partial y} = (0, 1, 2cy + bx + e)$$

The normal to the tangent plane is defined as

$$\vec{n} = \vec{v}_1 \wedge \vec{v}_2$$

The matrix of the first fundamental form is thus [2] :

$$F_1 = \begin{bmatrix} \langle \vec{v}_1, \vec{v}_1 \rangle & \langle \vec{v}_1, \vec{v}_2 \rangle \\ \langle \vec{v}_1, \vec{v}_2 \rangle & \langle \vec{v}_2, \vec{v}_2 \rangle \end{bmatrix}$$

The matrix of the second fundamental form is:

$$F_2 = \begin{bmatrix} \langle \vec{n}, \frac{\partial^2 V}{\partial x^2} \rangle & \langle \vec{n}, \frac{\partial^2 V}{\partial x \partial y} \rangle \\ \langle \vec{n}, \frac{\partial^2 V}{\partial x \partial y} \rangle & \langle \vec{n}, \frac{\partial^2 V}{\partial y^2} \rangle \end{bmatrix}$$

The matrix of the Weingarten endomorphism is

$$W = -F_1^{-1} F_2$$

The eigenvalues and the eigenvectors of W are respectively the principal curvatures k_1 and k_2 and the principal curvature directions \vec{t}_1 and \vec{t}_2 of the surface at vertex V .

In order to ensure the consistency of the orientation of the principal frame $(\vec{n}, \vec{t}_1, \vec{t}_2)$, we enforce:

$$\det(\vec{n}, \vec{t}_1, \vec{t}_2) > 0$$

Among the six neighbors of vertex V , we choose the vertex V_1 which maximizes $\langle \vec{V}\vec{V}_1, \vec{t}_1 \rangle$. Then, we estimate the derivative of the maximum curvature in the maximum curvature direction (denoted as dk_1) by finite differences, and set:

$$dk_1(V) = k_1(V_1) - k_1(V)$$

3.3 Zero-crossing extraction

The extraction of the zero-crossings of dk_1 is performed using a tracking algorithm inspired by the Marching Lines algorithm [10]. Here, we are dealing with regular hexagonal triangulations. On each facet F of the mesh, we apply the following algorithm :

- for each vertex V of F , determine the sign of the derivative $dk_1(V)$.
- if, for two neighbors V_1 and V_2 , $dk_1(V_1).dk_1(V_2) < 0$, there is a crest point on the edge $(V_1 V_2)$. Interpolate linearly dk_1 along the edge $(V_1 V_2)$ and find the location of the zero-crossing of dk_1 .
- another zero-crossing must appear on one of the two other edges of the facet. Locate it on the appropriate edge.
- draw a segment across the facet.

By applying this scheme to all the facets of the mesh, we can draw lines on the triangulation. They are guaranteed to be continuous, and either form a loop on the surface or cross

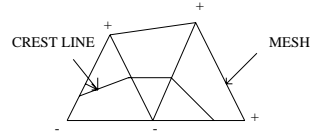


Figure 1. The zero-crossing extraction algorithm

the whole surface from one boundary to the other.

Fig. 1 shows the tracking of the crest points over three facets. The + and - signs on the vertices indicate the signs of dk_1 .

The algorithm links all the zero-crossings of dk_1 that can be found on adjacent edges. The crest line is thus composed of maxima and minima of the maximum curvature. A simple thresholding on the value of the interpolated maximum curvature of each zero-crossing, compared to the maximum value of this curvature on the whole surface, enables us to get rid of most of the spurious points.

3.4 Experimental results

We have tested our algorithm on several meshes representing terrain models. Fig. 2 shows the crest lines extracted from the mesh at a rather coarse scale (255 vertices and 448 facets). The neighborhood is taken to be equal to 2, i.e. the 6 neighbors of a vertex and all the neighbors of these neighbors are used for the quadric approximation ; this value for the neighborhood is experimentally a good trade-off between the smoothing of the surface and the accuracy of the results. But different thresholds on the maximum curvature value can be applied. The threshold has been set to 30% of the maximum value of the maximum curvature on the whole mesh. A high threshold (e.g. 50 %) would discard every feature except the main crest line on the top of the cliff.

Fig. 3 shows the results obtained on a mesh representing a terrain with two series of outcrops with two different thresholds (respectively 30 % and 50 %) and two different neighborhoods (repectively 2 and 3) . This scene is much more complex than the previous one, especially around the bumps on the left part. A larger smoothing (right image) yields more continuous and significant lines but can tend to move them to wrong locations. This is the scale-space effect described in [11].

Fig. 4 shows the extraction of a river bed. It is of course very hard to choose the right parameters of the algorithm so that only the river bed would be detected. Restricting our search to a neighborhood surrounding the valley, we could extract it quite efficiently.

A critical point of this approach is the level of refinement of the triangulation we use. At a coarse scale, we are able to

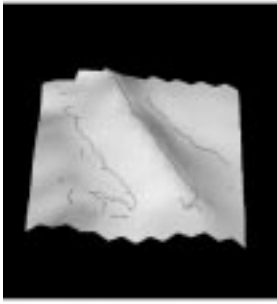


Figure 2. Extraction of a ridge and two valleys shown on a Gouraud shaded rendering of a terrain model

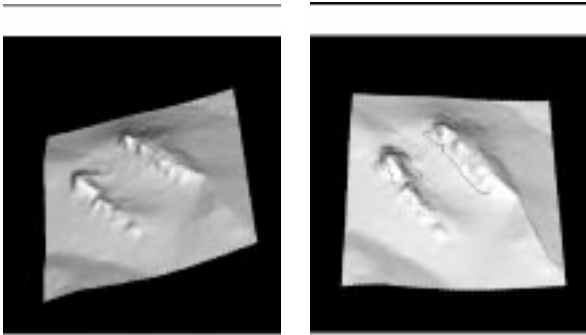


Figure 3. Outcrops

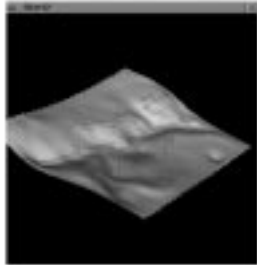


Figure 4. Extraction of a river bed

extract roughly some crest lines, which can help describe the surface in terms of global features. At a higher resolution, the local information can be too noisy to create significant lines. For instance, if the terrain shows an alignment of bumps, a coarse scale enables us to detect a line along this alignment, so that we can refine the mesh around this line, but if we look at the terrain at a finer scale, the bumps will prevent us from extracting a continuous line.

4 Using crest lines to improve the model

The next step is to use the information we have extracted on the mesh to derive a more accurate description of the surface in the areas where the differential information is meaningful, i.e in the river beds, the valleys, on the crests, etc... We first propose the following algorithm :

- extract some crest lines on a mesh.
- for each detected zero-crossing, find the closest vertex to this zero-crossing.
- move this vertex towards this zero-crossing so that the edges of the mesh coincide with the crest lines.
- optimize the new mesh using the algorithm of Section 2.
- restart the process with the new mesh.

Incorporating the differential information in the reconstruction process ensures that the model fits to the data (through the stereo term) and is consistent with the geometrical features extracted.

Fig. 5 shows the result using the mesh of Fig. 2, and superposes the main crest line obtained from the original mesh and the line obtained after 10 iterations of our algorithm. The edges of the mesh tend to coincide with the crest line obtained after 10 iterations. The new facets are crossed by the crest line extracted at the first iteration.

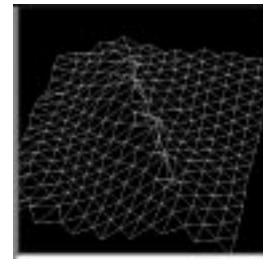


Figure 5. A mesh after optimization with incorporating feature extraction

We can then further refine the mesh in the areas that contain typical features. For that purpose, we apply the following algorithm:

- start from a regular coarse mesh which has already been optimized by the method described in Section 2.
- extract the main crest lines on this mesh.
- define a neighborhood around these lines.
- move the vertices of the mesh so that the edges coincide with the crest line.
- uniformly refine the mesh inside these neighborhoods with a uniform subdivision of the facets.
- reoptimize the finer mesh.

Fig. 6 shows the result of this local refinement around the crest line.

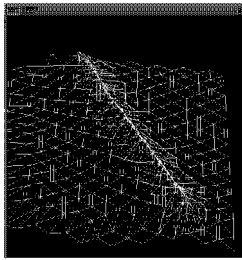


Figure 6. A mesh with the refinement of the neighborhood of the crest line

If we reapply the crest line extraction on the new area, the continuity of the line can be lost because the local information is too noisy. However, we would like to preserve the continuity of this line while taking advantage of more accurate information about the surface. A way of overcoming this problem would be to consider the crest line, whose initial shape is determined by the coarsest scale, as a snake which then can evolve under the following constraints:

- the crest line extracted at scale s_t must remain “close” enough to the line extracted at scale s_{t-1} .
- the crest line must remain on the terrain.
- the crest line must remain fairly smooth.

We have implemented a constrained optimization algorithm [3] that can simultaneously optimize the mesh and the ridge line and that we intend to customize for this purpose in future work.

5 Conclusion

In this work, we have provided a tool for automatic extraction of features of interest from the differential geometry point of view, like ridge lines, rivers, valleys. We have shown how we could insert this extraction in the terrain model optimization process in order to guide the surface reconstruction.

In future work, we intend to :

- apply finite element methods in the optimization scheme in order to be able to deal with irregular meshes introduced by a Delaunay triangulation.
- test a constrained optimization algorithm to simultaneously optimize the mesh and the ridge line.

The ultimate purpose of this research is to develop a global and semantic description of the surface from stereo images using local information. The extraction of typical features such as crest lines from image representations like triangulated meshes and their incorporation in the reconstruction process are therefore significant steps in image analysis and understanding, and crucial milestones for surface modeling.

References

- [1] I. Bricault and O. Monga. From volume medical images to quadratic surface patches. *To appear in Computer Vision and Image Understanding*, 1996.
- [2] M. P. do Carmo. *Differential Geometry of Curves and Surfaces*. Prentice-Hall, Englewood Cliffs, 1976.
- [3] P. Fua and C. Brechbuhler. Imposing hard constraints on soft snakes. In *European Conference on Computer Vision, Cambridge, U.K., II*, p. 495-506, 1996.
- [4] P. Fua and Y.G. Leclerc. Object-centered surface reconstruction: Combining multi-image stereo and shading. *International Journal on Computer Vision*, 1995.
- [5] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal on Computer Vision*, 1(4): 321-331, 1988.
- [6] J. Koenderink. *Solid Shape*. MIT Press, Boston, 1990.
- [7] O. Monga and S. Benayoun. Using partial derivatives of 3d images to extract typical surface features. *To appear in Computer Vision and Image Understanding*, 1996.
- [8] O. Monga, R. Lengagne, and R. Deriche. Extraction of the zero-crossing of the curvature derivative in volumic 3d medical images : a multi-scale approach. In *Conference on Computer Vision and Pattern Recognition, 1994, Seattle, USA*, 1994.
- [9] J. Ponce. *Représentation des objets tridimensionnels*. PhD thesis, Université d’Orsay (Paris Sud), 1988.
- [10] J.-P. Thirion and A. Gourdon. Computing the differential properties of isointensity surfaces. *Computer Vision and Image Understanding*, 61-2, 190-202, 1995.
- [11] A. Witkin. A multi-scale approach to extract zero-crossings of the laplacian. In *Proceedings of International Joint Conference on Artificial Intelligence*, 1982.