Using Differential Constraints to Reconstruct Complex Surfaces from Stereo

Section: Shape Representation and Recovery
Keywords: *3D reconstruction, deformable models, differential geometry, adaptive meshes*

Abstract

We propose an approach to constrain the reconstruction of complex surfaces to differential information. Both linear structures such as crest lines or scalar fields such as curvature values can be used to generate a reconstruction of the surface which is consistent with the differential properties. This method improves the accuracy of the reconstruction around the discontinuities and increases the compactness of the surface representation.

1 Introduction

3D reconstruction from a pair or a sequence of stereoscopic images is a major issue in many computer vision tasks and is a crucial step in many applications such as teledetection, interaction with virtual worlds, object recognition or data transmission. However, one of the keypoints of a reconstruction algorithm is its ability to deal with discontinuitites. Several studies have already been carried out in order to cope with the discontinuities. For instance, [7] presents a method to generate a depth map which preserves the depth discontinuities. Some authors, like [12], propose some algorithms to reconstruct some specific shapes, like ellipsoids. These methods, though powerful, are very limitative on the kind of objects we want to deal with. On the other hand, "blind" algorithms, like the one presented in [9], will perform nicely on the overall surface and produce a reconstruction which both fits to the data and respects some smoothing constraints, but will fail in accurately reconstructing the neighborhood of a discontinuity.
[8] shows that, in the case of terrain reconstruction, imposing some constraints on the models (whether surfaces or linear structures) one wants to optimize can both improve the accuracy of the reconstruction and ensure the mutual consistency of the models.
In this work, we are focusing on non polyhedral objects, so that the discontinuities we are dealing with are linked to differential properties of order 2 or 3. Our purpose is to find a general method to reconstruct with a good precision the areas around discontinuities, that we assume to be the regions of interest of the surface, since they contain some significant information which can be efficiently used for registration or recognition purposes. We basically want our method to be rather independent of the object we want to reconstruct. However, we allow a certain degree of interactivity, i.e. the user can choose, among all the regions of interest that the system has found according to differential properties, some areas that he assumes to be more relevant. For instance, in case we want to refine the reconstruction around linear structures like crest lines, the user can specify some constraints related to the number, the length or the orientation of the lines. Therefore, our approach lies between blind vision algorithms and model-based methods.
Our general framework is a reconstruction scheme based on deformable models as presented in [9], in which we incorporate differential infomation to guide the reconstruction.
Section 2 recalls the snake-like optimization process, investigated by [10] and [5] among others, that iteratively modifies the mesh in order to minimize an objective function. A finite difference version and a finite element version of this algorithm are both implemented to enable us to deal with both regular and irregular meshes.
Section 3 recalls the computation of differential properties on triangulated meshes and proposes a way to automatically extract crest lines on those meshes.
Section 4 presents two different ways of taking into account the differential properties of the surface in the reconstruction process.
In case we can accurately extract some crest lines on the surface, we use this structural information to guide the reconstruction, improve its accuracy and produce a more compact representation of the surface which reflects its differential properties, since more facets are generated in high curvature-areas and few facets in flat areas.
In case the crest line extraction is not completely satisfactory, we use the curvature field to constrain the reconstruction. Both methods generate a new mesh which is optimized using stereo and regularization constraints.
We show some experimental results of surface reconstruction using both methods, from terrain and face images.

2 Surface reconstruction using deformable models

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 [9]. 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 [10] by :

- introducing a quadratic regularization term $\mathcal{E}_D = 1/2 S^t K_S S$ where K_S is a sparse stiffness matrix,
- defining the total energy $\mathcal{E}_T = \mathcal{E}_D(S) + \mathcal{E}(S) = 1/2 S^t K_S S + \mathcal{E}(S)$,
- 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 [10] and is therefore more effective at propagating smoothness constraints across the surface than an explicit method such as conjugate gradient.

In this work, we have to deal with arbitrary triangulations, i.e. they can be regular and hexagonal (each vertex but the boundary ones has exactly six neighbors) or not. Using regular triangulations can be viewed as a limitative constraint but its advantage lies in the fact that it enables us to use a finite-difference based optimization scheme which is rather easy to implement and fast in terms of computation time and convergence properties.
In such a case, the stiffness matrix K is defined by identifying $\mathcal{E}_D$ with the equations :

$$\begin{aligned} e(i) = \tfrac{1}{2} \sum_{k=1}^{6} & (2x_i - x_{i,k} - x_{i,k+3})^2 + \\ & (2y_i - y_{i,k} - y_{i,k+3})^2 + \\ & (2z_i - z_{i,k} - z_{i,k+3})^2 \end{aligned} \quad (2)$$

$$\mathcal{E}_D = \sum_{i=1}^{N} e(i) \quad (3)$$

where (x_i, y_i, z_i) are the 3D coordinate of vertex i, and (i,k) denotes the $k-th$ neighbor of vertex i, if the g neighbors of i are ordered clockwise.

In case we deal with irregular triangulations (where each vertex can have an arbitrary number of neighbors), we have to implement a finite element version of the discretization of the surface and define a new regularization term for the optimization process.
Basically, the regularization energy term becomes [15] :

$$\mathcal{E}_D = \int\int_{\omega} \left| \frac{\partial^2 z}{\partial u^2} \right|^2 + 2 \left| \frac{\partial^2 z}{\partial u \partial v} \right|^2 + \left| \frac{\partial^2 z}{\partial v^2} \right|^2 du dv \quad (4)$$

where $(u,v) \rightarrow z(u,v)$ is a parameterization of the surface. Here, the parameterization is $(x,y) \rightarrow z(x,y)$ where (x,y,z) are the Cartesian coordinates of a vertex in the 3D space. If we want to deal with self-occluding objects and produce fully 3D reconstructions, we have to change the parametrization into a cylindric $(z(r,\theta))$ or spheric one $(z(r,\theta,\phi))$. A new stiffness matrix K is computed according to equation 4, and the optimization is performed again.
Another interest of the finite element method is the approximation of the surface $z(x,y)$ by a polynomial function (here of order 5). This analytical expression of the surface can be of great interest for a further modeling of the surface into a set of analytical patches (quadrics, cubics or splines as already investigated in [2] for instance).

We now show the result of this optimization process on terrain images (figure 1). In fact, we apply a hierarchical approach : we generate the depth map by a standard correlation algorithm, and triangulate the depth map in order to produce a coarse initial mesh(figure 2a). The initial mesh has to be rather close to the real surface, so that the surface optimization algorithm can perform well. The optimization of the initial mesh is performed with small images (typically, one edge of the mesh corresponds to five pixels of the images) and a rather high degree of smoothing. The result of this first optimization is refined and reoptimized using higher resolution images and a smaller degree of smoothing(figure 2b).

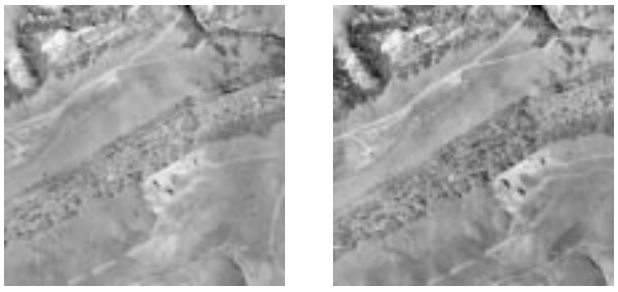
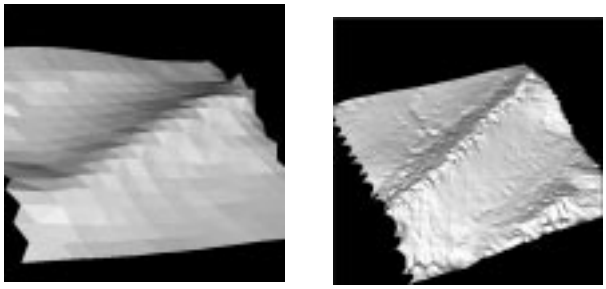


Figure 1: A pair of aerial images of a terrain



(a) (b)

Figure 2: The triangulation of the depth map providing the initialization of the optimization process (a) and the result of the optimization using a regular mesh (b)

3 Recovering differential properties on triangulated meshes

3.1 Purpose

One drawback of this optimization process is that the same degree of smoothing is applied to the whole surface, not discriminating flat areas rom curved ones. One way of circumventing that problem is to selectively refine the mesh in areas of interest. For instance, in the previous example, we may need more accuracy, i.e. more facets, in the neighborhood of the central ridge. In many cases, the regions of interest correspond to significant features from the point of view of differential geometry, like ridges, river beds, valleys in case of terrains, or orbits, nose ridges in case of faces. Of course, a user may want to specify by himself these areas, drawing a line or selecting a region on the surface. But an automatic detection of regions of interest can be more accurate and less strenuous than a manual one.

3.2 Curvature estimation

This problem thus amounts to recovering the differential properties of order 2 or 3 on a triangulation, which is not straightforward, since it is a piecewise planar approximation of the surface. This has already been thoroughly studied ([3],...). Here , we chose to compute the curvature field 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([13]). 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 $\vec{v_1} = \frac{\partial V}{\partial x}$ and $\vec{v_2} = \frac{\partial V}{\partial y}$. The normal to the tangent plane is defined as $\vec{n} = \vec{v_1} \wedge \vec{v_2}$. We compute the matrices of the two fundamental forms of the surface Φ_1 and Φ_2 (see [6]) and the matrix of the Weingarten endomorphism $W = -\Phi_1^{-1}\Phi_2$. 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:

$$det(\vec{n}, \vec{t_1}, \vec{t_2}) > 0$$

3.3 Crest line extraction

An accurate crest line extraction requires the computation of differential properties of order 3, since a crest line is defined as the set of zero-crossings of the derivative of the maximum curvature in the maximum curvature direction [14]. If k_1 and $\vec{t_1}$ denote respectively the maximum curvature and the maximum curvature direction, a crest point is thus defined by the equation:

$$dk_1 = < \vec{\nabla} k_1, \vec{t_1} > = 0$$

where $< .,. >$ denotes the inner product and $\vec{\nabla}$ is the gradient operator. The extraction of the zero-crossings of dk_1 is performed using a tracking algorithm inspired by the Marching Lines algorithm [16]. Among the neighbors of vertex V, we choose the vertex V_1 which maximizes $< \overrightarrow{VV_1}, \vec{t_1} >$. Then, we estimate the derivative of the maximum curvature in the maximum curvature direction by finite differences, and set:

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

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_1V_2). Interpolate linearly dk_1 along the edge (V_1V_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 which are guaranteed to be continuous.
Figure 3 shows the tracking of the crest points over three facets. The + and - signs on the vertices indicate the signs of dk_1.

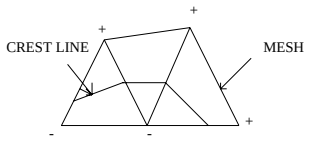


Figure 3: The zero-crossing extraction algorithm

4 Guiding the reconstruction with differential properties

4.1 Using crest lines to constrain the reconstruction

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...in the case of terrain models, or on orbits, or the nose ridge in the case of faces.
[4] presents a way of constraining the reconstruction of a polyhedral surface model from a range image to the surface characteristics extracted from the data.
In our approach, we want to deal with arbitrary complex surfaces and combine the differential constraints and the optimization process previously decribed.
[11] proposes a method that deforms the mesh according to the crest lines by moving the vertices so that the edges of the mesh coincide with the crest line. The new mesh in then reoptimized using the image information and the regularization constraint.
Increasing the level of refinement in the regions of interest would tend to achieve both purposes that we have defined : more accuracy around the discontinuities and a more compact representation of the surface, guided by the differential properties.
In case we are able to detect rather accurately a certain number of crest lines on the surface, we can directly use this structural information to govern our reconstruction.

Here, we propose the following algorithm :

- start from an initial mesh obtained by triangulating a depth map.
- optimize this regular mesh using a finite difference version of the algorithm of section 2.
- extract some crest lines on the optimized mesh (either automatically or manually). These lines will generate new points on the edges.
- generate a new mesh including these new points with smaller facets around the crest lines.
- optimize the new mesh using a finite element implementation of 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 of the surface.

We detect the crest lines on the optimized mesh and select the ones we want to keep (according to criteria based on length, orientation, number,...). Figure 4 shows all the lines of zero-crossings all over the mesh. Figure 5a shows the remaining crest lines after a thresholding on the value of the maximum curvature, and figure 5b is the result of the selection of the crest lines after adding a constraint on their length. This selection step can be performed interactively, i.e. the user can choose to keep one or several crest lines according to some constraints he has preliminarily defined.

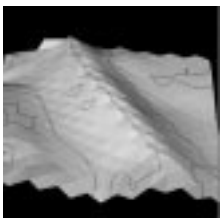
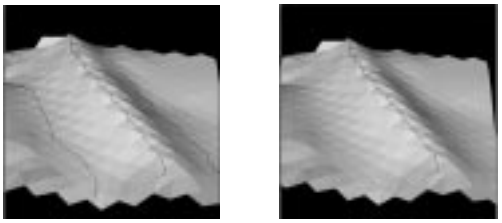


Figure 4: The extraction of all the zero-crossings



(a) (b)

Figure 5: The crest lines after applying a curvature-based constraint (a) and adding a length-based constraint (b)

We then generate a new mesh which includes the crest points, so that its edges will exactly coincide with the crest line of the previous mesh. We force the mesh generator to produce more facets around the line than elsewhere, by specifying the average length of the edges in all areas [1]. This both makes the mesh be in agreement with the geometrical feature we have extracted and be more accurate in the neighborhood of this feature, since the level of refinement will be higher in this area. The new mesh is then optimized with higher resolution images and a slighter degree of smoothing than the first one, and the whole process is reapplied on that mesh. We iterate this whole process several times, which little by little makes the

crest line and the mesh converge and be consistent with each other. This is in fact a multi-resolution approach where the resolution is the level of refinement of the mesh around the crest lines, associated with the resolution of the stereo images. We use the information extracted at one level of refinement to guide the reconstruction at the next level. This is also a crucial point in the selection of the crest lines: according to [17], thew most significant features are the ones which appear at the larger number of scales (or resolutions), whereas their precise locations are given at the finest resoltion. We apply this idea in the crest line detection: we keep the lines which appear at several scales and reject the ones which only appear at fine scales, since they are more probably due to noise.

We show in figure 6 and 7 the result of the optimization with incorporating the crest line information. We focused in this example on the main crest line in the center of the surface. Let us notice that, in this case, the extraction could be performed manually, since it is easy for a user to outline the ridge. We compare this result with the one obtained using only the minimization of the stereo energy combined with the regularization constraint (section 2),shown in figure 8. In the latter case, both the mesh and the crest line are very wrinkly and completely disconnected, whereas in the former case the mesh and the crest line are more in agreement with each other.

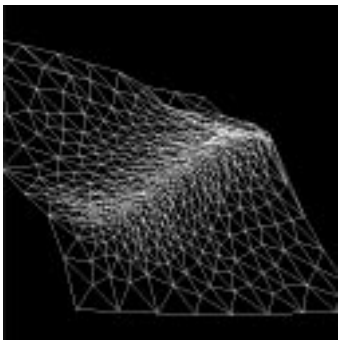


Figure 6: Optimization with crest line information: the mesh

We have also run our algorithm on another terrain example showing river beds (figures 9,10 and 11).

4.2 Using the curvature field to constrain the reconstruction

Sometimes, it is not straightforward to extract all the crest lines we would like to. For instance, on the face model shown in figure 12, we are able to easily detect the orbits but the nose ridge is much harder to retrieve, probably because the initial mesh and the optimized are too noisy for the crest line extraction algorithm to perform well in this area,as shown in figure 16. In that case, we may want to use curvature information which is more likely to be accurately computed, since it uses differential properties of order 2 of

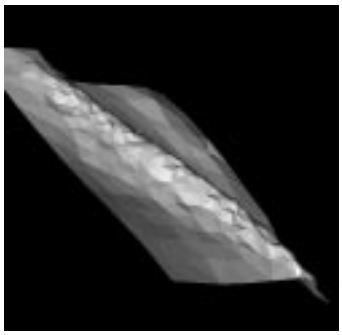


Figure 7: Optimization with crest line information: a shaded view with the crest line shown in black

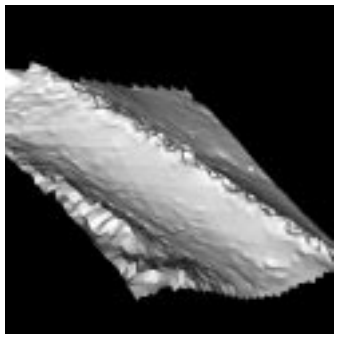


Figure 8: Optimization without crest line information: a shaded view with the crest line shown in black

the surface, which are less sensitive to noise than the crest point information.Therefore, the maximum curvature scalar field is used to govern the remeshing of the surface. Practically, we apply the following algorithm:

- start from an initial mesh obtained by triangulating a depth map.
- optimize this regular mesh using a finite difference version of the algorithm of section 2.

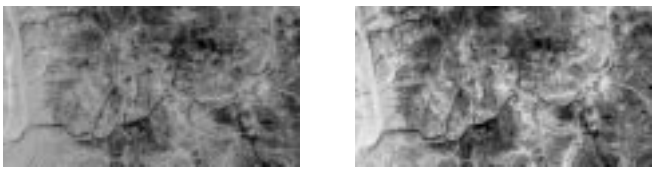


Figure 9: Left (a) and right (b) images of river beds.

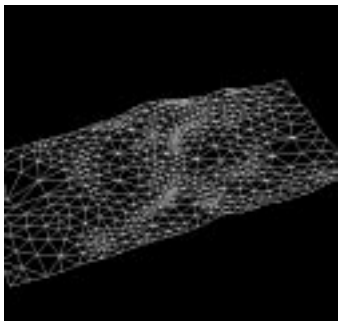


Figure 10: The optimization incorporating crest line information: the mesh

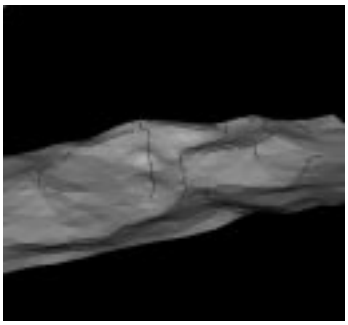
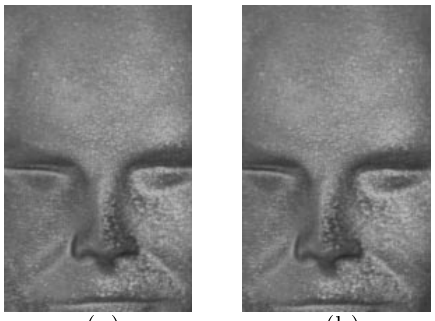


Figure 11: The optimization incorporating crest line information: a shaded view of the reconstructed surface consistent with the crest lines

- compute the maximum curvature field on the whole mesh.
- specify for each vertex a coefficient linked to the curvature value and governing the average length of the edges of the facets containing this vertex in the new mesh.
- generate a new mesh according to all these scalar values.
- optimize the new mesh using a finite element implementation of the algorithm of Section 2.
- restart the process with the result of the optimization if we want to change the levels of refinement.

Typically, we choose the scalar values governing the lengths of the new edges to be inversely proportional to the maximum curvature.
Let us notice that in this framework (like in the previous section), the mesh from which we compute the differential information has to be rather reliable so that the curvature information is meaningful. Otherwise, we might need another source of information, such as a manual specification of the crest line locations, or of the high curvature areas, to guide the reconstruction.

We show in the following figures different results obtained from a pair of images of a manufactured face model (figure 12). Figures 13,14,15 show the result of the optimization using stereo and regularization constraints only, with two different degrees of smoothing. A small amount of smoothing produces a very wrinkly surface even in the flat areas, whereas a stronger smoothing effect tends to erase the sharp areas on the orbits. Moreover, some large bumps also appear on the forehead, and the mouth is not recovered, proving that the stereo information is not important enough anymore. Figure 16 shows the best



(a) (b)

Figure 12: Left (a) and right (b) images of a face.

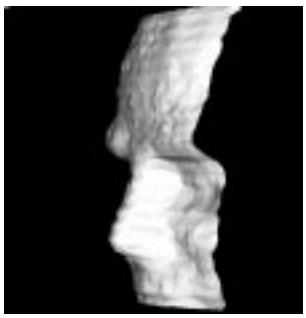


Figure 13: The reconstruction using stereo only (1)

crest line extraction we obtained on the face: the orbits are properly extracted, but we could not extract the nose(probably because the reconstruction is too noisy in this area). We thus have to rely on the curvature information. We then show the reconstruction using adaptive meshes guided by the curvature field, both before (figure 18) and after (figure 19) optimization. The optimization smooths the result but still conforms to the image data, whereas a simple gaussian smoothing (figure 20) vaguely flattens the surface and does not preserve the nose shape for example (the

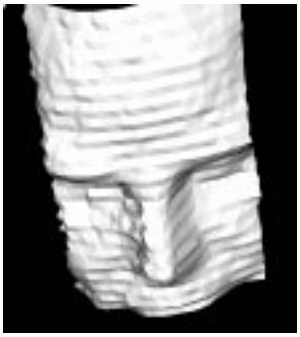


Figure 14: The reconstruction using stereo only (2)

Figure 15: The reconstruction using stereo only (3) with a high degree of smoothing

nose is too flat). Finally, we show a front view of the reconstructed face using adaptive meshes that can be compared to figure 14.

5 Conclusion

We have presented two ways of guiding the reconstruction of complex non-polyhedral surfaces with incorporating differential information in the optimization of a triangulated mesh. Differential information, whether structural in case of crest lines, or represented by a scalar field, in case of the curvature information,

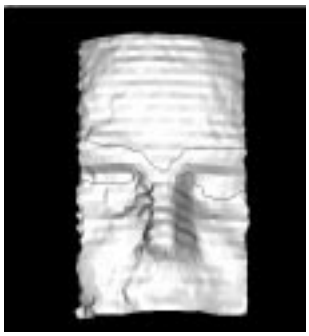


Figure 16: Crest line extraction on the face

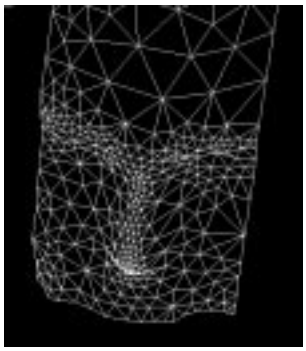


Figure 17: The reconstruction guided by differential information: the adaptive mesh

Figure 18: The reconstruction using adaptive mesh before optimization: a shaded view

Figure 19: The reconstruction using adaptive mesh after optimization: a shaded view

Figure 20: The reconstruction using adaptive mesh with no optimization but with gaussian smoothing: a shaded view

Figure 21: A front view of the face after the optimization of the adaptive mesh

is used to govern a mesh generation which is optimized using stero and regularization constraints. This method both increases the accuracy of the reconstruction around typical surface features and improves the compactness of the representation of the surface. We strongly believe that this can be a crucial step for registration, recognition or coding purposes.

References

[1] H. Borouchaki, M. J. Castro-Diaz, P. L. George, F. Hecht, and B. Mohammadi. Anisotropic adaptive mesh generation in two dimensions for cfd. In *5th International Conference on Numerical Grid in Computational Field Simulations, Mississippi State University, USA*, April 1996.

[2] I. Bricault and O. Monga. From volume medical images to quadratic surface patches. *To appear in Computer Vision and Image Understanding*, 1996.

[3] X. Chen and F. Schmitt. Intrinsic surface properties from surface triangulation. In *Proceedings, European Conference on Computer Vision, Santa Margherita Ligure, Italy*, May 1992.

[4] X. Chen and F. Schmitt. Surface modelling of range data by constrained triangulation. *Computer-Aided Design*, Vol.26, August 1994.

[5] L. D. Cohen and I. Cohen. Finite element methods for active contour models and balloons from 2-d to 3-d. In *Proceedings, Computer Vision and Pattern Recognition, Urbana Champaign, Illinois, USA*, June 1992.

[6] M. P. do Carmo. *Differential Geometry of Curves and Surfaces*. Prentice-Hall, Englewood Cliffs, 1976.

[7] P. Fua. A parallel stereo algorithm that produces dense depth maps and preserves image features. *Machine Vision Applications, 6(1)*, 1993.

[8] 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.

[9] P. Fua and Y. Leclerc. Object-centered surface reconstruction: Combining multi-image stereo and shading. *International Journal on Computer Vision*, 1995.

[10] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal on Computer Vision, 1(4): 321-331*, 1988.

[11] R. Lengagne, P. Fua, and O. Monga. Using crest lines to guide surface reconstruction from stereo. In *Proceedings, 13th International Conference on Pattern Recognition,ICPR96, Vienna*, August 1996.

[12] S. D. Ma and L.Li. Ellipsoid reconstruction from three perspective views. In *Proceedings, 13th International Conference on Pattern Recognition (ICPR96, Vienna)*, August 1996.

[13] O. Monga, N. Ayache, and P. Sander. From voxel to curvature. In *Proceedings, IEEE Conference on Computer Vision and Pattern Recognition*, June 1991.

[14] 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.

[15] W. Neuenschwander. *Elastic Deformable Contour and Surface Models for 2D and 3D Image segmentation*. PhD thesis, Swiss Federal Institute of Technology (ETH), Zürich, Switzerland, 1995.

[16] J.-P. Thirion and A. Gourdon. Computing the differential properties of isointensity surfaces. *Computer Vision and Image Understanding, 61-2, 190-202*, 1995.

[17] A. Witkin. A multi-scale approach to extract zero-crossings of the laplacian. In *Proceedings of International Joint Conference on Artificial Intelligence*, 1982.