3D Stereo Reconstruction of Human Faces driven by Differential Constraints

Richard Lengagne [a], Pascal Fua [b] and Olivier Monga [a]

[a] *INRIA Rocquencourt, Domaine de Voluceau BP105, 78153 Le Chesnay, FRANCE*

[b] *Computer Graphics Lab, EPFL, CH-1015 Lausanne, SWITZERLAND*

Keywords: stereo face reconstruction, differential geometry, deformable models, adaptive meshes, constrained optimization

Abstract

Conventional stereo algorithms often fail in accurately reconstructing a 3D object because the image data do not provide enough information about the geometry of the object. We propose a way to incorporate a priori information in a reconstruction process from a sequence of calibrated face images. A 3D mesh modeling the face is iteratively deformed in order to minimize an energy function. Differential information extracted from the object shape is used to generate an adaptive mesh. We also propose to explicitly incorporate a priori constraints related to the differential properties of the surface where the image information cannot yield an accurate shape recovery.

1 Introduction

3D face reconstruction is currently receiving a lot of attention in the Computer Vision and Computer Graphics communities. It is a thriving research field with many applications such as virtual reality, animation, face recognition, etc... In all these cases, the recovered model must be compact and accurate, especially around significant areas like the nose, the mouth, the orbits, etc... These areas can often be characterized in terms of their differential properties. Several attempts to deal with that problem have been made. In [DF94], the differential properties of the surface are inferred from a disparity map and used to modify the shape of a correlation window. In [LFM96], crest line extraction is performed on a 3D model and used to improve the reconstruction around sharp ridges. These methods improve the accuracy of the reconstruction but do not suffice if the initial 3D model is not reliable. For instance, it is well known that

bad lighting conditions or lack of texture can make correlation-based stereo fail. Consequently, the image information alone is not always sufficient to recover 3D shape. In [FB96], constraints on the depth of a given set of points on a surface mesh are applied in order to improve terrain reconstruction. In [LMF97], curvature information and structural features such as crest lines are extracted from the 3D model or interactively specified in order to generate an anisotropic surface mesh that reflects the geometric properties of the object. In this paper, we propose a further step towards incorporating a priori information in the reconstruction process from sets of sequences of calibrated face images. Differential information is used to constrain the topology of a mesh modeling the surface and the parameters of an analytical surface model, through the specification of low(high)-curvature areas, or structural features. Mathematically, this is achieved by constrained mesh optimization. We show preliminary results of this ongoing work, whose goal is to build 3D face models using entirely passive techniques.

2 The reconstruction process

2.1 An energy minimization scheme

Our reconstruction process is based on the iterative deformation of a 3D triangular mesh (i.e. a collection of vertices, triangular faces and edges) modeling the face in order to minimize an energy function E. The reconstruction process is thus treated as a snake-like process ([KWT88],[FL95],[LFM96]).
The initial mesh is computed by fitting a generic animation mesh to 3-D points derived from a correlation-based disparity map ([FM98]). It is then refined by minimizing an energy function that is the weighted sum of two terms: one stereo term E_{ext}, whose minimization makes the model fit to the image data (see [FL95] or [LFM96] for more details), and one regularization term E_{int}.
The stereo term is based upon the Lambertian assumption, which implies that the intensities of the two projections of a given 3D point onto the images are the same. If M_1 and M_2 denote the projections of a 3D point X on the left and the right image and if I_1 and I_2 denote the intensities of these projections, then the stereo term attached to X is defined as:

$$E_{ext}(X) = \frac{(I_1 - I_2)^2}{4} \tag{1}$$

Each facet is sampled into 3D points and the global stereo term is the sum of all the stereo terms attached to each point. This term can also be expressed in case of multi-image stereo. It is in fact slightly modified in order to only take into account the facets that are visible from the viewpoints of the image planes.

This term reflects the difference between these two intensities. Minimizing the stereo term will thus move the 3D point towards its right location in space. This process is based on correlation; consequently, in many well-known cases (lack of texture, lighting problems,...), it will fail in accurately recovering the 3D shape.
In order to convexify the energy function and to prevent it from being too wrinkly, we must add include a regularization term in this energy function. The regularization term is quadratic, which helps the convergence of the process. It is defined as:

$$E_{int} = \int\int (\frac{\partial^2 z}{\partial x^2})^2 + 2(\frac{\partial^2 z}{\partial x \partial y})^2 + (\frac{\partial^2 z}{\partial y^2})^2 dxdy \tag{2}$$

and expressed as $E_{int} = S^t K S$, where K is a stiffness matrix.
This optimization uses a finite-element scheme. The depth z of each surface point is expressed as a piecewise polynomial function of the two other coordinates x and y. This polynomial is of degree 5, which guarantees that the surface is piecewise C^1 (see [Neuen95], [ZT88]). The parameters of the optimization process are the depths of each vertex, as well as the 5 first and second-order partial derivatives of the depth with respect to x and y. Consequently, if the mesh is composed of n vertices, we come up with a $6n$-variable state vector:

$$S = ((z_1, ..., z_n), (\frac{\partial z_1}{\partial x}, ..., \frac{\partial z_n}{\partial x}), (\frac{\partial z_1}{\partial y}, ..., \frac{\partial z_n}{\partial y}),$$
$$(\frac{\partial^2 z_1}{\partial x^2}, ..., \frac{\partial^2 z_n}{\partial x^2}), (\frac{\partial^2 z_1}{\partial x \partial y}, ..., \frac{\partial^2 z_n}{\partial x \partial y}), (\frac{\partial^2 z_1}{\partial y^2}, ..., \frac{\partial^2 z_n}{\partial y^2})).$$

To compute the initial values of the partial derivatives, we have locally approximated the surface by a quadric and set the partial derivatives of the surface to the partial derivatives of the corresponding quadric.

The minimization of the total energy $E = \lambda_{int} E_{int} + \lambda_{ext} E_{ext}$ is then performed by differentiating E with respect to S, and embedding the system in a viscous medium, which leads to iteratively solving the dynamics equation $\frac{\partial E}{\partial S} + \alpha \frac{dS}{dt} = 0$, which can be rewritten as:

$$(K + \alpha I) S_t = \alpha S_{t-1} - \left.\frac{\partial E}{\partial S}\right|_{S_{t-1}} \tag{3}$$

where I is the identity matrix, α is the viscosity of the medium, and S_t is the state vector computed at time t.

The computation time can be very high if we keep a very large number of vertices. On the other hand, some typical features of the object must be recovered with high accuracy. Therefore, we have to reduce the number of vertices in featureless areas and to keep many points in the most significant areas of the face. Furthermore, this has to be achieved with as much automation as possible. For instance, we would like to keep many points in the nose area, the orbits, the mouth, i.e. areas which are likely to act as landmarks in an animation or a recognition process. All these areas can be characterized by geometrical properties of the surface, especially differential properties. Indeed, areas like the nose ridge or the orbits can be expressed in terms of high curvature areas or crest lines, whereas the cheeks or the forehead (where we would like a small number of facets) can be described as low curvature areas. We have thus chosen to refine the 3D model according to the differential properties of the surface that can be easily inferred from the analytical expression of the surface or estimated by a local quadric approximation.

2.2.1 The computation of curvature fields

Information about the computation of differential properties can be found in [DoCar76]. We briefly summarize the algorithm we use to compute the principal curvatures and the principal curvature directions. Here, we 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([Monga91]). The size of the neighborhood used for quadric-fitting is an important parameter of the crest line extraction algorithm, since increasing the neighborhood implies further smoothing of the surface.
In the quadric-fitting approximation, the depth 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 $\vec{v_1} = \dfrac{\partial V}{\partial x}$ and $\vec{v_2} = \dfrac{\partial V}{\partial y}$. 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 [DoCar76] for more details) 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:

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

The maximum curvature is defined as the principal curvature with the highest absolute value, and the maximum curvature direction is the principal curvature direction attached to the maximum curvature.

2.2.2 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 [Monga92]. If k_{max} and $\vec{t_{max}}$ respectively denote the maximum curvature and the maximum curvature direction, a crest point is thus defined by the equation:

$$dk_{max} = < \vec{\nabla} k_{max}, \vec{t_{max}} > = 0$$

where $< ., . >$ denotes the inner product and $\vec{\nabla}$ is the gradient operator. The extraction of the zero-crossings of dk_{max} is performed using a tracking algorithm inspired by the Marching Lines algorithm [Thir92]. Among the neighbors of vertex V, we choose the vertex V_1 that maximizes $< \overrightarrow{VV_1}, \vec{t_{max}} >$. Then, we estimate the derivative of the maximum curvature in the maximum curvature direction by finite differences, and set:

$$dk_{max}(V) = k_{max}(V_1) - k_{max}(V)$$

On each facet F of the mesh, we apply the following algorithm:

- for each vertex V of F, determine the sign of the derivative $dk_{max}(V)$.
- if, for two neighbors V_1 and V_2, $dk_{max}(V_1).dk_{max}(V_2) < 0$, there is a crest point on the edge (V_1V_2). Interpolate linearly dk_{max} along the edge (V_1V_2) and find the location of the zero-crossing of dk_{max}.
- 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 1 shows the tracking of the crest points over three facets. The + and - signs on the vertices indicate the signs of dk_{max}.

We typically threshold the result of crest line extraction according to the maximum curvature value in order to only keep significant lines. Moreover, we apply here a multiscale crest line extraction strategy, where the scale is set to the size of the neighborhood used to compute the differential properties.

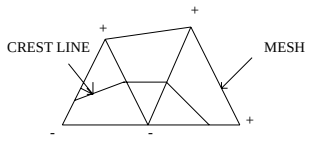

Fig. 1. The zero-crossing extraction algorithm

According to the multi-scale theory developed in [Wit82], the most significant features are detected at the coarsest scale (i.e. using the largest neighborhood, which is equivalent to further smoothing the underlying surface) and their most reliable locations are found at the finest scale (i.e. the smallest neighborhood). Moving in the scale space towards the coarsest scales implies a higher degree of smoothing of the data, thus displacing the zero-crossings from their true locations, while moving towards the finest scales can generate spurious extrema. Therefore, we detect significant crest lines at a coarse scale according to their lengths and the values of the maximum curvature along the lines, and track them down to the finest scale with finding the correspondent of each line at scale n in the set of lines at scale $n-1$.

2.2.3 The mesh generation algorithm

As described in [LMF97], we generate an adaptive mesh governed by the differential properties of the surface, that can be either the curvature fields or structural information such as crest lines.
If we generate a new mesh using curvature information, the algorithm can be summarized as follows:

- compute on the initial mesh the principal curvatures k_{max} and k_{min} and the principal curvature directions $\vec{t}_{max}$ and $\vec{t}_{min}$.
- specify for each vertex of the initial mesh the three parameters (two scalar values h_1 and h_2 and an angle θ) of an ellipse centered on the vertex which governs the generation of a new mesh.
- optimize the new mesh by minimizing the energy function $E = \lambda_{ext}E_{ext} + \lambda_{int}E_{int}$.

The algorithm completely remeshes a 2D domain (which is taken here to be a frontal projection of the face) according to the values of h_1, h_2 and θ. These values govern the local topology of the new mesh in the vicinity of the old vertex they are attached to. As shown in figure 2, the angle θ determines in which direction the new facet in the remeshed surface will be "elongated". This direction will be given by $\vec{t}_{min}$. In other terms, the edges of the new facets will be longer in the minimum curvature direction than in the maximum curvature direction (those two directions are orthogonal). This is rather intuitive: for instance, in the case of the nose ridge, the minimum curvature direction lies along this ridge. We want to capture as many details

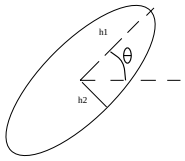

Fig. 2. The ellipse defining the local topology of the new mesh.

as possible in the direction orthogonal to this ridge, since there is a high curvature variation in that direction. Consequently, it is natural to generate longer edges in the minimum curvature direction (i.e. along the ridge) than in the maximum curvature direction (i.e. across the ridge). The scalar values h_1 and h_2 determine the average lengths of the edges in those two directions. They are decreasing functions of k_{max} and k_{min}, since we want more facets in low curvature areas. Typically, they are chosen as inverses of a second order polynomial function. h_1 is determined by the minimum curvature and h_2 is determined by the maximum curvature. This procedure uses a mesh generation software developed for the Computational Field Simulations ([BCGHM96]).

This scheme can also be used if we want to remesh the surface according to structural information such as crest lines than can be automatically detected or interactively specified. In this case, the angle θ is thus determined by the direction of the crest line, and h_1 and h_2 are fixed parameters.

Figure 3 shows a stereo pair of a face and the initial model obtained by the deformation of an animation mask. Our purpose is thus to capture more details in significant areas of the face, while preserving a reasonable number of vertices. Figure 4 shows an optimized anisotropic mesh of the face governed by curvature information. The nose ridge is well recovered since we have extracted high curvature values in this area and the principal curvature directions have oriented the facets along the ridge. However, the mouth is not very well recovered. Figure 5 shows a map of automatically extracted crest lines and the optimized mesh governed by crest line information. The crest line extraction algorithm ([LMF97]) ensures that the crest line lies inside a facet. The mesh of the face model has edges on the nose ridge. This is why this ridge has not been detected at the exact location by the crest line extraction algorithm. However, the purpose of this extraction is to specify areas of interest on the face and refine them, so we do not need a very accurate crest line extraction. In the example, we show the automatic extraction of the nose ridge, orbits, some lines on the lips and other lines that are not as intuitive but that can also describe the face geometry, such as cheek or forehead lines. In this case, the mouth is better recovered than using curvature information. We show in Figure 6 that an interactive outline of some crests can also help the reconstruction of key areas such as the orbits. We have shown examples where mesh topologies driven by curvature, automatically extracted crest lines and manually specified crest lines have been generated separately. An optimal reconstruction algorithm would merge all these kinds of information. This is

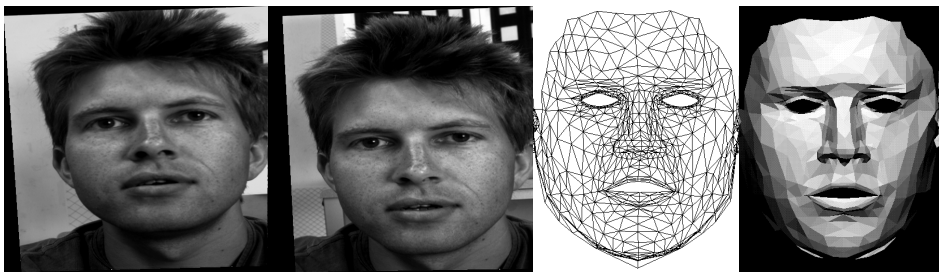

Fig. 3. A stereo pair of a face. The generic animation model fitted to correlation data: the mesh and a shaded view.

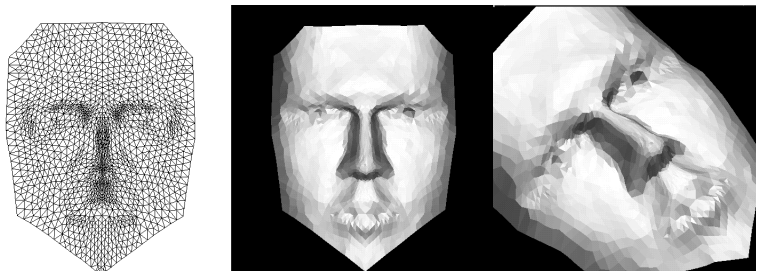

Fig. 4. The reconstructed surface using an anisotropic mesh governed by curvature information: the mesh and two shaded views.

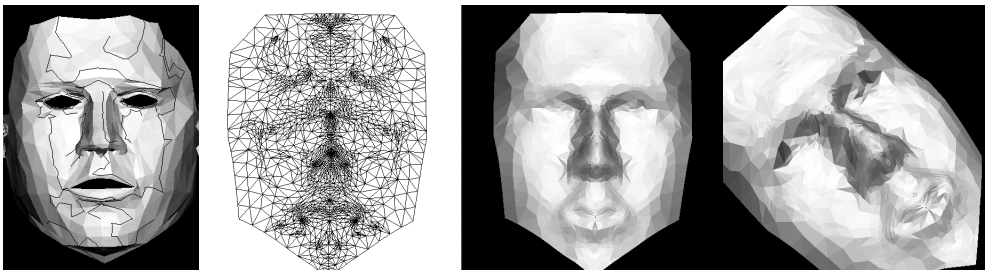

Fig. 5. The most significant crest lines automatically detected on the face model using a multi-scale extraction algorithm, and the reconstructed surface using an anisotropic mesh governed by crest line information: the mesh and two shaded views.

part of the software we are currently developing.

Governing the mesh topology by surface differential properties and running a correlation-based optimization algorithm on the adaptive mesh is sometimes not sufficient to accurately recover 3D shapes. For instance, in the above example, the shape of the eyes cannot be recovered accurately from stereo information alone because of specularity (see also figure 7). Similarly, the lips have not been reconstructed properly. Therefore, it seems necessary to incorporate in the reconstruction process extra information.

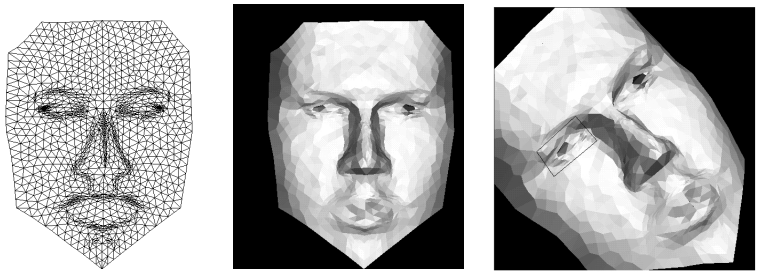

Fig. 6. The reconstructed surface using an anisotropic mesh governed by a priori knowledge about crest lines: the locations of the crest lines have been specified manually.

3 Incorporating a priori knowledge

When reconstructing an object, we have a rough idea about its shape, especially about typical features like crest lines, or about areas that can be labeled as "flat", "spherical", "cylindrical", etc... This kind of a priori knowledge can be of great interest where classical stereo methods fail. The a priori knowledge that a user can have about the shape he wants to reconstruct can be intuitive ("This region is flat, or spherical") or can rely on well-known geometric properties (anthropometric in case of face reconstruction). In any case, this a priori knowledge can very often be expressed in terms of differential properties. For instance, the knowledge "This area is flat" is obviously "translated" as:

$$\text{at each vertex, } k_{max} = k_{min} = 0$$

"This area is spherical" means:

$$\text{at each vertex, } k_{max} = k_{min}$$

We can also express "structural" knowledge such as "There is a crest line here", and interactively outline the crest on the surface (or, ideally, on the images). If we restrict our problem to the crest lines that are sets of maxima of the maximum curvature in the maximum curvature direction, the corresponding constraint can be expressed as:

$$\forall i \in \mathcal{V}, k_{max}(i) \geq k_{max}(j) \text{ and } k_{max}(i) \geq k_{max}(j')$$

where $\mathcal{V}$ is the set of vertices lying on the crest line and j and j' are surface points such that the directions respectively defined by (i, j) and (i, j') are orthogonal to the crest line.

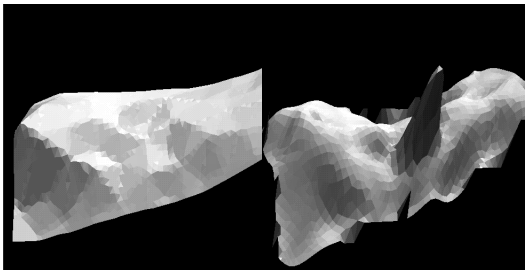

Fig. 7. The initial eye surface (left) and the reconstructed eye with the classical optimization algorithm (right). This area corresponds to the black spot in the eye in fig. 6.

Incorporating a priori knowledge in the reconstruction process can be achieved using constrained optimization, since all the constraints are expressed in terms of the partial derivatives of the surface, which are the parameters of the optimization process. We use an optimization algorithm already presented in [FB96] that decouples the projection step towards the constraint surface and the minimization of the energy function.

We show two examples where we have locally applied differential constraints to improve the reconstruction of specific facial features.

We have reconstructed one eye of the face shown in the previous section (see the small window on the right image of fig. 6), using the a priori assumption that the eye is spherical. We first constrain the topology of the mesh by manually outlining the eyelid, therefore generating more facets on the tip of the eyelid. The initial eye surface is computed by interpolation, since there is no information on the animation model in this area. We then minimize $E = \lambda_{ext}E_{ext} + \lambda_{int}E_{int}$ under the following constraints:

$$\forall i \in \mathcal{V}, k_{max}(i) = k_{min}(i) \tag{4}$$
$$\forall (i,j) \in \mathcal{V}^2, k_{max}(i) = k_{max}(j) \tag{5}$$
$$\forall (i,j) \in \mathcal{V}^2, k_{min}(i) = k_{min}(j). \tag{6}$$

where i denotes the i-th vertex and $\mathcal{V}$ the set of vertices lying on the eye surface.
We show in the results a fine isotropic mesh of the reconstructed eye surface after resampling the anisotropic mesh and using the polynomial surface approximation given by the finite element scheme. The surface has been rotated for visualization purposes, thereby inverting the signs of the curvatures.

We have also focused on lip reconstruction. Adding constraints to the optimization scheme expressing that the lips define crest lines on the face yields the final reconstruction shown in fig. 9. For this final result, we have subsampled the mesh that contains a little more than 500 vertices into a very fine triangulation containing more than 40000 vertices. The polynomial approxi-

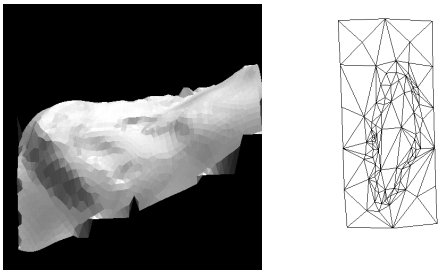

Fig. 8. The reconstructed eye after incorporating curvature-based constraints (left) and resampling the anisotropic mesh (right).

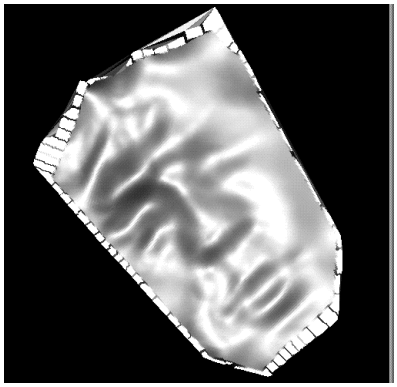

Fig. 9. The final reconstruction after incorporating crest line constraints on the lips

mation given by the finite element scheme is used to perform the subsampling. A direct optimization of such a fine mesh would have been computationally intractable. In this example, the forehead is not well reconstructed because of the presence of hair. The next step of the implementation of this algorithm is thus to incorporate a set of different constraints (involving curvatures, crest lines, or even normal orientations) applying simultaneously on different parts of the whole face. The main bottleneck of the constrained optimization scheme is the non-linearity of the constraints involving differential properties. We still have to improve this step, especially in terms of computation time (the constrained optimization of the above-mentioned mesh requires about 15 minutes on a SGI Indy workstation, whereas the unconstrained one only requires a few minutes).

The accuracy of the reconstruction should soon be measured since we intend to compare our results with the ones obtained with a 3D scanner.

4 Conclusion

We have proposed a way of palliating the lack of information extracted from stereo images during a 3D reconstruction task. We interactively reconstruct from stereo a complex 3D object like a face using a priori information about

its differential properties. We first try to design an optimal 3D mesh in terms of compacity (for computational reasons) and locations of the vertices (to capture the maximum information contained in the images) and then add differential constraints to the iterative deformation of the mesh where the image information is not sufficient. Our purpose is to develop an interactive image-based modeling software that takes into account some a priori knowledge that a user can have about the differential properties of the object to reconstruct.

References

[BCGHM96] H. Borouchaki, M. J. Castro-Diaz, P. L. George, F. Hecht and B. Mohammadi: *Anisotropic Adaptive Mesh Generation in Two Dimensions for CFD.* Proceedings of the 5th International Conference on Numerical Grid in Computational Field Simulations, 1996, Mississippi State University, USA.

[DF94] F. Devernay and O.Faugeras: *Computing Differential Properties of 3-D Shapes from Stereoscopic Images without 3-D Models.* Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), 1994, Seattle, USA.

[DoCar76] M. P. do Carmo: *Differential Geometry of Curves and Surfaces*, Prentice-Hall, Englewood Cliffs, 1976.

[FB96] P. Fua and C. Brechbuhler: *Imposing Hard Constraints on Soft Snakes.* Proceedings of the European Conference on Computer Vision (ECCV), 1996, Cambridge, U.K., II, p. 495-506.

[FL95] P. Fua and Y.G. Leclerc: *Object-Centered Surface Reconstruction: Combining Multi-image Stereo and Shading.* International Journal on Computer Vision, 1995, 16(1), p. 35-56.

[FM98] P. Fua and C. Miccio: *From Regular Images to Animated Heads: A Least Squares Approach.* Proceedings of the European Conference on Computer Vision, pages 188-202, Freiburg, Germmany, June 1998.

[KWT88] M. Kass, A. Witkin and D. Terzopoulos: Snakes : *Active Contour Models.* International Journal on Computer Vision, 1988, 1(4) p. 321-331.

[LFM96] R. Lengagne, P. Fua and O. Monga: *Using Crest Lines to Guide Surface Reconstruction From Stereo.* Proceedings of the 13th International Conference on Pattern Recognition (ICPR), 1996, Vienna, Austria.

[LMF97] R. Lengagne, O. Monga and P. Fua: *Using Differential Constraints to Reconstruct Complex Surfaces From Stereo.* Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), 1997, San Juan, Puerto Rico, USA.

[Monga91] O. Monga, N. Ayache and P. Sander: *From voxel to curvature.* Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, June 1991.

[Monga92] O. Monga and S. Benayoun: *Using Partial Derivatives of 3D Images to Extract Typical Surface Features.* Computer Vision and Image Understanding, vol. 61-2, March 1995, pp. 171-189.

[Neuen95] W. Neuenschwander: *Elastic Deformable Contour and Surface Models for 2D and 3D Image segmentation.* PhD Thesis of the Swiss Federal Institute of Technology (ETH), Zürich, Switzerland, 1995.

[Thir92] J.-P. Thirion and A. Gourdon: *Computing the Differential Properties of Isointensity Surfaces.* Computer Vision and Image Understanding, 61-2, 190-202, 1995.

[Wit82] A. Witkin: *A Multi-scale Approach to Extract Zero-crossings of the Laplacian.* Proceedings of International Joint Conference on Artificial Intelligence, 1982.

[ZT88] O.C. Zienkiewicz and R.L Taylor: *The finite element method*, Mc Graw-Hill, 1988.