Proceedings of IEEE Second International Conference on Automatic Face and Gesture Recognition (FG'96)
October 14-16 1996, Killington, USA.

From 2D Images to 3D Face Geometry

Richard Lengagne, Jean-Philippe Tarel, Olivier Monga
INRIA Rocquencourt
Domaine de Voluceau
BP 105
78153 Le Chesnay Cedex
FRANCE
Richard.Lengagne@inria.fr

Abstract

This paper presents a global scheme for 3D face reconstruction and face segmentation into a limited number of analytical patches from stereo images. From a depth map, we generate a 3D model of the face which is iteratively deformed under stereo and shape-from-shading constraints as well as differential features. This model enables us to improve the quality of the depth map, from which we perform the segmentation and the approximation of the surface.

1 Introduction

While a lot of work has already been carried out on face images [3], 3D face modeling is still very little used for recognition purposes. In this paper, we propose a global scheme which yields a description of the face surface in terms of given analytical shapes such as quadrics, or cubics. It seems to be a reasonable assumption to consider that a human face can be modeled with a good approximation by a few patches covering some significant regions that human perception would use for face recognition. The contribution of this paper is to present successive modules which produce a global segmentation of the face from stereo views, through the computation of a depth map, its optimization using a 3D mesh of the surface, and the extraction of differential features which guide us for the segmentation and the approximation process. Section 2 briefly explains how we obtain the depth map from a calibrated stereo pair. Section 3 shows how we can improve the quality of the depth map, via a 3D surface model which evolves under stereo, shape-from-shading and differential constraints. Section 4 uses the depth map to derive differential features which could guide a future segmentation of the surface. Section 5 develops the segmentation of the surface into a set of given patches, typically quadrics or cubics.

Fig. 1 briefly presents the whole process which derives a global face segmentation from a pair of images.

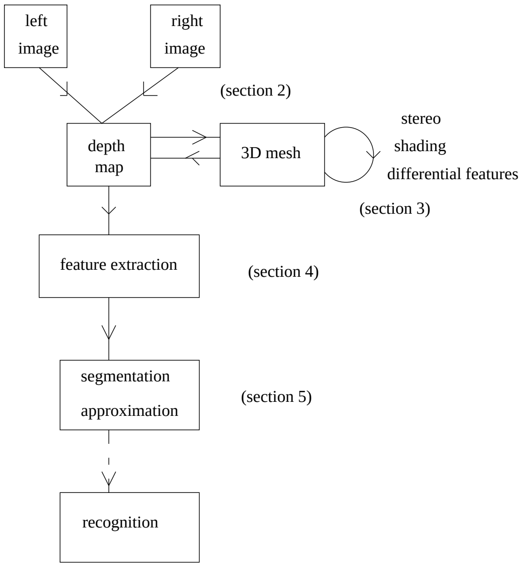

Figure 1. From a pair of images to face segmentation

2 From stereo pairs to depth maps

Our scheme starts with the acquisition of a calibrated stereo pair which gives the two perspective projection matrices M_i such that :

$$\begin{pmatrix} u_i s_i \\ v_i s_i \\ s_i \end{pmatrix} = M_i \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \qquad (1)$$

where X, Y, Z are the 3D coordinates of a point and u_i, v_i are its projections in image i.

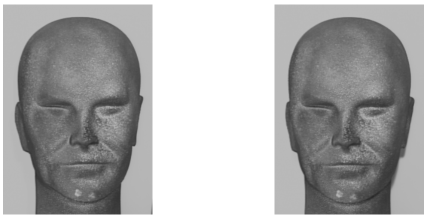

Figure 2. A stereo pair of a manufactured face

From a calibrated stereo pair of a face (fig. 2), we compute the disparity map (fig. 3) with a correlation algorithm [5] [1] that produces dense disparity maps with very few false matches which can be easily interpolated.

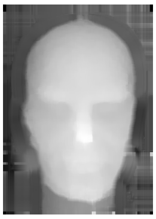

Figure 3. The depth map provided by a correlation algorithm

3 3D reconstruction with deformable models

From an organized cloud of 3D points, we can generate a triangulated mesh which covers the face. Typically, we sample our depth map so that the average distance between 2 vertices is approximately 5 pixels. For example, we are dealing with 250x350 images, and our triangulation is composed of 3500 points. The mesh will then be considered as a deformable model which can evolve under internal and external forces. Our mesh is regular and hexagonal, i.e. each vertex, except at the boundaries, has exactly six neighbors. The work of this section has been partly done in collaboration with P. Fua from SRI International, USA.

3.1 Snake-like optimization

We recover a model's shape by minimizing an objective function $\mathcal{E}(S)$ that embodies the image-based information. The variables of the objective function are the 3 coordinates of each vertex of the mesh, so $S = (X_1, X_2, ..., X_n, Y_1, Y_2, ..., Y_n, Z_1, Z_2, ..., Z_n)$.

The objective function can be expressed as the weighted sum of a stereo term and a shape-from-shading term [6]. The stereo term is most useful when dealing with highly-textured areas, whereas the shape-from-shading term is most useful in areas with constant or slowly varying albedo, which is a reasonable assumption when working on face images.

In all cases, $\mathcal{E}(S)$ typically is a highly non-convex function, and therefore difficult to optimize. However, it can effectively be minimized [7] 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 $\mathcal{E}_T = \lambda_D \mathcal{E}_D(S) + \mathcal{E}(S)$
- 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

$$\begin{aligned} K_S S_t + \alpha(S_t - S_{t-1}) &= -\left.\frac{\partial \mathcal{E}}{\partial S}\right|_{S_{t-1}} \\ \text{i.e.} \quad (K_S + \alpha I) S_t &= \alpha S_{t-1} - \left.\frac{\partial \mathcal{E}}{\partial S}\right|_{S_{t-1}} \qquad (2) \end{aligned}$$

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 [7] and is therefore more effective at propagating smoothness constraints across the surface than an explicit method such as conjugate gradient.

[1] implemented in Robotvis project, at INRIA Sophia Antipolis

In our case, λ_D has been set to a rather high value, so that the regularization term plays an important role in the optimization process. Our reconstructed surface needs to be rather smooth in order to extract crest lines and perform a global segmentation, as it will be explained in the following sections.

3.2 Crest line extraction on a triangulated mesh

We would like our mesh to be constrained by geometrical features such as orbits, or the nose ridge which are significant features in a human face, and can be useful for recognition purposes or can guide a future segmentation of the face. Our goal is thus to find a mathematical description of those features, extract them automatically on the mesh and constrain the mesh to coincide with those features. Since they usually correspond to high curvature areas, 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, because 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 describe the features in terms of crest points, previously defined as the zero-crossings of the derivative of the maximum curvature in the maximum curvature direction [9, 10].

We can attach to each point of the surface two principal curvatures and two principal curvature directions. 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 :

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

where $< .,. >$ denotes the inner product.
A crest line is the locus of these zero-crossings.
The notion of crest point uses a third order derivative of the surface $\mathcal{S}(u, v)$, and is therefore very sensitive to noise. We thus need to smooth the surface before starting any computation.

3.3 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 [8].
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 $\vec{v_1} = \dfrac{\partial V}{\partial x}$ 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 then derive the two fundamental forms Φ_1 and Φ_2 and 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:

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

Among the six 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:

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

3.4 Zero-crossing extraction

The extraction of the zero-crossings of dk_1 is performed using a tracking algorithm inspired by the Marching Lines algorithm [13]. Here, we deal with regular hexagonal triangulations. On each facet F of the mesh, we apply the following algorithm :

- for each vertex V of facet 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. They are guaranteed to be continuous, and either form a loop on the surface or cross the whole surface from one boundary to the other.

Fig. 4 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

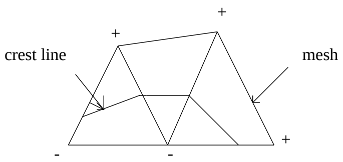

Figure 4. The zero-crossing extraction algorithm

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.

3.5 Guiding the 3D reconstruction with differential features

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 the regions with high curvature values (nose, orbits,...).

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 3.1.
- restart the process with the new mesh.

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

Fig. 5 shows a shaded view of the 3D mesh obtained from the depth map of Fig. 3, and the optimized mesh with some superimposed crest lines.

4 Extraction of differential features from depth maps

Another way of extracting differential features on the face surface is to deal with the depth map itself. This work is fully described in [11], and has been implemented by V. Prinet at INRIA-Rocquencourt. Since the intensity of each point (x, y) of the depth map is the depth $z(x, y)$ of that point, we can easily compute the partial derivatives of the surface $(x, y, z(x, y))$ with filtering techniques, and then derive the principal curvatures. This approach yields rather satisfactory results and we use it here to extract the parabolic lines, which are the loci of the zero-crossings of the Gaussian curvature $k_g = k_1 k_2$. These lines separate the areas which can be locally approximated by an ellipsoid ($k_g > 0$) from those which can be locally approximated by an hyperboloid ($k_g < 0$). This partition can be a first step towards a global approximation of the whole face surface by a set of quadrics as it will be explained in the next section.

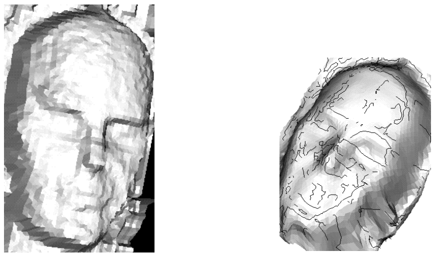

Figure 5. A shaded view of a mesh generated from the depth map, and the main crest lines superimposed on a Gouraud rendering of the final mesh

We also run the crest line extraction on the depth map. Fig. 6 shows the optimized depth map after running the process described in the previous section. Fig. 7 shows the partition of this map induced by both parabolic line and crest line extraction.

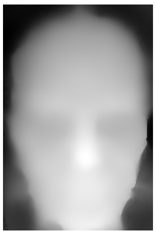

Figure 6. The final depth map obtained from the optimized mesh

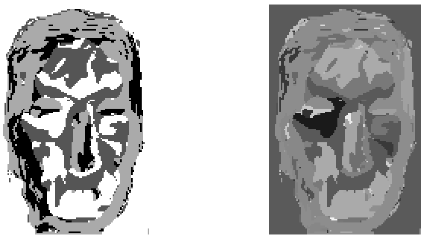

Figure 7. A partition of the face depth map according to parabolic lines and crest lines, before and after connexification

5 From depth maps to face segmentation

The segmentation in patches of a depth image provides a small description of the reconstructed data, as demonstrated by [4]. Moreover, in the particular case of a human face, the segmentation allows to select some parts of the face such as the nose, cheeks and the forehead.

The segmentation algorithm used is an extension to geometric clusters of the Fuzzy c-Means clustering method [1]. Let $I(u,v)$ be the depth image, and $\{F_k(u,v)\}_{k=1,..,c}$ a set of cluster prototypes. A cluster prototype is a function defined on the whole image, such as a plane, a quadric, or a cubic. Each prototype only fits a part of the depth image, defined as the pixel cluster. Cluster indexed by k collects all the pixels where the pixel depth is at the nearest euclidean distance of prototype $F_k(u,v)$. Then, a cluster is not a connex patch. Nevertheless, cluster prototype fitting and image partition are performed simultaneously.

Thus, our algorithm directly minimizes the depth distance between the depth data and functions $F_k(u,v)$ which divide the image in c parts. To do this in a robust way, the membership $u_k(u,v)$ of the pixel (u,v) to the cluster k is not 0 or 1, but it is a fuzzy membership between 0 and 1. Consequently, the algorithm minimizes the objective function below on the image $I(u,v)$:

$$e(F_k, u_k) = \sum_{k=1}^{c} \sum_{(u,v)} (u_k(u,v))^m (F_k(u,v) - I(u,v))^2 \tag{3}$$

with $\sum_{k=1}^{c} u_k(u,v) = 1$ and $u_k(u,v) \geq 0$.

Therefore, the segmentation algorithm is [12]:

- $\mathbf{step\ 0}$: Fix the number c of functions $F_k(u,v)$, the fuzzy exponent m and the function degree (plane, quadric, cubic). Initialize the functions F_k with the coarse segmentation obtained in the previous section.
- $\mathbf{step\ m}$: Generate a new partition of the image using the following equation of the fuzzy memberships:

$$u_k(u,v) = \frac{1}{\sum_{j=1}^{c} \left(\frac{|F_k(u,v) - I(u,v)|}{|F_j(u,v) - I(u,v)|} \right)^{\frac{2}{m-1}}} \tag{4}$$

- $\mathbf{step\ m'}$: Compute the new best F_k which minimizes the weighted mean-square error between points $(u, v, F_k(u,v))$ and $(u, v, I(u,v))$:

$$\sum_{(u,v) \in image} (u_k(u,v))^m (F_k(u,v) - I(u,v))^2$$

- $\mathbf{step\ m''}$: If the distance partition is not stable, go to step m. Otherwise each pixel is labeled with the index k of the nearest cluster prototype, and we connexify each region.

Fig. 8 and 9 successively show the segmentation into 7 quadric patches, and another segmentation into 7 cubic patches initialized with the previous segmentation. We can see that one region covers the forehead and eyebrow ridges, one covers the cheeks, another one approximates the nose and two other ones cover the two orbits.

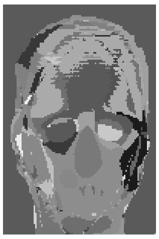

Figure 8. The segmentation of this region by a set of quadrics with the initialization given by Fig. 7

The main problem in this process is to find a trade-off between the quality of the approximation, ensured by a large number of patches, and the significance of segmentation, only ensured by a limited number of patches.

In future work, we intend to incorporate in the segmentation process itself some constraints related to the differential features, following the work already been carried out in [2]: for instance, it should be hard for two regions to be merged if they are separated by a parabolic line.

It would also be crucial to study how stable this segmentation is from a person to another ; for example, we wonder if the kind of quadric which approximates a given region of the face is rather invariant between different people.

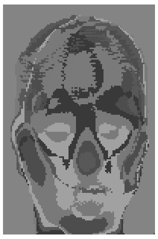

Figure 9. The segmentation of this region by a set of 7 cubics with the initialization given by Fig. 8

6 Conclusion

In this paper, we have presented a global scheme which can produce a segmentation of the human face into 3D patches from a pair of stereo images. This method uses various techniques such as correlation algorithm for 3D reconstruction, deformable model theory, feature extraction based on differential geometry and surface segmentation with geometric fuzzy clustering. All the techniques briefly described here are difficult problems in themselves and need further improvement. We think that such a scheme can be used for recognition purposes, or face modeling. For instance, a semantic description of the surface using a limited number of shapes and parameters could be of interest to match a given face to an element of a face database. On the other hand, we could think of synthesizing a full 3D model of the head using such an analytical description of the surface head, from a whole sequence of stereo images.

References

[1] J. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press, 1981.

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

[3] R. Chellappa, S. Sirohey, C. L. Wilson, and C. S. Barnes. Human and machine recognition of faces : a survey. *Technical report CAR-TR-731, CS-TR-3339, University of Maryland*, August 1994.

[4] O. Faugeras, M. Hébert, and E. Pauchon. Segmentation of range data into planar and quadratic patches. In *CVPR '83 (IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Arlington, VA)*, pages 8–13, June 1983.

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

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

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

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

[9] 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.

[10] 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.

[11] V. Prinet and O. Monga. Adaptative filtering and geometrical invariants in face's depth maps. In *Workshop of Geometrical Modelling and Invariants for Computer Vision, Xian, China*, April 1995.

[12] J. Tarel. Multi-objets interpretation. In *Proceedings, 13th International Conference on Pattern Recognition (ICPR96, Vienna)*, August 1996.

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