Generation of a 3-D Face Model from One Camera

Gérard Medioni,
University of South California
Los Angeles, CA
medioni@usc.edu

Bastien Pesenti
Geometrix, Inc.
San Jose, CA
bastienp@geometrix.com

Abstract

The generation of a fully textured 3-D model of a person's face presents difficult technical challenges, but has many applications in several fields, such as video games, immersive telepresence, and medicine. Current commercial systems rely on booth-like set-ups, equipped with laser-based scanners, or project a pattern on the subject's face.

The major drawbacks of such systems are the cost of the hardware they require, and the lack of operational flexibility. We present here a fully automatic system to generate a 3-D model from a sequence of images taken by a single camera. Unlike other methods, we do not use a generic 3-D face subject to deformation, but instead proceed in a fully bottom-up fashion.

The approach is a two-stage process. First, we estimate for each view the pose of the object with respect to the camera. This is accomplished by robust feature matching and global bundle adjustment. Then, we consider sets of adjacent views, which we treat as stereo pairs, and generate partial depth maps, which are then integrated into a single 3-D model. The texture is obtained by merging the images themselves. We describe the algorithm in detail, and show results on a number of real datasets.

1. Introduction

Many applications require the use of a 3-D face model. For instance, players in an interactive game may want to see their own face on the body of their hero. Facial animation and e-mail delivery by an avatar has also been proposed (see Eyematic [1] or LifeF/X [2]). Another example is demonstrated in the "Virtual Try-On" of eyeglasses in 3-D (See [4]).

The critical component for all these applications is the 3-D model acquisition phase. Active methods proceed by projecting laser [3], infrared, or other patterns on the face to produce very good results, but the hardware required reduces their operational flexibility. The space carving methodology [5] has emerged from the use of regular cameras in the recent past. It seems appropriate, but it requires many sensors.

Methods using two cameras only [6] have recently become robust enough to be presented at trade shows, such as Siggraph 2000. The ultimate challenge is to generate a high quality 3-D model from a *single* camera, as it opens the possibility of operation by the end-user at home, with a common webcam. This is the goal of the system we are presenting here.

As a preliminary step, the camera's internal parameters are computed by taking pictures of a known pattern, which is a standard technique. On-line implementations can be found in [7], [8].

We start by taking a sequence of images of a subject moving his/her head from one side to the other. Typically, we record 4-5 seconds at 15 frames per second.

We then match features between adjacent frames to produce an initial estimate of the motion, refine the matches with triplets of frames, and perform a step of global bundle adjustment.

In the following step, we consider sets of adjacent views, which we treat as stereo pairs, and generate partial depth maps. These are in the end integrated into a single 3-D model. The texture is obtained by merging the images themselves. The process is fully automatic, and generates a textured 3-D mesh of the face.

We start by reviewing the recent work most related to our approach in Section 2, then give an overview of the system in Section 3, with an illustrative running example. Section 4 describes the first stage, which performs pose estimation. Section 5 presents the 3-D mesh construction and integration. Section 6 shows some results, and Section 7 summarizes our contribution.

2. Related Work

Two recent research efforts stand out. In the first one, Pascal Fua built a system to reconstruct faces from video sequences [9], with an uncalibrated camera. His approach is based on a regularized bundle adjustment, and makes extensive use of a generic 3-D face model. This enables the recovery of the motion information. The final model is built by deforming the generic model.

In the second one, Zhengyou Zhang [10] has demonstrated a system to build a three dimensional model, using a single web cam to capture images. The 3-D model has further been integrated with a number of other elements, such as a text to speech animation module, to produce a complete animation-ready head.

Zhang extracts 3-D information from one stereo pair only, then deforms a generic face model. Camera poses are computed for the rest of the sequence, and used to generate a cylindrical texture.

1051-4651/02 $17.00 © 2002 IEEE

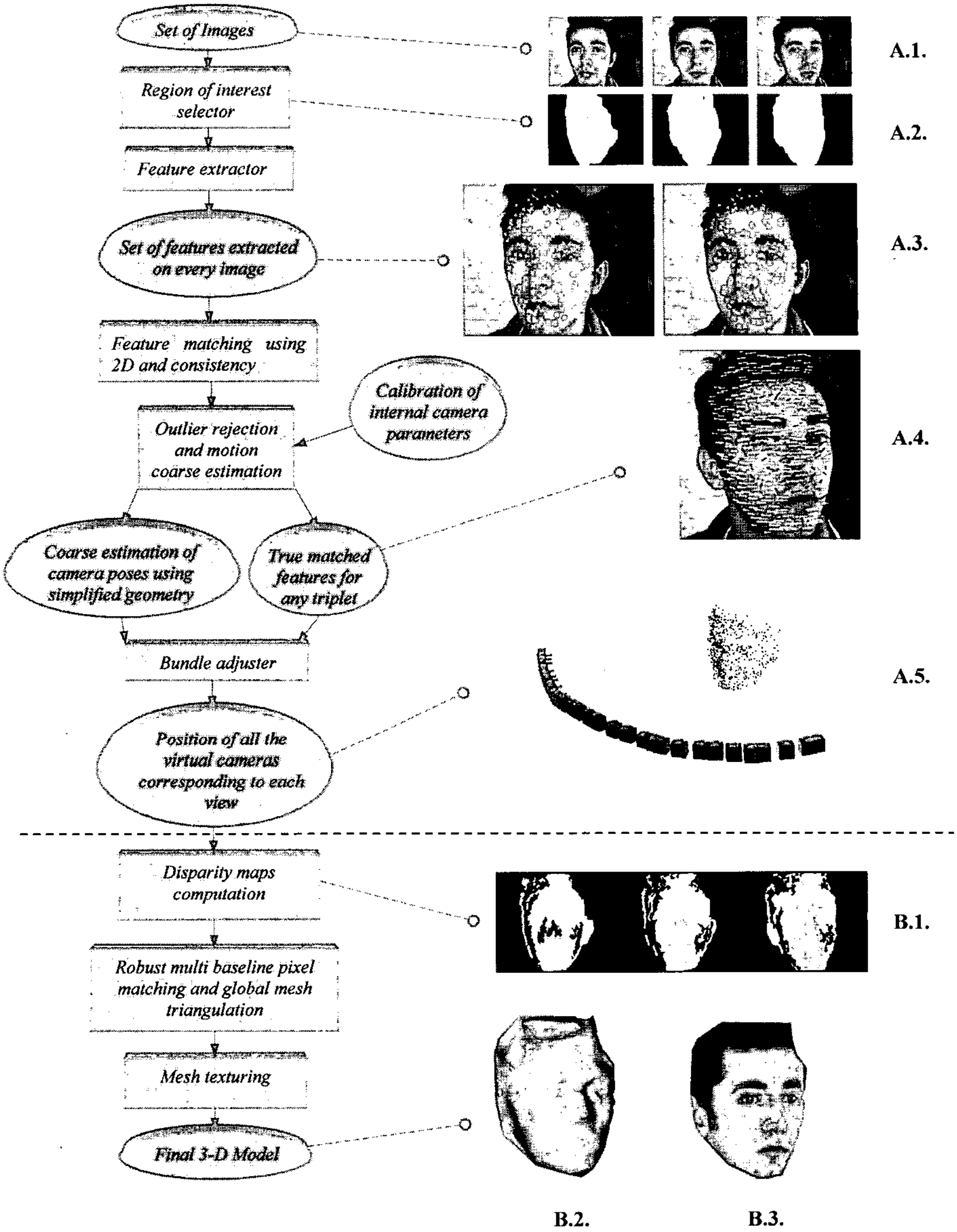
Set of Images
Region of interest selector
Feature extractor
Set of features extracted on every image
Feature matching using 2D and consistency
Calibration of internal camera parameters
Outlier rejection and motion coarse estimation
Coarse estimation of camera poses using simplified geometry
True matched features for any triplet
Bundle adjuster
Position of all the virtual cameras corresponding to each view
Disparity maps computation
Robust multi baseline pixel matching and global mesh triangulation
Mesh texturing
Final 3-D Model
A.1.
A.2.
A.3.
A.4.
A.5.
B.1.
B.2.
B.3.

Figure 1: Flow chart

These approaches both use a generic face in order to guide the reconstruction. This gives stability and speed to the numerical algorithms. The resulting mesh however, is a compromise between the generic face model and the real face. Furthermore, the generic face cannot accurately capture the hair portion of the head, account for a moustache, or an abundant beard.

3. System Overview

Once the images have been captured and the camera's internal parameters are known, the method proceeds as illustrated in the flowchart on Figure 1. In the first stage, we recover the relative position of the cameras to the subject's face. This is accomplished by identifying features on the face throughout the sequence, and adjusting the camera position using the re-projection error of these identified points. In the second stage, we build a three-dimensional mesh of the face, knowing the positions of the cameras. In the following two sections, they are described in detail.

4. Camera Pose Estimation

For convenience, we reformulate the problem by considering the face fixed, and the camera moving. We recover the motion of a virtual camera for each image of the sequence by computing a position matrix [R | T] .

This is a classical structure from motion problem. We solve it using feature extraction and matching. 2d3 addresses a similar problem in Boujou [11], which is a tool that recovers camera poses from images taken with a calibrated camera. It is feature based, and uses a robust computation of the trilinear tensor, as described in [14] to recover the motion of the camera. Unlike our tool, this method does not make any assumptions on the type of movement of the camera. We are using a simplified model of motion, more adapted to our problem (See Section 4.2).

4.1. Feature extraction and matching

The first step of the process is to capture a sequence of images of a face moving from side to side. (See A.1. on Figure 1). A wide angle of rotation ensures a good coverage of the face.

The face is segmented from the background using image differencing. The computed border does not need to be precise for the reconstruction to succeed. (See A.2. in Figure 1). This step reduces image processing operations outside the region of interest.

Once the area of interest is defined, a corner extractor is applied on the image portion of the face [12]. The goal of this operation is to identify features to be matched (See A.3. in Figure 1). We use an adaptive threshold inside the region of interest, in order to generate a somewhat uniform spatial distribution of features, leading to a more accurate pose estimation.

The matcher is correlation-based, and also checks for local neighborhood consistency. Many incorrect matches are likely to remain at this stage. They need to be handled in the subsequent steps.

4.2. Outlier Rejection

This critical step in the workflow computes a spatial system of camera poses and feature points accurate enough to be used as an initial starting point for the non-linear bundle adjuster. The algorithm takes overlapping triplets of matched features in successive images as input, and computes a robust estimate of a rigid motion. This robust estimation is performed with RANSAC (RANdom SAmple Consensus) [13].

To gain speed and stability, we assume a simplified model of rigid motion. For a triplet of images, the head is rotating around a fixed center, whose projection in the image should fall within the region of interest.

We use a special decomposition of the rotation transform proposed by Konderink and van Doorn in [14], particularly adapted to our problem. We decompose the rotation motion of the camera around the head into two basic rotations. First, a rotation around the camera principal axis and second, a rotation around an axis parallel to the image plane. Three angles, determining completely the rotation are estimated. θ is the angle of the rotation around the principal axis of the camera. φ represents the attitude of the complementary rotation axis, which is parallel to the image plane of the camera, and ψ represents the amount of rotation around this axis. The rotation matrix of the camera can then be expressed as defined in [14] $R = R_{\varphi,\Psi} \times R_{\vartheta}$.

The motion is estimated using a combination of RANSAC and non-linear least squares minimization. The convergence criterion is the back projection error of tracked features. This back projection error happens to be most sensitive to variations in θ, φ and then ψ. We therefore decompose the estimation of the rotation in two major steps. First, θ and φ are roughly estimated using two image frames only. ψ is estimated as a second pass, while θ and φ are refined as well. However, the impact of a variation about ψ on the back projection error is small. For this reason, it is necessary to work on a triplet of frames. We thus can use a cross back projection error, estimating the impact of the motion between frame one and two as a back projection error on frame three. As a drawback, two different motions have to be estimated in the same time.

Each pair of motions is estimated using the previous one as a starting point. It therefore assumes continuity of movement.

4.3. Bundle Adjustment

The coarse estimated system is fed to a non-linear bundle adjuster [15], which is a classical photogrammetry package. (See [16])

This bundle adjuster uses the filtered feature matches (See A.4. in Figure 1), which no longer contain outliers, in order to precisely position the cameras in space. The roughly estimated camera poses computed in the previous step are entered as the starting point of this non-linear estimation.

The bundle adjustment then iterates until it reaches a pre-set precision, which is a function of the back projection error for the 3D features. It does not make any assumption about a simplified model of motion, unlike what was done in the previous steps. It will therefore precisely position the cameras (See A.5. in Figure 1). Convergence to the correct solution depends mostly on the fact that all outlier matches are rejected in the previous step.

Tests on a turntable sequence have proven the camera pose accurate within 0.3 degrees.

5. Model Reconstruction

This is the second stage of the workflow. It computes a dense and complete 3-D model of the face from the images and the camera poses.

5.1. Dense Stereo on Selected Pairs

For each pair of successive selected camera poses, a dense stereo map is computed. The computation is performed on the image portion of the face only, since the background has no motion relative to the camera.

First, the images of a pair are rectified so their respective epipolar lines are parallel to each other, and correspond to scan lines. This is straight forward as we know the relative position of the two cameras. A disparity is then assigned to each pixel, describing the position of its corresponding feature in the other image. The computation of the dense disparity map proceeds by a low-resolution correlation search, followed by a multilevel propagation and refinement. Fully detailed explanation of theory and the algorithm is available in [6].

Points closer to the camera appear darker on the disparity map (See B.1. in Figure 1).

Tests on a dummy face for which we have ground truth values, show that two-view stereos produce an average error of 0.8mm given an angular basis of 10 degrees.

5.2. Dense 3-D Point Cloud Computation

In this step, corresponding points are tracked throughout the sequence, using dense stereo information, and triangulated using all the views they appear in. For robustness, consistency checks are performed, and correspondences that do not pass the test are rejected. The complete set of triangulated points gives a cloud of 3-D points, representing the face surface.

Due to the inherent noise in the disparity maps, exact consistency is hard to enforce when computing the cloud of points.

The computation of the cloud of points is followed by an additional consistency check. All 3-D points are re-projected on the original images. The points that do not project inside the regions of interest are deleted. This helps removing points that accidentally pass the motion consistency check.

5.3. Mesh Construction

Converting a cloud of points into a mesh is known to be a difficult problem, especially when dealing with surfaces containing holes or concavities. Fortunately, the human head is almost a star-shaped solid. That is to say, if a line is drawn from the center of the head, it will intersect once and only once the surface. This is not exactly true, but the small concavities in the ear or below the nose can be ignored considering the level of detail we seek. Even better, the face can be approximated as a solid that can be projected on a cylinder (see Figure 2).

Therefore, the cloud of 3-D points can be mapped onto a rectangular domain. The Z coordinate represents the distance from the surface to the axis of the cylinder. Using this property, the domain is partitioned in buckets; thickness is dealt with using a median filter inside the buckets. Finally, a mesh is obtained by using a 2-D Delaunay triangulation algorithm.

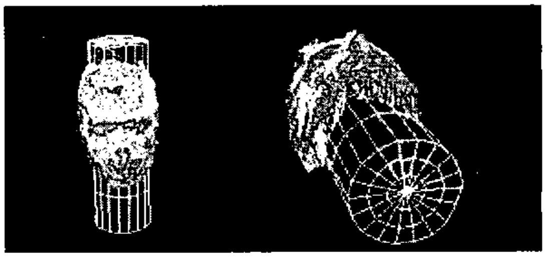

Figure 2: Cylinder

The mesh is then post processed using standard decimation and smoothing algorithms (See B.2. in Figure 1). It is textured with the original images, projected onto the mesh. (See B.3. in Figure 1).

The eyes of a person always move independently from the head during capture, so we cannot simply aggregate the texture images. Instead, we select the texture map around the eyes area from a single image only, typically the front view.

6 Results

We show on Figures 4 and 5 some results on real image sequences that have been computed using the technique presented in this paper.

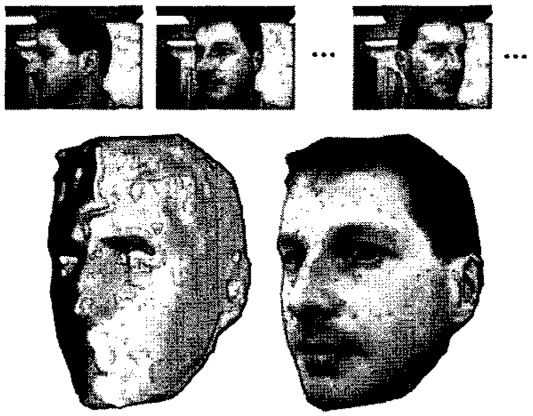

Figure 4: Mike's face

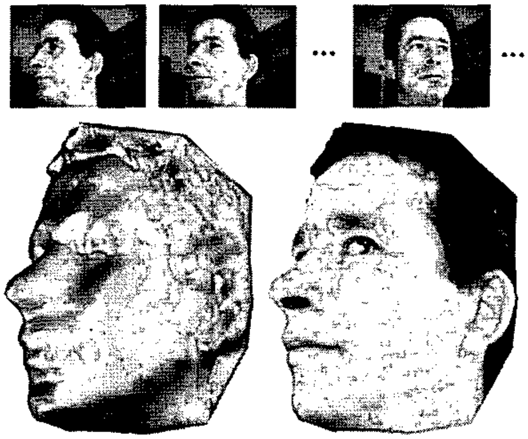

Figure 5: Arthur's face

7. Summary and Discussion

We have built a system to generate a 3-D model from a sequence of images of a face. This is achieved through the use of image processing, structure from motion, stereo matching, and graphics tools. The system has been tested on a large number of subjects, and gives good results in about 15 minutes of computation time on a standard PC. The face has precise biometric accuracy, and the visual quality is good.

More tests need to be conducted on lower quality cameras, as well as hostile lighting conditions, to reach the goal of having the system ready for users at home.

References

[1] Eyematic "Eyematic Expression", www.eyematic.com/products_xpression.html, Inglewood, CA.

[2] LifeFX, "Life FX Stand In Technology" www.lifefx.com/FaceOfTheInternet/product.html, Boston, MA.

[3] CyberWare, "Head & Face Color 3D Scanner Bundle" Monterey, CA. - www.cyberware.com

[4] Geometrix Inc, "FaceVision" www.geometrix.com/Facevision, San Jose, CA.

[5] N. Kutulakos, S. Seitz, "A Theory of Shape by Space Carving", *IJCV*, (38), No. 3, pp. 197-216, July 2000.

[6] G. Medioni, Q. Chen, "Building 3-D Human Face Models from Two Photographs", *The Journal of VLSI Signal Processing*, Kluwer Academic Publisher, pp 127-140, 2001.

[7] Z. Zhang, "A flexible new technique for camera calibration", *IEEE Transactions on PAMI*, 22(11), pp1330-1334, 2000.

[8] OpenCV, "The Open Source Computer Vision Library." - www.intel.com/research/mrl/research/opencv/

[9] P. Fua. "Regularized Bundle-Adjustment to Model Heads from Image Sequences without Calibration Data", *International Journal of Computer Vision*, 38(2), pp153-171, July 2000.

[10] Z. Liu, Z. Zhang, C. Jacobs, M. Cohen, "Rapid modeling of animated faces from video", *Journal of Visualization and Computer Animation*, 12(4), pp227-240, 2001.

[11] Boujou - www.2d3.com/2d3/products/boujou.shtml

[12] L. Shapiro, R. Haralick. "Image Matching – An Interest Operator", *Computer and Robot Vision Volume II* pp341-343, October 1992, Prentice Hall.

[13] M. Fischler and R. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image", *Communications of the ACM*, pp381-395, June 1981.

[14] J.J. Koenderink, A.J. van Doorn, "Affine Structure From motion", J. Optical Soc. Am. pp377-385. 1991.

[15] B. Triggs, P. McLauchlan, Richard Hartley and Andrew Fitzgibbon. "Bundle Adjustment – A Modern Synthesis", *Workshop on Vision Algorithms*, pp298-372, 1999.

[16] Photo Modeler Pro – www.photomodeler.com . EOS Systems Inc.