Rapid Modeling of Animated Faces From Video Images

[DEMO SUMMARY] *

Zicheng Liu, Zhengyou Zhang, Chuck Jacobs, Michael Cohen
Microsoft Research, One Microsoft Way, Redmond, WA 98052, USA
zliu@microsoft.com, zhang@microsoft.com

ABSTRACT

Generating realistic 3D human face models and facial animations has been a persistent challenge in computer graphics. We have developed a system that constructs textured 3D face models from videos with minimal user interaction. Our system takes images and video sequences of a face with an ordinary video camera. After five manual clicks on two images to tell the system where the eye corners, nose top and mouth corners are, the system automatically generates a realistic looking 3D human head model and the constructed model can be animated immediately. A user, with a PC and an ordinary camera, can use our system to generate his/her face model in a few minutes. We will demonstrate the system at the conference.

Keywords

facial animation, geometric modeling, computer vision

1. INTRODUCTION

One of the most interesting and difficult problems in computer graphics is the effortless generation of realistic looking, animated human face models. Animated face models are essential to computer games, film making, online chat, virtual presence, video conferencing, etc. So far, the most popular commercially available tools have utilized laser scanners. Not only are these scanners expensive, the data are usually quite noisy, requiring hand touchup and manual registration prior to animating the model. Because inexpensive computers and cameras are widely available, there is great interest in producing face models directly from images. In spite of progress toward this goal, the available techniques are either manually intensive or computationally expensive.

The goal of our system is to allow an untrained user with a PC and an ordinary camera to create and instantly animate his/her face model in no more than a few minutes. The user interface for the process comprises three simple steps. First, the user is instructed to pose for two still images. The user is then instructed to turn his/her head horizontally, first in one direction and then in the other. Third, the user is instructed to identify a few key points in the images. Then the system computes the 3D face geometry from the two images, and tracks the video sequences to create a complete facial texture map by blending frames of the sequence. The key observation is that even though it is difficult to extract dense 3D facial geometry from two images, it is possible to match a sparse set of corners and use them to compute head motion and the 3D locations of these corner points. We can then fit a linear class of human face geometries to this sparse set of reconstructed corners to generate the complete face geometry. In this paper, we show that linear classes of face geometries can be used to effectively fit/interpolate a sparse set of 3D reconstructed points. This novel technique is the key to quickly generating photorealistic 3D face models with minimal user intervention.

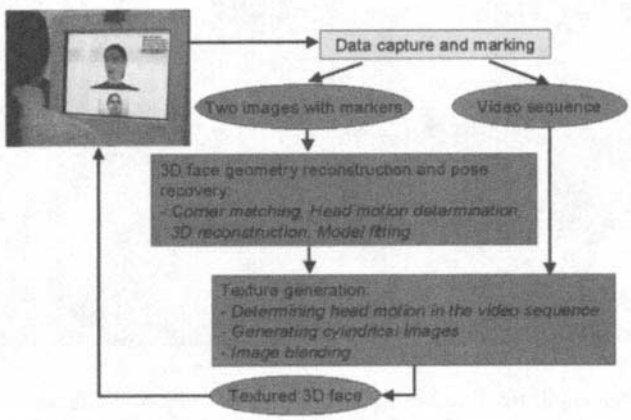

Figure 1: System overview

2. SYSTEM OVERVIEW

Figure 1 outlines the components of our system. The equipment include a computer and a video camera. We assume the intrinsic camera parameters have been calibrated, a reasonable assumption given the simplicity of calibration procedures [1].

The first stage is data capture. The user takes two images

*A full version of this paper is available as *Microsoft Research Report* MSR-TR-2000-11 at `www.research.microsoft.com/~zhang/Papers/TR00-11.pdf`

Figure 2: Side by side comparison of the original images with the reconstructed models of various people.

with a small relative head motion, and two video sequences: one with head turning to each side. Or alternatively, the user can simply turn his/her head from left all the way to the right, or vice versa. In that case, the user needs to select two approximately frontal views with head motion of 5 to 10 degrees, and edit the video into two sequences. In the sequel, we call the two images the *base images*.

The user then locates 5 markers in each of the two base images. The 5 markers correspond to the two inner eye corners, nose top, and two mouth corners.

The next processing stage computes the face mesh geometry and the head pose with respect to the camera frame using the two base images and markers as input. This is done through the following steps:

- Preprocessing: compute automatically a mask image to locate the approximate area of head motion.
- Feature matching and motion determination: a robust technique based on least-median-squares is used to match points of interest and simultaneously determine the head motion across images [2].
- 3D reconstruction: Matched points are reconstructed in 3D space.
- Fitting: the so-called *metrics*, i.e., the parameters which define the face mesh geometry, are estimated through fitting face mesh to 3D reconstructed points and also silhouettes.

The final stage determines the head motions in the video sequences, and blends the images to generate a facial texture map.

3. RESULTS

We have used our system to construct face models for various people. Figure 2 shows side-by-side comparisons of seven reconstructed models with the real images. The accompanying video shows the animations of these models. In all these examples, the video sequences were taken using ordinary video camera in people's offices. No special lighting equipment or background was used. After data-capture and marking, the computations take between 1 and 2 minutes to generate the synthetic textured head. Most of this time is spent tracking the video sequences.

For people with hair on the sides or the front of the face, our system will sometimes pick up corner points on the hair and treat them as points on the face. The reconstructed model may be affected by them. For example, the female in Figure ?? has hair on her forehead above her eyebrows. Our system treats the points on the hair as normal points on the face, thus the forehead of the reconstructed model is higher than the real forehead.

In the animations shown in the accompanying video[1], we have automatically cut out the eye regions and inserted separate geometries for the eye balls. We scale and translate a generic eyeball model. In some cases, the eye textures are modified manually by scaling the color channels of a real eye image to match the face skin colors. We plan to automate this last step shortly.

4. REFERENCES

[1] Z. Zhang. Flexible camera calibration by viewing a plane from unknown orientations. In *ICCV'99*, pp.666–673, 1999.

[2] Z. Zhang, R. Deriche, O. Faugeras, and Q.-T. Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence Journal*, 78:87–119, Oct. 1995.

[1] A 5-minutes video is available at `ftp://ftp.research.microsoft.com/Users/zhang/FaceModeling.mpg`