Model-based Face Capture from Orthogonal Images

Douglas R. Lanman
California Institute of Technology
Caltech MSC 649, Pasadena, CA 91126
dlanman@caltech.edu

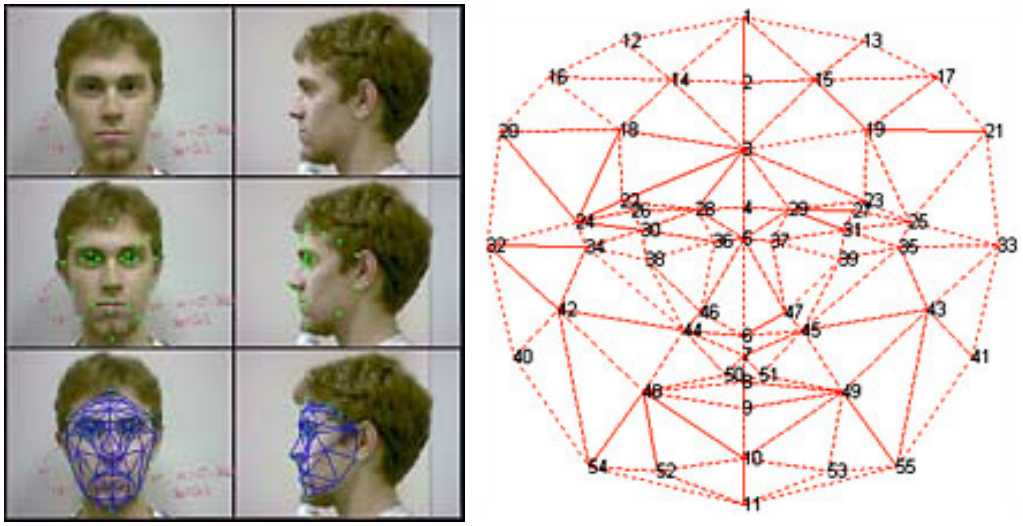
From a survey of techniques, a novel method was implemented for creating photo-realistic 3D facial models from 2D images of a subject. Starting from two orthogonal views, a user-assisted procedure was employed to recover 3D coordinates of a sparse set of chosen locations on the subject's face. A scattered data interpolation algorithm was then used to deform a generic face mesh to fit the particular geometry of the subject's face. A model texture map was created by combining both views. Using this technique, it is possible to generate highly realistic facial models. These models can be used for diverse applications including low-bandwidth video conferencing, computer animation, and multiple-view face recognition.

1 Introduction

Accurate mimicry of the human face, including both structure and expression, is a significant challenge. The fine details of the face, such as wrinkles and subtle color and texture variations, are essential for recognizing identity and interpreting expressions [5]. These features are difficult to imitate using traditional computer-based modeling and animating techniques. As an alternative to artificial synthesis of human faces, actual faces can be sampled to obtain geometric models and to control expression animation. While some facial modeling attempts utilize expensive calibrated 3D scanning equipment, others only require a simple PC camera. These photogrammetric techniques obtain an estimate of facial geometry from a series of camera images. By sampling actual human faces, photogrammetry provides realistic models, while reducing the cost inherent to computer-based modeling.

2 Methods

Human faces are similar in both structure and shape; as a result, a model deformation process can be combined with photogrammetric techniques to obtain a best-fit model for a given individual. The model deformation process requires certain features, corresponding to vertices in the generic model, to be identified. In this implementation the correspondences are manually selected by the user, however certain feature recognition algorithms could be applied for automated model creation. Once the model is deformed, a view-independent texture map is synthesized from the original image set to produce a complete representation.



(a) Modeling process (b) Generic model

Figure 1: Model deformation

2.1 Model Deformation

In order to deform a generic model using photogrammetric techniques, a set of images must be obtained that provide sufficient information to locate the relative displacement of certain facial features in three dimensions. One alternative, as proposed by Pighin et al., is to apply computer vision techniques to estimate viewing parameters (position, orientation, and focal length) for a series of cameras positioned around the subject [5]. This approach utilizes a linear least squares algorithm to provide a maximum likelihood estimate of the camera parameters, however initial estimates of intrinsic and extrinsic parameters are required [5]. As an alternative, two orthogonal images can be used to obtain an estimate of facial geometry [3]. This simplification removes the necessity of providing initial estimates of any camera parameters; instead, minimal calibration requirements are introduced: images must be separated by approximately 90° and camera focal lengths are assumed to be similar. While reducing the accuracy of the final model, the orthogonal images simplification limits the number of cameras required for image capture. Images can be obtained one at a time using a single camera or two cameras can be used simultaneously for increased model accuracy.

Figure 1(a) shows a set of typical input images as they are modified through the fitting process. The projection of a generic model onto the front image plane is shown in Figure 1(b). To begin the model fitting process, the user must select a set of feature vertices in the generic model. These vertices typically correspond to prominent components of the subject's face, including the outline, the eyes, the nose, and the mouth. After a vertex set is selected from the generic model, the corresponding positions are selected in the input images, as has been done in the second image pair in Figure 1(a). Having selected certain feature points, displacement vectors are computed between selected points and corresponding model vertices. Given the displacement vectors, a smooth interpolation function must

be constructed to provide displacement vectors for both feature and non-feature vertices in the generic model, allowing the model to be adapted to fit each image.

For this implementation, an interpolation function consisting of radial basis functions was used; this method is similar to that proposed by Pighin et al., although here the model deformation process occurs in two dimensions rather than three [5]. Given the set of user-selected features $\vec{p_i}$ and the corresponding set of model vertices $\vec{p_i}^0$, a generic interpolation function can be expressed as

$$f(\vec{p}) = \sum_{i=1}^{N} \vec{c_i} \phi(\| \vec{p} - \vec{p_i}^0 \|). \tag{1}$$

Appropriate model deformation can be demonstrated if the weighting function ϕ takes the form of a radial basis function:

$$\phi_{rb}(r;\lambda) = e^{-\frac{r}{\lambda}}, \text{ for } \lambda > 0. \tag{2}$$

The set of constant vectors $\vec{c_i}$ in Equation 1 can be determined by assuming that the interpolation function should map model vertices to corresponding user-selected features:

$$\text{choose } f(\vec{p_i}^0) = \vec{u_i} = \sum_{j=1}^{N} \vec{c_j} \phi_{rb}(\| \vec{p_i}^0 - \vec{p_j}^0 \|;\lambda). \tag{3}$$

In two dimensions, Equation 3 has the matrix formulation

$$\begin{pmatrix} u_{x,1} & \cdots & u_{x,N} \\ u_{y,1} & \cdots & u_{y,N} \end{pmatrix} = \begin{pmatrix} c_{x,1} & \cdots & c_{x,N} \\ c_{y,1} & \cdots & c_{y,N} \end{pmatrix} \begin{pmatrix} \phi_{rb}(\| \vec{p_1}^0 - \vec{p_1}^0 \|;\lambda) & \cdots & \phi_{rb}(\| \vec{p_1}^0 - \vec{p_N}^0 \|;\lambda) \\ \vdots & \ddots & \vdots \\ \phi_{rb}(\| \vec{p_N}^0 - \vec{p_1}^0 \|;\lambda) & \cdots & \phi_{rb}(\| \vec{p_N}^0 - \vec{p_N}^0 \|;\lambda) \end{pmatrix}. \tag{4}$$

The set of constant vectors $\vec{c_i}$ can be evaluated by multiplying Equation 4 by the right inverse of the radial basis function matrix:

$$\begin{pmatrix} c_{x,1} & \cdots & c_{x,N} \\ c_{y,1} & \cdots & c_{y,N} \end{pmatrix} = \begin{pmatrix} u_{x,1} & \cdots & u_{x,N} \\ u_{y,1} & \cdots & u_{y,N} \end{pmatrix} \begin{pmatrix} \phi_{rb}(\| \vec{p_1}^0 - \vec{p_1}^0 \|;\lambda) & \cdots & \phi_{rb}(\| \vec{p_1}^0 - \vec{p_N}^0 \|;\lambda) \\ \vdots & \ddots & \vdots \\ \phi_{rb}(\| \vec{p_N}^0 - \vec{p_1}^0 \|;\lambda) & \cdots & \phi_{rb}(\| \vec{p_N}^0 - \vec{p_N}^0 \|;\lambda) \end{pmatrix}^{-1}. \tag{5}$$

Having determined the form of the interpolation function, the generic model can be deformed to fit the projection of the subject's face in each image plane. Results of the fitting process are shown in the final image pair in Figure 1(a).

The preceding calculations provide two sets of 2D coordinates for each vertex in the model: $(x_f, y_f)_i$ for the front view and $(y_s, z_s)_i$ for the side view. To complete the structural modeling process, the vertex coordinates for each image plane can be combined into a single estimate of the subject's 3D facial structure according to Equation 6.

$$(x, y, z) = (x_f, \frac{y_f + y_s}{2}, z_s) \tag{6}$$

2.2 Texture Mapping

The model deformation process produces an estimate of the 3D structure of the subject's face. To produce a complete representation, it is necessary to create a texture map that can be applied to the fitted mesh. Typically, the 3D model is projected into two dimensions in some manner so that a texture map can be created that accurately represents the subject. For instance, by projecting the deformed model onto each image plane, a set of texture coordinates can be determined for each picture. These texture coordinates can be used in conjunction with each image to define a pair of texture maps. This process fails, however, to produce an accurate representation; if the front image is selected as the texture map, then features toward the side of the face will be distorted. Similarly, if the side image is selected, then features near the front of the face will be under-sampled due to the small projected area these regions have in the side image.

An automated texture mapping algorithm was designed for this implementation. As discussed, to create a texture map a set of texture coordinates in two dimensions must be determined by projecting the 3D fitted model. In the literature, it is common to map a human face onto the surface of a cylinder [4, 5]. The cylindrical projection equalizes the area of each facial region in the texture map, reducing the under-sampling that occurs due to small projected areas in the input image pair.

In order to calculate the cylindrical projection, the origin of the cylinder and its radius must be determined. Consider a coordinate system in which the $\hat{z}$-axis is directed outward from the nose and the $\hat{y}$-axis is oriented along the axis of the cylinder and toward the top of the head. If a complete head model is being constructed, then the origin of coordinates should be centered at the average of the model vertices. For a facial model, the origin should be selected so that it is behind all model vertices; this ensures that the projected coordinates lie on the front half of the cylinder. Once the origin is determined, the radius R of the cylinder is computed by determining the minimum bounding box for the projection of the model vertices in the $x-z$ plane. Given the set of deformed model vertices $(x, y, z)_i$ the texture coordinates $(s, t)_i$ are given by

$$(s, t)_i = (R \tan^{-1}\left(\frac{x_i}{z_i}\right), y_i) \tag{7}$$

With the texture coordinates determined, pixel colors must be assigned to the texture map to create a final model. The goal is to sample pixels from the two original images and assign them to the final texture map in a manner that maximizes the resolution of facial details; intuitively, pixels near the side of the face should be extracted from the profile image, whereas pixels near the front of the face should be transfered from the front image.

To begin the texture mapping process, a sampling rate for the texture map, expressed in pixels, should be selected. The texturing algorithm will then progress through each triangle and assign the color of all enclosed pixels; as a result, this implementation describes a texture mapping algorithm that scales linearly with the number of triangles in the mesh.

For a given triangle in the fitted model, the texture mapping algorithm begins by computing the surface normal and the angle θ_{normal} of its projection in the $x-z$ plane. Next, a minimum bounding box is found for the texture coordinates of the triangle. Recall that the vertices (p_0, p_1, p_2) of a triangle, expressing in texture coordinates (s, t), can be used as

an affine basis; the affine coordinates $(\lambda_0, \lambda_1, \lambda_2)$ for each pixel in the bounding box can be computed using Equation 8.

$$\begin{pmatrix} \lambda_0 \\ \lambda_1 \\ \lambda_2 \end{pmatrix} = \begin{pmatrix} p_{0_s} & p_{1_s} & p_{2_s} \\ p_{0_t} & p_{1_t} & p_{2_t} \\ 1 & 1 & 1 \end{pmatrix}^{-1} \begin{pmatrix} s_{pixel} \\ t_{pixel} \\ 1 \end{pmatrix} \tag{8}$$

If the affine coordinates satisfy $0 < (\lambda_0, \lambda_1, \lambda_2) < 1$, then the triangle encloses the pixel. In order to combine the color values for the corresponding pixels in the input images, the texture coordinates for the front and side images must be determined. The texture coordinates for the front image $(s_f, t_f)_i$ and for the side image $(s_s, t_s)_i$ are obtained, as discussed, by projecting the deformed model into each image plane. Because the affine coordinates of a pixel are invariant for these projections, the affine coordinates of a pixel in the final texture map can be used to determine the corresponding pixels in the front and side images.

Equation 9 can be applied to assign the pixel colors in the final texture map.

$$\begin{aligned} color_{texture\ map}(s_{pixel}, t_{pixel}) = \cos^2(\theta_{normal})\ color_{front} \left(\begin{pmatrix} \lambda_0 \\ \lambda_1 \\ \lambda_2 \end{pmatrix}^T \begin{pmatrix} p_{0_{s_f}} & p_{0_{t_f}} \\ p_{1_{s_f}} & p_{1_{t_f}} \\ p_{2_{s_f}} & p_{2_{t_f}} \end{pmatrix} \right) + \\ sin^2(\theta_{normal})\ color_{side} \left(\begin{pmatrix} \lambda_0 \\ \lambda_1 \\ \lambda_2 \end{pmatrix}^T \begin{pmatrix} p_{0_{s_s}} & p_{0_{t_s}} \\ p_{1_{s_s}} & p_{1_{t_s}} \\ p_{2_{s_s}} & p_{2_{t_s}} \end{pmatrix} \right) \end{aligned} \tag{9}$$

Equation 9 effectively combines the orthogonal images to produce a cylindrical texture map. The front and side image colors are weighted by $\sin^2(\theta_{normal})$ and $\cos^2(\theta_{normal})$, respectively; this ensures that regions of the face that are predominately facing a certain image plane will utilize the detailed texture information available from that image; for example, the side of the face and the nose will be predominately textured using the profile image, reducing the distortion present in previously discussed methods.

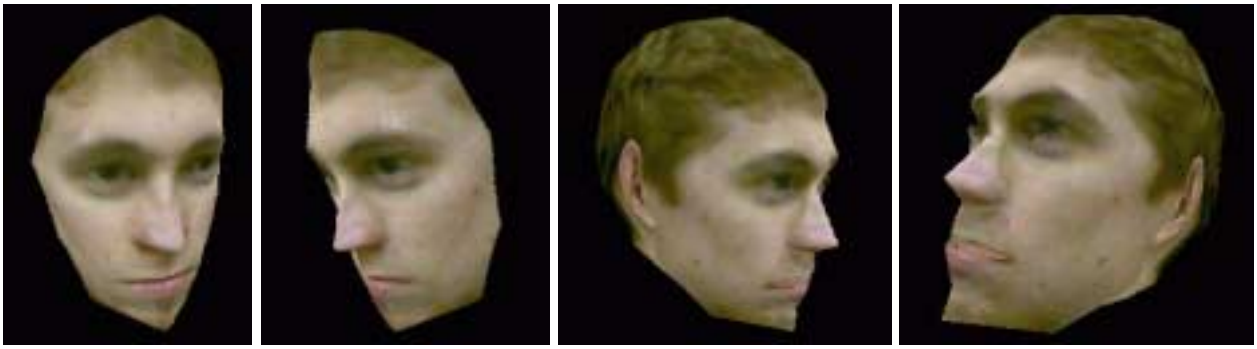
3 Results

The modeling process described in the previous section produces accurate models with a minimum of user interaction. Figure 2 shows a texture map created using a full head model and the input images in Figure 1(a). As discussed, the cylindrical texture map smoothly combines the input images and preserves the details visible in each. Figure 3 shows several views of models created using the deformation and texture mapping algorithms. Note that the model appears realistic for viewing conditions different than the the input images, one of the key tests for any photogrammetric method.

4 Discussion

It is important to note that the orthogonal image simplification translates to near-symmetric features in the fitted models and texture maps. In order to improve the realism of the facial

Figure 2: Cylindrical texture map



(a) (b) (c) (d)

Figure 3: Model views

models, three orthogonal images should be used, taken from the front, left, and right sides of the face.

An additional limitation of this method involves the problems inherent to the cylindrical representation. For full head models certain features are difficult to represent in the final texture map. For example, the region behind the ear is occluded. One possible remedy for this problem is to apply smoothing algorithms, such as Laplacian smoothing, to the texture coordinates to ensure that no triangles overlap in the final texture map [6]. Full head models introduce additional occlusion problems, including the region inside the mouth, which many not be completely visible in the image set. Several authors have resolved this difficulty by constructing separate models for problematic regions, such as the eyes and mouth, before capturing facial features [5]. The limitations of the cylindrical representation are not restricted to occlusion problems; certain features do not map onto the surface of the cylinder for a full head model; for example, the top of the head and the bottom of the chin do not project onto the surface of the cylinder. Additional techniques are required to include texture elements for these regions.

5 Conclusion

Model deformation represents a characteristic approach in photogrammetry; objects which have similar structures can be effectively modeled using a fitting algorithm. Generic models can be created from scratch and refined by averaging over many fitted models. As demonstrated, photogrammetric methods can be applied to generate models of human faces. Full head models have also been creating by modifying the generic model applied to the input images. Deformation methods have also been used by Debevec et al. to model architecture from a series of images [2]. In applications where it is difficult or expensive to obtain accurate estimates of the 3D structure of an object directly, a subset of features can be located and a default model can be adapted to fit the surface in a least squares sense. For example, medical imaging applications include modeling of internal organs from computed tomography (CT) images [1].

Robust methods for the creation of facial texture maps remains an unsolved problem. Recent research has focused on cylindrical texture maps [4, 5]. The automated texture mapping algorithm presented in this paper scales linearly with the number of triangles and can be easily implemented on a standard PC, with minimal computation time required to generate the final texture map. In addition, the proposed method allows the resolution, determined by the sampling rate, to be continuo-sly varied, producing models for a variety of applications including low-bandwidth video conferencing, computer animation, and multiple-view face recognition.

References

[1] J. L. Boes, T. E. Weymouth, and C. R. Meyer. Multiple organ definition in computed tomography using a bayesian approach for 3d model fitting. In *Proceedings of Vision Geometry IV*, 1995.

[2] P. E. Debevec, C. J. Taylor, and J. Malik. Modeling and rendering architechture from photographs. In *Proceedings of Siggraph '96*, 1996.

[3] H. Ip and L. Yin. Constructing a 3d individualized head model from two orthogonal views. *The Visual Computer*, pages 254–266, 1996.

[4] Z. Liu, Z. Zhang, C. Jacobs, and M. Cohen. Rapid modeling of animated faces from video. Technical Report MSR-TR-2000-11, Microsoft Research, 2000.

[5] F. Pighin, J. Hecker, D. Lischinski, R. Szeliski, and D. Salesin. Synthesizing realistic facial expressions from photographs. In *Proceedings of Siggraph '98*, 1998.

[6] G. Taubin. Geometric signal processing on polygonal meshes. In *Eurographics 2000 State of the Art Report (STAR)*, 2000.