

Coding of Facial Image Sequences by Model-Based Optical Flow

Malcolm Davis

Texas Instruments
8330 LBJ Freeway, MS 8374
Dallas, Texas 75243
malcolm.davis@ti.com

Mihran Tuceryan

Indiana Univ./Purdue Univ. Indianapolis
Dept. of Computer and Information Science
Indianapolis, Indiana 46202-5132
mtucerya@cs.iupui.edu

ABSTRACT

A model-based method for estimating the shape and motion of 3D objects appearing in a video is described. This technique is used for model-based video coding (video compression). The method is based on a new variant of optical flow and uses 3D computer graphics to represent and display an object. Though the algorithm is general, this work concentrates on videos depicting the human head and face because of its relevance to videotelephony and teleconferencing. Rigid body motion of the head and facial expressions (opening the mouth) are accommodated. Results obtained from videos of a moving person are described.

1. INTRODUCTION

The concept behind model-based video coding (video compression) is that models of 3D objects and their motion require less information to transmit than videos of those objects. As illustrated in Figure 1, this type of coder analyzes the video to obtain values for the parameters of these models and estimates of the 3D motion of modeled objects. These parameter values and motion estimates are transmitted and a video display of the modeled objects and their motion is synthesized using 3D computer graphics.

This work concentrates on model-based coding of heads and faces because such techniques

are most relevant for teleconferencing and visual communication. Inspired by the work of Waters [2] and Tang [3] a model of the human head and face has been developed which uses an approximate model of facial musculature to animate facial expressions. A model-based formulation of optical flow, derived from the work of DeCarlo and Metaxas [4, 5], provides estimates of the motion of the head model.

1.1. Model-Based Motion Estimation

In 3D model-based video coding, the 3D motion of the object depicted in the video must be determined. In a few methods for recognition, tracking, or coding of facial image sequences, optical flow is mapped directly onto the parameters of a 3D model in order to determine the motion of the modeled object. Most methods are specific to a particular motion model (e.g., rigid motion) and an assumed object model (e.g., a triangular mesh). It is possible, however, to modify the optical flow implementation (which usually assumes that the motion is planar) so that virtually any motion and object model can be accommodated in a regular and *automatic* manner [4, 5].

Parametric Representations:

First, consider a 3D object, such as a human face that appears in a video. This object can be represented by a 3D vector function, $\vec{s}(\vec{u}, \vec{q}_s) = [s_1(\vec{u}, \vec{q}_s) \quad s_2(\vec{u}, \vec{q}_s) \quad s_3(\vec{u}, \vec{q}_s)]^T$, which associates each value of the vector \vec{u} with a point on the surface of the object. The column vector, \vec{q}_s , contains values that control the shape of the object. The function $\vec{s}(\vec{u}, \vec{q}_s)$ is called a *parametric representation* of the object and the elements of \vec{u} are the domain of this representation.

As example, suppose that $\vec{s}(\vec{u}, \vec{q}_s)$ is a generic model of an average human face (instead of an ellipsoid). In this case, \vec{q}_s might contain param-

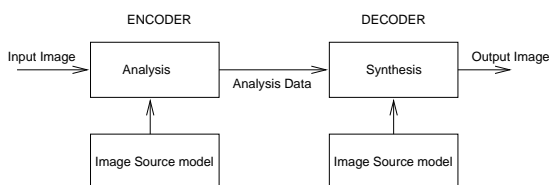


Figure 1: A block diagram of a model-based video coding system (from [1]).

eters which would (1) adapt the shape of this generic model to the shape of a individual’s face and (2) deform the face appropriately for facial expressions such as smiles, frowns, and raising the eyebrows.

Coordinate Transformations: The motion of a 3D object can be represented by a *coordinate transformation* which changes over time. A coordinate transformation maps (or transforms) each 3D coordinate location to another coordinate location. Examples of coordinate transformations include rotation, translation, perspective (projection), and deformations such as scaling, bending, and twisting. A coordinate transformation is represented as a vector function, $\vec{y} = \vec{r}(\vec{s}, \vec{q}_r)$, which transforms (maps) the coordinate location, \vec{s} , into a new location, \vec{y} . The transformation has parameters values (e.g., rotation angles), which comprise the elements of the vector, \vec{q}_r .

Model-Based Optical Flow: The well-known planar formulation of optical flow is based on the gradient constraint equation:

$$\nabla I(\vec{x}, t)^T \dot{\vec{x}} + I_t(\vec{x}, t) = 0$$

where $\nabla I(\vec{x}, t)$ is the gradient of $I(\vec{x}, t)$ with respect to the image coordinates \vec{x} and $I_t(\vec{x}, t) = \frac{\partial I(\vec{x}, t)}{\partial t}$. This equation can be extended to encompass arbitrary motions, with the result [4]:

$$\nabla I(\vec{x}, t)^T \mathbf{L}(\vec{u}, \vec{q}) \dot{\vec{q}} + I_t(\vec{x}, t) = 0. \quad (1)$$

where $\mathbf{L}(\vec{u}, \vec{q})$ is the *Jacobian matrix* of the coordinate transformation from object coordinates to camera coordinates, including animation or deformation of the object, with respect to the parameters of the transformation and the model, $\vec{q} = [\vec{q}_r^T \quad \vec{q}_s^T]^T$. This matrix, $\mathbf{L}(\vec{u}, \vec{q})$, is used to transform partial derivatives of \vec{q} into partial derivatives of \vec{x} : $\dot{\vec{x}} = \mathbf{L}(\vec{u}, \vec{q}) \dot{\vec{q}}$.

Equation (1) represents the fundamental principle for estimating 3D motion from optical flow. As in the typical application of optical flow, values for the spatial and temporal derivatives of the image, $\nabla I(\vec{x}, t)$ and $I_t(\vec{x}, t)$, are obtained using derivative filters (with Gaussian kernels). It is assumed that the Jacobian matrix and the transformation are known in advance, e.g., the object is a face and the transformation is a combination of rotation, translation, facial expressions, and perspective (projection). Only $\dot{\vec{q}}$ remains undetermined.

Least squares is used to solve for $\dot{\vec{q}}$ from the spatial and temporal derivatives of the image at a set of points, $\vec{x}_i, i = 1, 2, 3, \dots, N$. The resulting value for $\dot{\vec{q}}$ is the 3D estimate of the motion

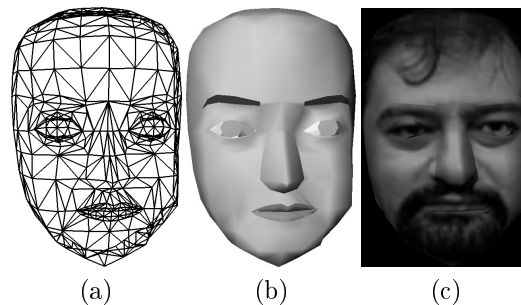


Figure 2: A computer graphics representation of a face as a 3D triangular mesh drawn: (a) as a wireframe (each line is an edge of a triangle); (b) as solid shapes with shading added; (c) with texture mapping overlaid.

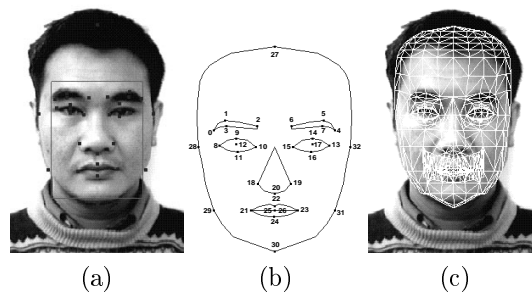


Figure 3: Customization of a generic face model to conform to a particular individual: (a) a set of facial features are detected; (b) the corresponding location of these features on the generic face model; (c) the generic face model is warped (deformed) to bring the two sets of features into approximate alignment.

occurring in the video at time t , expressed as the rate of change in the object position parameters, \vec{q} . The object (e.g., a face) position parameters, \vec{q} , at time t can be determined by numerically integrating $\dot{\vec{q}}$. Presently, this algorithm makes use of the Euler method.

The Face Model: The 3D computer graphics representation of the face is a 3D triangular mesh with texture mapping. The model is depicted in Figure 2. This 3D model of a typical human head and face is customized to fit the shape of the individual depicted in the video as illustrated in Figure 3. The jaw can rotate, i.e., the mouth can open and close. By a method analogous to that of Waters [2], major muscles of the face are approximated by a set of “actuators” that can contract and relax. These actuators are anchored to fixed locations (bone) at one end and, at the other end, are attached to vertices of the triangular mesh (skin) through a simulated flexible medium. By appropriate ac-

tivation of groups of “muscles” the face can be made to smile, frown, raise an eyebrow, and so on.

The Initial Pose: The position (pose) of the face in the first frame of the video is needed to initialize the algorithm. This initial pose is determined by using an optimization algorithm (gradient descent) to minimize the mean squared error between feature locations on the actual face and the modeled face.

Model-Based Coding: In model-based video coding, the values of the parameters defining the current face shape and orientation, \vec{q} (or \vec{q}), are encoded and transmitted for each frame in the video. Other information, such as the customization of the shape of the head model for the individual depicted in the video, is transmitted only at the beginning of communication.

2. RESULTS

The motion estimation algorithm described in this paper has been applied to video sequences depicting the head and shoulders. The motion of the head appearing in one video is tracked and used to create a second video depicting the computer-generated face model as it follows the motions in the first video. An example of rigid motion estimation is illustrated in Figure 4. In Figure 4(a), 5 frames extracted from a 100 frame video of M.T. are shown. The same frames from the computer-generated video are displayed in Figure 4(b). The fact that the video is computer generated is more apparent in Figure 4(c), where texture mapping has been disabled. A unique feature of this type of video coding is the ability for the person to appear differently at the receiver (decoder) than he does at the transmitter (encoder). This feature is illustrated in Figure 4(d) where a graphics model of S.K.’s head moves in synchronization with the video of M.T. The tracking of more complex motion is illustrated in Figure 5 which depicts several frames extracted from a video of M.D. as he turns his head and simultaneously opens his mouth.

It has been indicated that about 68 parameters are needed to encode facial expressions and head motion. Using this value, the estimated transmission (baud) rate of the video sequences in Figure 4 and Figure 5 is about 6,800 bits/sec (10 bits/parameter \times 68 parameters/frame \times 10 frames/sec) *without any encoding of the parameter values*. Applying a coding scheme, like arithmetic coding, to the parameter values would significantly reduce even this low rate.

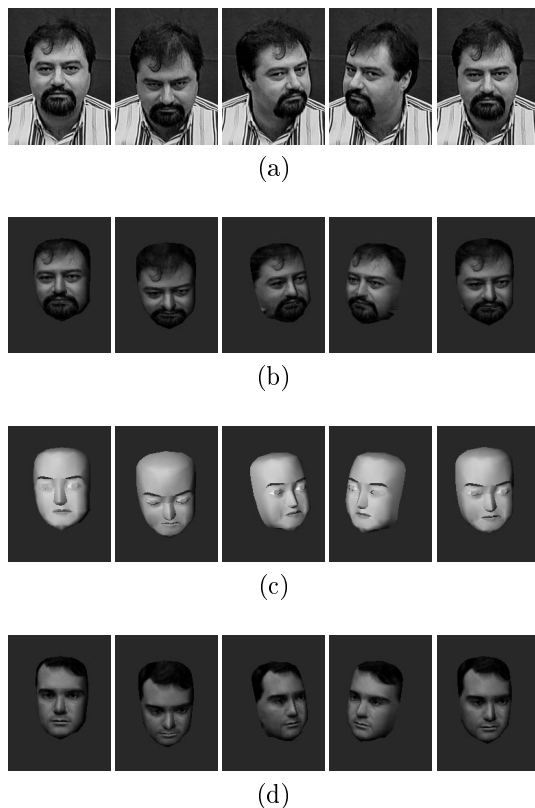


Figure 4: Model-based motion estimation of M.T.: (a) frames extracted from a video of M.T.; (b) the same five frames from a video where the computer-generated head image follows the motion of M.T.’s head; (c) the video of (b) with the texture map removed; (d) a computer-generated video where S.K.’s head follows the motion of M.T.’s head.

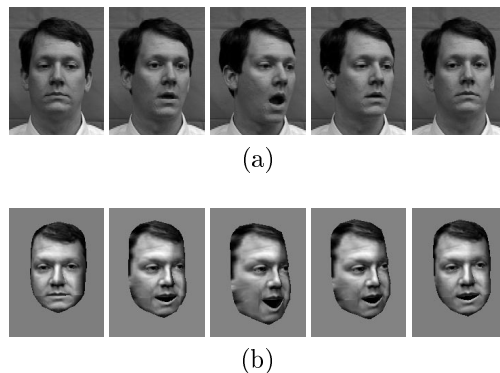


Figure 5: Model-based motion estimation of M.D.: (a) frames extracted from a video of M.D.; (b) the same five frames from a video where the computer-generated head image follows the motion of M.D.’s head.

Acknowledgement

The authors are grateful to Scott King for his contributions to the development of the face model and several handy software tools. Doug DeCarlo's frank discussions of his research are appreciated. Bruce Flinchbaugh proofread the manuscript.

3. REFERENCES

- [1] K. Aizawa and T. S. Huang, "Model-based image coding: Advanced video coding techniques for very low bit-rate applications," *Proceedings of the IEEE*, vol. 83, pp. 259–271, Feb. 1995.
- [2] K. Waters, "A muscle model for animating three-dimensional facial expression," *Computer Graphics*, vol. 21, pp. 17–24, July 1987.
- [3] L.-A. Tang, *Human Face Modeling, Analysis, and Synthesis*. PhD thesis, Electrical Engineering Department, University of Illinois at Urbana-Champaign, Urbana, Illinois, 1996.
- [4] D. DeCarlo and D. Metaxas, "The integration of optical flow and deformable models with applications to human face shape and motion estimation," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, (San Francisco, CA), pp. 231–237, IEEE Computer Society Press, June 18–20, 1996.
- [5] D. Metaxas and D. DeCarlo, "Deformable model-based face shape and motion estimation," in *Proceedings of the Second International Conference on Automatic Face and Gesture Recognition*, (Killington, VT), pp. 146–150, IEEE Computer Society Press, Oct. 14–16, 1996.