View Morphing based Facial Video Compression *

Shoma Chatterjee Subhashis Banerjee
K.K. Biswas
Department of Computer Science and Engineering
Indian Institute of Technology
New Delhi 110016
email: {*shoma,suban,kkb*}*@cse.iitd.ernet.in*

Abstract

In this paper we present a view morphing based technique for compression of facial video sequences. The facial image is triangulated and the vertices of the triangles serve as control points to be tracked across frames. We borrow the idea of view morphing [1] used in computer graphics and in an initial pre-interpolation stage we rectify the control points of the index frames by projecting them on to parallel image planes. It turns out that the control points of the intermediate frames can be reconstructed in the rectified domain, as linear combinations of control points in the index frames, if the optical centers of the intermediate views lie on the straight line joining the optical centers of the two end index frames. In a post-interpolation stage we de-rectify the control points back to the original configuration and carry out an intensity warping of each reconstructed triangle at the decoder. We account for the local regions, mainly eyes and lips, by interpolating from the index frame templates using an optic flow based procedure.

1 Introduction

In this paper we investigate the possibility of using view morphing [1] for low bit-rate compression of facial video sequences. In an earlier work [2] we presented an affine structure based facial video compression algorithm where the face is segmented and triangulated and the vertices of the triangles serve as control points which are tracked across frames and their affine structure is computed. The affine structure [3] of the control points remain invariant across frames and need to be transmitted to the decoder only once at the bootstrapping stage. Subsequently, we need to only transmit the affine bases corresponding to each frame which represent the affine projection matrices. The control points are reconstructed at the decoder by projecting the affine structure using the basis for each frame and for each triangular region the intensity profile is reconstructed by warping from the corresponding triangle in an index frame using a 2D local affine transformation for each triangle. The remaining non-affine motion is captured using predictive encoding of error images in a standard way.

Significantly higher compression can be achieved if instead of transmitting the basis for each frame to the decoder, the basis information of the intermediate frames can be reconstructed from those of the index frame, possibly using linear interpolation. In the affine case this is straightforward, at least in principle, because any image can be expressed as a linear combination of two views [4]. In this paper, we extend the video compression algorithm [2] to a projective model and derive methods to linearly approximate the image projection of control points from the two end frames of a group of frames.

We borrow the idea of view morphing [1] used in computer graphics and in an initial pre-interpolation stage we rectify the control points of the index frames by projecting them on to a common image plane. It turns out that the control points of the intermediate frames can be exactly reconstructed in the rectified domain, as linear combinations of control points in the index frames, if the optical centers of the intermediate views lie on the straight line joining the optical centers of the two end index frames. Thus, for small global motions between index frames, we can approximately reconstruct the control points in the rectified domain for each intermediate frame as linear combinations of the control points in the end frames. In a post-interpolation stage we rectify the control points back to the original configuration and carry out an intensity warping of each reconstructed triangle at the

*Shoma Chatterjee was supported by a C.S.I.R Research associateship during the course of this work. This work was supported by an Indo-Israel Scientific collaboration project funded by DST.

decoder. We account for the local regions, mainly eyes and lips, by interpolating from the index frame templates using an optic flow based procedure.

A key advantage in using a projective model of image formation, as opposed to an affine model, is that the model tends to be valid for a larger sequence window. Consequently, we can use a larger group of frames in each window set and use fewer index frames. No predictive encoding is required in this method and we obtain significantly higher compressions.

The paper is organized as follows. In Section 2 we briefly describe the theory of view morphing. In Section 3 we describe our scheme of applying view morphing to video compression. In Section 4 we describe the local reconstruction algorithm. In Section 5 we present some initial results of compression using view morphing. Finally, in Section 6, we conclude the paper.

2 View morphing

In what follows, we briefly describe view morphing [1]. Consider two projective cameras $\mathbf{\Pi}_1 = [\mathbf{I} \mid \mathbf{0}]$ and $\mathbf{\Pi}_2 = [\mathbf{I} \mid \mathbf{t}]$ with camera centers at $\mathbf{C}_1 = [\mathbf{0}, 1]^T$ and $\mathbf{C}_2 = [-\mathbf{t}, 1]^T$ respectively. Consider an arbitrary world point $\mathbf{P}$ and its two projections $\mathbf{p}_1 = \mathbf{\Pi}_1\mathbf{P}$ and $\mathbf{p}_2 = \mathbf{\Pi}_2\mathbf{P}$. A linear interpolation of the two points $\mathbf{p}_1$ and $\mathbf{p}_2$ can be expressed as

$$\begin{aligned} \mathbf{p}_s &= (1-s)\mathbf{p}_1 + s\mathbf{p}_2 && (1)\\ &= (1-s)\mathbf{\Pi}_1\mathbf{P} + s\mathbf{\Pi}_2\mathbf{P} && (2)\\ &= ((1-s)\mathbf{\Pi}_1 + s\mathbf{\Pi}_2)\mathbf{P} && (3)\\ &= [\mathbf{I} \mid s\mathbf{t}]\,\mathbf{P} && (4) \end{aligned}$$

Thus, there exists a physically valid camera $\mathbf{\Pi}_s = [\mathbf{I} \mid s\mathbf{t}]$ which gives the interpolated point $\mathbf{p}_s$ as the image of the world point $\mathbf{P}$. Moreover $\mathbf{\Pi}_s$ is a linear interpolation of $\mathbf{\Pi}_1$ and $\mathbf{\Pi}_2$, and its optical center $\mathbf{C}_s = [-s\mathbf{t}, 1]^T$ is also given by the linear combination of $\mathbf{C}_1$ and $\mathbf{C}_2$.

Conversely, given two end views with parallel image planes, any intermediate view on the same image plane can be linearly interpolated from the end views provided that the optical centers lie on a straight line.

2.1 Non-parallel views

In case the two views are non-parallel, the first camera can still be chosen as $\mathbf{\Pi}_1 = [\mathbf{I} \mid \mathbf{0}]$. A choice for $\mathbf{\Pi}_2$, consistent with the epipolar geometry between the two index frames, can be given as

$$\mathbf{\Pi}_2 = [\mathbf{M} \mid \mathbf{e}_2] \qquad (5)$$

where $\mathbf{M}$ represents a homography between the two views though an arbitrary plane in space and $\mathbf{e}_2$ represents the epipole in the second index frame. Such a choice is always possible and is consistent with the fundamental matrix $\mathbf{F}$ between the two views, and we have that $\mathbf{F} = [\mathbf{e}_2]_\times \mathbf{M}$ where $[\mathbf{e}_2]_\times$ is a skew-symmetric matrix representing a cross product with $\mathbf{e}_2$ [5].

Applying $\mathbf{M}^{-1}$ to $\mathbf{\Pi}_2$, we obtain a rectified camera

$$\hat{\mathbf{\Pi}}_2 = \mathbf{M}^{-1}\mathbf{\Pi}_2 = \left[\mathbf{I} \mid \mathbf{M}^{-1}\mathbf{e}_2\right]$$

Note that $\mathbf{\Pi}_2$ and the rectified camera $\hat{\mathbf{\Pi}}_2$ have the same optical center $\mathbf{C}_2 = \mathbf{M}^{-1}\mathbf{e}_2 = -\mathbf{M}^{-1}\mathbf{e}_2$. Hence, after applying the rectification homography $\mathbf{M}^{-1}$ to the points in the second image the two images become parallel and we can generate intermediate parallel views corresponding to a linear motion of the camera by linear interpolation. [1] suggests a further rectification step of aligning the epipolar lines in the two images, however it is not strictly necessary.

Further, for any interpolated view, a suitable non-parallel view corresponding to any camera with the same optical center can be generated by applying an inverse homography [1].

For affine cameras, the last row of the 3×4 camera projection matrix is [0 0 0 1] indicating that the optical center is a point at infinity. Consequently, linear interpolation of end frames will always yield physically valid views.

In case we have a static camera and motion of the object then the situation is equivalent. Consider a world point $\mathbf{P}$ in one view being transformed to a point $\mathbf{TP}$ in another view where $\mathbf{T}$ is a homography in 3-space. This is equivalent to static point $\mathbf{P}$ being viewed with a camera $[\mathbf{I} \mid \mathbf{0}]$ and another camera $[\mathbf{I} \mid \mathbf{0}]\,\mathbf{T}$ and the above analysis still applies.

3 Video compression using view morphing

For facial video compression we first fit a triangular mesh on to the face image and track the vertices of the triangles using a Kalman filter based face tracking algorithm [2, 6, 7].

Let us consider a group of frames $\{\mathbf{V}_1, \mathbf{V}_2, \ldots, \mathbf{V}_n\}$ where $\mathbf{V}_1$ and $\mathbf{V}_n$ are the two index (end) frames and $\mathbf{V}_1$ to $\mathbf{V}_n$ represent a monotonic motion of the camera.

Let us assume, for the time being, that the motion of the camera between two index frames is such that the optical centers move along a straight line. Taking pairs of views $(\mathbf{V}_1, \mathbf{V}_r), r = 2, \ldots, n$, we compute the fundamental matrices, $\mathbf{F}_{1r}$, from point correspondences (tracking). The epipoles $\mathbf{e}_{1r}$ and $\mathbf{e}_r$ are obtained as the right and left null spaces of $\mathbf{F}_{1r}$ [5].

Given the epipolar geometry between $\mathbf{V}_1$ and $\mathbf{V}_r$, for $r = 2, \ldots, n$, we can compute the camera projec-

tion matrices for $\{\mathbf{V}_1, \mathbf{V}_2, \ldots, \mathbf{V}_n\}$ in a common projective frame. We assume that the first camera to be $[\mathbf{I} \mid \mathbf{0}]$ which fixes the world origin to the camera center of the first view. Using the symmetry of the face, we choose four corresponding points, $\mathbf{p}_{1r}, \mathbf{p}_{2r}, \mathbf{p}_{3r}, \mathbf{p}_{4r}$, from the sequence $\mathbf{V}_r$ for $r = 1, \ldots, n$ such that the four points are coplanar. The choice is a made out of a careful analysis of the triangular mesh and is fixed once for all. For our initial experiments we have chosen the four points by putting special markers on the face. We show the four points in Figure 3. We compute homographies, $\mathbf{M}_r$, between $\mathbf{V}_1$ and $\mathbf{V}_r$ for $r = 2, \ldots, n$ from the point correspondences $(\mathbf{p}_{11}, \mathbf{p}_{21}, \mathbf{p}_{31}, \mathbf{p}_{41})$ and $(\mathbf{p}_{1r}, \mathbf{p}_{2r}, \mathbf{p}_{3r}, \mathbf{p}_{4r})$. $\mathbf{M}_r$, then, is the homography through the plane of the four points. We can choose the camera projection matrix for the r^{th} view to be $[\mathbf{M}_r \mid \mathbf{e}_r]$. Note that this choice is consistent with a single projective frame because $\mathbf{M}_r$ is computed using a corresponding plane in each view. The camera center for the r^{th} view is $(\mathbf{M}_r)^{-1}\mathbf{e}_r$. The homography $\mathbf{M}_r$ can also be computed from three point correspondences and the corresponding epipoles, because three points define a plane and every plane has a point which projects on to the corresponding epipoles [5]. However, this is computationally a little problematic, especially for affine projections, when the epipoles tend to be close to points at infinity.

Given the camera equations for each view, we can rectify each view on to parallel image planes by applying the homography $(\mathbf{M}_r)^{-1}$ to the r^{th} view. The homography does not change the camera centers, and the camera centers will not, in general, lie on a straight line. They will lie on a straight line only when the camera motion is a pure translation. However, for small motion between index frames the deviation of the camera centers from the line joining the camera centers of the end frame is small and we can still apply view morphing. We compute the view morphing parameter s_r for the r^{th} view as follows. We express a point in the r^{th} view as

$$\tilde{\mathbf{p}}_r = (1 - s)\mathbf{p}_1 + s\mathbf{p}_n$$

where $\mathbf{p}_1$ and $\mathbf{p}_n$ are corresponding points in the end frames and $\tilde{\mathbf{p}}_r$ is the point closest to the corresponding point $\mathbf{p}_r$ in the r^{th} view on the straight line joining $\mathbf{p}_1$ and $\mathbf{p}_n$. We take s_r as the average of all the s computed as above.

Note that the rectification of the index frames is required for the computation of s_r even for affine or orthographic projections in order to account for the camera motion. In fact, the above scheme can handle the projective or the affine cases alike.

We transmit the index frames along with the triangular mesh to the decoder. For reconstruction of the global motion at the decoder, we need to transmit only the parameter s_r and the image positions of the four distinguished points. At the decoder, we can reconstruct the triangular mesh for each intermediate frame in the rectified domain by linear interpolation from the index frames and subsequently de-rectify the reconstructed mesh by applying the inverse homography $\mathbf{M}_r$ which can be computed from the four distinguished points. We can save on the transmission of the image positions of the four distinguished points for the intermediate frame by approximating the de-rectification homography by linear interpolation of the de-rectification homographies of the index frames. We have observed in our experiments that this does not degrade the performance significantly.

After reconstructing the image positions of the mesh points in the intermediate frames we reconstruct the intensity information in each triangle by warping from the index frames using 2D affine transformations. This completes the global reconstruction of the video sequence.

4 Local reconstruction

The above scheme of global reconstruction cannot account for the independent motion of the eyes and the lips. We can, however, localize the regions of the mouth, eyes and the eye balls by identifying corresponding triangles of the mesh from the global reconstruction. In these independent regions we parameterize the optic flow [8] parameters in the affine form as

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} u_0 \\ v_0 \end{bmatrix} + \begin{bmatrix} u_x & u_y \\ v_x & v_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \qquad (6)$$

and compute the six affine parameters of the optic flow between an intermediate frame and the nearest index frame by linear least squares. For each local region we transmit the optic flow parameters to the decoder. At the decoder we reconstruct the local regions by intensity warping from the index frames using the optic flow parameters.

5 Results

In this section we present some initial experimental results of video compression using view morphing. For reliable computation of the epipolar geometry we detect corner features on the moving face and track these corners using a variant of a Kalman filter based tracking algorithm [2, 7]. We use the Kalman filter prediction parameters to detect breaks in monotonic motions and insert index frames after every eight frames or whenever there is a break in monotonicity. We also

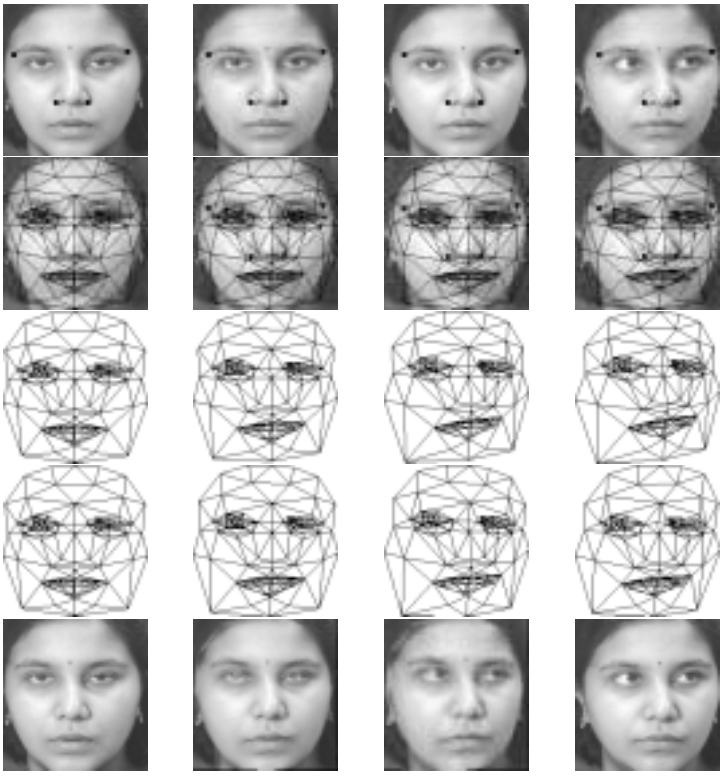

Figure 2: First Row: Original Frames, Second Row: Projected Mesh, Third Row: Rectified Interpolated Mesh, Fourth Row: De-Rectified Mesh, Fifth Row: Reconstructed Frames.

Figure 1: Epipolar Geometry between two index frames.

track the vertices of the triangulation using a Kalman filter based tracker and correlation of intensities.

We compute the epipolar geometry between the index and the intermediate frames after rejecting the outliers of the matches obtained. In Figure 1 we show the typical epipolar geometry between two index frames.

For reliable computation of the homographies $\mathbf{M}_r$, we have inserted special markers for the four coplanar points on the face. The four coplanar points were chosen using the symmetry of the face. In Figure 2 and Figure 3 we show some initial results of video compression using view morphing. In the first rows we show some of the original images of a sequence between two index frames and the position of the markers. In the second rows we show the tracking of the mesh superimposed. In the third rows we show the reconstruction of the control points in the rectified domain. The intermediate frames are interpolated from the rectified index frames. In the fourth rows we show the de-rectified mesh and in the last rows we present the reconstructed frames obtained after intensity warping of each triangle.

The local regions of the eyes and the lips are reconstructed by intensity warping from the index frames using optic flow parameters. In Figure 4 we show the reconstruction of local features.

We could achieve a bit-rate of nearly 45 kbits/sec by encoding a typical facial sequence using the above view morphing based compression algorithm.

6 Conclusion

In this paper we have presented some initial results for video compression using view morphing. We triangulate the facial image and track the vertices of the triangles (control points) through a sequence of frames.

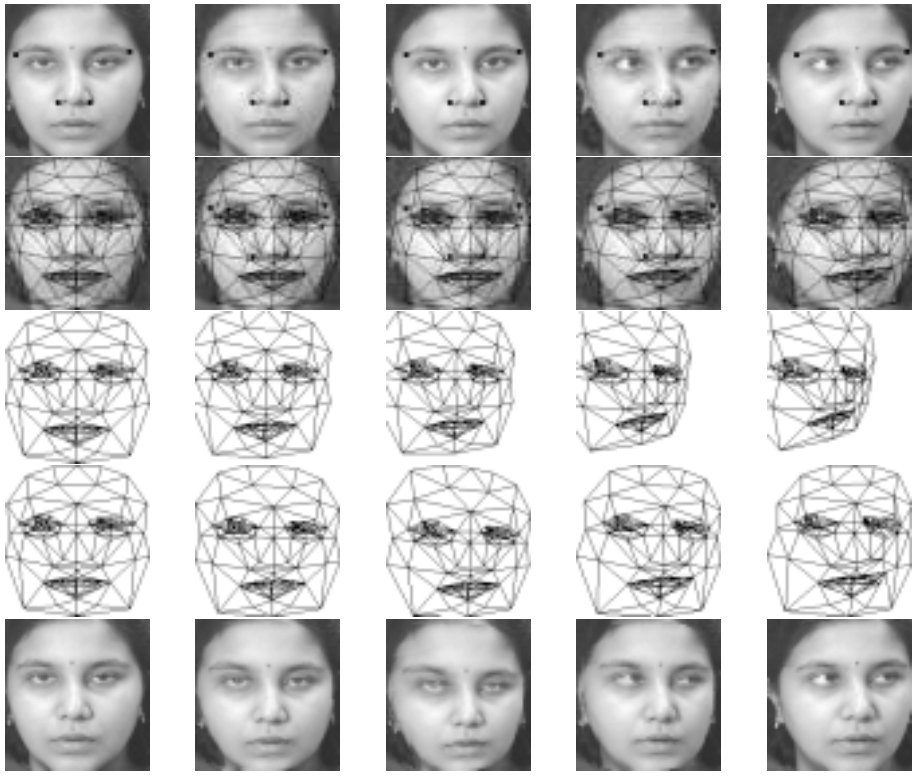

Figure 3: First Row: Original Frames, Second Row: Projected Mesh, Third Row: Rectified Interpolated Mesh, Fourth Row: De-Rectified Mesh, Fifth Row: Reconstructed Frames.

We first apply rectification transformations on the triangulation in each frame in a group of frames to obtain the control points in parallel image planes. This allows us to express the triangulation in the intermediate frames as linear combinations of the triangulation of the end frames. For reconstruction of the control points at the decoder we need to transmit only the control points of the index frames and the interpolation parameters for each intermediate frame. In a post interpolation stage at the decoder we apply an inverse homography to the control points of each image to convert them back to their original configuration. We re-create the intensity information by intensity warping in each triangle from the index frames. Finally we account for the independent local motions of the eyes and the lips using an optic flow based warping technique. The initial results are encouraging and demonstrate the applicability of the method.

References

[1] Steven M. Seitz and Charles R. Dyer, 'View Morphing,' in *Proc. SIGGRAPH'96*.

[2] Shoma Chatterjee, Subhashis Banerjee and K.K. Biswas, 'Affine Structure based facial image encoding,' *IEE Proceedings-Vision, Image and Signal Processing*, Vol. 146, No.4, pp.211-221,August 1999.

[3] J.J. Koenderink and A.J. Van Doorn: 'Affine Structure from Motion,' *J. of Opt. Soc. of Am. Series A*, 1991, Vol. 8, pp. 377-385.

[4] Shimon Ullman and Ronen Basri, 'Recognition by Linear Combinations of Models,' Massachusetts Institute of Technology, Artificial Intelligence Laboratory, A.I. Memo No. 1152, Aug., 1989.

[5] Richard Hartley and Andrew Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2000.

[6] I.D. Reid, and D.W. Murray: 'Active Tracking of foveated feature clusters using Affine Structure,' *Intl. J. Computer Vision*, 1996, **18**,(1), 41-60.

Figure 4: Left: Original Frame, Right: Locally Reconstructed Expressions.

[7] G. Manku, P. Jain, A. Aggarwal, L. Kumar and S. Banerjee, 'Object Tracking Using Affine Structure for Point Correspondences,' *Proceedings of 1997 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 704-709, June, 1997.

[8] B. K. P. Horn and B. G. Schunck, 'Determining Optical Flow,' *Artificial Intelligence*, Vol. 17, pp. 185-203, 1981.