ABSTRACT

Computer graphics models are becoming more realistic and detailed, using millions of triangles to represent a model is currently a norm. Rendering cost and real-time requirement restrict the level of detail graphics models can have. On the other hand, image morphing has cost proportional to image size, independent of the models complexity. If the motion between two frames are not too large, morphing can be a cheaper alternative to rendering complex scenes.

One of the main problems in morphing is that there are many different paths a morph can take, and hence the result could be physically impossible or awkward looking. The view morphing technique proposed by Seitz [13] showed that by rectifying the images before the morph, physical correctness of the result can be guaranteed.

This paper presents a summary of available image morphing techniques, validation of Seitz technique and some future research directions for morphing in graphics. An alternative rectification technique by Hartley [4] is also presented as a comparison.

Keywords: Image morphing, image metamorphosis, projective rectification, view synthesis, image warping.

1. INTRODUCTION

With the advancement of digital laser scanners, the acquisition process for detailed three-dimensional (3D) models has been greatly simplified. Modern laser scanners have resolution up to sub-millimeter, the computer graphics models of real objects acquired are becoming more and more realistic. At the same time, the number of polygons in graphics models is also increasing exponentially. Using millions of polygons to represent one model is currently a norm.

As the rendering time is proportional to the number of polygons processed by the graphics hardware, the rendering process is a computationally very expensive task due to the large number of polygons involved. This is particularly crucial for applications that require real-time rendering such as those in the gaming industry. When frame rate takes the ultimate precedence, the models have to be simplified and image quality is compromised.

Image morphing transforms a base image into a target image with smooth transitions in between. It is traditionally used for creating the special visual effect of fluid transformation between two objects. The computational cost of a morph is proportional to the image size.

Since rendering is very expensive and successive frames usually look similar, morphing could be a cheaper alternative to rendering. Instead of rendering all 30 frames in a second for real-time requirements, only a few keyframes need to be rendered, all the transition frames can be obtained by morphing. In this manner, the frame generation time of the rendered keyframes is proportional to the complexity of the scene, which varies; and the frame generation time of all other transition frames is proportional to the image size, which is usually fixed. The time required for the later is usually shorter than the former in complex scenes.

So, we would like to have a smaller number of rendered frames and a larger number of morphed frames to speed up the frame generation. However, if the two rendered frames are too different, it is likely that the morphed transitions will look unnatural or have large patches of holes due to occlusion. We also need to ensure that the path that the morphing transition takes, which is usually a linear path, is as close to the real path as possible. The view morphing work by Seitz [13] demonstrates that unless special care is taken, morphing between two images of similar a 3D shape does not necessarily produce images that preserve the 3D shape, as shown in Fig.1. Computer vision principles - the epipolar constraints were proposed to be used to ensure the morph transition goes through the same path as the path a legitimate camera movement would take. This is achieved by adding a prewarping and a postwarping stage. Before the normal morphing, the two images are prewarped into a parallel view configuration, under which linear interpolation of pixel flow produces physically valid views. After the morphing, the resultant image is postwarped according to its new interpolated camera configuration.

13th School of Computer Science & Software Engineering Research Conference, Yanchep, Western Australia, 20th–21st September 2004. ©2004 is asserted by the author(s).
2. IMAGE MORPHING

Image morphing or image metamorphosis is a technique that combines two-dimensional (2D) interpolation of shape and color to produce smooth transition between images. It is essentially image warping followed by cross dissolving. At each step of the morphing, two images are geometrically interpolated (or warped) such that two different shapes in the images are deformed into one interpolated shape. Then the two warped images are blended together with different weight depending on the stage of transitions. The result is a smooth fluid transition of two images.

Many techniques have been proposed in image morphing since the pioneering work of Smythe in 1988. Wolberg has surveyed the field extensively in [17, 16]. The various morphing techniques differ fundamentally in two major aspects: feature specification and warp generation. More recent literature extended the morphing field in transition control [7], generalization to multiple images [8, 19, 20] and shape preservation techniques [13, 10, 18].

2.1 Available Techniques

The earliest morphing was based on mesh warping developed at Industrial Light Magic by Smythe in 1988 [14]. Meshes were used to mark feature points and warp generation was done with straightforward bicubic spline interpolation. Beier and Neely improved the feature specification by using line pairs in their field morphing approach [2], at a cost of computationally more expensive warp generation stage, and occasional unexpected distortion due to incompatible interference of multiple lines.

Later approaches loosened the restriction of feature primitives to allow all points, lines and curves to be used, and energy minimizing splines (snakes) were introduced to help users place features near edges [7]. The lines and curves can be point sampled and hence the features can be considered to be represented by a set of points. As a result, the warp generation is formulated into a scattered data interpolation problem, which is well surveyed by Ruprecht and Müllai 11 and Wolberg [15]. Thin plate splines [9], radial basis functions [1] and multilevel freeform deformation [7] are approaches under this category.

All approaches above require accurate manual feature specification to generate visually aesthetic results. Gao and Sederberg [3] proposed a work minimization approach that can potentially automate the feature specification process if the input images are sufficiently similar. They assigned cost functions to the bilinear B-spline warp and pixel intensity recolouring, and use the warp that has the global minimum sum of cost as the solution.

In more recent morphing literature, transition control was introduced to vary the rate of warping and colour blending between images [7]. Different parts of the base image can transform into the target image at different rate, creating more possibilities of transformation and visual satisfaction. Morphing among multiple images was also attempted [8, 19, 20]. Lee et al. proposed a polymorph framework that demonstrated non-uniform blending of features from several input images. They showed an example of a facial image that has its eyes, nose, mouth and hair derived from three input images.

View morphing proposed by Seitz [13] utilizes the epipolar constraints from computer vision to ensure that the in-between images are shape preserving and physically valid. The technique added a prewarp stage and a postwarp stage for a normal morphing such that the morphing is done under parallel view configuration, which preserves 3D shapes. Xiao and Shah extended Seitz's work into three view using trifocal tensor constraints [20]. Dynamic scenes view morphing was demonstrated to be possible by segmentation of moving object and morphing in different layers [10, 18].

2.2 Applicable Techniques

The morphing for graphics aims to automatically filling in the non-keyframes between rendered frames. Automatic feature specification is required and techniques like [3] may be utilized. However the current frame generation speed of about 10 seconds for a 256 × 256 image is too low for our application. Faster automatic morphing technique might be developed by combining existing techniques and the geometric information available in the graphics domain.

Since the features are obtained automatically, they are most likely to be point samples. Thus the warp generation will be a scattered data interpolation problem. Many techniques in the field [11, 15] are readily available to be used.

The images in our context are rendered images of computer animation at different instance of time. They are likely to be very similar and represent animation of the same objects. So shape preservation is very important. View morphing techniques [13] and its various extensions [10, 18] are likely to be useful.
3. VIEW MORPHING

Seitz’s work in view morphing [13] demonstrated that unless special care is taken, morphing between two images of similar 3D shape does not necessarily produce image that preserve the 3D shape. It also showed that morphing two parallel views of a scene with linear interpolation produces another parallel view of the same scene (Fig.2). That is, parallel view condition is a special case under which a linear morph is shape preserving. This section describes Seitz approach to ensure 3D shape preservation of morphing by changing a general view condition into this special case.

3.1 Parallel View

Following convention, a point in 3D space with Euclidean coordinates \((X, Y, Z)\) is expressed in homogeneous coordinates as \(\mathbf{X} = [sX, sY, sZ, s]^T\) and a Euclidean image point \((x, y)\) is expressed as \(\mathbf{x} = [sx, sy, s]^T\). A camera is represented by a \(3 \times 4\) camera projection matrix \(\mathbf{P}\) of the form \(\mathbf{P} = [\mathbf{M}] - \mathbf{MC}\). The vector \(\mathbf{C}\) is the position of camera optical center and the \(3 \times 3\) matrix \(\mathbf{M}\) encodes the intrinsic camera parameters and the orientation of the camera image plane \(I\) with respect to the world coordinate system. An image point \(\mathbf{x}\) is a projection of a scene point \(\mathbf{X}\) through a camera \(\mathbf{P}\), expressed mathematically as:

\[
\mathbf{x} = \mathbf{P} \mathbf{X}
\]

In a parallel view condition, suppose that first camera is at the world origin \((0, 0, 0)\), with focal length \(f_1\), the second camera is at \((C_x, C_y, 0)\) with focal length \(f_2\), we can write the respective camera projection matrices, \(\hat{\mathbf{P}}_1\) and \(\hat{\mathbf{P}}_2\) as:

\[
\hat{\mathbf{P}}_1 = \begin{bmatrix}
    f_1 & 0 & 0 & 0 \\
    0 & f_1 & 0 & 0 \\
    0 & 0 & 1 & 0
\end{bmatrix}
\]

\[
\hat{\mathbf{P}}_2 = \begin{bmatrix}
    f_2 & 0 & 0 & -f_2 C_x \\
    0 & f_2 & 0 & -f_2 C_y \\
    0 & 0 & 1 & 0
\end{bmatrix}
\]

Let \(\mathbf{x}_1 \in I_1\) and \(\mathbf{x}_2 \in I_2\) be projections of a scene point \(\mathbf{X}\) =

3.2 Non-parallel View

When two images to be morphed are not in the parallel view condition, the 3D shapes of objects will not be preserved in the morphed images, as shown in Fig.1. To ensure the 3D shapes are preserved, which is an important criterion for visual realism, Seitz proposed a three steps algorithm of prewarping, morphing, and postwarping (Fig. 3).

1. **Prewarp**: reproject non-parallel images \(I_1\) and \(I_2\) into parallel views \(\hat{I}_1\) and \(\hat{I}_2\).

2. **Morph**: create \(\hat{I}_o\) by linearly interpolating position and color of corresponding points in \(\hat{I}_1\) and \(\hat{I}_2\) using Eq.1 or any image morphing technique that approximate it.

3. **Postwarp**: reproject \(\hat{I}_o\) to the desired intermediate camera configuration and yield image \(I_o\).

When the prewarp and postwarp are correctly applied, the problem is reduced to that of parallel view.
The prewarping stage is an image reprojection operation that rectifies two non-parallel image planes onto a pair of parallel image planes (See Fig. 3, black solid arrows). The technique is closely related to the stereo views rectification techniques for simplifying 3D shape reconstruction.

Let $I$ and $\hat{I}$ be two views that share the same optical center, with projection matrices $P = [M] - [MC]$ and $\hat{P} = [M] - [M\hat{C}]$. The projections of scene point $X$ into $x \in I$ and $\hat{x} \in \hat{I}$ are related by the following transformation:

$$
x = P X
\hat{x} = \hat{P} X
$$

Thus the projective transformation matrix that projects the image plane $I$ onto $\hat{I}$ is given by the $3 \times 3$ matrix

$$
H^{-1} = \hat{M} M^{-1}
$$

We seek a matching pair of projective transformations $H_1$ and $H_2$ such that the prewarped first image $\hat{I}_1 = H_1^{-1} I_1$ and the prewarped second image $\hat{I}_2 = H_2^{-1} I_2$ are parallel to the $C_1 C_2$ baseline. In the computation of this two views reprojection, we utilize the fundamental matrix, which encodes important epipolar geometry of the two views.

The fundamental matrix is a $3 \times 3$, rank 2 matrix $F$ such that for every pair of corresponding image points $x_1 \in I_1$ and $x_2 \in I_2$,

$$
x_1^T F x_2 = 0
$$

$F$ is defined up to a scale factor and can be computed from 8 or more corresponding points [6]. However, since full information of the camera parameters is available in the graphics domain, the fundamental matrix can be computed accurately using the two camera calibration matrices $P_1$ and $P_2$ as shown below

$$
F = [e_2]_x P_2 P_1^+
$$

where $[e_2]_x$ is the skew symmetric matrix of $e_2$, $P_1^+$ is the pseudo inverse of $P_1$,

$$
[e_2]_x = \begin{bmatrix}
  0 & -e_2^z & e_2^y \\
  e_2^z & 0 & -e_2^x \\
  -e_2^y & e_2^x & 0
\end{bmatrix}
$$

$$
P_1^+ = P_1^T (P_1 P_1^T)^{-1}
$$

$$
e_2 = P_2 C_1 = [e_2^x e_2^y e_2^z]^T
$$

$$
P_1 C_1 = 0
$$

A sufficient condition for two views to be parallel is to have the fundamental matrix in the form

$$
\hat{F} = \begin{bmatrix}
  0 & 0 & 0 \\
  0 & 0 & -1 \\
  0 & 1 & 0
\end{bmatrix}
$$

The transformations $H_1^{-1}$ and $H_2^{-1}$ reproject $I_1$ and $I_2$ into parallel view condition if the following equation is satisfied.

$$
(H_2^{-1})^T F H_1^{-1} = \hat{F}
$$

There is a range of homographies $H_1$ and $H_2$ that satisfy Eq. 5, corresponding to different choices of parallel planes. And there are various techniques for computing the warp matrices $H_1$ and $H_2$. Two techniques, one by Seitz [13, 12], which can be computed with only the fundamental matrix; and the other by Hartley [4], which requires the fundamental matrix and point correspondence, are described.

### 3.3 First Prewarping Method - Seitz

Let $E$ be a plane parallel to $C_1 C_2$. Applying a 3D rotation to rotate image planes $I_1$ in depth about a line $d_i = [d_i^x d_i^y d_i^z]^T$, which is the intersection of $I_1$ and $E$, can make $I_1$ and $E$ co-planar. Rotating about any axis parallel to $d_i$ can make all image planes parallel to $C_1 C_2$ Fig. 4.

Figure 4: A pair of views $I_1$, $I_2$ can be made parallel to the baseline ($C_1 C_2$) by rotating them about any axis parallel to $d_1$ and $d_2$ respectively. $d_i$ is the intersection of image planes $I_i$ and plane $E$ that is parallel to $C_1 C_2$.

Seitz prewarping method can be described geometrically as first rotating image planes $I_i$ in depth about $d_i$ to make them become parallel views, then applying an image rotation to line up the epipolar lines.

The calculation starts with a choice of the arbitrary rotation axis $d_i = [d_i^x d_i^y d_i^z]^T \in I_i$. Although any axis of rotation will satisfy the equation mathematically, in practice, it is important to choose $d_i$ that will minimize nonlinear distortion. One approach is to choose a rotation axis that will minimize the angle of rotation, $\theta_i$.

The rotation matrix $R_{\theta_i}^d$ specified by rotation axis $d_i = [d_i^x d_i^y d_i^z]^T$ and rotation angle $\theta_i$, is given by:

$$
R_{\theta_i}^d = \begin{bmatrix}
  t d_i^x d_i^x + c & t d_i^x d_i^y - d_i^z s & t d_i^x d_i^z + d_i^y s \\
  t d_i^y d_i^x + d_i^z s & t d_i^y d_i^y + c & t d_i^y d_i^z - d_i^x s \\
  t d_i^z d_i^x - d_i^y s & t d_i^z d_i^y + d_i^x s & t d_i^z d_i^z + c
\end{bmatrix}
$$

where $c = \cos \theta_i$, $s = \sin \theta_i$, $t = 1 - c$. 


The epipoles $e_1 = [e_1^x, e_1^y]^T$ are a projection of optical center of the other camera in $I_2$, or equivalently, the intersection of $O_1 O_2$ and the image plane $I_1$.

To warp $I_1$ and $I_2$ to be parallel to $O_1 O_2$, the new epipole $e_1$ will be at infinity. Thus we seek $R_{d_1}$ such that

$$
[ e_1^x, e_1^y, 0 ] = e_1 = R_{d_1} e_1
$$

since $d_1 = 0$, from the third row of Eq.6, the angle of rotation $\theta_1$ is deduced as:

$$
\theta_1 = \tan^{-1}\left(\frac{e_1^y}{d_1^2 e_1^2 - d_1^2 e_1^y}\right)
$$

To minimize the warping distortion, $\theta_1$ is to be minimized. Choosing the rotation axis orthogonal to $e_1$ will produce an optimal value for $\theta_1$.

$$
d_1 = \frac{1}{\sqrt{(e_1^x)^2 + (e_1^y)^2}} \begin{bmatrix} -e_1^y \\ e_1^x \\ 0 \end{bmatrix}
$$

The corresponding rotation axis for $I_2$ is given by

$$
d_2 = \begin{bmatrix} -y \\ x \\ 0 \end{bmatrix} \text{ where } \begin{bmatrix} x \\ y \\ z \end{bmatrix} = Fd_1
$$

Once $I_1$ and $I_2$ are warped to be parallel, they are rotated to align their corresponding epipolar lines. The rotation angles $\phi_1, \phi_2$ and their respective rotational matrices $R_{\phi_1}, R_{\phi_2}$ are given by

$$
\phi_i = -\tan^{-1}\left(\frac{e_i^y}{e_i^x}\right)
$$

$$
R_{\phi_i} = \begin{bmatrix} \cos \phi_i & -\sin \phi_i & 0 \\ \sin \phi_i & \cos \phi_i & 0 \\ 0 & 0 & 1 \end{bmatrix}
$$

After the 3D rotation in depth and 2D image rotation, the fundamental matrix has the form (up to a scale factor):

$$
\hat{F} = R_{\phi_2} R_{d_2}^T F R_{d_1} R_{\phi_1} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & a \\ 0 & 1 & b \end{bmatrix}
$$

Finally get $\hat{F}$ into the form of $\hat{F}$, $I_2$ is translated and vertically scaled by the matrix,

$$
T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -a & -b \\ 0 & 0 & 1 \end{bmatrix}
$$

The result can be verified by $T^{-1}^T \hat{F} = \hat{F}$.

In summary, the prewarping transforms $H_1$ and $H_2$ are

$$
H_1 = R_{\phi_1} R_{d_1}^T
$$

$$
H_2 = TR_{\phi_2} R_{d_2}^T
$$

### 3.4 Second Prewarping Method - Hartley

Hartley [4] also presented a method to rectify two views such that their corresponding epipolar lines are parallel and aligned. The strategy is to first find a transformation $H_2$ that will send the epipole of the second image $e_2$ to infinity. Then it seeks a matching transformation $H_1$ that minimizes the disparity between all point pairs. This section only presents a summary of some derivations and important equations, detail proofs are to be found in the original paper [4].

At the first stage, the transformation $H_2$ is obtained by applying an image rotation similar to Eq.11 to the image to position the epipole on the x-axis, then applying a projective transformation $G$, to map the epipole to infinity.

The rotation angle for positioning the epipole $e_2 = [e_2^x, e_2^y, 1]^T$ on x-axis is

$$
\phi = -\tan^{-1}\left(\frac{e_2^y}{e_2^x}\right)
$$

After the image rotation, the new epipole $\hat{e}_2 = [e_2^x, 0, 1]^T$ lies on x-axis. The following transformation $G$ will map $\hat{e}_2$ to the point at infinity $[e_2^x, 0, 0]^T$:

$$
G = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1/e_2^x & 0 & 1 \end{bmatrix}
$$

A translation $T$ is needed to shift origin to the center of image $x_0 = [x_0, y_0]^T$ to minimize distortion,

$$
T = \begin{bmatrix} 1 & 0 & -x_0 \\ 0 & 1 & -y_0 \\ 0 & 0 & 1 \end{bmatrix}
$$

The transformation $H_2$ can then be computed from the epipole $e_2$ and image center coordinate $x_0$ as

$$
H_2 = G R_{\phi} T
$$

At the second stage, a matching transformation $H_1$ that minimizes the sum of squared distance between corresponding point pairs

$$
\sum d(x_1, x_2)^2
$$

need to be found.

Let $x_1$ be a point on the first image $I_1$, then $e_1 \times x_1$ is the epipolar line in $I_1$, and $Fx_1$ is the epipolar line in the second image $I_2$. Transformation $H_1$ and $H_2$ are matching pair if and only if $H_1^T (e_1 \times x_1) = H_2^T (Fx_1)$, where $H_1^T$ is the matrix of cofactor for $H_1$. Since this must hold for all $x_1$,

$$
H_1^T (e_1 \times x_1) = H_2^T (Fx_1)
$$

$$
[H_1 e_1]_x H_1 = [H_2 e_2]_x H_2 M
$$

$$
H_1 = (1 + H_2 e_2^T H_2 M)
$$

$$
= A H_2 M
$$
where \( A = I + H_2 e_2 a^T \)
a is an arbitrary 3-vector
\( M = [e_2]F + e_2 v^T \)
v is an arbitrary 3-vector

Since we are only interested in the case when \( H_2 \) takes the epipole \( e_2 \) to infinity \([e_2 \ 0 \ 0]^T\), \( A = I + [e_2 \ 0 \ 0]^T a^T \) is of the form

\[
A = \begin{bmatrix}
a & b & c \\
0 & 1 & 0 \\
0 & 0 & 1 \\
\end{bmatrix}
\]

(19)

which represents an affine transformation. This is a special case of Eq.18.

If we write \( \hat{x}_{1i} = H_2 M x_i \) and \( \hat{x}_{2i} = H_2 x_i \), the minimization problem in Eq.17 is to find \( A \) such that

\[
\sum_i d(A \hat{x}_{1i}, \hat{x}_{2i})^2
\]

(20)
is minimized.

Let \( \hat{x}_{1i} = [\hat{x}_{1i1}, \hat{x}_{1i2}, 1]^T \) and \( \hat{x}_{2i} = [\hat{x}_{2i1}, \hat{x}_{2i2}, 1]^T \), \( (\hat{y}_{1i} - \hat{y}_{0i}) \) is a constant, the quantity to be minimized may be written as

\[
\sum_i (a \hat{x}_{1i1} + b \hat{x}_{1i2} + c - \hat{x}_{2i2})^2
\]

(21)

### 3.5 Triangulation Base Morphing

The morphing stage of the three steps algorithm can be done using any image morphing algorithm that linearly interpolates feature positions and colors of images. Many of the techniques mentioned in section 2 can be used, but for simplicity, a triangulation based approach was implemented.

All feature points in \( I_1 \) are triangulated using Delaunay triangulation technique. The same vertices structure is used for feature points in \( I_2 \) so that the two images have a matching set of triangles. Then each triangle is warped into its corresponding triangle via an affine transformation \( A_{triangle.i} \).

Affine transformation has six degrees of freedom and can be easily computed by the three vertices correspondence, with each vertex providing two constraints.

Let \( v_{ij,k} \) be a triangle vertex in homogeneous coordinates. The index \( i \) is triangle pair index, \( i \in [1, 2, \cdots, i_{max}] \), \( i_{max} \) is the total number of triangle pairs; index \( j \) denotes either the first or second triangle in the pair, \( j \in [1, 2]; \) and index \( k \) denotes the vertex number, \( k \in [1, 2, 3] \). The transformation \( A_{triangle.i} \) that warps the second triangle into the first triangle in the \( i \)-th triangle pair can be deduced as

\[
A_{triangle.i} = [v_{i,1,1}, v_{i,1,2}, v_{i,1,3}] [v_{i,2,1}, v_{i,2,2}, v_{i,2,3}]^{-1}
\]

To get the transition images for morphing. The intermediate positions of the triangle vertices \( v_{i,j,k} \) are obtained via linear interpolation of \( v_{1,k} \) and \( v_{2,k} \). \( j \) is the transition index, \( j \in [1, 2, \cdots, j_{max}] \) where \( j_{max} \) is the number of transition steps.

\[
v_{i,j,k} = \frac{(1-\alpha)}{j_{max}+\alpha} v_{i,1,k} + \alpha v_{i,2,k}
\]

with \( \alpha = \frac{j}{j_{max}+1} \)

By going through and generating a warp matrix for each triangle pair, the warp matrix for the whole image can be obtained. The two images are then warped into two matching intermediate images, \( I_1' \) and \( I_2' \). The final image \( I_j \) is a weighted sum of the intermediate images.

\[
I_j = (1-\alpha) I_1' + \alpha I_2'
\]

(22)

### 3.6 Postwarping

The postwarping stage warps the morphed intermediate images, which have image planes parallel to the baseline, into the final images of desired intermediate camera configurations, \( P_\alpha = [M_\alpha - M_\alpha C_\alpha] \). The camera matrices \( P_\alpha = [M_\alpha - M_\alpha C_\alpha] \) of morphed intermediate images can be computed with Eq.1. Following Eq.2, the transformation \( H_\alpha = M_\alpha M_\alpha^{-1} \) warps \( I_\alpha \) into the final image \( I_\alpha \).

The desired intermediate camera matrix \( P_\alpha \) can be obtained in several ways. A natural way is to interpolate the orientation of the image planes by a single axis of rotation. Let \( N_1 \) and \( N_2 \) be the image plane normals, the rotation axis \( D \) and angle of rotation \( \theta \) are given as

\[
D = N_1 \times N_2
\]

\[
\theta = \cos^{-1}(N_1 \cdot N_2)
\]

(23)

However, Seitz specified the transformation by interactively specifying four control points in the output image. Xiao et al. [20] developed an automatic postwarping method for their tri-view morphing. The method uses five corresponding points to find the least distortion postwarp.

### 4. VALIDATION RESULTS

Seitz work in view morphing is implemented and validated in this experiment. Two prewarping methods described in section 3.3 and section 3.4 and were implemented. The results are different but similar, and both valid. The morphing was done using a simple triangulation method as described in section 3.5. The postwarping was specified with explicitly providing the desired camera matrices, which correspond to those used to generate the ground truth.

Fig.5 shows the desired view transformation sequence, i.e. the ground truth. Fig.1 was generated using 30 point correspondence, \( 5 \times 5 \) point grid on each of the two planes. The figure clearly shows that straight lines are not preserved under linear morphing. Fig.6 was generated using only the 6 corner points as correspondence. Because the triangulation morphing method transforms one triangle to another triangle, all the straight lines are preserved. However, the features are misaligned due to the difference between projective transformation, which is the real transformation during a change of view point, and affine transformation, which is the triangulation morphing.

Fig.7 is the result of Seitz’s prewarping, and Fig.8 is the result of Hartley’s prewarping. Note that both methods successfully rectify the images to have horizontally aligned epipolar lines. The features are all aligned in the morphed sequence. Only 6 point correspondences are needed to compute the morph sequence. Fig.9 is the postwarped result of Fig.7. It is visually the same as the desired view transformation sequence of Fig.5.
Figure 5: The desired view transformation sequence

Figure 6: Features misalign when only the corner points are selected. Straight lines are preserved.

Figure 7: Prewarped and morphed result by Seitz’s prewarping method

Figure 8: Prewarped and morphed result by Hartley’s prewarping method

Figure 9: Postwarped Result of Fig.7
5. DISCUSSIONS & FUTURE WORK

This experiment validated Seitz's view morphing algorithm. With correct point correspondences, view morphing preserves 3D shapes in transitional images. Hartley's and Seitz's methods of rectifying images are both valid, provided that the epipole are outside of the image.

Seitz's method computes the transformations by changing camera projections and works well. Hartley's method produces prewarped image of least disparity according to the point correspondence, the result is non-symmetrical even when the two images are mirrored images. Further investigation into Hartley methods revealed that although to views are rectified to have aligned epipolar lines, it does not necessarily represent a valid camera view. As a result, although the images can be morphed with correct point correspondence, no proper postwarping transformations could be found to rectify the morphed images back to any sensible camera views. And thus, Hartley's method is not appropriate for this application in view morphing.

When the epipole is inside or very close to the image, parallel view rectification fails or produces seriously distorted images because the epipole is moved to infinity. However, there are many cases where the epipole is inside or close to the image. Alternative technique needs to be developed to ensure shape preserving morphing in such cases.

Since point correspondence is computed from features, the prewarping stage and warp generation methods determine the correctness of the point correspondence and the number of features needed to achieve the correctness. In this experiment, it was observed that when two images are rectified into parallel view condition, all points on the same 3D plane in one image can be correctly matched up with their corresponding points in another image by an affine transformation. This means that only the corners of the plane need to be specified as features and affine transformation can be applied to generate the full point correspondence. The correspondence may not be correct if two views are not in parallel (as in Fig.6) or another warp generation method is used.

This gives rise to several problems to be further investigated.

- Firstly, what features are important and what features are redundant?
- Secondly, what is the error of the point correspondence produced by different warp generation techniques?
- In combination, what is the minimum set of features that will produce a correct point correspondence (to some tolerance of error) using a certain warp generation method?

The ground truth of point correspondence can be obtained in the graphics domain, and compared with the result produced by different warp generation techniques to deduce an error measurement.

The overall problem we are aiming to solve is to replace rendering in non-critical frames by morphing for faster complex computer graphics scenes generation. Physically correct morphing, which is shown in this paper is one of the key problems under morphing for graphics. Other key problems include automatic feature correspondence, morphing behaviour at edges and fast implementation.

The automatic feature correspondence is a hard problem that is not fully solved. It is fortunate in our case, because the images are rendered, we have all the information of camera parameters and point positions in 3D space. But it is unfortunate that in the graphics rendering pipeline, all this information are considered useless at some stage and is lost once the images are rendered. So, we will need to find a way to retaining some data from the rendering process or use other ways such as watermarking of the mesh to solve the matching problem.

Another problem in morphing that is constantly ignored is the behaviour of points at edges that cannot be matched in the other image. Computer vision constraints can probably be used to estimate the path and rate of how unmatched points should travel in the morph. Finally, efficiency is always a key attribute in graphics computation, so some optimization of computation on hardware will be worth to investigate.

6. REFERENCES


