Projective Transformations for Image Transition Animations

TzuYen Wong Peter Kovesi Amitava Datta
The University of Western Australia
School of Computer Science & Software Engineering
35 Stirling Highway, Crawley, W.A. 6009, Australia
{wongt, pk, datta}@csse.uwa.edu.au

Abstract

Transformation of image patches is a common requirement for 2D transition animations such as shape interpolation and image morphing. It is usually done by applying affine transformations to triangular patches. However, the affine transformation does not model the perspective transformation frequently found in images. Hence, such techniques can only produce approximate results and usually use an excessively large number of triangles to compensate for this shortcoming. This paper proposes the application of projective transformations on quadrilateral image patches as a solution to this problem. We address the issues of appropriate decomposition and interpolation of projective transformation matrices to produce a natural looking transition animation for a single quadrilateral as well as for shapes made up of multiple quadrilaterals.

1. Introduction

Matrix animation and polar decomposition techniques [7] enable decomposition of an affine transformation matrix into rotation, scaling and translation matrices for meaningful interpolation and animation. However, planar surfaces in images are more often related by a general projective transformation than its affine transformation subset, and hence using the correct transformation group is an obvious rational approach as supported by the principles for a good morph outlined by Gomes [3].

The contribution of this paper can be separated into two parts. The first is the extension of matrix decomposition and interpolation techniques to operate on projective transformations, and this enables physically-valid interpolation of quadrilaterals in 2-dimensional space. The second contribution is the usage of quadrilateral patches and projective interpolation in shape interpolation and image morphing.

Suppose there is a pair of corresponding projected planar shapes or points for which we want to generate a transition, an affine transformation will not produce the correct result as shown in Fig. 1, hence we need to find their projective relationship (called a homography), and then decompose and interpolate that projective matrix.

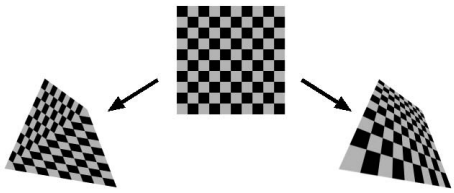

Figure 1. Comparison of piecewise affine transformation (left) and perspective transformation (right) of a square checkerboard

The key idea is to extract the last row of the homography which represents the vanishing line of the planar region in homogeneous coordinates, and interpolate the vanishing line vector with the vanishing line at infinity as a vector rotation. The remainder of the projective component is an affine transformation matrix which can be decomposed using singular value decomposition and interpolated with affine matrix animation techniques [7].

When quadrilaterals can be projectively interpolated, more complex shapes consisting of multiple quadrilaterals can be projectively interpolated also. However, the problem is that interpolated quadrilaterals of different projective transformations will become disconnected. To enforce boundary connectivity, we assign a force proportional to the distance between each disconnected vertex and analytically deduce the final position of all quadrilaterals when the force equilibrium is reached.

2. Related Work

Shoemake [7] describes a meaningful affine transformation matrix animation scheme that extracts and interpolates the rotation component of an affine transformation matrix using polar decomposition. Alexa utilises this technique for his as-rigid-as-possible shape interpolation and image morphing based on triangles [2]. He also provides an extensive review on triangular mesh morphing [1]. This work lays down the ground theories and approaches for a meaningful decomposition of transformation matrices and natural looking interpolations, but they are unable to fully model perspective effects. Another drawback of shape interpolation using triangles is that the quality of the transition strongly depends on avoiding having irregular triangles in the mesh through complex isomorphic dissections.

Projectively correct interpolation is usually handled via three dimensional graphics techniques such as three dimensional transformation and texture mapping or using two dimensional view morphing techniques [5, 6, 9]. Techniques in the former involve getting into an additional dimension and hence complexity while techniques in the latter are not suitable for multiple planer surface pairs of different relationships due to its prewarp and postwarp operations.

3. Overview of Projective Interpolation

Let $\mathbf{p}$ and $\mathbf{q}$ be the vertices of two quadrilaterals in homogeneous coordinates (stored as $[3 \times 4]$ matrices) related by a $[3 \times 3]$ projective transformation, a homography $\mathbf{H}$.

$$\mathbf{q} \;=\; \mathbf{H}\,\mathbf{p} \tag{1}$$

The homography $\mathbf{H}$ can be computed using the normalised direct linear transformation algorithm by Hartley [4].

If we want a smooth transition from $\mathbf{p}$ to $\mathbf{q}$ (with transition constant $\alpha : 0 \rightarrow 1$), linear interpolation of their vertices ($(1-\alpha)\mathbf{p} + \alpha\,\mathbf{q}$) and direct interpolation of the $\mathbf{H}$ matrix ($[(1-\alpha)\mathbf{I} + \alpha\,\mathbf{H}]\,\mathbf{p}$) do not yield satisfying results in general cases [7], and if the two quadrilaterals are projectively related, dividing them into two triangles and affinely interpolating them will not produce a projectively correct result, as shown in Fig. 1.

Rather than interpolating the elements of $\mathbf{H}$ directly, a smooth transition can be achieved by appropriately interpolating the decomposed components of $\mathbf{H}$ separately. The issues to be dealt with here is that there are some choices to be made in the decomposition and interpolation.

3.1. Decomposition

The homography $\mathbf{H} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}$ has 8 degrees of freedom, with 9 entries in the matrix. Thus the matrix can be multiplied by any non-zero value and still represents the same transformation. To have a unique normalised representation, we pre-multiply $\mathbf{H}$ with a normalisation constant η.

$$\eta = \frac{sign(h_{33})}{\sqrt{h_{31}^2 + h_{32}^2 + h_{33}^2}} \tag{2}$$

Note that the entries h_{31}, h_{32}, h_{33} represent the vanishing line of the planar region in homogeneous coordinates. This normalisation constant η is chosen to make the decomposed projective matrix have a vanishing line vector of unit magnitude and that avoids unnatural interpolation results, as will be further explained in section 4.3.

The normalised projective matrix $\eta\mathbf{H}$ can be decomposed into simple transformation elements

$$\eta\mathbf{H} = \mathbf{T}\,\mathbf{R}_\theta\,\mathbf{R}_{-\phi}\,\mathbf{S}\,\mathbf{R}_\phi\,\mathbf{P} \tag{3}$$

where $\mathbf{P}$ is a pure projective matrix $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \eta h_{31} & \eta h_{32} & \eta h_{33} \end{bmatrix}$ with a vanishing line vector, $\mathbf{v_v} = \eta[h_{31}, h_{32}, h_{33}]^T$ of unit magnitude; $\mathbf{R}_{\pm\phi}$ are rotation matrices to align the axis for horizontal and vertical scaling of $\mathbf{S}$; $\mathbf{R}_\theta$ is another rotation matrix to orientate the shape into its final orientation; and lastly $\mathbf{T}$ is a translation matrix.

$\mathbf{P}$ is extracted by simply taking the last row of $\eta\mathbf{H}$, the remainder of the transformation is an affine matrix $\mathbf{H_A}$ that can be further decomposed into $\mathbf{T}\,\mathbf{R}_\theta\,\mathbf{R}_{-\phi}\,\mathbf{S}\,\mathbf{R}_\phi$.

$$\begin{aligned} \mathbf{H_A} &= \eta\mathbf{H}\,\mathbf{P^{-1}} \\ &= \mathbf{T}\begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0^T} & 1 \end{bmatrix} \end{aligned} \tag{4}$$

$\mathbf{T}$ is extracted by taking the 3rd column of $\mathbf{H_A}$, and $\mathbf{A}$ is extracted by taking the top left $[2 \times 2]$ submatrix of $\mathbf{H_A}$ which represents a non-homogeneous affine transformation. $\mathbf{A}$ can be further decomposed to produce the non-homogeneous version of $\mathbf{R}_\theta\,\mathbf{R}_{-\phi}\,\mathbf{S}\,\mathbf{R}_\phi$ using the singular value decomposition (SVD).

$$\begin{aligned} \mathbf{A} &= \mathbf{U}\,\mathbf{D}\,\mathbf{V^T} \\ &= (\mathbf{U}\,\mathbf{V^T})\,\mathbf{V}\,\mathbf{D}\,\mathbf{V^T} \end{aligned} \tag{5}$$

where $\mathbf{D}$ is a diagonal matrix satisfying $D_{11} < D_{22}$, and $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices.
$\mathbf{S} = \begin{bmatrix} \mathbf{D} & \mathbf{0} \\ \mathbf{0^T} & 1 \end{bmatrix}$, $\mathbf{R}_\theta = \begin{bmatrix} \mathbf{UV^T} & \mathbf{0} \\ \mathbf{0^T} & 1 \end{bmatrix}$ and $\mathbf{R}_\phi = \begin{bmatrix} \mathbf{V^T} & \mathbf{0} \\ \mathbf{0^T} & 1 \end{bmatrix}$.

Note that the projective transformation is position-dependent compared to the position-independent affine transformation. Care must be taken to ensure natural intermediate shapes by translating the source quadrilateral centroid to the origin and the target quadrilateral accordingly before the decomposition takes place. The significance of this procedure will be further explained in Section 4.1.

3.2. Interpolation

Let $\mathbf{u}_\alpha$ be the vertices of the interpolated quadrilaterals with $\alpha \in [0,1]$. $\mathbf{u_0} = \mathbf{p}$, are the source vertices and $\mathbf{u_1} = \mathbf{q}$ are the target vertices.

$$\mathbf{u}_\alpha = \mathbf{H}_\alpha \, \mathbf{p} \tag{6}$$

where $\mathbf{H}_\alpha = \mathbf{T}_\alpha \mathbf{R}_{\theta\alpha} \mathbf{R}_{-\phi} \mathbf{S}_\alpha \mathbf{R}_\phi \mathbf{P}_\alpha$ is the interpolated homography, with $\mathbf{H_0} = \mathbf{I}$, the identity matrix and $\mathbf{H_1} = \eta\mathbf{H}$.

The progressive change from identity matrix $\mathbf{I}$ to projective matrix $\mathbf{P}$ is achieved by interpolating their vanishing line vectors. Note that the vanishing line vector of $\mathbf{I}$ is $\mathbf{v_{vI}} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$ at infinity. We find the axis of rotation $\mathbf{v_{axis}}$ by taking the cross product of $\mathbf{v_{vI}}$ and $\mathbf{v_v}$. Then we find the cosine of the angle of rotation γ between the two vectors by taking the dot product of the two vectors. Finally we interpolate γ to get the intermediate vanishing line vector in $\mathbf{P}_\alpha$.

$$\begin{aligned} \mathbf{v_{axis}} &= \eta \begin{bmatrix} h_{32} & -h_{31} & 0 \end{bmatrix}^T \\ \gamma &= \cos^{-1}(\eta h_{33}) \\ \mathbf{P}_\alpha &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -h_{31}\sin(\alpha\,\gamma) & -h_{32}\sin(\alpha\,\gamma) & \cos(\alpha\,\gamma) \end{bmatrix} \end{aligned} \tag{7}$$

The rotations $\mathbf{R}_{\pm\phi}$ align the axes of scaling to the x and y directions and should not be interpolated. The scaling, rotation and translation interpolations are as follows.

$$\mathbf{S}_\alpha = (1-\alpha)\mathbf{I} + \alpha\mathbf{S} \tag{8}$$

$$\mathbf{R}_{\theta\alpha} = \begin{bmatrix} cos(\alpha\,\theta) & -sin(\alpha\,\theta) & 0 \\ sin(\alpha\,\theta) & cos(\alpha\,\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{9}$$

$$\mathbf{T}_\alpha = (1-\alpha)\mathbf{I} + \alpha\mathbf{T} \tag{10}$$

4. Normalisation

In applying a projective transformation, $\begin{bmatrix} sx' \\ sy' \\ s \end{bmatrix} = \begin{bmatrix} (h_{11}/h_{33}) & (h_{12}/h_{33}) & (h_{13}/h_{33}) \\ (h_{21}/h_{33}) & (h_{22}/h_{33}) & (h_{23}/h_{33}) \\ (h_{31}/h_{33}) & (h_{32}/h_{33}) & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$, the value of s is $(\frac{h_{31}x + h_{32}y}{h_{33}} + 1)$.

Empirically, we have found that large variation of the s value among the transformed vertices of a quadrilateral usually leads to unnatural excessive distortion of the transformed quadrilateral. A rule of thumb is to keep the magnitude of $\frac{h_{31}x + h_{32}y}{h_{33}}$ smaller than 0.5. Hence to keep s in the stable range, normalisations are needed and are presented in the following subsections.

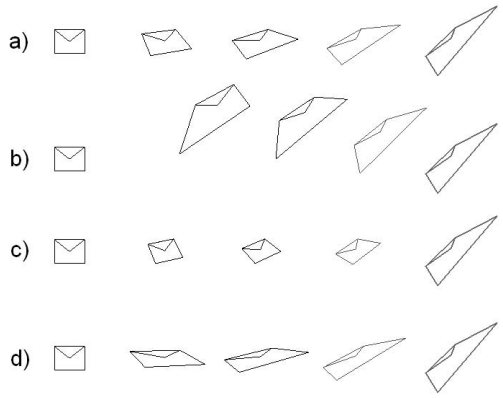

Figure 2. Projective interpolation of a plane and effects of normalisations.
(a) Perspectively correct transition with translation and vanishing line normalisation. (b) The source quadrilateral is translated away from the origin by about 75% of the quadrilateral size. (c) The source quadrilateral is scaled down such that the coordinate magnitudes are of the order of 0.1. (d) The homography is multiplied by 10, so that magnitude of the vanishing line vector is 10.

4.1. Translation Normalisation

If $\mathbf{p}$ and $\mathbf{q}$ are translated by $\mathbf{T_p}$ and $\mathbf{T_q}$ into $\mathbf{p^t}$ and $\mathbf{q^t}$, $\mathbf{p^t}$ and $\mathbf{q^t}$ are related by a new homography $\mathbf{H^t}$. All elements of $\mathbf{H^t}$ are different from elements of $\mathbf{H}$ except for h_{31} and h_{32}.

$$\begin{aligned} \mathbf{H^t} &= \mathbf{T_q}\,\mathbf{H}\,\mathbf{T_p^{-1}} \\ &= \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} + \begin{bmatrix} h_{31}t_{qx} & h_{32}t_{qx} & -h_{a13} \\ h_{31}t_{qy} & h_{32}t_{qy} & -h_{a23} \\ 0 & 0 & -h_{a33} \end{bmatrix} \end{aligned}$$

with $h_{a33} = h_{31}t_{px} + h_{32}t_{py}$
$h_{a13} = (h_{a33} + h_{33})t_{qx} + h_{11}t_{px} + h_{12}t_{py}$
$h_{a23} = (h_{a33} + h_{33})t_{qy} + h_{21}t_{px} + h_{22}t_{py}$

It is worth noting that h_{a33} is affected by the translation of the source quadrilateral but not of the target quadrilateral. The undesirable range of s can sometimes occur if $h_{a33} \simeq h_{33}$ due to translation of the source quadrilateral away from origin. Hence, in our implementation, the source quadrilateral is translated to be centered on the origin to avoid this possible instability.

4.2. Scale Normalisation

If $\mathbf{p}$ and $\mathbf{q}$ are scaled by $\mathbf{S_p}$ and $\mathbf{S_q}$ into $\mathbf{p^s}$ and $\mathbf{q^s}$, $\mathbf{p^s}$ and $\mathbf{q^s}$ are related by a new homography $\mathbf{H^s}$.

$$\begin{aligned}\mathbf{H^s} &= \mathbf{S_q}\,\mathbf{H}\,\mathbf{S_p^{-1}} \\ &= \begin{bmatrix} (\frac{s_{qx}}{s_{px}})h_{11} & (\frac{s_{qx}}{s_{py}})h_{12} & (s_{qx})h_{13} \\ (\frac{s_{qy}}{s_{px}})h_{21} & (\frac{s_{qy}}{s_{py}})h_{22} & (s_{qy})h_{23} \\ (\frac{1}{s_{px}})h_{31} & (\frac{1}{s_{py}})h_{32} & h_{33} \end{bmatrix}\end{aligned}$$

Again the vanishing line vector is only affected by the scaling of the source quadrilateral. The value of s will not normally be a problem due to scaling unless there is excessive scaling down of the source quadrilateral such that the coordinate magnitudes are of the order of 0.1 or smaller.

4.3. Vanishing Line Normalisation

The normalisation of $\eta\mathbf{H}$ to ensure the vanishing line vector $\mathbf{v_v}$ be of unit magnitude is also essential to avoid unnatural distortions.

When the normalisation of $\eta\mathbf{H}$ is applied, making $\mathbf{v_v}$ unit magnitude, its interpolation with the vanishing line of the identity matrix $\mathbf{v_{vI}}$, which is also of unit magnitude is simply an orientation interpolation, without any scale difference. Hence the interpolated homography $\mathbf{H}_\alpha$ has only one scaling component from $\mathbf{S}_\alpha$.

However, if normalisation of $\eta\mathbf{H}$ is not applied and $\mathbf{v_v}$ is of magnitude $\frac{1}{\|\eta\|}$, its interpolation with $\mathbf{v_{vI}}$ is then an orientation interpolation and also a scale interpolation. The interpolated homography $\mathbf{H}_\alpha$ will have two scaling components, one from $\mathbf{S}_\alpha$ and another from $\mathbf{P}_\alpha$. However, combining the two interpolated scaling matrices is not the same as interpolating the combined scaling matrices as the process is non linear. The former usually yields undesirable results. The proof is as follow.

Let $\mathbf{S_3} = \mathbf{S_1}\,\mathbf{S_2}$. The interpolations with the identity matrix $\mathbf{I}$ are:

$\mathbf{S_3}(t) = (1-t)\mathbf{I} + t(\mathbf{S_1}\mathbf{S_2})$

$\mathbf{S_1}(t)\,\mathbf{S_2}(t) = ((1-t)\mathbf{I} + t\mathbf{S_1})((1-t)\mathbf{I} + t\mathbf{S_2})$

Thus, $\mathbf{S_1}(t)\,\mathbf{S_2}(t) \neq \mathbf{S_3}(t)$ for all $\mathbf{S_1} \neq \mathbf{I}$ or $\mathbf{S_2} \neq \mathbf{I}$

5. Interpolation of Multiple Quadrilaterals

The projective interpolation of quadrilaterals can be used as a primitive where triangles are traditionally used, such as in image transition applications. It has the advantages of preserving the physical correctness of projective transformation and requiring a smaller number of primitives.

Similar to triangle interpolation, the interpolation of adjacent quadrilaterals suffer from disconnected boundaries when they have different transformations. We propose a closed form force-equilibrium post processing scheme to line up the disconnected intermediate adjacent quadrilaterals and to minimise distortions.

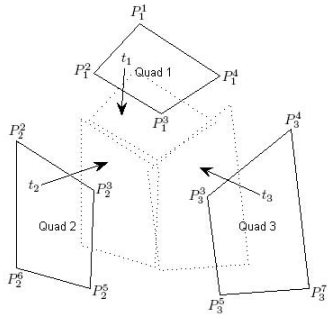

Figure 3. Force-equilibrium quadrilateral alignment

5.1. Force-equilibrium Scheme

Assuming that we have N point pairs and N_q quadrilateral pairs from the source and the target mesh, each quadrilateral i is described by a list of 4 point indices $\mathrm{QInd_i}$. $i \in \{1, \cdots, N_q\}$ and $\mathrm{QInd_i} \in \{1, \cdots, N\}$. Let P_i^j represent the intermediate interpolation of point j in quadrilateral i. $j \in \mathrm{QInd_i}$. For each point j in $\mathrm{QInd_i}$, we can derive a list of quadrilaterals that share that point, we define it as $\mathrm{SQ_j}$. $\mathrm{SQ_j} \in \{1, \cdots, N_q\}$.

In the Figure 3 example, the point indices are $\mathrm{QInd}_1 = \{1,2,3,4\}$, $\mathrm{QInd}_2 = \{2,3,5,6\}$, and $\mathrm{QInd}_3 = \{3,4,7,5\}$. The shared point lists are $\mathrm{SQ}_1 = \{1\}$, $\mathrm{SQ}_2 = \{1,2\}$, $\mathrm{SQ}_3 = \{1,2,3\}$, $\mathrm{SQ}_4 = \{1,3\}$, $\mathrm{SQ}_5 = \{2,3\}$, $\mathrm{SQ}_6 = \{2\}$, and $\mathrm{SQ}_7 = \{3\}$.

We define a translation term for each quadrilateral, t_i. So the translated version of the vertices of quadrilateral i is $(P_i^j + t_i)$. To align the quadrilaterals, we assign a force proportional to the distance between each translated disconnected vertex, and we minimise the total sum of the forces F_i^t acting on each quadrilateral.

$$\begin{aligned}F_i^t &= \sum_j^{\mathrm{QInd_i}} \sum_k^{\mathrm{SQ_j}} \Big((P_i^j + t_i) - (P_k^j + t_k) \Big) \\ &= \sum_j^{\mathrm{QInd_i}} \sum_k^{\mathrm{SQ_j}} (P_i^j - P_k^j) + \sum_j^{\mathrm{QInd_i}} \sum_k^{\mathrm{SQ_j}} (t_i - t_k) \quad (11)\end{aligned}$$

At force equilibrium, F_i^t equals to zero. So we have N_q equations. They can be rewritten into matrix form as

$$\mathbf{C}\,\mathbf{T_{eq}} = \mathbf{F_P}$$

where

$$\mathbf{C} = \begin{bmatrix} c_{11} & c_{12} & \dots & c_{1N_q} \\ c_{21} & c_{22} & \dots & c_{2N_q} \\ \vdots & \vdots & \ddots & \vdots \\ c_{N_q1} & c_{N_q2} & \dots & c_{N_qN_q} \end{bmatrix}$$

$$c_{rc} = \begin{cases} \sum_j^{\mathrm{QInd_i}} \sum_k^{\mathrm{SQ_j}} 1 & \text{for } r{=}c{=}i \\ -\sum_j^{\mathrm{QInd_i}} \sum_k^{\mathrm{SQ_j}} 1 & \text{for } r{=}i, c{=}k \\ 0 & \text{otherwise} \end{cases} \quad (12)$$

$$\mathbf{T_{eq}} = \begin{bmatrix} t_1 & t_2 & \dots & t_{N_q} \end{bmatrix}^T \quad (13)$$

$$\mathbf{F_P} = \begin{bmatrix} -\sum_j^{\mathrm{QInd_1}} \sum_k^{\mathrm{SQ_j}} \left(P_1^j - P_k^j\right) \\ -\sum_j^{\mathrm{QInd_2}} \sum_k^{\mathrm{SQ_j}} \left(P_2^j - P_k^j\right) \\ \vdots \\ -\sum_j^{\mathrm{QInd_{N_q}}} \sum_k^{\mathrm{SQ_j}} \left(P_{N_q}^j - P_k^j\right) \end{bmatrix} \quad (14)$$

Note that $\mathbf{C}$ has a rank of $(N_q - 1)$, so we need another constraint for a unique solution. Hence we specify that the net translation is zero.

$$\sum_{i=1}^{N_q} t_i = 0$$

The translations for all quadrilaterals can be obtained by

$$\mathbf{T_{eq}} = \mathbf{C^{-1}}\,\mathbf{F_P} \quad (15)$$

The final vertex position is taken as an average of the translated disconnected vertex positions at force-equilibrium.

6. Results and Discussions

Figure 2 illustrates the single transformation projective interpolation with several normalisations. We started with the source and the target quadrilaterals and our algorithm generates the transitions. Figure 2(a) shows the perspectively correct transition with translation and vanishing line normalisation. Figure 2(b) shows the unnatural path and intermediate shapes when the source quadrilateral is translated away from the origin. Figure 2(c) shows the unnatural intermediate shapes when the source quadrilateral is scaled down such that the coordinate magnitudes are of the order of 0.1. Figure 2(d) shows the unnatural intermediate shapes when the homography is multiplied by 10 so that magnitude of the vanishing line vector is 10.

From the results in Figure 2, it is worth noting that there are many paths and shapes a projective interpolation can go through from the source to the target. This depends on the translation and scaling of the source points and scaling of the homography before the decomposition. Translation has an especially strong effect on the quality of the paths and shapes while scaling affects the temporal linearity of the transitions.

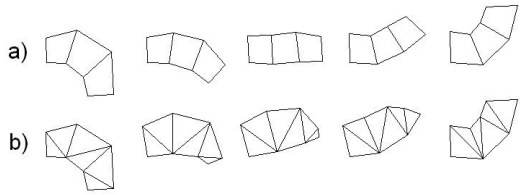

Figure 4. Non-rigid object shape interpolation. a) Projective interpolation with quadrilaterals, b) Alexa's as-rigid-as-possible affine interpolation with triangles.

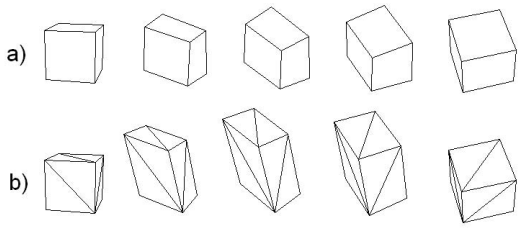

Figure 5. Rigid object shape interpolation. a) Projective interpolation with quadrilaterals, b) Alexa's as-rigid-as-possible affine interpolation with triangles.

Defining the quality of the transition is, of course, subjective. A quantitative measure of morph quality based on epipolar geometry is presented by Wong et al [8] which we hope to eventually apply here.

Figure 4 shows a comparison of our method with Alexa's as-rigid-as-possible approach when applied to a non-rigid shape interpolation. We did not do the isomorphic dissection for the triangles to illustrate the shortcomings of the affine interpolation approach. The transition generated by our algorithm is more rigid and natural. Similarly, Figure 5 shows the comparison between the two approaches through an interpolation between two views of a cube. The affine triangle approach yields an unnatural result, while the projective quadrilateral approach produces a much better approximation.

Figure 6 shows the single transformation projective interpolation applied on an image of sign post with 5 pairs of separate homographies for 5 planar surfaces. The effect is that the signs are swapped with each other and appear like flying boards in the transition. This could not be achieved using affine interpolation, and our algorithm works without having any 3D information other than perspectively related quadrilaterals.

Figure 6. Projective interpolation applied on a real image of sign post with 5 pairs of separate homographies for 5 planar surfaces. The effect is that the signs are swapped among each other and appear like flying boards.

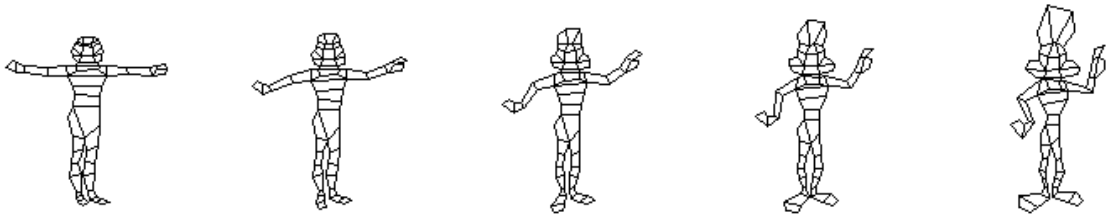

Figure 7. Projective shape interpolation from da Vinci's Vitruvian Man to Coyote.

Figure 7 shows the projective shape interpolation applied on a more complex shape transformation. The transition is smooth and convincing although it only uses 47 quadrilateral pairs. The number of triangles required for the same shape interpolation using Alexa's as-rigid-as-possible method would be many times more.

7. Conclusion and Future Directions

In this paper, we successfully extend matrix decomposition and interpolation techniques so that, rather than just being applicable to affine transformation, they can be also applied to general projective transformation. This enables perspectively correct interpolation of image patches to be generated without 3D information.

We suggest normalisation of the projective matrix before decomposition and position normalisation of the source points for better interpolation results. We present an efficient force-equilibrium method to handle disconnected intermediate quadrilaterals and enable quadrilaterals to be used as primitives for more complex shape interpolation and morphing applications.

Future research directions include the optimisation of the source position for optimal interpolation and exploration of applications of this technique for shape interpolation and image morphing.

References

[1] M. Alexa. Recent advances in mesh morphing. *Computer Graphics Forum*, 21(2):173–198, 2002.

[2] M. Alexa, D. Cohen-Or, and D. Levin. As-rigid-as-possible shape interpolation. In K. Akeley, editor, *Siggraph 2000, Computer Graphics Proceedings*, pages 157–164. ACM Press/Addison-Wesley Publishing Co., 2000.

[3] J. Gomes, L. Darsa, B. Costa, and L. Velho. *Warping and morphing of graphical objects*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1998.

[4] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[5] S. M. Seitz and C. R. Dyer. Physically-valid view synthesis by image interpolation. In *Proc. Workshop on Representation of Visual Scenes*. IEEE Computer Society Press, June 1995.

[6] S. M. Seitz and C. R. Dyer. View morphing. *Computer Graphics*, 30(Annual Conference Series):21–30, 1996.

[7] K. Shoemake and T. Duff. Matrix animation and polar decomposition. In *Proceedings of the conference on Graphics interface '92*, pages 258–264, San Francisco, CA, USA, 1992. Morgan Kaufmann Publishers Inc.

[8] T. Wong, P. Kovesi, and A. Datta. Towards quantitative measures of image morphing quality. In *DICTA*, page 19. IEEE Computer Society, 2005.

[9] J. Xiao, C. Rao, and M. Shah. View interpolation for dynamic scenes. In *EUROGRAPHICS 2002*, pages 360–372, Saarbrücken, Germany, 2002.