Parametric View-Synthesis

Miles E. Hansard and Bernard F. Buxton

Department of Computer Science, University College London,
Gower Street, London, WC1E 6BT.
Tel. +44 (0)171 380 7294
Fax. +44 (0)171 387 1397
{m.hansard, b.buxton}@cs.ucl.ac.uk

Abstract. We present a simple procedure for synthesising novel views, using two or more basis-images as input. It is possible for the user to interactively adjust the viewpoint, and for the corresponding image to be computed and rendered in real-time. Rather than employing a 3D model, our method is based on the linear relations which exist between images taken with an affine camera. We show how the 'combination of views' proposed by Ullman and Basri [19] can be appropriately *parameterised* when a sequence of five or more images is available. This is achieved by fitting polynomial models to the coefficients of the combination, where the latter are functions of the (unknown) camera parameters. We discuss an alternative approach, direct image-interpolation, and argue that our method is preferable when there is a large difference in orientation between the original gaze directions. We show the results of applying the parameterisation to a fixating camera, using both simulated and real input. Our observations are relevant to several applications, including visualisation, animation, and low-bandwidth communication.

1 Introduction

The World Wide Web presents many novel opportunities for the display of information from remote sources [10]. For example, consider the possibility of a *virtual museum*, in which visitors can interactively inspect the exhibits. There are two particularly important factors in the design of software for such applications: Firstly, the visual appeal of the experience may be more important than the veridicality of the display. Secondly, the widespread adoption of a visualisation method would depend on the flexibility of the data-capture process.

These needs are well served by *image-based* approaches [21], as opposed to the acquisition and rendering of 3D models. Ideally, we would like to use a small number of input images to specify a scene, within which we can manipulate a 'virtual viewpoint'. Although this question has been addressed before [7], we aim to present a method which is more general than direct image-interpolation [12], while being less complicated than tensor-reprojection [1].

Our approach is based on the *linear combination of views* theory, which was originally proposed by Ullman and Basri [19] in the context of object-recognition. As has been noted elsewhere, the orthographic camera employed by Ullman

D. Vernon (Ed.): ECCV 2000, LNCS 1842, pp. 191–202, 2000.

and Basri provides a point of contact with certain methods of visual motion estimation [15,13]. In particular, Tomasi and Kanade [17] have shown that the structure-from-motion problem can be defined by assembling the feature coordinates from each available image into a single matrix of measurements. The implicit 3D structure and motion can then be recovered as a particular factorisation of this *joint-image*.

These considerations are important, because the joint-image approach extends naturally to the treatment of any number of views, obtained under perspective projection [18]. However, (triplets of) perspective views are related by trilinear equations, rather than by factorisation or direct combination [14,3]. We emphasise that the present work is restricted to an *affine* imaging model, which admits of a more straightforward treatment.

2 Linear Combinations of Views

Consider a series of images, $\mathcal{I}_t$, taken with an affine camera $\mathbf{C}$ at 'times' $t = 1, 2, \ldots, T$. If we allow the position, orientation and intrinsic parameters of the camera to vary from one picture to the next, we can describe the T different projections of a particular feature as

$$\begin{bmatrix} x_t \\ y_t \end{bmatrix} = \mathbf{C}_t \begin{bmatrix} \mathrm{x} \\ \mathrm{y} \\ \mathrm{z} \\ 1 \end{bmatrix} \qquad t = 1, 2, \ldots, T. \tag{1}$$

where $[x_t \ y_t]^\top$ are the coordinates of the scene-point $[\mathrm{x\ y\ z\ 1}]^\top$, as it appears in the t-th image. In accordance with the affine imaging-model, $\mathbf{C}_t$ is a 2×4 matrix.

Given the T images, Ullman and Basri [19] have shown that it is also possible to obtain $[x_t \ y_t]^\top$ without direct reference to the 3D scene. This follows from the fact that the geometry of an affine view can be represented as a linear combination of '$1\frac{1}{2}$' other affine views. In practice, we chose to employ an overcomplete representation, which allows for a symmetric treatment of the basis images [2]. For convenience we will define the basis images as $\mathcal{I}' \equiv \mathcal{I}_1$ and $\mathcal{I}'' \equiv \mathcal{I}_T$, although we could in principle make different choices and, likewise, we could employ more than two basis images [6]. The coordinates of a particular feature in the target image, $\mathcal{I}_t$, are then expressed as

$$\begin{aligned} x_t &= a_0 + a_1 x' + a_2 y' + a_3 x'' + a_4 y'' \\ y_t &= b_0 + b_1 x' + b_2 y' + b_3 x'' + b_4 y''. \end{aligned} \tag{2}$$

We can form these equations for every feature in $\mathcal{I}_t$, and subsequently obtain the coefficients a_n and b_n expressing the least-squares estimate of the target geometry. However, it should be noted that the numerical rank of this system is dependent on the scene, the imaging model, and the relationship between the camera matrices $\mathbf{C}_t$, $\mathbf{C}'$ and $\mathbf{C}''$ (corresponding to $\mathcal{I}_t$, $\mathcal{I}'$ and $\mathcal{I}''$, respectively).

For example, if $\mathcal{I}_t$ and $\mathcal{I}'$ are related by an affine transformation of the image-plane, then $\mathcal{I}''$ is redundant. For this reason, the matrix *pseudoinverse* is used to obtain stable estimates of the coefficients a_n and b_n.

Several other issues must be addressed before equation (2) can be used as a practical view-synthesis procedure. Firstly, we must identify five or more corresponding features in the T images — this process was performed manually in the present study. Secondly, we must derive a means of *rendering* the synthetic image, while resolving those features in $\mathcal{I}'$ and $\mathcal{I}''$ which are *occluded* from the new viewpoint. These problems will be considered in §5.

A more serious limitation is that the coefficients a_n and b_n cannot be estimated without reference to the 'target' coordinates $[x_t \ y_t]^\top$. In other words, equation (2) only allows us to synthesise those pictures which we have *already* taken.

As outlined in the introduction, we would rather regard the T existing images as *samples* taken from a continuous motion of the camera. Our aim then, is to generate a convincing *movie*, $\mathcal{I}(\tau)$, where certain values of the parameter τ will yield the original basis images, while intermediate values will yield plausible intermediate views.

3 Image Interpolation

To formalise the definition of $\mathcal{I}(\tau)$, suppose that $0 \leq \tau \leq 1$. For consistency with the two basis images, we impose the following conditions:

$$\tau = 0 \Longrightarrow \begin{cases} x(\tau) = x' \\ y(\tau) = y' \end{cases} \qquad \tau = 1 \Longrightarrow \begin{cases} x(\tau) = x'' \\ y(\tau) = y'' \end{cases} \tag{3}$$

These requirements can be satisfied by a linear interpolation procedure, performed directly in the image domain:

$$\begin{bmatrix} x(\tau) \\ y(\tau) \end{bmatrix} = (1-\tau) \begin{bmatrix} x' \\ y' \end{bmatrix} + \tau \begin{bmatrix} x'' \\ y'' \end{bmatrix}. \tag{4}$$

The problem with such a scheme is that the intermediate images are not at all constrained by the conditions imposed via the basis views. Moreover, equation (4) is by no means guaranteed to produce a convincing movie. For example, consider what happens when the two cameras, $\mathbf{C}'$ and $\mathbf{C}''$, differ by a rotation of 180° around the line of sight; as τ is varied, every feature will travel linearly through the centre of the image. Consequently, at frame $\mathcal{I}(\frac{1}{2})$, the image collapses to a point.

In fact, Seitz and Dyer [12] have shown that if $\mathbf{C}'$ and $\mathbf{C}''$ represent *parallel* cameras[1] then direct interpolation will produce a valid intermediate view. However, because it is difficult to ensure that the uncalibrated cameras are in this configuration, it becomes necessary to *rectify* the basis images.

[1] If $\mathbf{C}''$ can be obtained by displacing $\mathbf{C}'$ along a direction orthogonal to its optic-axis, then the two cameras are said to be parallel.

The disadvantage of this method is that computationally, it is both complicated and expensive. The epipoles must be estimated in order to compute the rectifying homographies[2] — and these transformations remain underdetermined without the use of further constraints. Once these have been specified, $\mathcal{I}'$ and $\mathcal{I}''$ must be rectified, and the interpolation performed. Finally, the novel view has to be de-rectified. In order to generate a valid movie, the rectifying transformations (as well as the images themselves) would have to be interpolated. A detailed discussion of the rectification process can be found in [4].

Finally, we note that Pollard et al. [9] have extended just such an interpolation scheme to the three-camera case, and demonstrated good results without performing the rectification stage. However, this is not viable when there is a large difference in orientation between $\mathbf{C}'$ and $\mathbf{C}''$, as is the case in many applications.

4 Parametric Synthesis

In this section we will describe an alternative approach to the generation of $\mathcal{I}(\tau)$. It is based on the observation that the coefficients a_n and b_n in equation (2) are functions of the camera parameters [15]. It follows that if the camera-motion can be parameterised as $\mathbf{C}(\tau)$, then there must exist functions $a_n(\tau)$ and $b_n(\tau)$, which link the virtual viewpoint to $\mathcal{I}_t$, via (2). For example, suppose that we have a prior model of an orthographic camera, which is free to rotate through the range 0 to ϕ in the horizontal-plane, while fixating a centrally positioned object. If τ is used to specify the viewing angle, then we can define the parameterised camera-matrix

$$\mathbf{C}(\tau) = \begin{bmatrix} \cos(\tau\phi) & 0 & \sin(\tau\phi) & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}. \tag{5}$$

For images generated by such simple cameras, closed-form expressions can be obtained for $a_n(\tau)$ and $b_n(\tau)$ [19]. However, real, uncalibrated images are unlikely to be consistent with such a model. For this reason, we propose to *derive* a parameterisation from the original image sequence.

Because the constraints given in (3) can be applied to the view-synthesis equation (2), the unknown functions $a_n(\tau)$ and $b_n(\tau)$ must satisfy the following requirements:

$$\begin{aligned} a_1(0) &= 1, & a_3(1) &= 1, \\ a_n(0) &= 0,\ n \neq 1, & a_n(1) &= 0,\ n \neq 3, \\ b_2(0) &= 1, & b_4(1) &= 1, \\ b_n(0) &= 0,\ n \neq 2, & b_n(1) &= 0,\ n \neq 4. \end{aligned} \tag{6}$$

[2] In principle, affine imaging produces parallel epipolar lines within each image. However, this cannot be assumed in the treatment of real images.

These conditions allow us to propose functional forms for the variation of the coefficients. Where possible, we begin with *linear* models:

$$a_1(\tau) = b_2(\tau) = 1 - \tau, \tag{7}$$

$$a_3(\tau) = b_4(\tau) = \tau. \tag{8}$$

The remaining coefficients are tied to zero at both extremes of τ. A simple hypothesis generates the equations

$$a_n(\tau) = \alpha_n \tau(1-\tau),\ n \neq 1,\ 3, \tag{9}$$

$$b_n(\tau) = \kappa_n \tau(1-\tau),\ n \neq 2,\ 4, \tag{10}$$

where α_n and κ_n are (possibly negative) scalars.

4.1 Estimating the Viewpoint Parameter

We now have prototypes for the variation of the coefficients, as functions of the viewpoint parameter, τ. The constraints imposed above ensure that the basis-views are correctly represented, but we also have $T-2$ intermediate samples from each of these functions, in the form of the a and b coefficients expressing each 'keyframe', $\mathcal{I}_t$, in terms of $\mathcal{I}'$ and $\mathcal{I}''$. However, we do *not* know the actual value of τ corresponding to each image $\mathcal{I}_t$ — we will refer to this unknown as τ_t, with $t = 1, \ldots, T$, as before. In practice, we can estimate τ_t by assuming that our simple hypotheses (7–8) are valid, in a least-squares sense, for a or b, or for both. For example, we can posit that $a_3(\tau) = \tau$, and that $a_1(\tau) = 1 - \tau$. From this overcomplete specification we can obtain an estimate, $\hat{\tau}_t$ of τ_t according to

$$\hat{\tau}_t = \min_{\tau} \Big((a_3 - \tau)^2 + (a_1 + \tau - 1)^2 \Big),$$

where the a coefficients (estimated via the pseudoinverse, as indicated in §2) are particular to each target image $\mathcal{I}_t$. The solution of the least-squares problem is, of course

$$\hat{\tau}_t = \frac{1}{2}(1 - a_1 + a_3), \tag{11}$$

subject to the validity of equations (7) and (8). The use of the a coefficients in the above procedure is governed by the expectation that they will show significant variation during a horizontally oriented motion of the camera. If the path were closer to a vertical plane, then it would be preferable to use the b coefficients instead. In general, we could use a least-squares estimate (if necessary, weighted) over all four coefficients in (7–8), where the latter would be estimated via the pseudoinverse, as before.

4.2 Modelling the Coefficients

Should the the hypothesised linear and quadratic equations (7–10) prove poor approximations to the behaviour of the coefficients, we can add further terms,

obeying (6), as follows. Consider the *cubic* models

$$a_1(\tau) = 1 - \tau + \epsilon(1-\tau)^3, \qquad a_3(\tau) = \tau + \zeta\tau^3, \tag{12}$$

$$b_2(\tau) = 1 - \tau + \nu(1-\tau)^3, \qquad b_4(\tau) = \tau + \xi\tau^3 \tag{13}$$

and

$$a_n(\tau) = \alpha_n\tau(1-\tau) + \beta_n\tau^2(1-\tau) + \gamma_n\tau(1-\tau)^2, \quad n \neq 1, 3, \tag{14}$$

$$b_n(\tau) = \kappa_n\tau(1-\tau) + \lambda_n\tau^2(1-\tau) + \mu_n\tau(1-\tau)^2, \quad n \neq 2, 4. \tag{15}$$

We summarise the fitting procedure as follows: For each target $\mathcal{I}_t$, we use equation (2) to compute the coefficients of the linear combination, a_n and b_n. Next we use equation (11) to estimate the value of τ_t corresponding to each $\mathcal{I}_t$, where the latter are viewed as 'keyframes' in the movie, $\mathcal{I}(\tau)$. This enables us to use our $T-2$ samples $\{a_n, b_n\}$ to estimate the functions $a_n(\tau)$ and $b_n(\tau)$.

For example, from (14), we have the following cubic hypothesis for the variation of coefficient $a_{n\neq 1,3}$ over the T target images

$$\begin{bmatrix} \hat{a}_n(\hat{\tau}_1) \\ \hat{a}_n(\hat{\tau}_2) \\ \vdots \\ \hat{a}_n(\hat{\tau}_T) \end{bmatrix} = \begin{bmatrix} \hat{\tau}_1(1-\hat{\tau}_1) & \hat{\tau}_1{}^2(1-\hat{\tau}_1) & \hat{\tau}_1(1-\hat{\tau}_1)^2 \\ \hat{\tau}_2(1-\hat{\tau}_2) & \hat{\tau}_2{}^2(1-\hat{\tau}_2) & \hat{\tau}_2(1-\hat{\tau}_2)^2 \\ \vdots & \vdots & \vdots \\ \hat{\tau}_T(1-\hat{\tau}_T) & \hat{\tau}_T{}^2(1-\hat{\tau}_T) & \hat{\tau}_T(1-\hat{\tau}_T)^2 \end{bmatrix} \begin{bmatrix} \alpha_n \\ \beta_n \\ \gamma_n \end{bmatrix}. \tag{16}$$

Such an equation can be solved by standard least-squares techniques, such as Cholesky decomposition. Note that the system is exactly determined when $T = 5$, resulting in three 'keyframe' equations per a_n, and likewise three per b_n.

5 Rendering

Section 4 described a method for estimating the positions of image features in novel views. A significant advantage of the linear-combinations approach is that good results can be obtained from a small number of control-points (we typically used around 30–70 points per image).

However, when we come to *render* the novel view, we must obtain a value for every pixel, not just for the control-points. One way to fill-in $\mathcal{I}(\tau)$ is to compute the regularised *optic-flow* from the basis images [1], although this incurs a large computational cost. Furthermore, the estimation of optic-flow is error-prone, and any outlying pixels will noticeably disrupt the new image. We therefore prefer to use a simple *multiview-warping* scheme [2], as described below.

5.1 Image Triangulation and Warping

In order to render the novel view, we use the estimated positions of the *control-points* to determine a mapping from each of the basis *images*. This can be achieved by performing a systematic triangulation of the control points, as they appear in the current frame of $\mathcal{I}(\tau)$. The triangulation is then transferred to the

control-points in each basis view, thus ensuring the consistency of the mapping over all images.

We employed a constrained Delaunay[3] routine [16], which allows us to guarantee that different image-regions will be represented by different triangles. Furthermore, the constraints can be used to effectively separate the object from the background, by imposing an arbitrary boundary on the triangulation.

Once the triangulation has been transferred to $\mathcal{I}'$ and $\mathcal{I}''$, we can use it to warp each basis-image into registration with the novel view $\mathcal{I}(\tau)$. In fact, this is a special case of the *texture-mapping* problem, and was implemented as such, using the OpenGL graphics library [11]. Because each mapping is piecewise-affine, the intra-triangle interpolation is linear in screen-space, which simplifies the procedure. Finally, bilinear interpolation was used to resolve the values of non-integer coordinates in $\mathcal{I}'$ and $\mathcal{I}''$.

5.2 Computation of Intensities

In the previous section, we showed how the basis views can be warped into registration, such that corresponding image regions are aligned. The novel view can now be rendered as a weighted sum of the (warped) basis images [6]:

$$\mathcal{I}(\tau) = w'\mathcal{I}' + w''\mathcal{I}''. \tag{17}$$

As will be described below, the weights w' and w'' are also functions of τ, although this property will not be made explicit in the notation.

If the present frame of $\mathcal{I}(\tau)$ happens to coincide with either $\mathcal{I}'$ or $\mathcal{I}''$, then the other basis image should not contribute to the rendering process. This imposes the following requirements on w' and w'':

$$\mathcal{I}(\tau) = \mathcal{I}' \Longrightarrow \begin{cases} w' = 1 \\ w'' = 0 \end{cases} \qquad \mathcal{I}(\tau) = \mathcal{I}'' \Longrightarrow \begin{cases} w' = 0 \\ w'' = 1 \end{cases} \tag{18}$$

In fact, it makes physical sense to impose the additional constraint $w' + w'' = 1$, such that equation (17) becomes a barycentric combination. Using the results of §4, it would be possible to define the weights as $w' = 1 - \tau$ and $w'' = \tau$. However, this would be unsatisfactory, because τ specifies the notional 3D viewpoint, which may be a poor measure of the 2D image relationships. For this reason, we follow [2], and use equation (2) to derive appropriate weights. Specifically, we define distances d' and d'' of $\mathcal{I}(\tau)$ from $\mathcal{I}'$ and $\mathcal{I}''$ respectively:

$$d'^2 = a_3(\tau)^2 + a_4(\tau)^2 + b_3(\tau)^2 + b_4(\tau)^2 \tag{19}$$

$$d''^2 = a_1(\tau)^2 + a_2(\tau)^2 + b_1(\tau)^2 + b_2(\tau)^2. \tag{20}$$

[3] Once the triangulation has been transferred to another point-set, only the *adjacency* properties are necessarily preserved.

As in [2,6], we then compute the weights according to

$$w' = \frac{d''^2}{d'^2 + d''^2} \qquad w'' = \frac{d'^2}{d'^2 + d''^2}. \tag{21}$$

Having satisfied conditions (18), we can now compute the novel view, using equation (17). This potentially slow operation was implemented via the OpenGL accumulation buffer, which operates in hardware (where available). Clearly the method generalises immediately to colour images, by treating each spectral band as a luminance component [2].

5.3 Hidden-Surface Removal

An issue which must be addressed by all view-synthesis schemes is the treatment of missing or inconsistent data during the rendering process. Clearly, it is only possible to portray those features in $\mathcal{I}(\tau)$ which were present in at least one basis-image. If a new feature does appear, then the method described above renders a mixture of the (spatially) corresponding regions in the basis-images, thereby ensuring that, though it may be incorrect, each frame of $\mathcal{I}(\tau)$ is at least continuous.

The problem of consistency is more tractable, as it is possible to remove any parts of the basis-images which are *occluded* from the novel viewpoint. Moreover, although occlusion events are generated by the 3D nature of the scene, it does not follow that we have to explicitly recover the 3D structure in order to resolve them. Rather, it is sufficient to obtain the *affine depth* [5] of each control-point. Geometrically, this quantity can be understood as the relative deviation from a notional world plane, where the latter is defined by three distinguished points in the scene. It is straightforward to compute the affine depth, using the positions of the control points in two basis-images, together with the constraint that the three distinguished points have affine depths equal to zero. Once the measurements have been made at the control-points, we use the standard z-buffer algorithm to interpolate the depth over each triangle, and to remove the occluded regions.

A further consequence of this process is that the combination of views (2) can be recomputed at each of the $T - 2$ keyframes, excluding those control-points which occupy hidden surfaces in $\mathcal{I}_t$. This should lead to a more accurate synthesis, because the occluded points are likely to be *outliers* with respect to the original estimate [2].

6 Results

The methods which have been described were tested by simulation, using 3D models defined in the OpenGL environment. Several scenes (comprising arrangements of cubes) were imaged, while the viewpoint was subject to systematic variation. Both perspective and affine projections were recorded, and the matrix pseudoinverse was used to solve equation (2). As well as exact control of the ca-

mera model, the simulation procedure provides noiseless image coordinates for each control-point, and allows us to monitor occlusion effects.

The graphs in figures 1 and 2 show how the coefficients of the linear combination evolve as the (perspective) camera rotates through 90° in a horizontal plane. In the case of *real* images, it is likely that these curves would be disrupted by occlusion effects. As described in §5.3, it would be possible to avoid this, although we have not yet implemented the necessary re-estimation procedure.

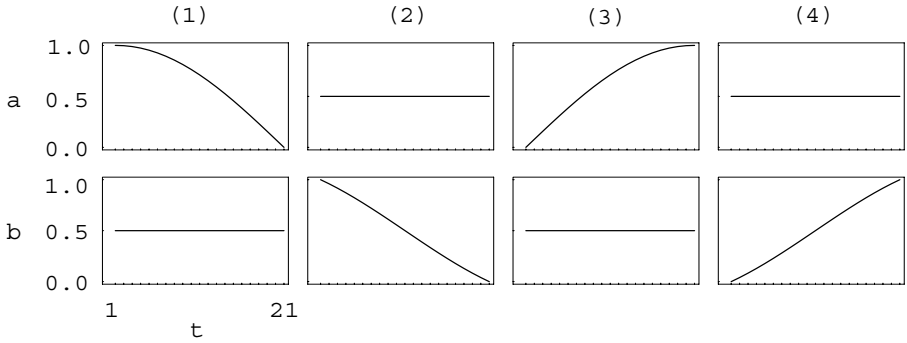

Fig. 1. Variation of the linear-combination coefficients (top; a_1,a_2,a_3,a_4; bottom; b_1,b_2,b_3,b_4). The simulated camera was rotated through 90° around the vertical axis of a group of six cubes. $T = 21$ pictures were taken in total. The functions obey the conditions (6), as expected, though $a_{2,4}$ and $b_{1,3}$ (which were zero) have been shifted up by 0.5 for the purpose of this display. The nonzero variation of the $b_{2,4}$ coefficients is attributable to perspective effects.

When plotted on the same axes, the fitted models (12–15) are indistinguishable from the measured curves shown in figures 1 and 2. For reasons of space, we proceed directly to the final error-measures, produced by using the fitted models to position the control-points in the existing T images. The two graphs in figure 3 show the r.m.s. discrepancy (in x and y respectively) between the estimated and actual control-points. For $t = 1$ and $t = 21$ the error is zero, because $\mathcal{I}' = \mathcal{I}_1$ and $\mathcal{I}'' = \mathcal{I}_{21}$ respectively.

We have not yet implemented the estimation of τ_t (as described in §4.1) from a real image sequence. However, preliminary results have been obtained by applying the functions $a_n(\tau)$ and $b_n(\tau)$, obtained by simulation, to real pairs of basis-images. For example, we produced the novel views shown in figure 4 by applying the functions shown in figures 1 and 2 to the two framed images. Once the polynomial models $a_n(\tau)$ and $b_n(\tau)$ have been obtained, they can be resampled to yield an arbitrary number of intermediate images.

In the light of our comments in §3, we note that the example shown in figure 4 is rather straightforward, in that the camera-motion is extremely simple.

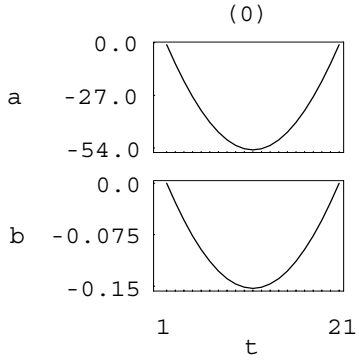

Fig. 2. Variation of coefficients a_0 (top) and b_0 (bottom). These are the constant terms in the linear combination (2), which are not of comparable magnitude to the coefficients plotted in fig. 1.

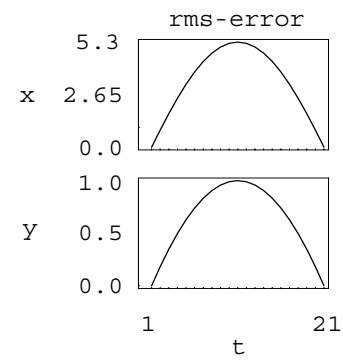

Fig. 3. Mean error of the parametric-synthesis, measured in x (top) and y (bottom). The units are pixels, where the original images were of size 256×256.

Nonetheless, rotation around the object is a typical requirement in visualisation applications.

As far as the image-quality is concerned, the results seem promising; in particular, we note that the left and right edges of the mask are appropriately culled, according to the viewpoint. The constrained triangulation also seems to perform adequately, although there is some distortion of the top-right edge of the mask during the early frames. This may indicate that insufficient control-points were placed around the perimeter.

7 Conclusions and Future Work

We have described a simple method of parametric view-synthesis, developed with the requirements of Web-based visualisation in mind. Our results are also relevant to other applications of the linear combinations of views theory, including animation [8] and low-bandwidth video-transmission [6].

In this outline we have concentrated on simple motions of the camera, because these are typical of visualisation applications. In principle, the same approach could be applied to general conjunctions of rotation and translation. However, it is to be expected that arbitrary motions will add complexity to the functions $a_n(\tau)$ and $b_n(\tau)$, which may demand the addition of further polynomial terms to the model (12–15). Arbitrary camera trajectories may also require the *speed* of the parameterisation to be regulated.

In future, it may be possible to extend the present method to cover a *region* of the view-sphere, rather than just a particular path of the camera; this would require two parameters, $\{\theta, \phi\}$, in place of τ. Because of the increased variety of intermediate images, such an extension would also require the use of more

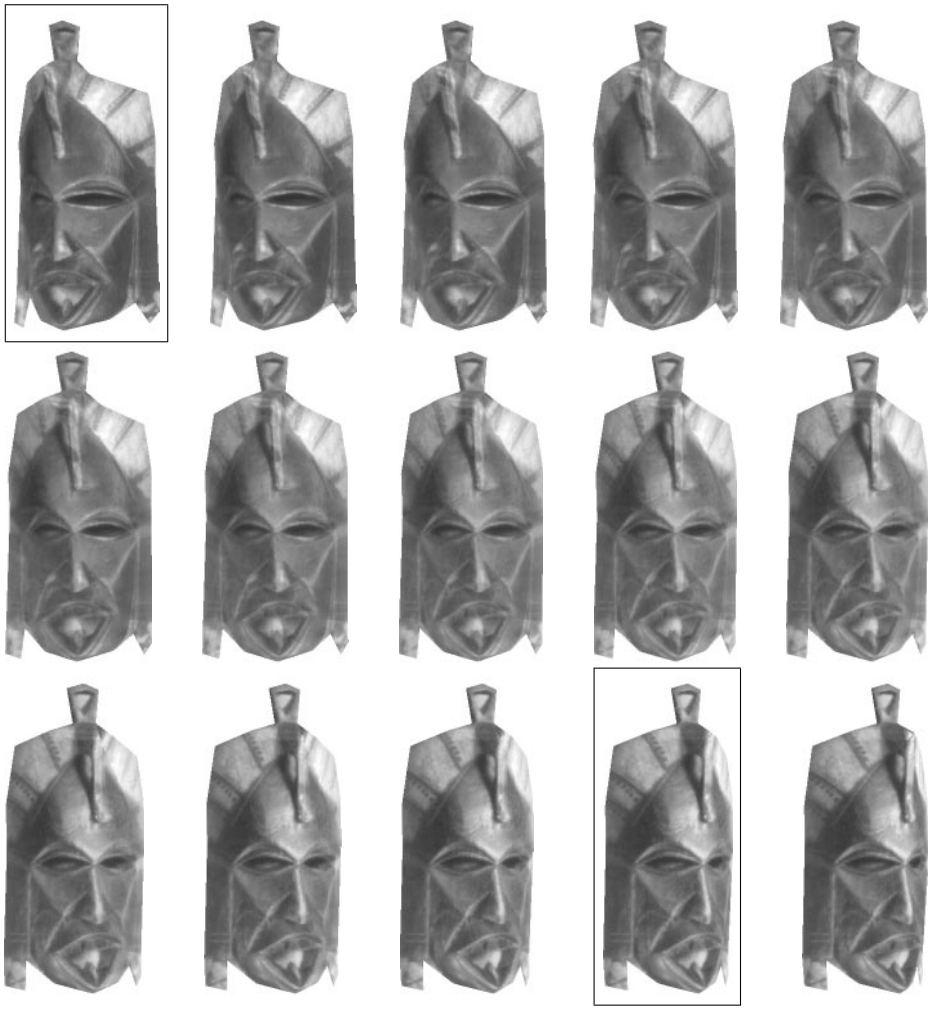

Fig. 4. A simple example of parametric view-synthesis, using 74 control-points. The basis-images are shown framed; the other thirteen views were generated as described in the text. The viewpoint is 'extrapolated' slightly in the final frame.

than two basis-views. This leads to the general question of how the quality of the results is related to the number of basis images employed, and whether it is possible to select appropriate basis-views automatically.

If these issues can be resolved, it may be possible to extend the methods described above to the perspective case, via the trifocal tensor [3,15].

References

1. S. Avidan and A. Shashua. Novel View Synthesis by Cascading Trilinear Tensors. *IEEE Transactions on Visualization and Computer Graphics 4(4), pp. 293–306, 1998.*
2. B.F. Buxton, Z. Shafi, and J. Gilby. Evaluation of the Construction of Novel Views by a Combination of Basis Views. *Proc. EUSIPCO '98, pp. 1285–1288, 1998.*
3. R. I. Hartley. Lines and Points in Three Views and the Trifocal Tensor. *International Journal of Computer Vision, 22(2), pp. 125–140, 1995.*
4. R. I. Hartley. Theory and Practice of Projective Rectification. *Submitted to: International Journal of Computer Vision, 1998.*
5. J.J. Koenderink and A. J. Van Doorn. Affine Structure from Motion. *Journal of the Optical Society of America 8(2), pp. 377–385, 1991.*
6. I. Koufakis and B. Buxton. Very Low Bit-Rate Face Video Compression using Linear Combination of 2D Face Views and Principal Components Analysis. *Image and Vision Computing 17, pp. 1031–1051, 1998.*
7. S. Laveau and O. Faugeras. 3-D Scene Representation as a Collection of Images and Fundamental Matrices. *INRIA Technical Report 2205, 1994.*
8. T. Poggio and R. Brunelli. A Novel Approach to Graphics. *MIT Technical Report 1354, 1992.*
9. S. T. Pollard, M. Pilu, E.S. Hayes and A. Lorusso. View-Synthesis by Trinocular Edge-matching and Transfer. *Proc. BMVC '98, pp. 770–779, 1998.*
10. R. M. Rohrer and E. Swing. Web-Based Information Visualisation. *IEEE Computer Graphics and Applications, pp. 52–59, 1997.*
11. M. Segal and K. Akely. The OpenGL Graphics System: A Specification. (Version 1.2.1). *Silicon Graphics, 1999.*
12. S. Seitz and C.R. Dyer. Physically-Valid View Synthesis by Image Interpolation. *Proc. IEEE Workshop on the representation of visual scenes, pp. 18–25, 1995.*
13. L.S. Shapiro, A. Zisserman, and J.M. Brady. 3D Motion Recovery via Affine Epipolar Geometry. *International Journal of Computer Vision, 16, pp. 147–182, 1995.*
14. A. Shashua. Algebraic Functions for Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 17(8), pp. 779–789, 1995.*
15. A. Shashua. Trilinear Tensor: The Fundamental Construct of Multiple-View Geometry and its Applications. *Proc. International Workshop on Algebraic Frames For The Perception-Action Cycle, 1997.*
16. J. R. Shewchuk. Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator. *Proc. First Workshop on Applied Computational Geometry, pp. 124–133, 1996.*
17. C. Tomasi, T. Kanade. Shape and Motion from Image Streams under Orthography — A Factorization Method. *International Journal of Computer Vision 9(2), pp. 137–154, 1992.*
18. B. Triggs. Matching Constraints and the Joint Image. *Proc. of the ICCV, pp. 338–343, 1995.*
19. S. Ullman and R. Basri. Recognition by Linear Combinations of Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 13(10), pp. 992–1006, 1991.*
20. T. Werner, T. Pajdla and V. Hlaváč. Efficient 3D Scene Visualization by Image Extrapolation. *Proc. ECCV '98, vol. 2, pp. 382–395*
21. L. Williams and S.E. Chen. View Interpolation for Image Synthesis. *Proc. SIGGRAPH '93, pp. 279–288, 1993.*