Environment Map Morphing

Russell Manning
Charles Dyer

Technical Report #1423

January 2001

Environment Map Morphing

Russell A. Manning Charles R. Dyer

Department of Computer Sciences
University of Wisconsin
Madison, Wisconsin 53706

Technical Report #1423
December 2000

Abstract

The techniques of static and dynamic view morphing [19, 12] are extended to allow for (1) interpolation between arbitrary, uncalibrated environment maps (complete or partial), (2) camera motion directly towards or away from the scene (so that the epipole is visible), and (3) views of dynamic scenes in which an object's vanishing point is visible inside the object. It is shown that, by using an *image cylinder* instead of an image plane, the scanline property [18] can be preserved and the morphing process can be made continuous for environment maps. Environment map interpolation together with layering provides an uncalibrated, sprite-like graphics primitive similar to "billboarding" techniques.

This work is sponsored by the National Science Foundation under Grant No. IIS-9530985 and by the Defense Advanced Research Projects Agency (DARPA) and Rome Laboratory, Air Force Materiel Command, USAF, under agreement number F30602-97-1-0138.

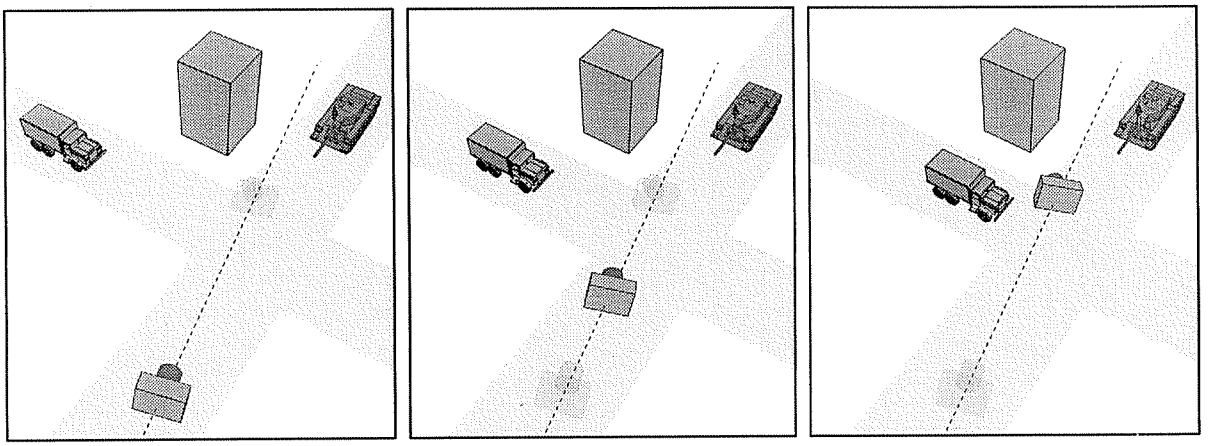

Figure 1: View interpolation involves synthesizing the view from the camera in the middle frame starting with only the two reference views from the cameras in the left and right frames. In this case, the epipole will be visible in the first view.

1 Introduction

In this paper we consider the problem of view interpolation [4] (Fig. 1) when the two input views are uncalibrated environment maps [8]. We also discuss extending our earlier techniques for simultaneous view and motion interpolation [12] to the case of uncalibrated environment maps. Of special interest are scenarios in which epipoles are visible or in which moving objects contain their own vanishing points.

There has been much interest recently in producing and using large-field-of-view images. The machine vision community has studied large-field-of-view cameras like Columbia's OmniCam [14] and various mosaicing techniques (e.g., [22, 16, 5]). The commercial sector has produced products like QuicktimeVR [3] and IPIX [10].

Working with large-field-of-view images offers several advantages. First is the immersive power of such views. Human vision covers nearly a half space when peripheral vision is considered, and conventional photographs and display devices can fill only the center portion of that wide field. For much greater realism in flight simulators and other virtual reality applications, it is necessary to fill the entire half-space of human vision. This is also a principle behind large-field-of-view movie theaters like OmniMax. A second advantage is the greater data capturing capability of large-field-of-view cameras, especially in surveillance applications. A third advantage is the greater expressive power of large-field-of-view images. For instance, it is easier to understand a scene portrayed as a large mosaic than as a series of small, disjoint images. In a related manner, it is easier to solve correspondence problems for large-field-of-view image pairs than for conventional small-field-of-view pairs because there will typically be a lower percentage of points that are only visible in one view.

In this paper we will use the terms *large-field-of-view image* and *environment map* interchangeably. Our use of these two terms is one of "intended application" rather than mathematical precision. First, we are interested in views that cover more than a half space, or pairs of views that together cover more than a half space, so that conventional techniques involving projection onto a single image plane can not be directly applied. Second, we are interested in cases of arbitrary camera motion and arbitrary vanishing points for moving objects in the scene, even for small-field-of-view images. In such cases the more general environment map framework is the most appropriate.

Strictly used, the term *environment map* refers to a representation of all the light that reaches a point in space at a given instant in time from every possible direction. The techniques described in this paper apply equally well to partial environment maps that only contain the light from some directions rather than all directions.

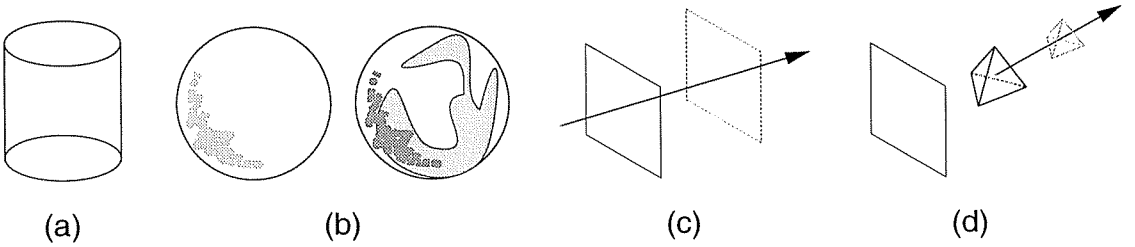

Figure 2: We seek methods for interpolating between completely general reference views such as (a) cylindrical environment maps, (b) complete or partial spherical environment maps, (c) pairs of reference views where ego motion is into the scene, and (d) dynamic scenes where an object's vanishing point is visible within the object. All such reference views will be termed "environment maps" to indicate their generality.

Large-field-of-view images typically are not complete environment maps. We will use the term environment map to mean either the partial or complete case.

With the growing availability of large-field-of-view images, it is natural to consider extending existing machine vision techniques to such views. There has been no published work on view interpolation for uncalibrated environment maps, and the closest related work is for static scenes only. McMillan and Bishop's plenoptic modeling work [13] allowed for arbitrary view interpolations between two closely-spaced cylindrical environment maps. Special assumptions were made about the acquisition of the reference views, thus allowing for automatic calibration of the cameras; once the cameras were calibrated, the problem reduced to scene reconstruction by triangulation. Avidan and Sashua [1] discussed arbitrary view synthesis from two calibrated views of an object. Their technique involved finding a calibrated trilinear tensor for the views and then recovering the rotation between the two views directly from the tensor. A dense correspondence between the views was also assumed, which was facilitated by having closely-spaced reference cameras. The paper concentrated on views of an isolated object, segmented from the background, and did not discuss application to environment maps. Many other techniques that could be used for view interpolation also require either known camera calibration or the ability to recover camera calibration (e.g., [6, 21, 7]) or else make special assumptions about the scene (e.g., assuming a polyhedral scene [11]) and will not be discussed further. Seitz and Dyer [19] provided a practical technique for view interpolation between uncalibrated images. However, their work explicitly excludes pairs of views that contain their own epipoles and thus is not directly applicable to most large-field-of-view images nor to cases of ego motion towards or away from the scene. All of the work cited above assumes static scenes and so can not be directly applied to environment-map reference views of dynamic scenes.

Fig. 2 shows the kinds of reference views we seek to address in this paper. The reason these views have not been addressed in our earlier work is the difficulties that arise when an epipole is contained in one of the views, or alternatively, when an object contains its own vanishing point (the duality of epipoles and vanishing points is explained by the fixed camera formulation given in Appendix A). The interpolation process used by our techniques requires that each camera basis be changed so that the epipole (or vanishing point) is contained in the plane $z = 0$; this is called rectification. In the new rectified coordinate system, the $z = 0$ plane will split the view or the object under consideration, leading to the difficulties discussed in Section 3.2. We refer to line formed where the $z = 0$ plane intersects the view (in the rectified coordinate system) as the *line of rectification* (see Fig. 3). Note that the problems mentioned above arise whenever the line of rectification is visible (alternatively, whenever it intersects the moving object), not just when the epipole is visible. Thus for the cases given in Fig. 2, the line of rectification always causes problems. Our goal in this paper is to extend static and dynamic view morphing to get around problems caused by the line-of-rectification.

2 Notation and preliminary concepts

The two reference views will be denoted by the letters A and B. Anything associated with a given reference view will be identified by the corresponding letter. For a vector $\phi \in \Re^3$, $(\phi)_x$ will denote the x-component of ϕ. Capital script letters like $\mathcal{R}$ will denote 3×3 matrices.

A position in space exists independently of what coordinate system is used to measure its location. If ω denotes a specific position in space, then ω_A will denote that position as measured in coordinate system A, which is the coordinate system associated with camera A. Note that ω is just a symbol, while ω_A is 3-dimensional vector.

The concept of rectification is primary in our approach to view interpolation. We will say two camera views are *strongly rectified* if the following three conditions are met: (1) the image planes of the two cameras are the same plane in space, (2) the x-axis of each camera is the same line in space (and thus is the line connecting the two optical centers, which defines the epipoles), and (3) for any point ω in space, $(\omega_A)_y = (\omega_B)_y$ and $(\omega_A)_z = (\omega_B)_z$. Seitz and Dyer [19] showed that two cameras can be strongly rectified when the fundamental matrix for the camera pair is known; this important fact will be used implicitly later on.

When the image planes of the two cameras are just parallel in space (not necessarily the same plane) and also parallel to the line connecting the optical centers of the cameras, we will say the two cameras are *weakly rectified*. No additional conditions need to be met; however, the cameras must be "pointed" in the same direction, which can be formally defined by requiring that the $z > 0$ half-space for one camera completely contain the $z > 0$ half-space for the other camera. Two cameras can be weakly rectified without knowledge of their fundamental matrix; for instance, if two conjugate directions (e.g., two vanishing points) can be identified in the reference views, then the two image planes of the cameras can be made parallel to both directions simultaneously. Manning and Dyer [12] showed that weak rectification is sufficient for many types of static and dynamic view interpolations.

We will use the term "image plane" as a formal concept referring to the plane $z = 1$ in local camera coordinates; this clarification is important when considering environment maps that span more than 180°. More specifically, we will refer to the planes $z = 1$ and $z = -1$ as the *positive image plane* and the *negative image plane*, respectively.

3 Environment map interpolation

In this section, we discuss two techniques for environment map interpolation. The first, a direct extension of existing techniques of "view morphing," can be applied in more circumstances; it can also be used in conjunction with the view morphing techniques for dynamic scenes introduced in [12]. The second technique involves an "image cylinder" rather than image planes and thus has the advantage of using a single, continuous projection surface that preserves the scanline property [18].

3.1 Interpolating individual points

We begin by explaining the interpolation process for one point ω, which is viewed as the conjugate pair ω_A and ω_B. The process can then be applied to every other conjugate pair in the same manner. When a dense correspondence between the reference views is known, this completely solves the view interpolation problem. In practical applications with widely-separated reference views, only a sparse set of conjugate points and conjugate line segments will have been determined (e.g., by hand); these conjugate points can be interpolated to guide a morphing process in creating a virtual view.

Assume that a pair of weakly rectifying transformations $\mathcal{R}_A$ and $\mathcal{R}_B$ for two reference views of a static scene are known. Keep in mind that each reference view is an environment map so that, for instance, ω_A

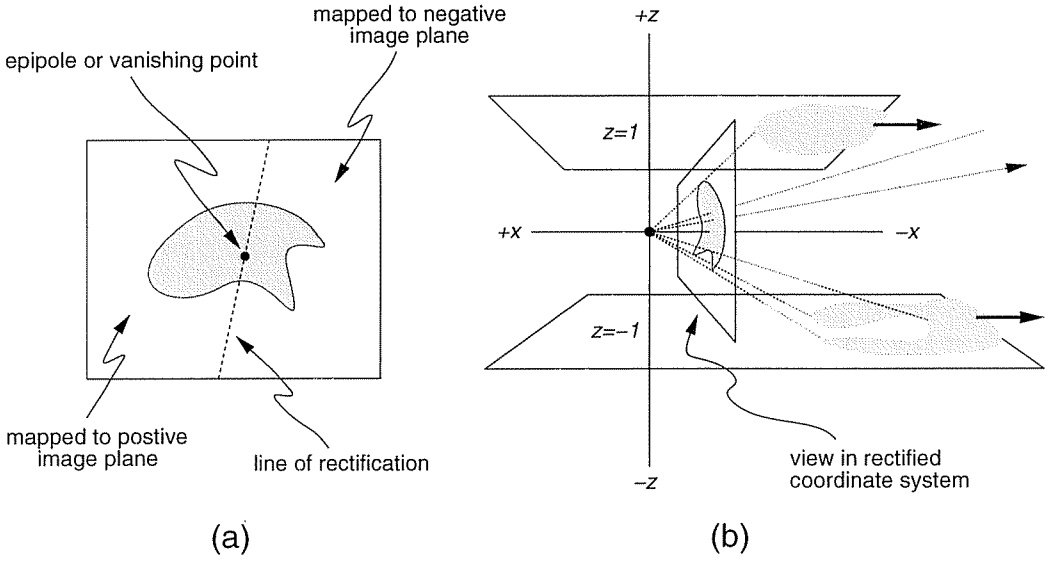

Figure 3: (a) Reference view where the line of rectification intersects the subject matter. (b) Overhead view looking down the y-axis. After rectification, the view is projected onto the positive and negative image planes. This leads to problems of numerical instability and morph discontinuities along the line of rectification.

is some nonzero vector in $\Re^3$ that represents a ray in space pointing from camera A's optical center towards position ω.

Thus $\overline{\omega}_A = \mathcal{R}_A \omega_A$ is the vector in the rectified coordinate system for view A. Without loss of generality, we will assume that $(\overline{\omega}_A)_z \neq 0$. In practice, if $(\overline{\omega}_A)_z = 0$ then ω_A can be slightly repositioned.

Note that either $(\overline{\omega}_A)_z$ and $(\overline{\omega}_B)_z$ are both greater than 0 or both less than 0 (since the $z = 1$ planes are parallel and the $z > 0$ half-spaces nontrivially overlap in the weakly rectified coordinate systems). Consequently, we can project $\overline{\omega}_A$ and $\overline{\omega}_B$ onto the positive image plane $z = 1$ (or negative image plane $z = -1$) by dividing through by the absolute value of each vector's z-component. At this stage, straight-forward linear interpolation of the projected points creates a virtual view. That is, if this process is repeated for all known conjugate pairs, then a physically-accurate new view of all the points as seen through a single virtual camera can be created (the mathematics, adapted from [12], is summarized in Appendix A).

3.2 Extending static and dynamic view morphing

If a dense correspondence between the two reference views can be determined, then the technique of the previous section can be applied to each conjugate pair to produce a dense virtual environment map. How to determine the dense correspondence is the "correspondence problem." For closely-spaced reference views, the correspondence problem can be reasonably solved provided that sufficient texture or shading cues are available.

For widely-spaced reference views, the correspondence problem is harder to solve automatically. In addition, it is usually not feasible to manually determine a dense correspondence. Hence as a practical matter we assume that only a sparse correspondence has been determined and is provided to the interpolation algorithm.

As described in the previous section, a sparse correspondence can be interpolated to produce a virtual view of the scene. Of course, the virtual view will consist only of the sparse set of conjugate points. To fill in the remaining virtual environment map, a 2D morphing algorithm (e.g., [2]) can be used, with the interpolated sparse-correspondence serving as "control points" to guide the morph. The image planes $z = 1$ and $z = -1$ are morphed separately.

Two serious problems can arise from this straight-forward method (see Fig. 3). First, if the environment maps, when projected onto the positive and negative image planes, extend far beyond the control points, then the morphing process will not be well-controlled for those distant regions. As noted in Section 1, this problem always arises when the line of rectification intersects the viewed region. A related problem is the issue of numerical stability for image regions that correspond to z-coordinates near 0 (and thus these regions can be near infinity in the x and y directions when projected onto the image planes).

A second problem is that of discontinuity along the line of rectification in the virtual view. This problem arises because the positive and negative image planes are morphed separately and then the results are combined to produce the new environment map. A related issue for some morphing algorithms occurs when the control features extend across the line of rectification. For instance, the Beier-Neely algorithm uses conjugate line segments to control morphing; when a segment crosses the line of rectification it must be artificially broken into two pieces. Furthermore, the length of these pieces, and of control segments in general that are near the line of rectification, can be arbitrarily large after projection onto the image planes. This poses a serious problem since the Beier-Neely algorithm uses the control-segment length as a weighting factor.

Often partial environment maps can be represented adequately as single flat images (for example, when a small-field-of-view reference camera is used but ego motion is towards the scene). In such cases, the problems listed above can be reasonably avoided as follows: First the sparse set of correspondences is correctly interpolated using the two-image-plane method, thus creating a virtual view. Next the interpolated points (currently on two image planes) are projected onto a single flat image that corresponds roughly to the size and shape of the original reference views. We will refer to this reprojection step as *unrectification*. A good way to achieve unrectification is by the transform

$$(1-s)\mathcal{R}_A^{-1} + s\mathcal{R}_B^{-1} \tag{1}$$

where s is the interpolation factor. Finally, the virtual view is filled-in by applying a morphing algorithm directly to the two flat original reference views, using the interpolated points in the virtual view to control the process.

The approach described above works because the morphing algorithm is just an approximate method for completing the virtual view in the first place. In the case of the Beier-Neely algorithm, the control segments retain their original length leading to better weighting factors than morphing on the rectified image planes.

When the reference views are environment maps represented by complete cylinders, this approach still has the problem of discontinuities along the "cut edge" of the unrolled cylinders. This can be fixed by modifying the 2D morphing algorithm to measure distance correctly across the cut edge.

Dynamic view morphing consists of a series of separate static view morphing steps (one for each layer or object identified in the scene). As such, the techniques described above can be used to perform each separate static view morphing step, thus extending dynamic view morphing to cases where the line of rectification intersects a layer's subject matter.

Note that if a moving object does not contain its own vanishing point but the line of rectification intersects the object anyway, then it is sometimes possible to prepend a rotation around the z-axis to the rectification transformation and thus rotate the line of rectification so it no longer touches the object. In this way, the two-image-plane rectification process can be avoided.

3.3 Image cylinder and the scanline property

The use of two separate image planes causes problems for the morphing step of virtual view creation. We now show how a single "image cylinder" can replace the duo of positive and negative image planes, and we show how the important scanline property [18] holds for this surface.

Recall that a position in an environment map is a 3-dimensional vector $\phi = [x, y, z]$. The corresponding ray on the positive (or negative) image plane is $[x/|z|, y/|z|, z/|z|]$. Now consider scaling ϕ by $1/\sqrt{y^2+z^2}$

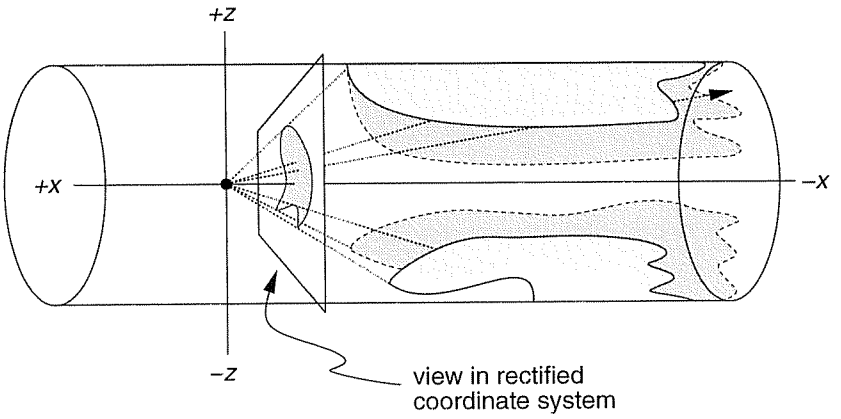

Figure 4: Projection onto the image cylinder.

instead of by $1/|z|$. The resulting vector is a point on the cylinder $y^2 + z^2 = 1$. We will refer to this cylinder as the *image cylinder* (Fig. 4).

Image cylinders have been used before as an alternative to image planes. In a technique called "pipe projection," Rousso et al. [16] used image cylinders to create mosaics from video sequences in which the camera motion was towards the scene (or in any other direction). Roy et al. [17] discussed benefits that arise from using image cylinders for rectification rather than image planes.

In order to use the image cylinder to interpolate environment maps, the two reference views must be strongly rectified; there is no concept analogous to weak rectification (i.e., just making the image planes parallel) for image cylinders. Recall that strong rectification can be achieved using the fundamental matrix. After strong rectification, conjugate points will have the same y and z coordinates. Consequently the conjugate points will both be scaled by the same factor $1/\sqrt{y^2 + z^2}$ during projection onto the image cylinder, and the projected vectors will also have the same y and z coordinates. Every (y, z) pair on the circle $y^2 + z^2 = 1$ defines a line in the x-direction on the image cylinder, and a pair of conjugate points will thus be projected to the same line parallel to the x-axis. During the interpolation process, the interpolated conjugate point will stay on this line for any interpolation factor s (because only the x coordinate will change). Such lines are referred to as *scanlines*.

The importance of preserving scanlines arises when considering monotonic scenes [19]. Informally, *monotonic scenes* are those in which every scene point that is visible in one view is visible in the other view and throughout the entire interpolation sequence. Thus linear ordering of points on each scanline is preserved, which has two benefits. First, the correspondence problem is simplified because not only is a corresponding point on the same scanline (which equals the point's epipolar line), its location must also be consistent with whatever ordering information has already been determined. Second, for regions of a scanline that are monochromatic, it is only necessary to determine the corresponding points for the two endpoints of the region in order to create perfect intermediate images. In this latter case, the dense correspondence problem is avoided altogether. Perfect scene reconstruction would not be possible from such information (since a dense correspondence is necessary) but completely accurate intermediate images can nonetheless be synthesized.

To create virtual views with image cylinders, control points can be interpolated on the image cylinder and then unrectified and used to guide a morphing algorithm on flat images as described in Section 3.2. Alternatively, a 3D morphing algorithm could be devised to complete the virtual view directly on the image cylinder. Such an algorithm would need to compute distance based on Euler angles in 3D rather Euclidean distance on the cylinder surface because of the large distortions that can occur for image points near the rectified x-axis. Equivalently, all three images (the two reference views and the incomplete interpolated view) could be projected onto the surface of a unit sphere and a special morphing algorithm could be applied directly on the sphere's surface. This

latter process is entirely bounded and thus avoids the problems of numerical instability that can arise with image planes and, to a lesser extent, with image cylinders.

Dynamic view morphing [12] can be performed with the image cylinder if the "camera-to-camera transformation" is known (because a separate image cylinder could be used for each object and the camera-to-camera transformation would allow for conversion between the various bases) or in the "parallel motion" case (where all objects share the same fundamental matrix). The "planar-parallel motion" case can only be used with the dual image plane techniques of the previous subsection.

4 Experimental results

Fig. 6 and Fig. 7 show results produced by the methods of this paper. Fig. 6 demonstrates the case of camera motion towards the scene. The top and bottom frames of the strip are the original reference images and the middle frames are two synthesized views excerpted from a continuous interpolation sequence. The scene was static and the same camera was used for both reference images; for the second reference view the camera was rotated downward. The epipoles are contained in both views. Approximately 20 conjugate line segments were identified by hand to guide the algorithm. These control segments were interpolated along the image cylinder and then reprojected into the virtual frame. The reprojection step involved unrectifying each virtual view using the transformation given in Eq. 1. The remaining points in the virtual view were computed using the Beier-Neely algorithm applied to the original images and guided by the interpolated control segments.

Fig. 7 shows frames from an interpolation sequence created using two nearly-complete, widely-separated, cylindrical environment maps as reference views. The two original environment maps are shown in Fig. 5; the first view is from a courtyard surrounded by buildings, the second is from a balcony on the third floor of one of the buildings. The scene is static. Each view was created by mosaicing together approximately 20 images taken from an uncalibrated camera. The camera was allowed to rotate and tilt up and down, and each mosaic covers approximately 270°.

From the two reference mosaics, a short movie was created containing QuicktimeVR-style panning around the individual mosaics combined with interpolation sequences that move between the two reference views. The interpolation sequences were created by interpolating the control segments along an image cylinder and then reprojecting them onto a virtual environment map. The virtual environment map was a cylinder identical to the two reference view cylinders. Unrectification was accomplished via Eq. 1. The virtual environment map was completed using the Beier-Neely algorithm applied directly to the unrolled environment map cylinders (that is, the environment map cylinders were treated as standard flat images). The environment maps are uncalibrated and thus very distorted in their unrolled form. To create the final movie sequence, only a window into the virtual environment map was displayed in each frame; these windows correspond roughly to the size and shape of the original camera views that were used to create the mosaics and thus appear natural and undistorted like typical camera images. To assist the morphing step and create added realism, the reference environment maps were divided manually into 5 layers corresponding to disjoint regions of the views (see [12] for a discussion of layering). Each layer was morphed separately and the results were recombined. Approximately 100 conjugate line segments were identified by hand as the control segments.

When creating the two reference mosaics, no special care was taken to ensure that all the images shared the same optical center, radial distortion was not corrected for, and in general the mosaics were only approximations of true environment maps. The fact that the mosaics contain severe distortions should be kept in mind when considering the results in Fig. 7: The sythetic movie was created directly from these two mosaics, so any errors in the source mosaics (such as bad registration of adjacent views) shows up in the final movie. Furthermore, in several sections of the mosaics the scene has been greatly shrunken (e.g., the row of flower pots in the first mosaic and in parts of the second), making it difficult to place correct conjugate line segments in these regions. This shrinking also made it difficult to produce a completely accurate mask for the different layers; some errors

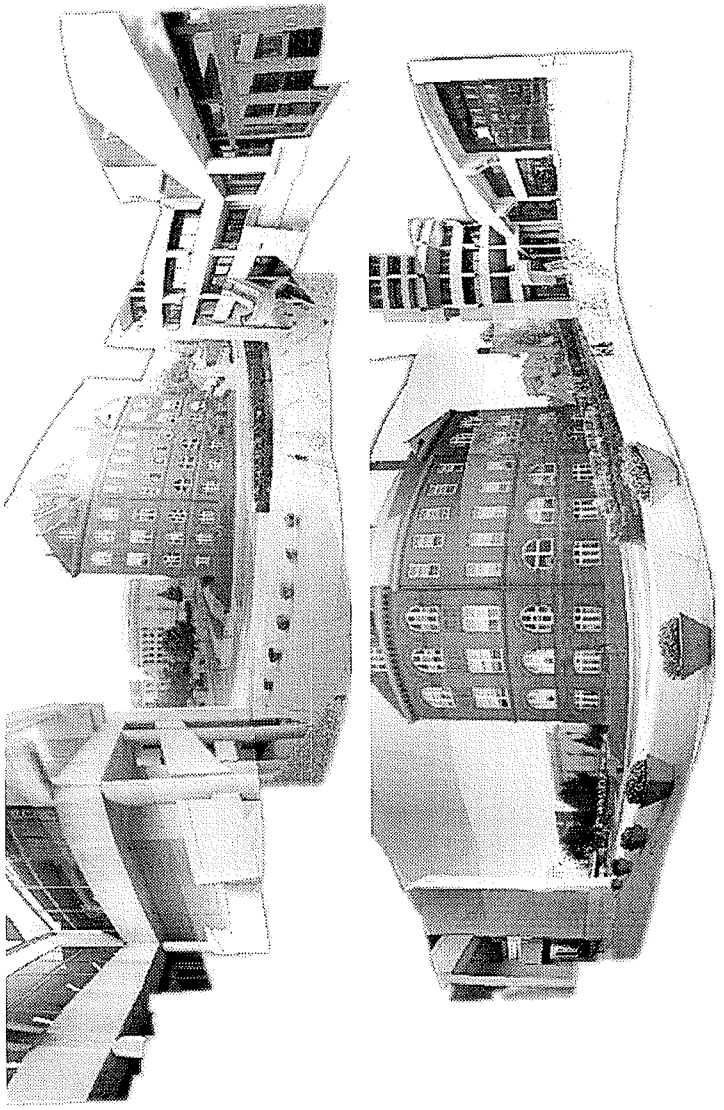

Figure 5: Uncalibrated environment map mosaics used to create the movie shown in Fig. 7.

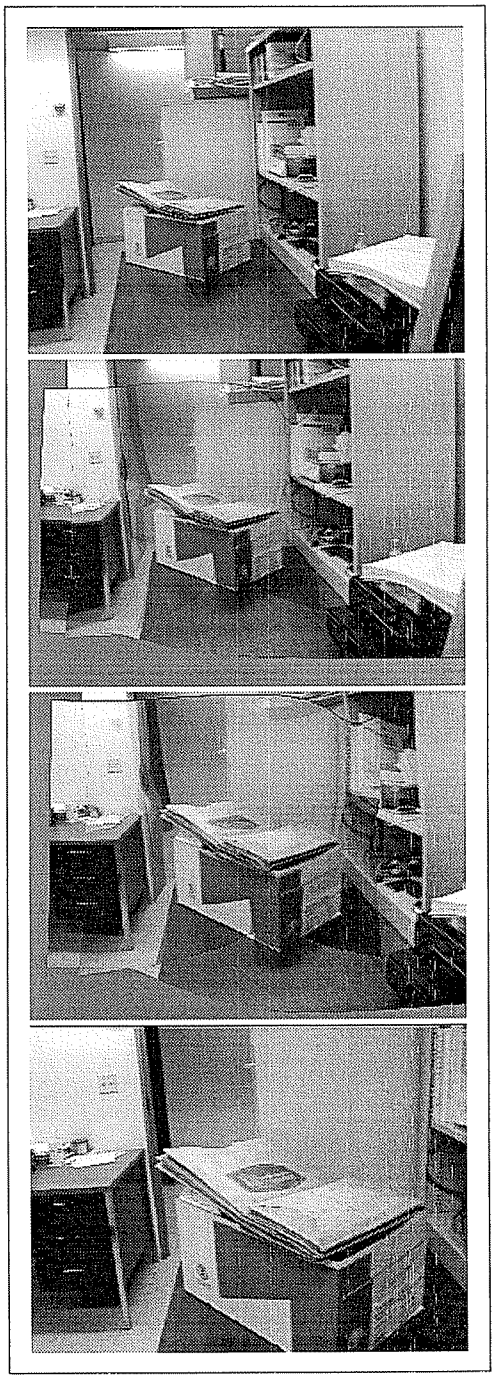

Figure 6: View interpolation with motion into the scene.

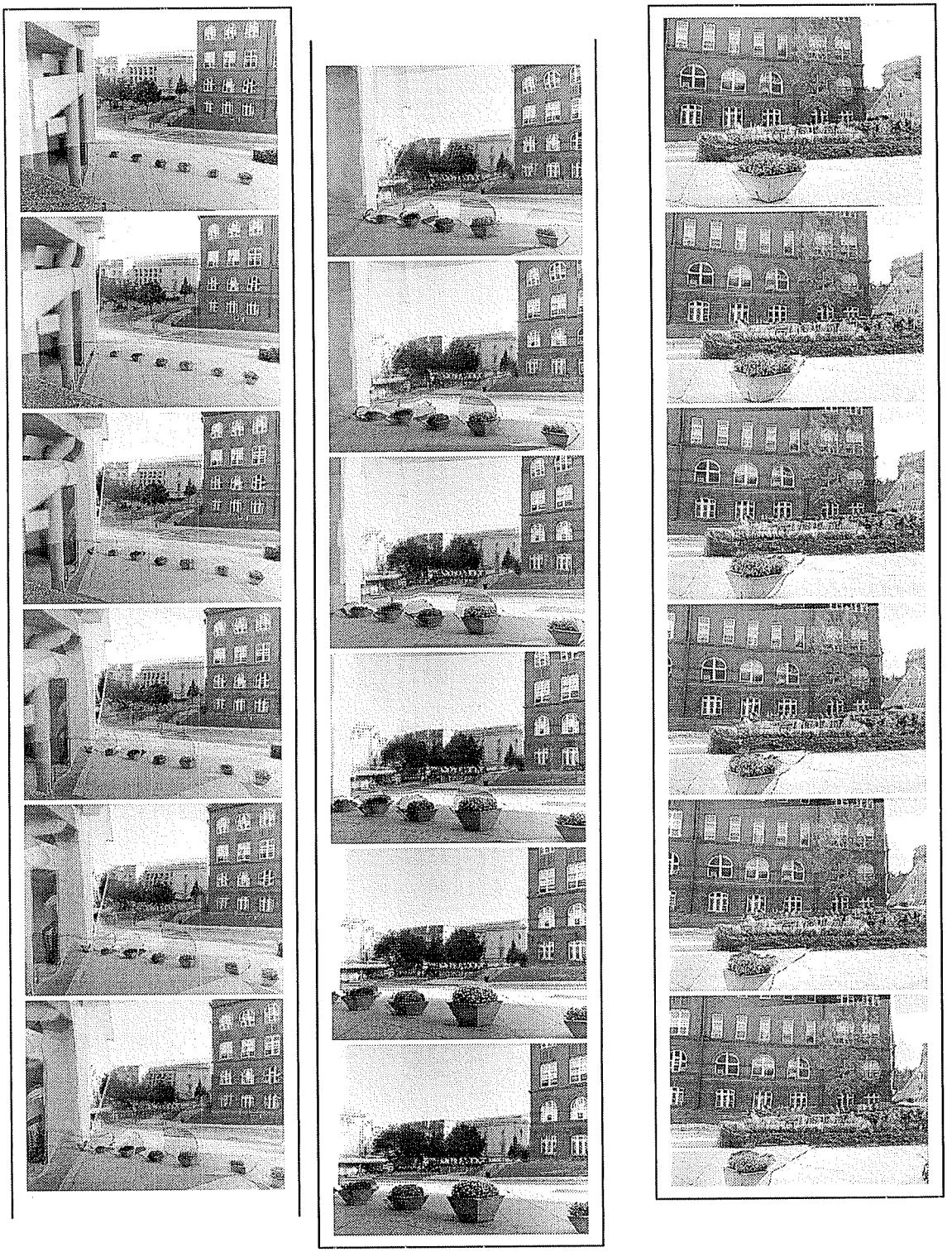

Figure 7: Excerpts from an extended movie showing several back-and-forth transitions between two viewing positions. The initial viewing direction and final viewing direction were different for each transition, creating a "panning" effect as the virtual camera position moved. The two strips on the left form a complete transition from one viewing position to the other. The strip on the right shows part of a second transition in which the final viewing direction was changed.

from the masking are evident in the figure. Nevertheless, an overall sense of movement in space is evident (particularly when the sequence is viewed as a movie), which demonstrates the robustness of the environment map morphing technique to noise. For input data with this level of noise, scene reconstruction (e.g., [15]) is out of the question.

Note that in the second example, when finding the fundamental matrix between the reference views to use in the strong rectification step, it was unnecessary to use the standard numerical stabilization technique of Hartley [9] because the conjugate points were spread over a very wide field of view.

Finally, the use of layering, when combined with view morphing, creates a sprite-like graphics primitive analogous to some "billboarding" techniques such as sprites-with-depth [20]. Layering has an advantage over sprites-with-depth in that it is an uncalibrated technique.

5 Conclusion

By generalizing the static view morphing technique of Seitz and Dyer, we have demonstrated a practical and robust method for interpolating between two environment maps. The method works for uncalibrated cameras, widely-separated views, and sparse correspondences. The generalizations can also be applied to dynamic view morphing, thus allowing for interpolation of both viewpoint and scene motion between two environment maps of a dynamic scene (provided each object in the dynamic scene undergoes a total motion equivalent to a rigid translation).

A scanline property for environment maps has been demonstrated via the use of an image cylinder rather than the standard image plane. For monotonic scenes the scanline property means that physically-correct virtual environment maps of a scene can be created even when the available information is insufficient for complete 3D scene reconstruction. Use of an image cylinder also prevents the rectified images from becoming unbounded in the y-direction.

The problem of discontinuity along the line of rectification in the morphing step has been addressed, and two potential solutions discussed. The first and easiest approach is to treat all views as flat or "unrolled" and then apply the 2D morphing algorithm directly. The second is to utilize the scanline property and morph directly in 3D along the surface of the image cylinder or, alternatively, on the surface of a unit sphere.

Finally, the combination of environment map interpolation and layering has been offered as an uncalibrated sprite-like graphics primitive.

Appendix A

Let $\mathbf{e}$ be the displacement vector between the two reference cameras. By subtracting $\mathbf{e}$ from the motion vector of every element in the scene, we get a new formulation of the scene in which both reference cameras share the same optical center. This is called the *fixed-camera formulation*. Given only the two reference views, it is impossible to disambiguate whether the fixed-camera formulation matches the actual manner in which the views were captured.

Since under this interpretation both cameras share the same optical center, which we can take to be the origin of the world coordinate system U, each camera can be represented by a 3×3 camera matrix. Let $\mathcal{T}_{UA}$ denote camera A's matrix.

Assuming the fixed camera formulation, let $\mathbf{q}$ be a scene particle that undergoes a translation of $\mathbf{u}$ between when it is viewed by camera A and when it is viewed by camera B. Assume cameras A and B have been weakly rectified with respect to the motion of $\mathbf{q}$, and let ξ denote the z-coordinate of $\mathbf{q}$ as measured in camera A's basis. The interpolation of the projection of $\mathbf{q}$ into both cameras is:

$$(1-s)\frac{1}{|\xi|}\mathcal{T}_{UA}\mathbf{q}_U + s\frac{\lambda}{|\xi|}\mathcal{T}_{UB}(\mathbf{q}+\mathbf{u})_U \tag{2}$$

where λ is a scale factor arising from the fact that the two camera's image planes are parallel but possibly at different depths from their shared optical center. Now define a virtual camera V by the matrix

$$\mathcal{T}_{UV} = (1-s)\mathcal{T}_{UA} + s\lambda\mathcal{T}_{UB} \tag{3}$$

Then the linear interpolation (2) is equal to the projection of scene point $\mathbf{q}(s)$ onto the image plane of camera V, where

$$\mathbf{q}(s) = \mathbf{q} + \mathbf{u}(s) \tag{4}$$

$$\mathbf{u}(s)_V = s\lambda\mathbf{u}_B \tag{5}$$

For a scene object that consists of many particles that all undergo the same translation $\mathbf{u}$, each particle can be interpolated separately to create a physically accurate virtual view of the entire object through the single camera V.

References

[1] Shai Avidan and Amnon Shashua. Novel view synthesis in tensor space. In *Proc. Computer Vision and Pattern Recognition Conf.*, pages 1034–1040, 1997.

[2] Thaddeus Beier and Shawn Neely. Feature-based image metamorphosis. In *Proc. SIGGRAPH 92*, pages 35–42, 1992.

[3] Shenchang Eric Chen. Quicktime VR — An image-based approach to virtual environment navigation. In *Proc. SIGGRAPH 95*, pages 29–38, 1995.

[4] Shenchang Eric Chen and Lance Williams. View interpolation for image synthesis. In *Proc. SIGGRAPH 93*, pages 279–288, 1993.

[5] James Davis. Mosaics of scenes with moving objects. In *Proc. Computer Vision and Pattern Recognition Conf.*, pages 354–360, 1998.

[6] Paul E. Debevec, Camillo J. Taylor, and Jitendra Malik. Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. In *Proc. SIGGRAPH 96*, pages 11–20, 1996.

[7] A. W. Fitzgibbon and A. Zisserman. Automatic 3D model acquisition and generation of new images from video sequences. In *Proceedings of European Signal Processing Conference (EUSIPCO '98), Rhodes, Greece*, pages 1261–1269, 1998.

[8] Ned Greene. Environment mapping and other applications of world projections. *IEEE Computer Graphics and Applications*, 6(11):21–29, November 1986.

[9] Richard I. Hartley. In defence of the 8-point algorithm. In *Proc. Fifth Int. Conf. on Computer Vision*, pages 1064–1070, 1995.

[10] Interactive Pictures Corporation, Inc. IPIX, version 1.0, 1997.

[11] Bjrn Johansson. View synthesis of piecewise planar objects. In *Proc. Eleventh Scandinavian Conf. on Image Analysis*, 1999. To appear.

[12] Russell A. Manning and Charles R. Dyer. Interpolating view and scene motion by dynamic view morphing. In *Proc. Computer Vision and Pattern Recognition Conf.*, volume 1, pages 388–394, 1999.

[13] Leonard McMillan and Gary Bishop. Plenoptic modeling. In *Proc. SIGGRAPH 95*, pages 39–46, 1995.

[14] Shree K. Nayar. Catadioptric omnidirectional camera. In *Proc. Computer Vision and Pattern Recognition Conf.*, pages 482–488, 1997.

[15] M. Pollefeys. *Self-Calibration and Metric 3D Reconstruction from Uncalibrated Image Sequences*. PhD thesis, Katholieke Universiteit Leuven, Belgium, 1999.

[16] B. Rousso, S. Peleg, I. Finci, and A. Rav-Acha. Universal mosaicing using pipe projection. In *Proc. 6th Int. Conf. on Computer Vision*, pages 945–952, 1998.

[17] Sebastien Roy, Jean Meunier, and Ingemar J. Cox. Cylindrical rectification to minimize epipolar distortion. In *Proc. Computer Vision and Pattern Recognition Conf.*, pages 393–399, 1997.

[18] Steven M. Seitz. *Image-Based Transformation of Viewpoint and Scene Appearance*. PhD thesis, University of Wisconsin, Madison, WI, 1997.

[19] Steven M. Seitz and Charles R. Dyer. View morphing. In *Proc. SIGGRAPH 96*, pages 21–30, 1996.

[20] Jonathan Shade, Steven Gortler, Li wei He, and Richard Szeliski. Layered depth images. In *Proc. SIGGRAPH 98*, pages 231–242, 1998.

[21] Gideon Stein. *Geometric and Photometric Constraints and Structure from Three Views*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, June 1998.

[22] Richard Szeliski. Video mosaics for virtual environments. *IEEE Computer Graphics and Applications*, 16(2):22–30, 1996.