In Proceedings of 1997 Symposium on Interactive 3D Graphics, Providence, RI, April 27-30, 1997, pp. 7-16.

Post-Rendering 3D Warping

William R. Mark Leonard McMillan Gary Bishop

Department of Computer Science*
University of North Carolina at Chapel Hill

Abstract

A pair of rendered images and their Z-buffers contain almost all of the information necessary to re-render from nearby viewpoints. For the small changes in viewpoint that occur in a fraction of a second, this information is sufficient for high quality re-rendering with cost independent of scene complexity. Re-rendering from previously computed views allows an order-of-magnitude increase in apparent frame rate over that provided by conventional rendering alone. It can also compensate for system latency in local or remote display.

We use McMillan and Bishop's image warping algorithm to re-render, allowing us to compensate for viewpoint translation as well as rotation. We avoid occlusion-related artifacts by warping two different reference images and compositing the results. This paper explains the basic design of our system and provides details of our reconstruction and multi-image compositing algorithms. We present our method for selecting reference image locations and the heuristic we use for any portions of the scene which happen to be occluded in both reference images. We also discuss properties of our technique which make it suitable for real-time implementation, and briefly describe our simpler real-time remote display system.

CR Categories and Subject Descriptors: I.3.3 [Computer Graphics]: Picture/Image Generation – Display Algorithms; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism – Virtual Reality; I.3.2 [Computer Graphics]: Graphics Systems – Distributed/Network Graphics.

Additional Keywords: image-based rendering, post-rendering warping, image compositing and reconstruction, remote display, 3D warp.

1 Introduction

Interactive 3D graphics applications continually demand the ability to display more complex geometric models using less expensive rendering hardware. Most progress to date has depended on improvements in semiconductor technology, visibility culling and level-of-detail management. We explore a different but complementary approach, by exploiting frame-to-frame coherence to completely avoid conventional rendering of most frames.

In typical immersive applications, the viewpoint changes gradually, so that adjacent frames are very similar. Most frames can be generated by using an image warp to extrapolate from nearby conventionally rendered frame(s). We refer to the frames rendered in the conventional manner as *reference frames*, and the frames produced from image warping as *derived* frames. Our technique produces several derived frames for each reference frame.

Performing image warping as a post-processing step to conventional rendering (*post-rendering warping*) can compensate for latency as well as increase frame rate. The two uses of post-rendering warping are very closely related. In either case, image warping must compute derived frames at new viewpoints by extrapolation from reference frames rendered at other viewpoints.

We use McMillan and Bishop's planar-to-planar, forward mapped image warping algorithm [27] to compute derived frames from reference frames. This warp uses a per-pixel disparity value as part of the warp computation. The disparity value is a form of depth information that is easily computed from the 1/Z values in a standard Z-buffer. We refer to this warp as a *3D warp* because it relies on both disparity information and image coordinates.

Unlike simpler warps, the 3D warp correctly accounts for both viewpoint rotation and translation. The ability to account for translation is crucial when using post-rendering warping to compensate for significant latencies. However, the ability to account for translation leads to two difficulties. Because adjacent pixels may be moved different distances by the warp, image reconstruction is more difficult than it is for other warps. Additionally, we must cope with the *occlusion* problem. Although the 3D warp will correctly warp all pixels which are in the reference frame, no information is available about objects which are occluded in the reference frame. Because translation can expose objects or portions of objects in a scene, the derived frame will, in general, be incorrect in the sense that it will not match a conventionally rendered frame at that same viewpoint. Figure 1 illustrates this problem.

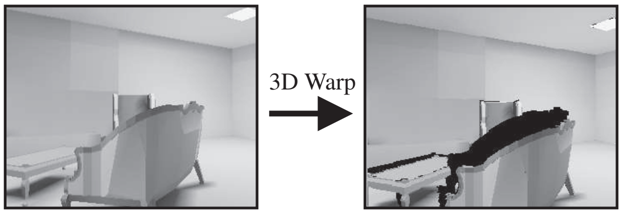

Figure 1: *The 3D warp can expose areas of the scene for which the reference frame has no information (shown here in black).*

We overcome this problem by warping multiple reference frames to produce each derived frame (Figure 2). If a region of the scene is visible in any one of the reference frames, we can correctly place it in the derived frame. Two appropriately chosen reference frames are sufficient to resolve most potential occlusion problems. For any areas of the derived frame which remain occluded in both of the

*CB #3175, Sitterson Hall; Chapel Hill, NC 27599-3175 USA.
email: {billmark | mcmillan | gb}@cs.unc.edu
www: http://www.cs.unc.edu/~{billmark | mcmillan | gb}
phone: +1.919.962.{1917 | 1797 | 1886}

In Proceedings of 1997 Symposium on Interactive 3D Graphics, Providence, RI, April 27-30, 1997, pp. 7-16.

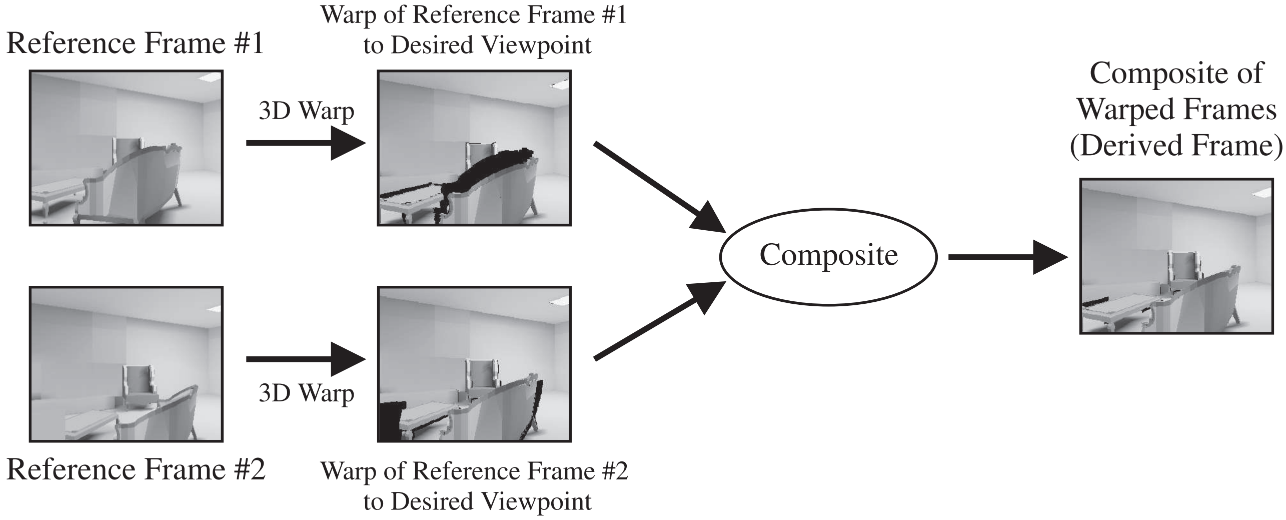

Figure 2: *A single warped frame will lack information about areas occluded in its reference frame. Multiple reference frames can be composited to produce a more complete derived frame.*

reference frames, we use a heuristic technique to estimate the correct surface.

The most straightforward approach to reconstruction for the forward-mapped 3D warp is to write a single derived pixel for each warped reference pixel. As Figure 3 indicates, this solution is inadequate. Pinholes appear in the derived frame for surfaces whose normal has rotated towards the user in the derived frame. We can consider these surfaces to be slightly under-sampled in the reference frame.

Figure 3: *The simplest form of reconstruction, writing a single derived pixel for each reference pixel, causes holes to appear in surfaces that are slightly under-sampled.*

We eliminate these pinholes by treating the reference frame as a mesh. A straightforward implementation of the mesh technique interacts poorly with multi-reference-frame compositing. Our algorithm overcomes this problem by detecting discontinuities in the mesh and treating them differently during the compositing step.

2 Applications

2.1 Increased Frame Rate

The primary focus of this paper is the use of post-rendering 3D warping to increase the frame rate of a graphics system. We have built a software test-bed which demonstrates this application in simulation. Reference frames are generated using conventional rendering at 5 frames/sec, and then warped to produce derived frames at 60 frames/sec. This system uses our multi-reference-frame compositing and reconstruction algorithm to minimize artifacts from the 3D warp. As a result, the derived frames are almost indistinguishable from conventional 60 frames/sec rendering. The use of post-rendering warping also gives this system a low latency for viewpoint change—the latency is equal to the time needed to perform the 3D warp.

The cost of post-rendering warping is determined by display resolution rather than model complexity. Thus, for sufficiently complex models, post-rendering warping will provide better price/performance than conventional rendering alone. In this paper, we concentrate primarily on demonstrating that post-rendering 3D warping can produce imagery similar in quality to conventionally rendered imagery, but we discuss performance issues as well.

2.2 Low-Latency Remote Display

Our second application uses the 3D warp to compensate for network latency when images are generated at one location, and viewed at another location by a user who controls the viewpoint (Figure 4). Network latency is a problem in such systems, and for transcontinental systems and systems using satellite links, this latency has a significant lower bound imposed by the speed of light.

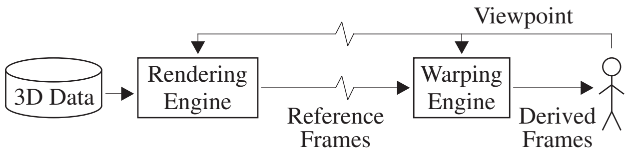

Figure 4: *Remote display system.*

In such a system 3D warping only compensates for latency-based errors that are caused by changes in the viewing transform, not those that are caused by changes in the underlying scene. Changes in a dynamic scene will still appear in time-delay. But, if the reference frames are transmitted across the network link at the display frame rate, this time-delay will only be noticeable when the user is interacting with the changing elements of the scene.

We have built a proof-of-concept real-time system which demonstrates the remote-display application. Reference frames (four sides of a cube with a common viewpoint) are generated at one location using an SGI RE/2, and transmitted uncompressed over an ATM link at $\frac{1}{4}$ frame/sec to the user-side computer. The user-side computer uses a software-only 3D warp to compensate for the network and rendering latency, and to increase the apparent frame rate. This 3D warp runs at 7 frame/sec, using three 200 MHz R4400 processors, to produce derived frames with $\frac{1}{2}$ of NTSC resolution. The system can also generate stereo imagery without any additional information from the rendering computer, by warping for both the left and right eye viewpoints.

The derived frame quality is poorer than what we get from our test-bed system, because the real-time system uses only a single reference-frame viewpoint—it doesn't composite reference frames from multiple viewpoints. The derived frame quality is also poorer because the system uses a simple splat-like reconstruction algorithm rather than the mesh-based algorithm used by our test-bed system. Further details about this system, including its reconstruction and clipping algorithms, can be found in [22].

The remainder of the paper is organized as follows: The next section discusses previous work. Section 4 provide details of our multi-image compositing and reconstruction techniques. In section 5 we describe how reference-frame viewpoints are chosen. Section 6 discusses our test-bed system. Finally, section 7 considers some limitations of our current work and discusses promising directions for future work.

3 Previous Work

In this section we discuss the previous work that our current efforts build upon. A good overview of the mathematical aspects of image warping is provided by Wolberg's book [39] on 2D image warps.

3.1 Post-Rendering Warping

Previous researchers have investigated a variety of different post-rendering warping algorithms to decrease latency and/or increase frame rate. The earliest work focused on latency reduction and used image shifting (i.e. translation in image X and Y). The rendered image is shifted either electro-optically [8–10, 31, 33], or in the frame-buffer just prior to scan-out [25].

A 2D affine transform is more general than image shifting, because it allows for scaling and image-plane rotation as well as image-plane translation. Hofmann proposed that 2D affine transforms could be used to avoid re-rendering parts of a scene, and discusses the conditions under which these transforms provide an adequate approximation to the desired image [16]. Microsoft's Talisman architecture [36] composites independent image layers at video rates in front-to-back order using a per-layer affine transform. Any given image layer is re-rendered only when the residual error after applying its affine transform exceeds a desired threshold.

To compensate for arbitrary changes in view direction, a 2D projective transform is required. Regan and Pose's address re-calculation pipeline implements this transform in hardware, with the goal of both reducing latency and (in conjunction with their priority rendering technique) increasing frame rate [29, 30]. 2D projective transforms can also be applied to selected parts of a scene by mapping recently rendered imagery onto large textured polygons before performing final rendering [3, 32].

3.2 Warping From Stored Images

Image warping can be used to display previously stored imagery. This imagery can be either rendered off-line, or acquired from the real world by cameras. Lippman's movie maps system [20] allows virtual movement through a city along controlled routes. He proposes the use of both image scaling and 2D projective transforms to interpolate between stored images. QuickTime VR's panoramic player [11] employs a 2D cylindrical-to-planar warp to allow arbitrary view directions from a single viewpoint.

The Lumigraph [14] and Light Field Rendering systems [19] take a very different approach to image-based rendering from the systems we have just discussed. Rather than storing a limited number of images and then warping to interpolate between them, they store a dense, regular sampling of the set of all possible light rays. Many of 3D image warping's problems (especially those related to occlusion) are minimized, but they are replaced with the challenge of storing and accessing an enormous dataset.

3.3 3D Warps

A 3D warp allows for changes in both view direction and in viewpoint. Our 3D warping algorithm is an extension of the planar-to-planar 3D image warp [27] which was developed earlier by two of this paper's authors. This earlier paper uses incremental evaluation of the 3D warp equations and an occlusion-compatible warping order to achieve real time performance, but does not address reconstruction issues in detail. The algorithm is demonstrated on pre-rendered synthetic imagery. A later system [28] used a 3D cylindrical-to-planar warp for acquired imagery. Similar warping calculations have been used to accelerate ray tracing of animations and stereo pairs [1, 2, 7]. Greene and Kass [15] generated simplified image-like representations of a scene by retaining only those polygons which were visible in a Z-buffered image from the reference viewpoint. To minimize cracking, polygons which occupied a pixel in this image but whose true projected screen-space area was smaller than a pixel were enlarged to fill the pixel.

Chen and Williams' work on view interpolation [12] was the first to discuss many of the problems with which we are concerned in our current work. Their system uses a 3D warp in a pre-processing step to compute the movement of pixels between reference frame viewpoints. The run-time warp is not a full 3D warp. Instead, it linearly interpolates the warp vector calculated in the pre-processing step to determine the derived-frame pixel coordinates. The interpolation parameter is determined from the location of the derived-frame viewpoint with respect to the reference frame viewpoints. Performance is optimized by grouping pixels with similar pre-computed warp vectors into blocks and computing a single motion vector for the entire block. The system composites multiple warped reference images using depth information, by organizing pixel blocks into a fixed visibility order. Reconstruction is very simple, always writing a single derived-pixel for each reference-pixel. A post-process step fills in empty derived-image pixels by interpolating from nearby filled pixels. This post-process does not fix cases where background objects are visible through missing pixels in foreground objects.

There are several systems which use 3D warps similar to those developed by McMillan and Bishop. None of these systems were concerned with real-time performance. Laveau and Faugeras [17, 18] explore the use of a partially inverse-mapped 3D warp. Max and Ohsaki [24] use several multi-layered reference images to to minimize occlusion artifacts. Their system uses deferred shading to correctly compute view-dependent shading. It is also different from most other systems because it uses parallel-projection reference images. More recent work by Max [23] uses a hierarchy of reference images of differing spatial resolution. Szelinski [34, 35] used equations similar to the 3D warp equations as part of a technique to extract depth from acquired imagery.

3.4 Layering Techniques for 2D Warps

Affine and 2D projective warps can not by themselves compensate for viewpoint translation if objects in the scene are not all coplanar.

In Proceedings of 1997 Symposium on Interactive 3D Graphics, Providence, RI, April 27-30, 1997, pp. 7-16.

But, these warps can partially compensate for translation if the scene is separated into layers, and a separate affine or 2D projective warp is applied to each layer. Different layers can also then be re-rendered at different rates. This technique is an alternative to a full 3D warp. Variations on it are used by Talisman [36], Regan and Pose (priority rendering) [30], and by Greene and Kass [15]. This last system actually renders the near geometry in the standard fashion into every derived frame—only the far geometry is represented in image form. The texture-based simplification systems mentioned earlier [3, 32] can also be considered to be layered systems.

3.5 Predictive Tracking

Prediction of future viewpoint and view direction can be used in place of, or in conjunction with, image warping to compensate for latency. Predictive tracking has been combined with image shifting by [9, 25, 31, 33]. Predictive tracking alone has been used by many systems; see [5] for an example and references to other work. Predictive tracking becomes less accurate as the prediction interval increases [6], and thus becomes less usable (especially by itself) as latencies increase.

4 Compositing and Reconstruction

4.1 Reconstruction

The straightforward reconstruction technique of writing a single derived-image pixel for each reference-image sample does not produce images of adequate quality. We have explored two reconstruction techniques that are more sophisticated. The first (Figure 5a) treats each reference pixel independently, but varies the size of the reconstruction kernel depending on the disparity and normal-vector orientation of the reference pixel. This approach is a form of splatting [37]. We use this technique in our real-time remote display system, but rough edges on under-sampled surfaces and occasional pinholes harm the visual quality of the derived frame.

The second technique (Figure 5b) treats the reference frame as a mesh. The 3D warp perturbs the vertices of the mesh, possibly causing folds. Reconstruction occurs by rendering the perturbed mesh triangles into the derived frame. The original pixel colors are thus linearly interpolated across the reconstructed mesh element. This second technique is the one used by our software test-bed.

A problem with the pure mesh-based reconstruction model occurs at silhouette boundaries between foreground and background objects. If the reference-frame mesh is treated as completely continuous, then surfaces are implied at these silhouettes even though the surfaces almost never actually exist. When the reference frame is warped, these implied surfaces manifest themselves as "rubber sheets" stretching from the edge of the foreground object to the background object behind it (Figure 6). The implied surfaces can hide objects behind them which are visible in a different reference frame, or even elsewhere in the same reference frame. In order to prevent this false occlusion, we need to treat these mesh triangles differently from true surfaces. Our compositing algorithm does this.

4.2 Compositing

The compositing algorithm combines two warped reference frames to generate a single derived frame. At a more detailed level, the algorithm must decide, for each pixel in the derived frame, which of the two warped reference frames will determine that pixel's color. A compositing algorithm could blend the contributions from the two reference frames in some cases, but our algorithm always makes a binary decision.

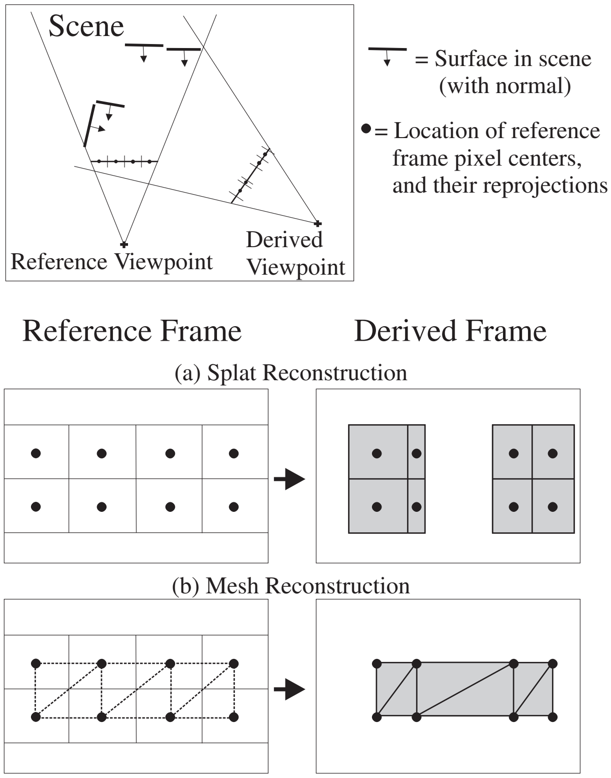

Figure 5: *Two different reconstruction strategies, splat-based and mesh-based. The box at the upper left shows a top view of the scene being rendered. The lower right portion of the diagram shows the derived frames obtained using the two reconstruction techniques.*

In practice, our algorithm does not directly compare the two potential contributors from the two reference frames. Instead, the comparisons are done incrementally. The reference frames are warped one at a time and composited into a derived-image frame-buffer.

A reference frame is warped by sequentially warping each of its mesh triangles. A mesh triangle's vertices are transformed using the 3D warp equations, and it is rasterized and composited into the derived image's frame-buffer. Each of the rasterized pixels is conditionally written. In other words, the key per-pixel compositing decision is whether to keep the pixel already in the frame-buffer, or whether to overwrite it with the contribution from the reference frame that is currently being warped. The incremental nature of this process is similar to that of conventional Z-buffering, where the pixel already in the frame-buffer is compared with the pixel from the

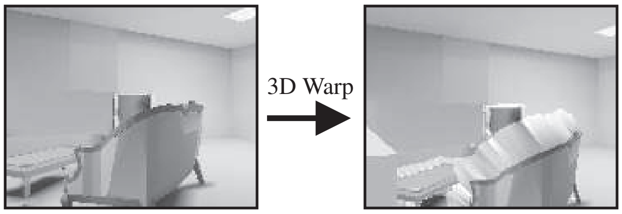

Figure 6: *Pure mesh-based reconstruction causes "rubber sheets" to stretch between foreground and background objects. The right-side image shows these sheets, in a lighter shade.*

polygon being rendered.

The 3D warp can produce folds in the warped reference frame (i.e. multiple reference frame pixels are warped to the same location). Therefore, the compositing algorithm also arbitrates between multiple potential contributions from the *same* reference frame, as well as contributions from different reference frames. The incremental nature of the compositing algorithm handles this detail naturally.

The most important task of the compositing algorithm is to distinguish between surfaces that actually exist, and the artificial surfaces that are implied by the mesh model at foreground/background silhouettes. We do not want to let an implied surface incorrectly occlude an actual surface, as would happen in some instances if we relied only on depth information for our compositing decisions.

To disambiguate between the connected and disconnected mesh cases, we define the notion of *connectedness*. Each triangle in the mesh is designated as either low-connectedness or high-connectedness, indicating whether or not the triangle is believed to represent part of a single actual surface. We will discuss the details of the algorithm for determining connectedness later—for now it is sufficient to know that it is determined for each mesh triangle in the reference frame, just after rendering of the reference frame is completed.

In cases where multiple reference frames contain information about the *same* connected surface, we want to generate the derived frame from the reference frame that best samples the surface. Our algorithm makes this determination using a *confidence* value. The confidence value represents the ratio between a reference-frame pixel's projected solid angle in the reference frame to its projected solid angle (prior to compositing) in the derived frame. The projected solid angle in the derived frame depends on the orientation of the surface to which the pixel belongs. Surfaces which are highly oblique in the reference frame but less so in the derived frame will have low confidence values, indicating that they are under-sampled. In our test-bed, we approximate the ratio of projected solid angles with the more cheaply computed ratio of image plane areas.

During the compositing process, the derived-image frame-buffer holds color, Z, connectedness, and confidence information for every derived-frame pixel. As each reference-frame mesh triangle is warped, the compositing algorithm compares the Z, connectedness, and confidence values of each of the warped triangle's pixels with those of the pixels already in the frame-buffer. This comparison determines whether or not each new pixel should replace the pixel already in the frame-buffer. The decision tree of the algorithm for each pixel pair is summarized as follows:

Compare connectedness:

1. If both pixels have high connectedness (both belong to valid surfaces), then compare warped Z values:
   (a) If warped Z's are different,
   ⇒ Pixel with closer Z is stored.
   (b) If warped Z's are the same (within a tolerance),
   ⇒ Pixel with greatest confidence is stored.
2. If only one pixel has high connectedness (one pixel belongs to a valid surface),
   ⇒ Pixel with high connectedness is stored.
3. If neither pixel has high connectedness (neither pixel belongs to a valid surface),
   ⇒ Pixel with greatest confidence is stored. Z's are ignored.

Case #3 needs some further explanation. Selection of this case indicates that we most likely do not have truly valid information about the derived-frame pixel. Both of the mesh triangles which are potential contributors to this pixel were determined not to belong to actual surfaces. In this case, we choose the pixel with the greatest confidence. Using confidence rather than Z to disambiguate generally yields the correct result in instances where one of the pixels belongs to an "almost-connected" triangle.

The reconstruction process uses a special heuristic when reconstructing the low-connectedness mesh triangles. Rather than linearly interpolating the triangle's vertex colors, we flat-shade the triangle with the color of the vertex which is furthest away from the reference frame viewpoint (Figure 7). This heuristic works correctly for the most common case of an occluded surface which continues behind the occluder. We are essentially filling in with the local background color. Such triangles are normally removed in the composition process, but if no high-connectedness mesh triangle is warped to the same point, then the low-connectedness triangle will be visible. This situation indicates that this particular portion of the scene was occluded in *all* of the reference frames.

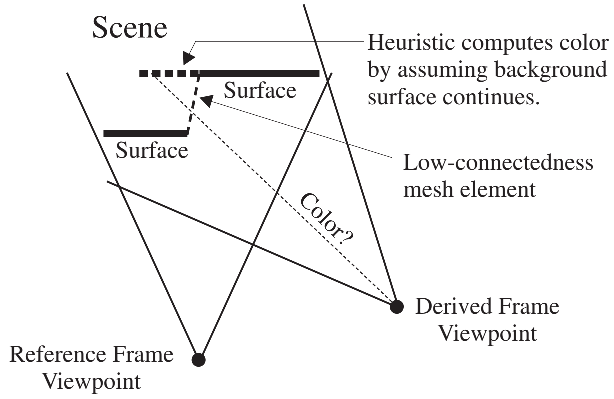

Figure 7: *Low-connectedness mesh triangles are flat shaded with the color of the vertex that is furthest away.*

4.3 Connectedness Calculation

The binary connectedness values required by the compositing algorithm are computed for each triangle in the reference frame mesh. This computation is done only once for each reference frame after rendering, not each time the reference frame is warped.

Our goal in computing connectedness is to determine whether the three vertices of a reference-frame mesh triangle lie on a single consistent surface. To aid us in this determination, our reference-frame renderer generates and stores (linearly interpolated) normal vectors for each pixel, in addition to the usual color and Z information. The connectedness-computing algorithm decides whether or not these mesh triangle's vertex normals are consistent with the triangle's vertex Z values, using the following algorithm.

For two of the three sides of the triangle, we do the following: First, to avoid singularities in our equations, we convert the Z values for the two vertices on that edge into range values (distance from center of projection). We also transform the normal vectors for the two vertices into a coordinate system which has its Z-direction matching that of the vector from the center of projection to the first vertex. Then, we use the component of the transformed normals that is in the direction of the edge to compute a quadratic range function which is consistent with the normals (Figure 8):

$$r = A + Bx + Cx^2 \tag{1}$$

$$A = r(0) \qquad B = \left.\frac{\partial r}{\partial x}\right|_{x=0} \qquad C = \frac{\left.\frac{\partial r}{\partial x}\right|_{x=vert2} - \left.\frac{\partial r}{\partial x}\right|_{x=0}}{2 \cdot vert2}, \tag{2}$$

where r is range and x is distance along the triangle edge, with $x = 0$ at the first vertex, and $x = vert2$ at the second vertex. The derivatives of range are extracted from the normal vectors. We use this range function, computed from the normals, to *predict* the difference in range between the two vertices. This predicted difference is then compared with the actual difference, which is known from the true range values. If the relative error (ratio between the absolute error and the predicted difference) is greater than a pre-determined threshold value, we decide that the two vertices do not belong to the same surface and we designate the triangle as having low connectedness.

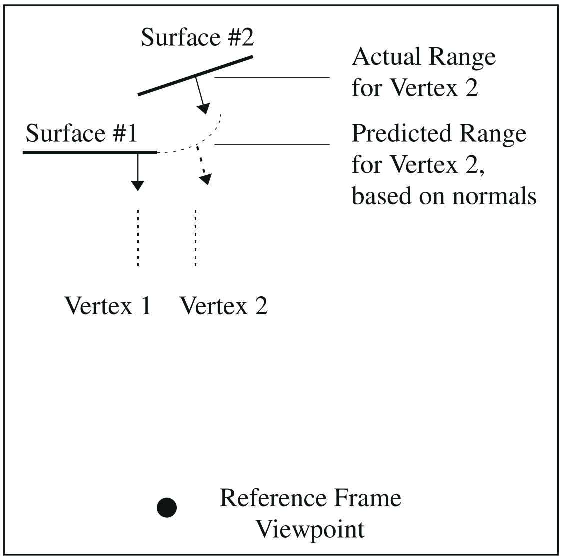

Figure 8: *Connectedness is computed by determining whether mesh normals are consistent with differences in Z values.*

With this test alone, we found that extremely sharp corners would (correctly) fail the test. However, we would like to treat these corners as a connected part of the mesh. So, we augment our original test with a second one: If the two vertices fail the first test, but their difference in Z values is small with respect to the difference in their X/Y coordinates, then the triangle is still deemed to have high connectedness.

Our connectedness calculation is expensive, and probably unnecessarily complex. We believe that future efforts will be able to simplify it substantially while still accomplishing the basic task of determining whether or not a given mesh element should be treated as part of a single, connected surface.

5 Reference Frame Viewpoint Selection

Our system warps and composites two reference frames to produce each derived frame. One reference frame's viewpoint is located near a previous viewer position, and the second frame's viewpoint is located near a future viewer position. When the viewer passes this "future" viewpoint, the system starts using a new "future" reference frame, and discards the old "past" reference frame. When reference frame viewpoint locations are plotted, they are located at fairly regular intervals along (or nearby) the viewer's path. For example, in Figure 9, reference frames A and B were warped to generate derived frame 3.

Since the future reference frame eventually becomes the past reference frame, we get double use out of the reference frames. It is this reuse of the reference frames that allows us to always warp and

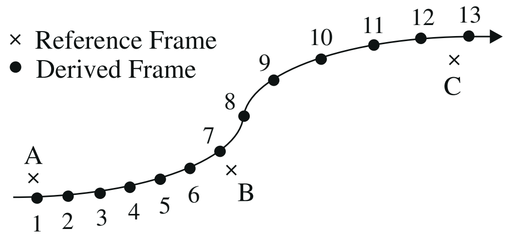

Figure 9: *Derived frames are computed by warping two reference frames, one near a past position of the viewer, and one near a future position of the viewer.*

composite two reference frames, without having to render them any more often than if we were only warping a single reference frame.

Two reference frames will not always be sufficient to completely avoid occlusion artifacts. But in practice, our choice of reference frame viewpoints produces very few artifacts. The following property will help to explain why: For a *single* convex occluder, we can guarantee that a derived frame will be free of occlusion artifacts if its viewpoint lies on the 3-space line through the two reference frame viewpoints (Figure 10a).

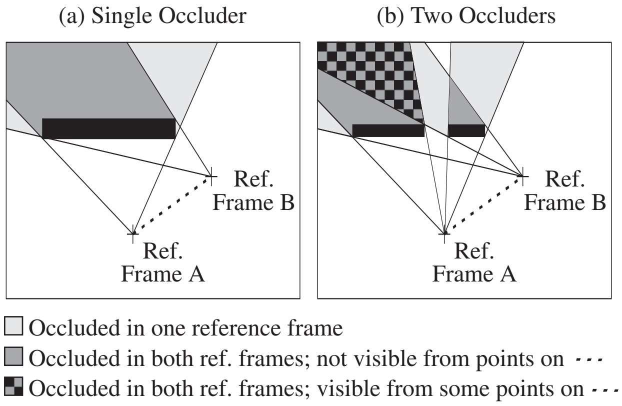

Figure 10: *For a single convex occluder, if a point is visible from a viewpoint on the line between A and B, then it is guaranteed to be visible from point A or point B. For multiple occluders no such guarantee can be made.*

In real scenes there are of course many occluders, but we have found that most occlusion problems will be avoided if this viewpoint condition is met. Intuitively, the reason is that with only a small viewpoint change, it is unlikely that the same portion of the scene will be occluded, then visible, then occluded again by a *different* occluder. Figure 10b shows such a case. This figure depicts an extreme example because the viewpoint-to-viewpoint distance is large relative to the viewpoint-to-object distance. Greene and Kass [15] discuss a similar two-occluder case.

The question of how to choose reference frame viewpoints is then reduced to attempting to insure satisfaction of this viewpoint-on-line condition. The problem can be exactly solved if viewpoint motion is linear and perfectly predictable. One reference frame is generated at the current viewpoint, and another is generated at an appropriate future viewpoint. Viewpoints for derived frames will then fall on this line.

Unfortunately, viewpoint motion is neither linear nor perfectly predictable. But, both of these conditions are approximately true over short intervals of time. To the extent that they are not true, we rely on our heuristic technique for low-connectedness triangles to handle occlusion artifacts in directions perpendicular to the line between the two reference frames. We discuss the topic of prediction error in more detail in the next section, which describes our test-bed system.

Max's multi-layered Z-buffer [24] is an alternative to our technique of choosing two different reference frame viewpoints. Our two-viewpoint technique is particularly well suited to post-rendering warping, because it takes advantage of the system's knowledge about likely derived-frame viewpoints to select which surfaces in the scene to sample. However, our technique often samples the same surface twice, whereas the multi-layered Z-buffer does not. Generating reference frames for our technique requires fewer changes to existing rendering hardware than the multi-layered Z-buffer requires.

6 Test-bed System

The color plates and the accompanying videotape were generated using our software test-bed. We used an Abekas to transfer frames produced by the test-bed to videotape. Figure 11 (next page) is a conceptual diagram of the system.

6.1 Viewpoint Motion

We simulate the user's motion through the model with a B-spline curve. The maximum movement speed is 1.0 m/sec (typical walking speed). Each derived frame is generated by warping and compositing the two closest reference frames. As stated earlier, one of these reference frames is near a past viewpoint, and the other is near a future viewpoint.

Figure 12 shows the time-line for reference frame rendering and for the warping that produces derived frames. Because we always require one reference frame which is at a future position, and we must know a reference frame's viewpoint when we *start* to render it, we must predict future position 400 msec in the future for a 200 msec reference-frame rendering time.

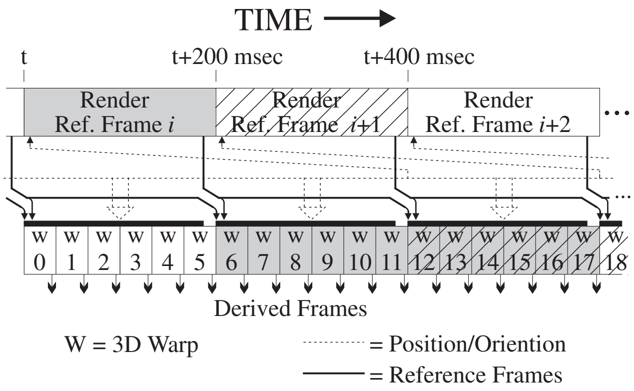

Figure 12: *Time-line (in simulated time) for the software test-bed. This diagram shows reference frames rendered at 5 frames/sec, and derived frames generated at 30 frames/sec. The test-bed can also generate derived frames at 60 frames/sec. Note that position information is shown moving backwards in time in order to render the reference frames. This backwards movement is achieved using prediction.*

Although our simulator knows the entire past and future viewpoint path, an actual system would have to determine the future positions using imperfect prediction. In order to simulate prediction error, our test-bed adds a random perturbation to the known future position before rendering each reference frame. The magnitude of the random perturbations in each axis is evenly distributed between zero and a maximum; thus the average perturbation is one-half the maximum. Our accompanying videotape shows video sequences using average per-axis perturbations of 2 cm and 5 cm for the simulated prediction interval of 400 msec.

In an actual head motion prediction system with a shorter prediction interval of 100 msec, Azuma and Bishop [6] cited an average per-axis error of 0.36 centimeters, although their peak error (over the entire sequence) was 15 centimeters. In theory, quadrupling the prediction interval increases average error by a factor of 16, so we would expect an actual system to have an average prediction error of about 5.7 cm.

Prediction for our application may be able to achieve lower average error than systems which use motion prediction for latency reduction alone (without any form of post-rendering warp). The reason is that there is a tradeoff in motion prediction between reducing average error and reducing perceptually disturbing high-frequency jitter [5]. A post-rendering warp can eliminate the jitter, and thus the predicter can be optimized for low average error.

Our rendered images have a large field-of-view to allow for head rotation and pixel movement due to translation. They currently have a $60°$ vertical field of view, and a $90°$ horizontal field of view. The derived frame has a $45°$ vertical by $60°$ horizontal field of view. In an actual system, the extra size needed in the reference frames would depend primarily on the maximum rate of head rotation and on the ability of head rotation to be predicted. Typical rates of head rotation are less than $100°$/sec (=$40°$/400 msec) [4].

6.2 Anti-aliasing

Chen and Williams [12] pointed out that reference frames should not be anti-aliased, because the blending of foreground and background colors at silhouette edges is view dependent. Furthermore, the 3D warp requires a single disparity value, and the disparity value of a blended pixel is ambiguous. We implement the same approach to this problem suggested by [12]: Our system generates reference frames at high resolution, warps them, then averages groups of warped samples to produce the derived frame. In other words, we are performing super-sampled anti-aliasing where the averaging is deferred until after the warp. Max [23] uses a similar technique, but with a coverage mask rather than full super-sampling, so Z's and normals are not super-sampled.

The super-sampling in the derived frame is on a 2x2 grid. We chose the angular resolution (that is, the number of pixels per degree) of the reference frames so that for most surfaces we expect a one-to-one ratio between reference frame pixels and derived frame super-samples. This super-sampling strategy allows us to reduce aliasing artifacts which are introduced at either the rendering or the warping stage. A 3x3 or 4x4 super-sampling grid would yield even better results.

7 Discussion

7.1 Performance

Although our test-bed system is not designed to run in real-time, it is important to consider the potential of the test-bed's algorithm for future real-time implementation. Our more primitive real-time system indicates that real-time performance for 3D warps is within reach. However, it would be much more difficult to achieve real-time performance of our test-bed's reconstruction algorithm without making some modifications. In particular, the use of linearly interpolated triangles for reconstruction is very expensive.

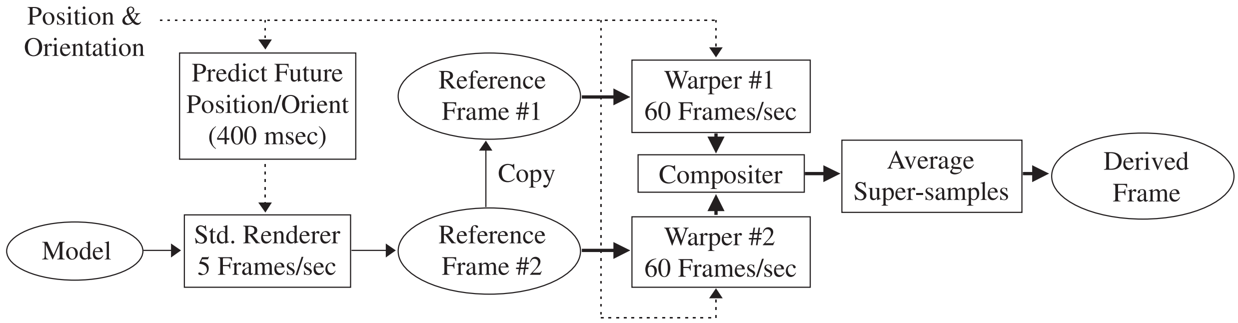

Figure 11: *Conceptual diagram of software test-bed. The frame rates are in simulated time.*

Fortunately, linearly interpolated triangles are unnecessary for most of the reference frame. The vast majority of the mesh triangles cover only one or two derived-frame pixels. For these triangles, the full triangle rendering machinery is not necessary. These extremely small triangles can be handled by a less expensive, more splat-like, approach. Alternatively, they could be rendered using table lookup [26]. The much less numerous large triangles can still be rendered using standard techniques.

Even greater efficiency and better bounds on computation cost can be obtained if the large triangles are not rendered at all. These large triangles are the low-connectedness "rubber-sheet" triangles that stretch between foreground and background objects. We are currently investigating the use of a post-processing blurring step that will allow us to avoid rendering these triangles. We are also exploring the use of a hybrid mesh-splat technique that would combine the performance advantages of splat-based reconstruction with the quality advantages of mesh-based reconstruction.

Even with improvements in the reconstruction efficiency, it is likely that in the short term our algorithm would need to use hardware designed with the algorithm in mind. The end of this section discusses some properties of the algorithm which make it amenable to cost-effective hardware implementation.

In the longer term, such hardware may not be necessary. One of the fundamentally attractive properties of post-rendering warping is that the work required for it is proportional only to the number of pixels in the display (and partially on the field of view)—it is *independent* of scene complexity.[1] So as scene complexity grows, the work for rendering will grow with it, but the work for warping will not. Thus, image warping is asymptotically cheaper than conventional rendering.

The cost of graphics systems, and even computer systems in general, is increasingly being determined by their required memory bandwidth and memory-access latency. The 3D warp has several important properties which lend it to relatively low-cost implementation:

1. The reference frames are traversed in a completely regular order, so that reads from these frames are coherent and can be grouped into large, efficient transfers from memory.

2. The derived frames are generated in an almost-regular order. At any one time, there is a small working set of possible memory locations which might be written. The particular location that is written depends upon the disparity value stored with the reference-frame pixel. An appropriately designed system can maintain this working set in a small cache, and use only large block transfers to main memory. [21]

3. The computations involved are very regular. Many of the necessary expressions in the warp can be evaluated incrementally (for example, replacing multiplies with adds). [27]

The regular memory access pattern of the 3D warp is very different from that encountered in standard rendering, where (without pre-sorting) any triangle can touch any pixel in the frame-buffer.

7.2 Limitations & Future Work

There are a number of areas besides reconstruction efficiency in which we believe we can make major improvements to our post-rendering warping technique. One such area is the connectedness calculation. The current calculation is probably more complex (and thus expensive) than necessary, and we are exploring a simpler alternative.

We could also add more flexibility to our algorithm that chooses reference-frame viewpoints. The current algorithm is most appropriate when the user is moving fairly rapidly, so that old reference-frame images are of no use. During periods of very little movement, it would be reasonable to render and warp several (more than two) reference frames in a small cloud surrounding the user's position. Even when the user is moving rapidly, it would be useful to supplement the reference frames chosen by our algorithm with reference frames rendered from a few carefully pre-chosen viewpoints. For example, in an architectural model supplementary reference frames could be generated in nearby doorways. These supplementary frames would have a good view of objects in adjacent rooms which are likely to be occluded in the standard reference frames.

Our technique does not correctly handle view dependent shading—specular highlights will jump around at the reference frame rate. Max [24] has already solved this problem in a 3D warp by using deferred shading [38]. It might make sense to implement partially deferred shading, in which only the most strongly view dependent parts of the shading calculation are deferred. Our technique also has problems with scenes which are not static. In such scenes the moving objects will move in a jerky manner. We believe that we could incorporate moving objects by augmenting the reference frames with per-pixel motion vectors, similar those proposed by [13]. Occlusion artifacts would probably be more pronounced with this strategy, because the reference-frame viewpoint choice algorithm can not be guaranteed to work for even a single moving occluder.

It would be interesting to explore the use of a 3D warp in conjunction with image-layering techniques. The image-layers in Talisman could be re-rendered less often if they were warped using a 3D warp rather than an affine warp. Regan and Pose's priority rendering would probably also benefit from a 3D warp. Finally, if some polygons could be rendered directly into the derived-frame (similar to Greene and Kass's technique), then nearby objects and moving objects could be represented more accurately. But this modification

[1] This is not strictly true for our current test-bed system, since rapidly alternating disparity values could cause many large triangles to be rendered as part of the reconstruction process. The modifications of the reconstruction technique that we are investigating would eliminate this possibility.

has a penalty—it would increase latency and complicate system design.

The piecewise linear reconstruction kernel of a mesh-based reconstruction algorithm is certainly not optimal. There is interesting research to be done by mathematically investigating 3D warp reconstruction. Our connectedness enhancement to the mesh-based algorithm has some undesirable properties. In particular, it shaves $1/2$ of a pixel off of the edge of all objects at foreground/background silhouettes. It can therefore cause objects in the reference frame that are only a single pixel wide to disappear.

We have made some interesting observations about the occlusion artifacts that we do observe with our system. These artifacts can be divided into two categories: Those that are due to violation of the "single-occluder" assumption, and those are that due to violation of the "viewpoint-on-line" condition (i.e. that derive-frame viewpoints be on the 3D line between reference-frame viewpoints). Somewhat surprisingly, we have observed very few artifacts from the first category, although certainly scenes can be constructed for which they would be more common. Most of the artifacts we have seen result from violations of the second condition.

Violations of the "viewpoint-on-line" condition have two different causes. The first is an actual viewpoint path that is not linear. The second is an error in the position prediction used to determine reference-frame viewpoints. In the video sequences produced by our test-bed, the simulated prediction error accounts for most of the "viewpoint-on-line" violations. The number and severity of occlusion artifacts from our technique is thus strongly dependent on the accuracy of position prediction. We need to more thoroughly determine the accuracy we can expect from position prediction.

Because the error in position prediction increases approximately as the square of the prediction interval, occlusion artifacts from our technique could be substantially reduced in exchange for increased latency. For example, if a latency of 200 msec was acceptable (the same that one would get with the 5 Hz conventional rendering alone), then the position prediction error could be reduced by about a factor of four. The 60 Hz frame rate would still be maintained. In an animation where the path is completely predetermined, prediction error can be eliminated entirely.

In order to definitively resolve many of the questions related to occlusion and position prediction, it will be necessary to construct an actual real-time system that uses a multi-reference-frame 3D warp, and test it with real users performing real tasks. However, many of the questions could be reasonably addressed by gathering tracker data from users exploring a model using a powerful conventional rendering engine, then simulating those same movements in the same model with our test-bed.

8 Conclusion

We have demonstrated that a 3D warp, when used in conjunction with two reference frames and our reconstruction and compositing algorithm, can produce derived frames of near reference-frame quality. In simulation, the technique can increase a system's frame rate from 5 Hz to 60 Hz. It can also compensate for rendering-system latency, and allow low-latency remote display of imagery.

The technique differs from Regan and Pose's priority rendering and Talisman's approach in that only two reference frames are used at any one time to produce the final output, rather than a larger number of frames or surfaces which are rendered at varying rates. The reference frames are produced by a slightly enhanced standard rendering engine which outputs linearly interpolated world-space pixel normals as well as 1/Z and RGB information. This information is already maintained internally by rendering engines that allow automatic generation of texture coordinates from normals.

This paper makes several new contributions. We have demonstrated in simulation that the 3D warp can be used for real-time post-rendering warping, thus compensating for viewpoint translation as well as rotation. We have shown that two properly chosen reference frames can eliminate most occlusion artifacts, and described how to choose the reference-frame viewpoints. Our compositing/reconstruction algorithm combines careful reconstruction with compositing of reference frames from different viewpoints. Finally, we have outlined the current obstacles to real-time implementation of our algorithm and sketched out approaches to these problems.

Our technique does not always produce perfect derived frames, but their quality generally is very good. Approximation is a long-accepted tradition in computer graphics; our technique is an addition to the stable of approximation strategies. For a large class of applications, we believe it has the promise of providing a performance improvement which substantially outweighs its artifacts.

9 Acknowledgements

We would like to thank our sponsors. This research was supported primarily by DARPA, under contract #DABT 63-93-C-0048. William Mark was supported in part by fellowships from the Microsoft Corporation and the Link Foundation. Leonard McMillan was supported in part by a Division, Inc. fellowship. Additional support for this work was provided by the NSF/DARPA Science and Technology Center for Computer Graphics and Scientific Visualization (contract #ASC-8920219).

We are fortunate to work in a stimulating and cooperative environment. In particular, we would like to thank Fred Brooks, Henry Fuchs, Steve Molnar, Turner Whitted, Nick England, David Ellsworth, Ken Weaver, David Harrison, Peggy Wetzel, Marc Olano, Mary Whitton, Andrei State, Carl Mueller, Bill Dally, and Dan Aliaga, who all provided useful advice and/or assistance. Finally, we thank the entire UNC Walkthrough group for providing the architectural model we used to demonstrate our systems.

References

[1] Stephen J. Adelson and Larry F. Hodges. Stereoscopic ray-tracing. *The Visual Computer*, 10(3):127–144, 1993.

[2] Stephen J. Adelson and Larry F. Hodges. Generating exact ray-traced animation frames by reprojection. *IEEE Computer Graphics and Applications*, 15(3):43–52, 1995.

[3] Daniel G. Aliaga. Visualization of complex models using dynamic texture-based simplification. In *Proceedings of IEEE Visualization 96*, pages 101–106, October 1996.

[4] Ronald Azuma. *Predictive Tracking for Augmented Reality*. PhD thesis, University of North Carolina at Chapel Hill, 1995. Available as UNC-CH Computer Science TR95-007, at http://www.cs.unc.edu/Research/tech-reports.html.

[5] Ronald Azuma and Gary Bishop. Improving static and dynamic registration in an optical see-through hmd. In *Computer Graphics Annual Conference Series (Proceedings of SIGGRAPH 94)*, pages 197–204, Orlando, Florida, July 1994.

[6] Ronald Azuma and Gary Bishop. A frequency-domain analysis of head-motion prediction. In *Computer Graphics Annual Conference Series (Proceedings of SIGGRAPH 95)*, pages 401–408, Los Angeles, CA, August 1995.

[7] Sig Badt, Jr. Two algorithms for taking advantage of temporal coherence in ray tracing. *The Visual Computer*, 4(3):123–131, 1988.

[8] Denis R. Breglia, A. Michael Spooner, and Dan Lobb. Helmet mounted laser projector. In *The 1981 Image Generation/Display Conference II*, pages 241–258, Scottsdale, Arizona, Jun 1981.

[9] Dick Burbidge and Paul M. Murray. Hardware improvements to the helmet mounted projector on the visual display research tool (VDRT) at the naval training systems center. In *Proceedings SPIE*, volume 1116, pages 52–60, Orlando, Florida, Mar 1989.

[10] CAE Electronics, Ltd. Wide-field-of-view, helmet-mounted infinity display system development. interim report AFHRL-TR-84-27, US Air Force Human Resoures Laboratory, Operations Training Division, Dec 1984.

[11] Shenchang Eric Chen. QuickTime VR — an image-based approach to virtual environment navigation. In *Computer Graphics Annual Conference Series (Proceedings of SIGGRAPH 95)*, pages 29–38, Los Angeles, California, August 1995.

[12] Shenchang Eric Chen and Lance Williams. View interpolation for image synthesis. In *Computer Graphics Annual Conference Series (Proceedings of SIGGRAPH 93)*, pages 279–288, Anaheim, California, August 1993.

[13] John P. Costella. Motion extrapolation at the pixel level. Unpublished paper available from http://www.ph.unimelb.edu.au/˜jpc, January 1993.

[14] Steven J. Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F. Cohen. The lumigraph. In *Computer Graphics Annual Conference Series (Proceedings of SIGGRAPH 96)*, pages 43–54, New Orleans, Louisiana, August 1996.

[15] Ned Greene and Michael Kass. Approximating visibility with environment maps. Technical Report #41, Apple Computer, November 1994.

[16] Georg Rainer Hofmann. The calculus of the non-exact perspective projection. In *Proceedings of the European Computer Graphics Conference and Exhibition (Eurographics '88)*, pages 429–442, Nice, France, Sep 1988.

[17] S. Laveau and O. D. Faugeras. 3-D scene representation as a collection of images. In *Proc. of 12th IAPR Intl. Conf. on Pattern Recognition*, volume 1, pages 689–691, Jerusalem, Israel, October 1994.

[18] Stephane Laveau and Olivier Faugeras. 3-D scene representation as a collection of images and fundamental matricies. Technical Report RR #2205, INRIA, February 1994. Available from ftp://ftp.inria.fr/INRIA/tech-reports/RR/RR-2205.ps.gz.

[19] Marc Levoy and Pat Hanrahan. Light field rendering. In *Computer Graphics Annual Conference Series (Proceedings of SIGGRAPH 96)*, pages 31–42, New Orleans, Louisiana, August 1996.

[20] Andrew Lippman. Movie-maps: An application of the optical videodisc to computer graphics. *Computer Graphics (Proceedings of SIGGRAPH 80)*, 14(3):32–42, July 1980.

[21] William R. Mark. Efficient two-phase architecture for disparity-based image warping: Overview and memory bandwidth analysis. UNC COMP 290-012 (Graphics Architecture) final project writeup, May 1996.

[22] William R. Mark, Gary Bishop, and Leonard McMillan. Post-rendering image warping for latency compensation. Technical Report UNC-CH TR96-020, Univ. of North Carolina at Chapel Hill, Dept. of Computer Science, January 1996. Available at http://www.cs.unc.edu/Research/tech-reports.html.

[23] Nelson Max. Hierarchical rendering of trees from precomputed multi-layer z-buffers. In Xavier Pueyo and Peter Schröder, editors, *Rendering Techniques '96: Proceedings of the Eurographics Rendering Workshop 1996*, pages 165–174, Porto, Portugal, June 1996.

[24] Nelson Max and Keiichi Ohsaki. Rendering trees from precomputed z-buffer views. In Patrick M. Hanrahan and Werner Purgathofer, editors, *Rendering Techniques '95: Proceedings of the Eurographics Rendering Workshop 1995*, pages 45–54, Dublin, Ireland, June 1995.

[25] Tomasz Mazuryk and Michael Gervautz. Two-step prediction and image deflection for exact head tracking in virtual environments. *Computer Graphics Forum (Eurographics '95)*, 14(3):C29–C41, 1995.

[26] Leonard McMillan. A list-priority rendering algorithm for redisplaying projected surfaces. Technical Report UNC-CH TR95-005, University of North Carolina at Chapel Hill, Dept. of Computer Science, 1995. Available at http://www.cs.unc.edu/Research/tech-reports.html.

[27] Leonard McMillan and Gary Bishop. Head-tracked stereoscopic display using image warping. In S. Fisher, J. Merritt, and B. Bolas, editors, *Proceedings SPIE*, volume 2409, pages 21–30, San Jose, CA, Feb 1995.

[28] Leonard McMillan and Gary Bishop. Plenoptic modeling: An image-based rendering system. In *Computer Graphics Annual Conference Series (Proceedings of SIGGRAPH 95)*, pages 39–46, Los Angeles, CA, August 1995.

[29] Matthew Regan and Ronald Pose. An interactive graphics display architecture. In *Proceedings of IEEE Virtual Reality Annual International Symposium*, pages 293–299, Seattle, Washington, September 1993.

[30] Matthew Regan and Ronald Pose. Priority rendering with a virtual reality address recalculation pipeline. In *Computer Graphics Annual Conference Series (Proceedings of SIGGRAPH 94)*, pages 155–162, Orlando, Florida, July 1994.

[31] Bruce Riner and Blair Browder. Design guidelines for a carrier-based training system. In *Proceedings of IMAGE VI Conference*, pages 65–73, Scottsdale, Arizona, Jul 1992.

[32] Johnathan Shade, Dani Lischinski, David H. Salesin, Tony DeRose, and John Snyder. Hierarchical image caching for accelerated walkthroughs of complex environments. In *Computer Graphics Annual Conference Series (Proceedings of SIGGRAPH 96)*, pages 75–82, New Orleans, Louisiana, August 1996.

[33] Richard H. Y. So and Michael J. Griffin. Compensating lags in head-coupled displays using head position prediction and and image deflection. *Journal of Aircraft*, 29(6):1064–1068, Nov-Dec 1992.

[34] Richard Szeliski. Image mosaicing for tele-reality. Technical Report CRL 94/2, Digital Equipment Corp. Cambridge Research Lab, May 1994. Available at http://www.research.digital.com/CRL/publications/crl-rr.html.

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

[36] Jay Torborg and James T. Kajiya. Talisman: Commodity realtime 3D graphics for the PC. In *Computer Graphics Annual Conference Series (Proceedings of SIGGRAPH 96)*, pages 353–364, New Orleans, Louisiana, August 1996.

[37] Lee Westover. Footprint evaluation for volume rendering. *Computer Graphics (Proceedings of SIGGRAPH 90)*, 24(4):367–376, August 1990.

[38] Turner Whitted and David M. Weimer. A software testbed for the development of 3D raster graphics systems. *ACM Transactions on Graphics*, 1(1):43–58, January 1982.

[39] George Wolberg. *Digital Image Warping*. IEEE Computer Society Press, Los Alamitos, California, 1992.