Image-Based Rendering with Occlusions via Cubist Images

Andrew J. Hanson   Eric A. Wernert [*]

Computer Science Department
Indiana University
Bloomington, IN 47405 USA

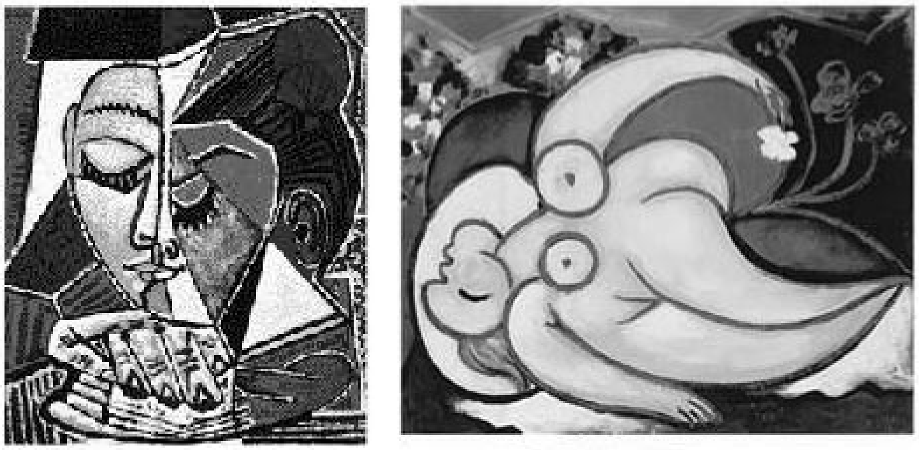


Figure 1: (a) "Head of Woman Reading" and (b) "Sleeping Nude" by Pablo Picasso. (©1998 Estate of Pablo Picasso/Artists Rights Society, N.Y.)

Abstract

We attack the problem of image-based rendering with occlusions and general camera motions by using distorted multiperspective images; such images provide multiple-viewpoint photometry similar to the paintings of cubist artists. We take scene geometry, in contrast, to be embodied in mappings of viewing rays from their original 3D intercepts into the warped multiperspective image space. This approach allows us to render approximations of scenes with occlusions using time-dense and spatially sparse sequences of camera rays, which is a significant improvement over the storage requirements of an equivalent animation sequence. Additional data compression can be achieved using sparse time keyframes as well. Interpolating the paths of sparse time key-rays correctly in image space requires singular interpolation functions with spatial discontinuities. While there are many technical questions yet to be resolved, the employment of these singular interpolation functions in the multiperspective image space appears to be of potential interest for generating general-viewpoint scene renderings with minimal data storage.

CR Categories: I.3.6 [Computer Graphics]: Methodology and Techniques—Interaction Techniques. I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism. I.3.8 [Computer Graphics]: Applications.

Keywords: Image Based Rendering; Occlusions

[*]Email: {hanson, ewernert}@cs.indiana.edu

1 Introduction

We address the problem of generating image sequences of a static scene with occluded objects using a single source image. Our approach is to abandon the geometric relationship between source-image pixels and scene geometry used in conventional image-based rendering (IBR); instead, we substitute more abstract source images inspired by the multiperspective content of cubist art, as represented in the Picasso paintings of Figure 1. There are two basic steps in the process: first is the selection of an arrangement of source pixels that covers the scene areas to be viewed; the second is the determination of key-frame projections of camera rays into this source image. Since, as in the examples of Figure 1, the geometric relationships of the source pixels may be arbitrarily stretched, distorted, or even torn, the entire geometric content of image production is shifted to the locations of the camera-ray intersections warped into the source image. The rendering of image sequences from spatially sparse and temporally sparse keyframes can in principle be accomplished by interpolating camera rays directly in the source image.

The "cubist" source images that we attempt to exploit typically contain occluded data and geometric distortions that may not be attributable to a real camera; in addition, we normally impose the condition that *no scene point appears more than once*. This distinguishes our construction from those whose source images are obtained from a sequence of physically plausible cameras, and which may contain multiple copies of the same scene element (see, e.g., the multiperspective panorama approach [16]). The most natural domains for our approach are thus terrain and terrain-like scenes that can be "shrink-wrapped" with a single continuous sheet, though other topologies can also be treated. The fact that geometry is removed from the image data is both a strength, in that the image can be warped to suit our needs, and a weakness, in that choosing this warping is a nontrivial task. Finding camera ray positions is also costly because in principle it requires having an accurate geometric model of the scene and the transformation to the source image.

There are two distinct mechanisms by which we can achieve the data compression characterizing successful IBR methods. No compression at all results if we raytrace each image pixel into the source image at each time step of the animation. We might just as well prestore the entire image sequence. By doing a spatially sparse ray sampling with dense time, we can achieve one level of data compression using texture mapping methods. The more challenging problem is to take sparse *time* samples as well; temporal interpolation of such key-frame time samples in source-image space produces a smooth time sequence without the associated cost per frame.

Hints about the relationship between desirable image warpings and camera ray interpolation can be determined by examining exact frame-by-frame projections of a moving camera's pixel rays into the 3D scene and their paths in the (warped) source image. Acceptable animations of highly occluded scenes can be constructed by choosing key frames whose interpolants closely follow the exact paths. This procedure requires only the single source image plus a chosen set of ray intersections.

A visual summary of the entire process is presented in Figure 2, which is explained in detail later.

A key feature of our approach is that it separately encodes all of a scene's photometry in a single source image in a form distinct from the geometry. This is significant for visualization applications since data in this source image can be manipulated using a variety of standard 2D image filters (e.g., color mapping, contrast tuning, edge-detection, etc.), thereby creating a complete re-rendering of the animation at the cost of a single image filtering. The option of storing per-pixel normal and position information with the source image provides additional flexibility for recomputing lighting effects.

In summary, our main result is the introduction of "cubist" multiperspective source images and the mapping of viewing rays from camera keyframes into this distorted scene to produce accurate occluded-scene animations. The benefits include potential data compression in both the spatial and temporal domains and the ability to easily modify a range of rendering attributes.

Background: Image Based Rendering. Image-based rendering (IBR) is based on the idea that prerendered pixel arrays can be associated with sufficient scene geometry to allow the accurate reconstruction of digital images appropriate to some constrained set of moving camera parameters. Thus, IBR avoids the expense of storing a full time-step array of images by enforcing camera constraints (e.g., restricting the camera motion to pivot about a fixed focal center) that enable the use of prerendered texture maps embodying very complex image models (including real-life photographs) [1, 11, 15]. The challenge of the IBR approach is to produce realistic images while using less image data than a direct animation sequence and less geometric knowledge about the scene than one would need to directly render each changing viewpoint of a textured polygon model, e.g., with textures derived from a radiosity approach or actual photographs. Unfortunately, when one attempts to move beyond the now-traditional constraints of fixed-focal center with only rotation and zoom camera motions, straightforward texture warping as a substitute for geometry breaks down and it has been found necessary to bring in additional geometry. Major obstacles to straightforward image warping include the tearing and folding of the image when general camera motions are attempted in the presence of occlusions. One approach to handling these difficulties is to obtain images from neighboring viewpoints to fill in the holes [2, 13]. The distortion maps of vanishing points in the image of a moving camera also need further image-plane geometry in the form of specially structured meshes, as investigated in [8, 7, 17]; foreground geometry can be extracted into a separate layer and the images recomposited using masks. The warping of images to account for changes in the camera model has been studied in [10]. Other methods require the addition of depth or range data [2, 11] for each source image used for rendering, or adapt simplified polygon models to the task [3]. The multiperspective panorama approach [16] achieves some extra flexibility by using a single image as a repository for composited information relevant to many viewpoints on a single constrained camera animation path. Alternative approaches based on light fields can require vast amounts of data storage [4, 9, 14].

2 Concepts

In this paper, we study the idea of IBR based on a flattened, possibly torn, wrap-around texture image (the source image) and the paths in time of camera rays mapped from their 3D scene intersection points to this single warped image. We also investigate the feasibility of sparse spatial camera-ray sampling and sparse camera-ray time samples whose interpolations can have *spatial singularities:* such singularities naturally occur wherever occlusions require jumps across unseen texture patches (see, e.g., Figure 12(c)). The ability to tolerate sparse space or time samples of the camera ray grids can greatly reduce data storage requirements. With these concepts, we are able to select camera motions and generate acceptable image sequences for many scenes containing significant geometric effects and occlusions.

The basic idea is to flatten 3D image data, e.g., of mountains, buildings, people, or even bridges, to a single 2D texture, and to create interpolations across this virtual terrain that accurately encode geometric distortion as well as the appearance and disappearance of occlusion edges. The use of such a flattened image is very similar to a method employed extensively by the cubist school of painting (see the Picasso paintings in Figure 1), which depicts objects *as they are conceptually represented in the mind* and not the way they are seen; all sides of a person or object may be presented to the viewer at once on a 2D canvas. While the viewer of a Picasso is expected to reconstruct the object being represented using the imagination, we approximate the same result using complex mappings of camera rays to the warped source image, and by analyzing the behavior that would be needed in the corresponding keyframe interpolation methods. Our approach is also closely related to methods of classical topology for representing 2D surfaces embedded in 3D; as shown in Figure 3, a complex surface may be represented by a flat polygon with certain edges identified; in our philosophy, such objects can in principle be handled if they are in contact with the ground plane by mapping them into a flattened torn or cut-out sheet, as shown in Figure 4.

We observe that the original IBR concept of simplifying continuous image generation by constraining the camera motion to a single center of rotation can, in our approach, be extended to much more general motions; the interesting feature of this is that if there are very complex areas whose interpolations we cannot handle smoothly, we can revert to forbidding certain camera parameter ranges, thus restricting the user's freedom to sets of views that have reduced rendering complexity criteria; this is a good example of an adaptation of the constrained navigation paradigm to the IBR arena, and is in some sense complementary to the multiperspective panorama concept [16]. The result is to provide a natural context for the exploitation of constraint manifolds that reduce the scope and complexity of the rendering task as well as enabling simplified navigation through the scene (see, e.g., [6]).

The procedure we propose is then summarized as follows: Instead of assigning texture to each facet of the geometric structures,

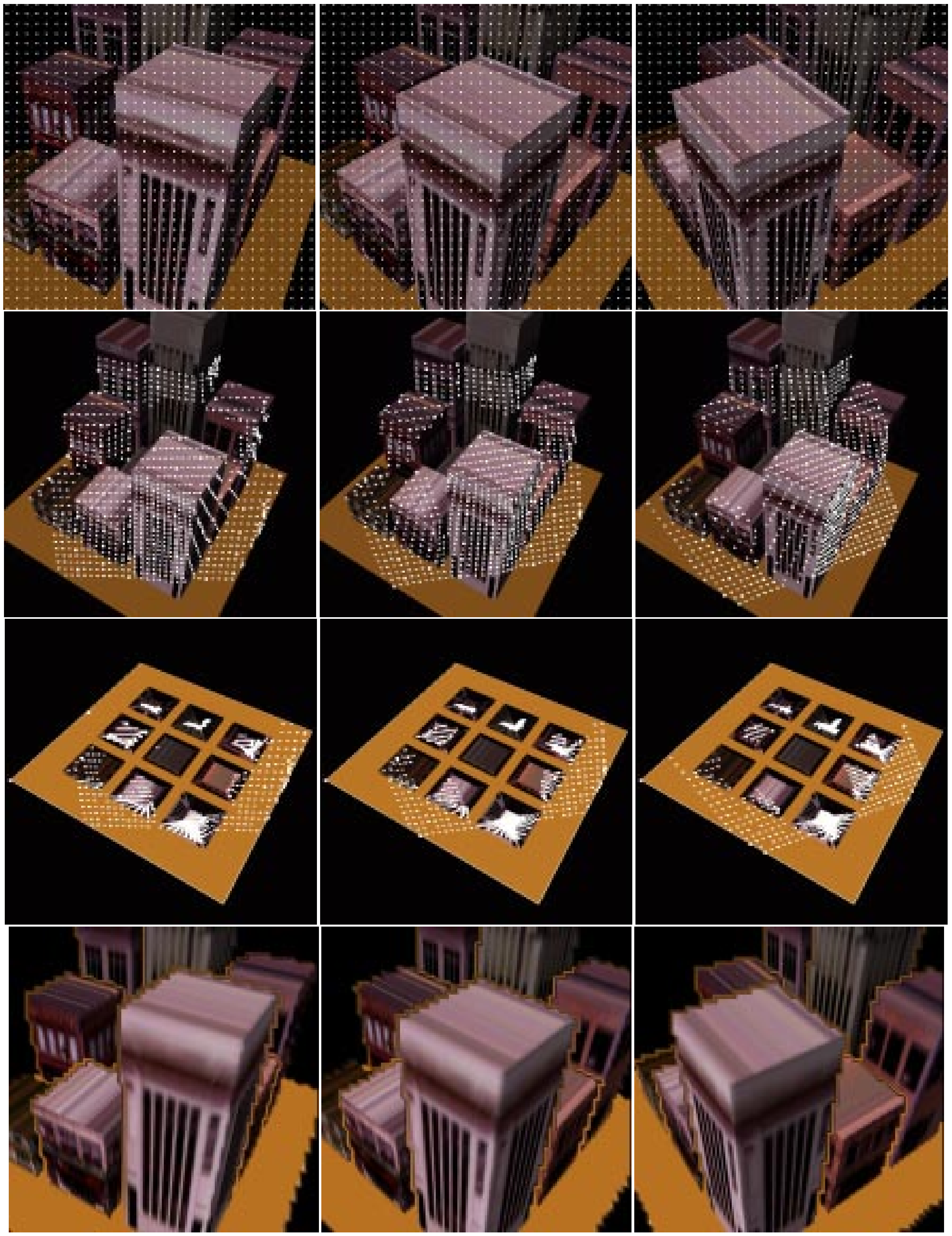


Figure 2: Frames of an animation illustrating the projection of camera rays into the 3D scene model followed by a warp to the distorted 2D source image. In the first row, the 32×32 array of dots represents the paths of rays from the camera focal point through sampled image pixel centers. In the second row, the ray intersections are seen obliquely, and the ray paths clearly make large jumps in the real spatial geometry as they pass from the occlusion edge of a nearby building to the face of a more distant building. In the third row, after the points where the rays intersect the geometry are warped to the source image, the geometry is embodied in the positioning of the ray intersections. The bottom row shows the scene as it appears reconstructed from the data using our method with a sample density of 64×64.

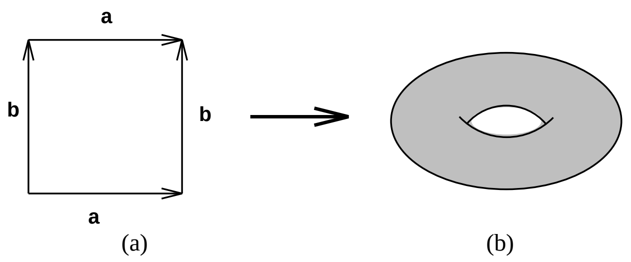


Figure 3: Representations of the torus, a surface of genus one. (a) This abstract representation with opposite edges identified as marked precisely embodies our proposal to represent photometry independently from geometry. (b) The corresponding geometric object that the camera rays would be traced to is this toroidal surface, which results from attaching the edges together as marked.

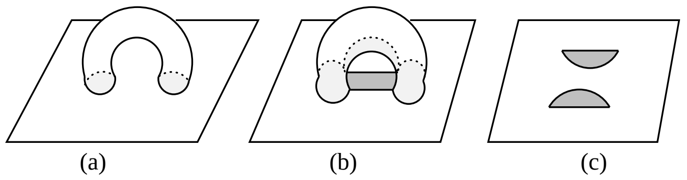


Figure 4: Example of a possible deformation of an arch into the global source image. (a) Sketch of a tubular arch attached to background plane. (b) Cutting a slit all the way around the inner hole, separating it slightly to leave two exposed edges, each shaped like a half disc. (c) Flattening the now strip-like arch down into the image plane, leaving two discontinuous cuts.

of which there may be millions in the scene, we use camera-ray casting followed by a warp to assign a geometric property to each pixel of an "unwrapped," possibly torn, flattened texture that represents (uniquely) every pixel viewable from every viewpoint in the camera navigation domain; this geometric assignment may lead to distortions and singularities, but it does not require multiple images or range data like other approaches. The fundamental issue is whether or not the per-pixel data structure associated with our source image data structure and the time taken to perform the singular warp transformation is more or less efficient than a direct textured polygon rendering. Some of the needed features might in principle be supportable via imaginative usage of present graphics hardware capabilities.

3 Fundamental Methods.

In a typical IBR method, the camera view plane is geometrically related to the prestored source image, and that image is globally warped as necessary to produce a precise representation of what the camera would see as points in the projected 3D scene. We propose the following paradigm to extend the applicability of the method to scenes with occlusion whose photometry is encapsulated in a "cubist" source image:

- **Warp Scene Texture to Single Image.** The simplest example of the type of warped texture we require is a simple terrain map texture, with a pixel intensity $I(x, y)$ assigned to each facet of the 2D scalar field $z = f(x, y)$ defining the terrain (see Figure 5). Objects with vertical or overhanging faces like buildings must be distorted so that the pixels originally assigned to the roof are shrunk to accommodate wall textures as shown in Figure 6. Scenes with non-trivial topology can be handled using one of several variations on Figure 3 or Figure 4. Note that the basic data model required by our method contains *only* RGB color data at each pixel of the source image; one may store additional geometric information to achieve specific effects, but this is not required.

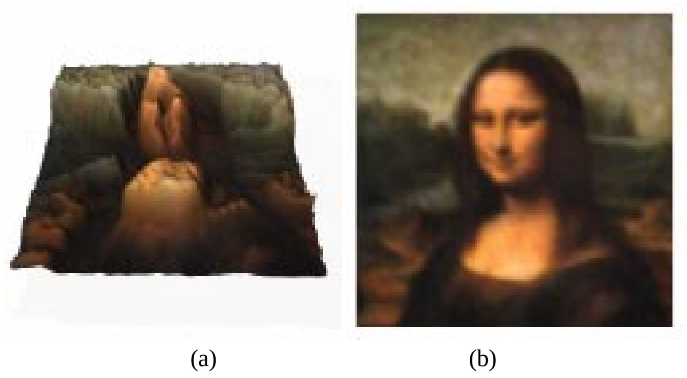


Figure 5: Terrain models are perfectly adapted to our method: (a) Viewed obliquely, a non-pathological terrain image has both texture and topography described by some single-valued elevation function $z = f(x, y)$. (b) Viewed from an appropriate point (possibly of infinite altitude), we see a single 2D texture map that can be used to texture all the polygons of the terrain.

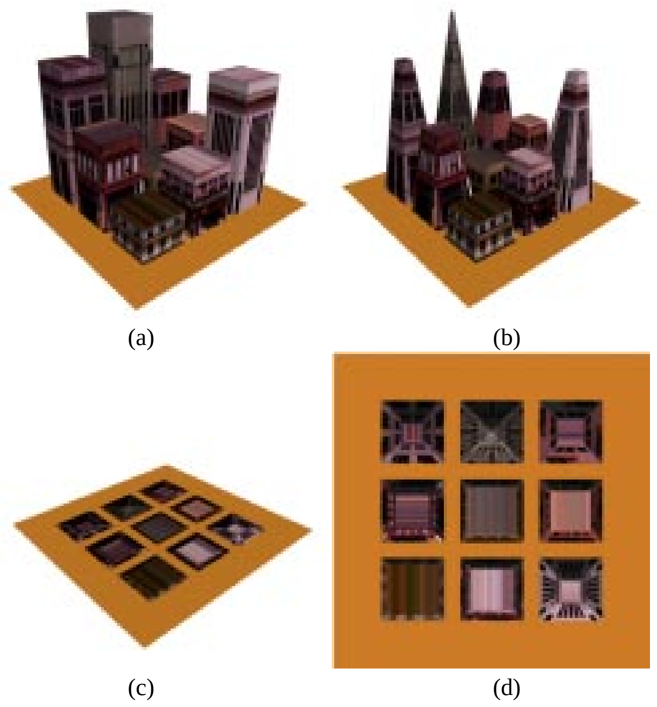


Figure 6: Demonstration of scene warping for polyhedral objects such as conventional buildings. (a) Oblique view of normal scene geometry. (b) The first stage of warping for a building is to taper the top to form a frustum, thus exposing the vertical walls to the view of a distant aerial camera. (c) Flattening the deformed scene into a plane preserves all the viewable texture information. (d) The 2D source image contains all viewable pixels in the 3D scene, but has lost geometric accuracy.

- **Project Camera Rays to Scene.** To construct a key-frame data structure, pass the camera rays through each image pixel (or a sampled subset) and record intersection points with the true 3D scene geometry. This is closely related to the construction of a shadow map, and is illustrated in the first and second rows of Figure 2: when we trace rays from the eyepoint through each sampled pixel to the scene geometry, an oblique view shows the shower of rays skipping past occluded polygons to strike more distant objects. For certain applications, samples more heavily weighted around the center of attention might be preferred to regularly-spaced pixel samples.

- **Warp Ray Intersections from Scene to Source Image.** The source image is a rectangular array resulting from a rubber-sheet warping of all the "shrink-wrapped" scene texture to a (possibly torn) 2D image. The ray intersections form a warped grid in this space, with large gaps wherever an occlusion boundary occurred in the original projection. If there are no occlusions, there is still a non-trivial warping affect that becomes singular at the instant an occlusion boundary appears. Thus each point of the camera ray grid stores only the texture coordinates of the corresponding source image point. The third row of Figure 2 shows the source-image mapping of the rays. Figure 2 as a whole shows typical changes in the ray locations after small camera motions.

- **Fill in Sparse Space/Time Samples.** Sparse spatial samples can be used to generate much richer detail by using the source image as a texture map and interpolating appropriately in space (see the fourth row of Figure 2). Sparse time samples of camera ray intersections with the scene can also be filled in to produce dense-time animations using interpolation methods; because of discontinuities in the interpolation when occlusions are encountered, this can be a challenging task. The time needed to reconstruct an image is independent of the resolution of the source image and is dependent on the resolution of the keyframes and the complexity of the temporal interpolation method.

Image-Based Spatial Compression and Color Manipulation. If minor distortions are acceptable for the purpose of an inexpensive visualization, one can replace the image derived from sampling *all pixels* at the ray intersections in the source image by an equivalent image defined by the *texture coordinates* at the ray intersections; using only a sparse sample of camera rays can give a high quality result with only a fraction of the data storage for rays. Figures 11(a) and 11(b) illustrate the use of this method; the sparse pixel samples in (a) give a clearly inferior image to that in (b) computed using a texture map based on the identical ray samples. Another bonus for data that requires transformations such as color mapping for visualization applications is that we can easily reassign or modulate the colors in the image, add contour lines, and so on; this visualization application is illustrated in Figure 7, which shows a B2 bomber wing's unwrapped density field as a source image and several visual transformations that can be performed when the image is rendered.

Just as various advantages are achieved in other IBR techniques by adding information such as depth or range, we can get new capabilities by adjoining surface normal information to the base image on a per-pixel basis as in Figures 7(d) and 11(c); this allows us to independently add approximate shading and specular highlight enhancement interactively while varying either the camera motion or the light direction. In principle, one might even simulate a mirror by following a set of ray bounces and treating them in IBR mode as well.

Animating Time Sequences. We are guaranteed to get geometrically accurate samples of the source image if we create one set of camera pixels per minimal camera displacement; however, this is not very useful, since if there is one ray per pixel, the amount of data stored is at least as great as if we had stored the corresponding stack of completely rendered images and played them back as an (uncompressed) MPEG movie. This is why we demonstrated in Figure 11(b) that we could convert from a dense set of camera rays to a sampled set (say every 4th image pixel) and get better results by using the more sparsely sampled set as texture coordinates. The result is that with 1/16 of the data, we get a good approximation to a full animation.

However, to get accurate images, we had to store camera pixel arrays for every minimal time step. What happens when we also reduce the sampling rate in time and interpolate temporally to fill in the intermediate steps? The answer is very interesting from the point of view of camera-constraint technology: In Figure 8, we show five frames of a simple geometric image with a cone on a plane. Images (a) and (e) are keyframes, while the three intermediate images (b), (c), (d) are interpolations. Rotating the camera around the cone gives quite reasonable intermediate images, even though, in principle, this should be a very hard case due to the infinite aspect graph of a conical object. Contrast this with Figure 9, which also has three interpolated frames in the middle of two keyframes; the path now is a rigid camera translation, and the intermediate frames are badly distorted. The lesson is that *camera motion constraints can be chosen to optimize the effectiveness of simple interpolations*.

There is also a converse approach to optimizing the appearance of an interpolated time sequence. Recall first that we have made no compelling argument for any unique way to warp an image so that, cubist-style, all viewed pixels appear in the source image; there are many ways. We may thus select a camera path constraint and a mapping to the source image pixels, and then attempt to find an optimal *rewarping* of the source image to provide better performance. In the next section, we give some of the details of how to begin such an analysis.

4 Analysis: Deducing Acceptable Approximations

Our guiding principle is to use low-storage, singular interpolation methods to transform as accurately as possible among constrained key-frames. This appears adequate to produce reasonable renderings of occluded scenes using single source images, and thus to achieve most of the desired features of IBR. In practice, the deduction of appropriate interpolations is difficult: to understand reliable results about the situation, we can study very closely spaced sequences of projected camera grids in a source image to see *exactly* how the interpolations among more widely-spaced camera key-frames should be carried out. We may then later attempt to adjust our heuristic interpolations to approach more closely the veridical image data.

To illustrate our application of this procedure, we begin by showing in Figure 12(a) four time frames of a typical scene with a cone-shaped object, where a column of five camera rays has been selected for study. Figure 12(b) shows the locations of these pixels as a function of time in the flattened source image. Figure 12(c) shows the comparison in base-image-space between a cubic interpolation among pixels in image space (red) and the frame-accurate camera motion (blue). The blue paths simulate a "perfect" time interpolation among a set of sparse keyframes. Using a cubic interpolation produces a badly distorted animation unless the camera motion is constrained so that the interpolations match, or the scene is so featureless that the image-space interpolations cannot differ. In general, one has massive discrepancies such as those shown in Figure 12(c). Nevertheless, this fact itself can now be used to guide further development of this work; the next step is to find families of

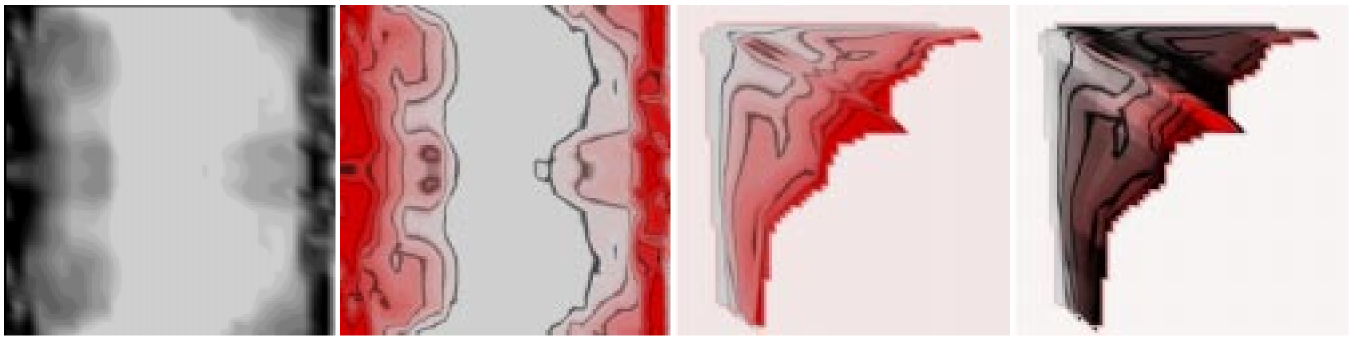


Figure 7: (a) Pressure density on a B2 bomber wing unwrapped to give a flat source image. (b) A useful color map of this density data (this is not modern art). (c) Unlit rendering of the wing with color-map texels indexed by a 64×64 camera keyframe. (d) Lit rendering of the wing, where the stored density data has been augmented by the local surface normals. (Data from IRIS Explorer distribution.)

interpolations, possibly combined with rewarping the source image, that minimize the difference between the bare spline interpolation and the veridical path.

5 Adjustment of Heuristics among Scene Types

The complete determination of an appropriate interpolation for sparse time frames is in general an unsolvable problem. However, with the philosophy of applying domain-dependent constraints on the camera motion to scenes with various characteristics, we can list a variety of cases and speculate on how to handle them appropriately:

These families fall into the following categories:

- **Fixed focal point.** Traditional IBR works perfectly if you do not move the camera's focal point, even if there are occlusions.
- **Match image symmetry.** If the image, for example, has vertical columns in the image foreground, one can move the camera along vertical paths and have negligible evidence of new data appearing behind the columns; there is one simple family of singularities at the column edges, and the interpolation can directly accommodate that using prestored edge data.
- **Pyramid: Finite Aspect Graph Occlusions.** It is relatively straightforward to handle pyramid-like objects, whose full texture map can be captured in a single aerial or top-down image, and whose aspect graphs are finite.
- **Cones: Infinite Aspect Graph Objects.** When one extends the allowed objects to cones, one has an infinite aspect graph, so that the occlusions must continually roll across to accommodate new views.
- **Terrain with Occlusions.** This is similar to the cones problem, but contains uncontrollable numbers of extra occlusions.
- **Buildings.** No more than three of five visible building sides can be viewed at once by a real camera; with appropriate warping into a frustum, the building texture can be made easily into a warped image that is visible to cameras on all sides.
- **Arches and Loops.** These require topological tearing and unfolding of the image in nontrivial ways (see Figures 3 and 4). The images correspond very closely to the cubist Picasso images in the introduction, and the interpolation must be adjusted to jump over the required slits in the image to avoid unacceptable effects.

6 Conclusion

We have studied an approach to image-based rendering in which we assign photometric properties to each pixel of an "unwrapped," possibly torn, flat texture (the source image) that represents every pixel viewable from every viewpoint in the camera navigation domain. The scene geometry is then embodied in the process of casting camera rays into the 3D scene followed by the warping of those intersection points into the source image. Depending upon our knowledge of the space interpolation and time interpolation characteristics of a particular domain, we can achieve animations with free camera motion but much lower storage requirements than a full animation sequence of images; the use of a single source image also confers many additional advantages since no rerendering is required to incorporate color transformations and lighting effects. Future work would include further studies of the nature of singular image-space ray interpolations, possibly accompanied by complementary rewarping of the source image.

Acknowledgments

This research was made possible in part by NSF infrastructure grant CDA 93-03189 and the support of the Indiana University Advanced Information Technology Laboratory.

Note added: since this paper was written, two papers have appeared that utilize related but not identical concepts that may also be of interest to the reader: the multiple-center-of-projection approach [12], and multiple viewpoint rendering [5].

References

[1] Shenchang Eric Chen. Quicktime VR - an image-based approach to virtual environment navigation. In Robert Cook, editor, *SIGGRAPH 95 Conference Proceedings*, Annual Conference Series, pages 29–38. ACM SIGGRAPH, Addison Wesley, August 1995. Held in Los Angeles, California, 6–11 August 1995.

[2] Shenchang Eric Chen and Lance Williams. View interpolation for image synthesis. In James T. Kajiya, editor, *Computer Graphics (SIGGRAPH '93 Proceedings)*, volume 27, pages 279–288, August 1993.

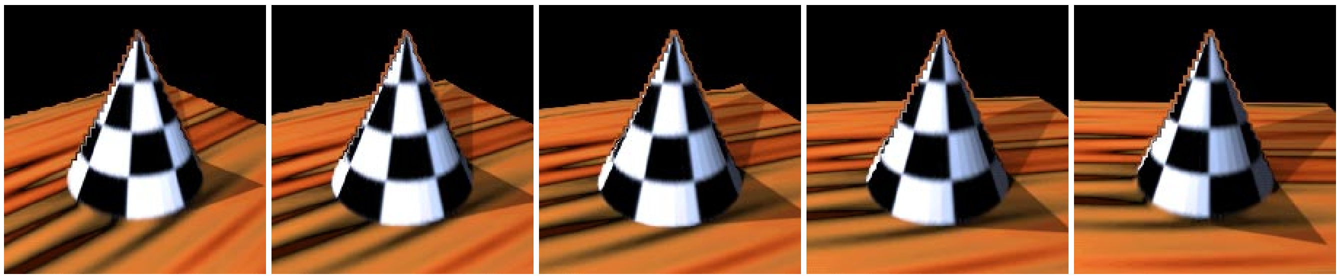


Figure 8: Rendering a cone from several viewpoints is challenging because its aspect graph is not finite. Nevertheless, for appropriately constrained camera paths, the linear source-image interpolations (b),(c),(d) between the keyframe views (a) and (e) are relatively stable.

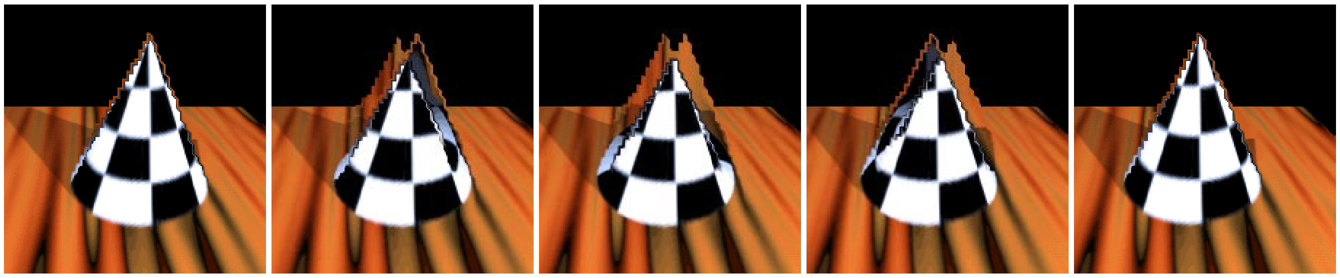


Figure 9: If we choose a less appropriate camera path such as this one with rigid translational motion, the linear source-image interpolations (b),(c),(d) between the keyframe views (a) and (e) are unacceptably distorted.

[3] L. Darsa, B.C. Silva, and A. Varshney. Navigating static environments using image space simplification and morphing. In *Proceedings of 1997 Symposium on Interactive 3D Graphics*, pages 25–34, 1997.

[4] Steven J. Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F. Cohen. The lumigraph. In Holly Rushmeier, editor, *SIGGRAPH 96 Conference Proceedings*, Annual Conference Series, pages 43–54. ACM SIGGRAPH, Addison Wesley, August 1996. Held in New Orleans, Louisiana, 4–9 August 1996.

[5] M. Halle. Multiple viewpoint rendering. In *SIGGRAPH 98 Conference Proceedings*, Annual Conference Series. ACM SIGGRAPH, Addison Wesley, 1998.

[6] A. J. Hanson and E. Wernert. Constrained 3D navigation with 2D controllers. In *Proceedings of Visualization '97*, pages 175–182. IEEE Computer Society Press, 1997.

[7] M. Hirose, S. Watanabe, and T. Endo. Generation of wide-range virtual spaces using photographic images. In *Proceedings of VRAIS '98*, volume 5, pages 234–241, 1998.

[8] Youichi Horry, Ken ichi Anjyo, and Kiyoshi Arai. Tour into the picture: Using a spidery mesh interface to make animation from a single image. In Turner Whitted, editor, *SIGGRAPH 97 Conference Proceedings*, Annual Conference Series, pages 225–232. ACM SIGGRAPH, Addison Wesley, August 1997.

[9] Marc Levoy and Pat Hanrahan. Light field rendering. In Holly Rushmeier, editor, *SIGGRAPH 96 Conference Proceedings*, Annual Conference Series, pages 31–42. ACM SIGGRAPH, Addison Wesley, August 1996. Held in New Orleans, Louisiana, 4–9 August 1996.

[10] W. R. Mark, L. MacMillan, and G. Bishop. Post-rendering 3d warping. In *Proceedings of 1997 Symposium on Interactive 3D Graphics*, pages 7–16, 1997.

[11] Leonard McMillan and Gary Bishop. Plenoptic modeling: An image-based rendering system. In Robert Cook, editor, *SIGGRAPH 95 Conference Proceedings*, Annual Conference Series, pages 39–46. ACM SIGGRAPH, Addison Wesley, August 1995. Held in Los Angeles, California, 6–11 August 1995.

[12] P. Rademacher and G. Bishop. Multiple-center-of-projection images. In *SIGGRAPH 98 Conference Proceedings*, Annual Conference Series. ACM SIGGRAPH, Addison Wesley, 1998.

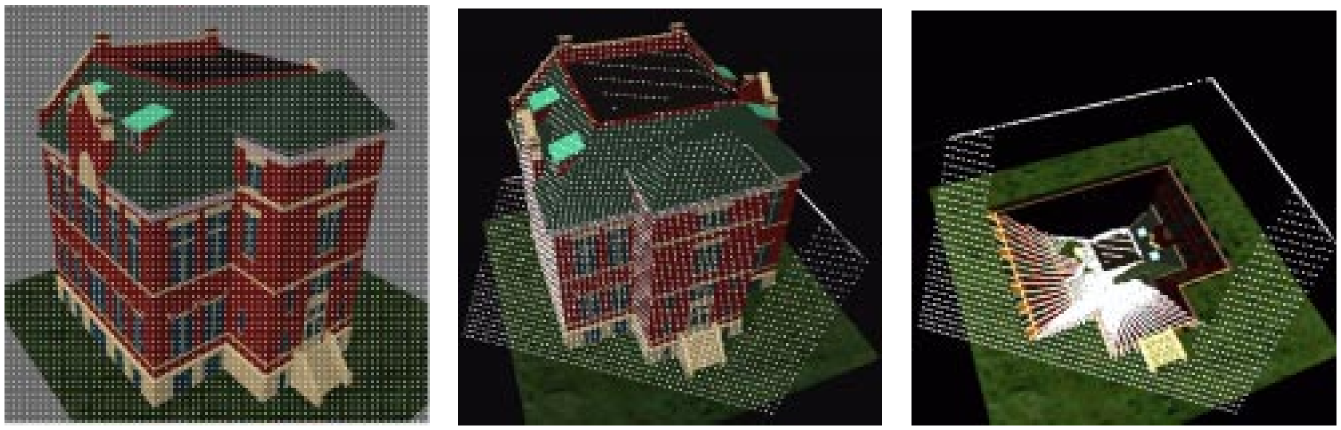
[13] Steven M. Seitz and Charles R. Dyer. View morphing: Synthesizing 3D metamorphoses using image transforms. In Holly Rushmeier, editor, *SIGGRAPH 96 Conference Proceedings*, Annual Conference Series, pages 21–30. ACM SIGGRAPH, Addison Wesley, August 1996. Held in New Orleans, Louisiana, 4–9 August 1996.

[14] P. P. Sloan, M.F. Cohen, and S.J. Gortler. Time critical lumigraph rendering. In *Proceedings of 1997 Symposium on Interactive 3D Graphics*, pages 17–23, 1997.

[15] Richard Szeliski and Heung-Yeung Shum. Creating full view panoramic mosaics and environment maps. In Turner Whitted, editor, *SIGGRAPH 97 Conference Proceedings*, Annual Conference Series, pages 251–258. ACM SIGGRAPH, Addison Wesley, August 1997.

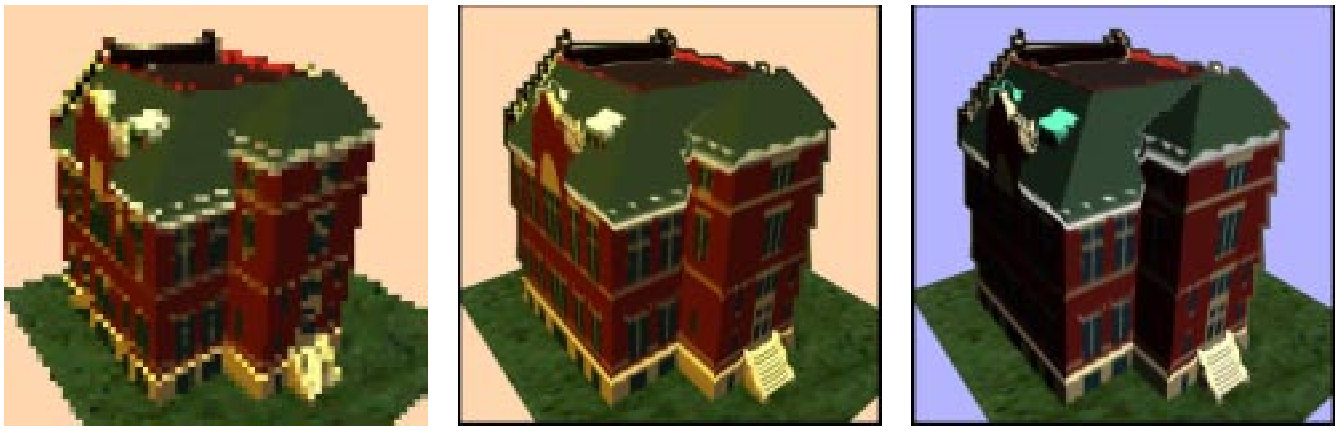
[16] Daniel N. Wood, Adam Finkelstein, John F. Hughes, Craig E. Thayer, and David H. Salesin. Multiperspective panoramas for cel animation. In Turner Whitted, editor, *SIGGRAPH 97 Conference Proceedings*, Annual Conference Series, pages 243–250. ACM SIGGRAPH, Addison Wesley, August 1997.

[17] Z. Zhu, G. Xu, and X. Lin. Constructing 3D natural scene from video sequences with vibrated motions. In *Proceedings of VRAIS '98*, pages 105–112, 1998.



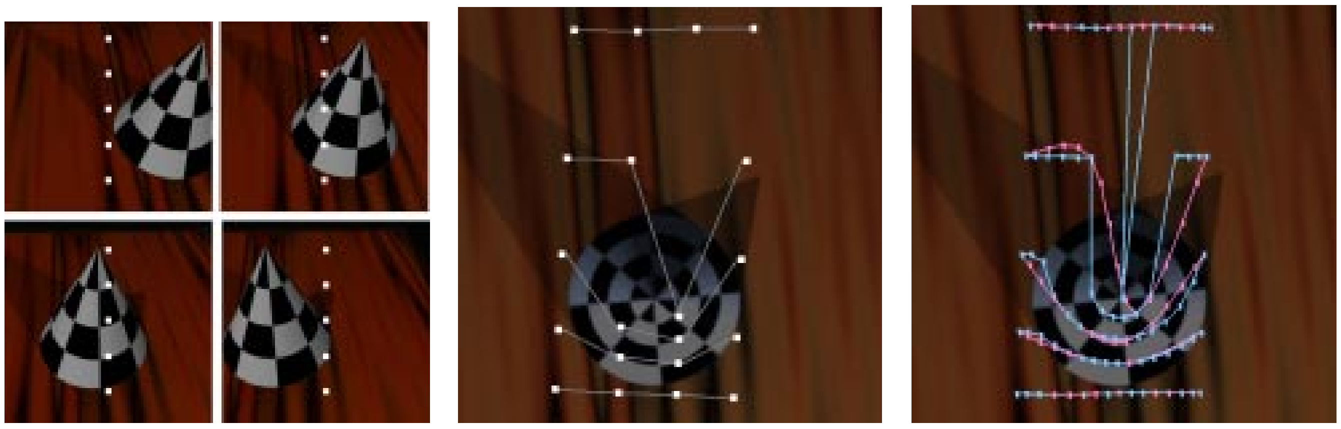
(a) (b) (c)

Figure 10: (a) Standard 3D rendering of building model with camera rays marked. (b) Oblique view of rays. (c) Warp of camera rays to an 640×640 flattened source image. (Owen Hall model courtesy of Indiana University Architect's Office.)



(a) (b) (c)

Figure 11: (a) Scene reconstruction from 64×64 sampled image points. (b) Scene reconstruction using the same 64×64 ray intersections as texture coordinates in the source image. (c) Tailored visualization achieved by altering color and lighting characteristics of source image.



(a) (b) (c)

Figure 12: Examining interpolation errors. (a) Four time keyframes of a five-pixel column of the camera ray array. (b) Locations of the keyframe pixels in the source image. (c) Trace of veridical pixel paths (blue) and approximate cubic interpolation (red) between keyframes.