Image-Based Visualization from Widely-Separated Views

Charles R. Dyer*
Department of Computer Sciences
University of Wisconsin
Madison, WI 53706
dyer@cs.wisc.edu
http://www.cs.wisc.edu/vsam

Abstract

This report describes image-based visualization research in support of video surveillance and monitoring systems. Our primary goal is to develop methods so a user can interactively visualize a 3D environment from images captured by a set of widely-separated cameras. Results include view interpolation of dynamic scenes, coarse-to-fine voxel coloring for efficient scene reconstruction, and recovering scene structure and camera motion.

1 Introduction

The goal of this project is to enhance human and automated surveillance capabilities by developing new technologies that enable scene visualization by a virtual camera. The methods being developing will enhance capabilities for monitoring areas of interest and for assessing objects' dispositions, as best determined by operator viewing preferences and task-specific targets and activities. Examples of military activities of this type include battlefield and facility visualizations and flybys, mission rehearsal and planning, site analysis, treaty monitoring, and accident analysis.

Our approach is image based in that the input is a set of images or video, and no auxiliary data sources such as terrain data or site models are usually assumed. Instead, images are leveraged to use the rich information they supply about scene structure and, by definition, photorealistic appearance. The challenge is to obtain much of the flexibility of geometry-based rendering in terms of viewing position and orientation, ability to change lighting, ability to virtually modify the scene itself, and so on.

We assume that views are captured by multiple cameras that are widely separated and arbitrarily positioned around the environment. The views from the cameras are partially overlapping so that multiple cameras view most scene points. The 3D scene can be arbitrarily complex. Output is a set of images to be viewed by a person or used as input to other image-understanding algorithms.

Three approaches are currently being studied. The first is called *view morphing* and is a method for view interpolation based on two uncalibrated views. The second approach is called *voxel coloring* and is a novel method for combining many widely-separated, calibrated views. The third approach is a new structure-from-motion method called *projected error refinement*. Results in these three areas are summarized below.

2 View Morphing

When input views are widely separately, view interpolation has required solving the difficult correspondence problem between points in the two images. To lessen this problem,

*This work is sponsored in part by the Defense Advanced Research Projects Agency (DARPA) and Rome Laboratory, Air Force Materiel Command, USAF, under agreement number F30602-97-1-0138.

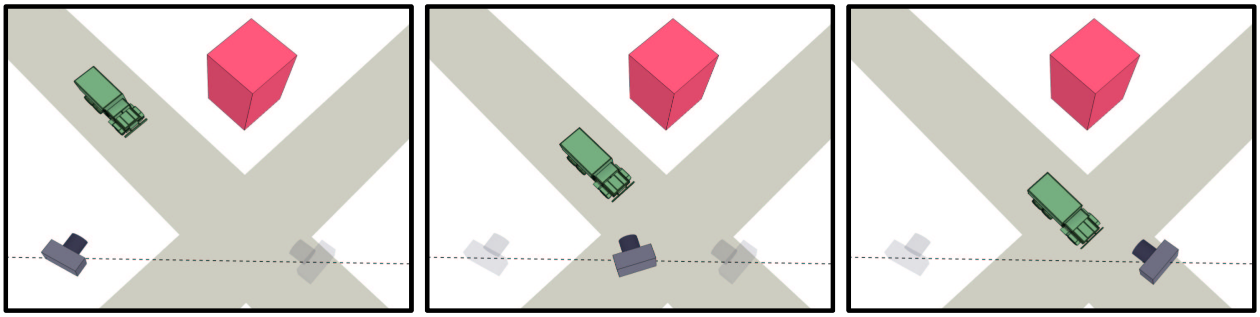

Figure 1: Dynamic view morphing scenario. Two input views (left and right) and virtual view (center) showing intermediate viewpoint and interpolated position of the moving truck.

Figure 2: Dynamic view morphing example. Input is the two images on the left and right, showing a dynamic scene from two viewpoints. Notice that the box in the foreground has moved along the edge of the table. The two center views were generated using our dynamic view morphing algorithm.

yet still generate photorealistic virtual views, we developed an extension to image morphing called view morphing [Seitz and Dyer, 1996b, Seitz and Dyer, 1996a, Seitz and Dyer, 1997c, Seitz, 1997]. The method correctly handles 3D projective camera and scene transformations, producing photorealistic image sequences for a virtual camera. The algorithm prewarps the two input images prior to computing an image morph, and then postwarps the interpolated images. No camera calibration or knowledge of 3D shape is required. The ability to synthesize changes both in viewpoint and image structure enables a wide variety of interesting 3D effects via simple image transformations.

2.1 Dynamic View Morphing

Our view morphing method, like other image-based rendering methods developed to date, assumes a static scene. If, on the other hand, the two input views are captured at different times and if the scene has changed in the interim, then we would like to produce an interpolation of both viewpoint and scene motion. That is, assuming the scene contains some number of moving objects, we would like to synthesize a sequence of images that smoothly transitions from the first image's viewpoint at time 0 to the second image's viewpoint at time 1. No knowledge of the real motions is assumed; we only require that each object's motion is equivalent to a rigid translation from its position at time 0 to its position at time 1. The algorithm, which we call *dynamic view morphing* [Manning and Dyer, 1998a, Manning and Dyer, 1998b], generates virtual views for one possible, physically-correct interpolation. Under certain conditions the objects are guaranteed to move along straight-line trajectories, or along straight-line trajectories at constant velocity, which are often the most plausible interpolations. Figure 1 illustrates an example scenario. Figure 2 shows the result of the algorithm for a scene containing a moving box. This technique can be used in applications such as filling-in gaps in video, performing smooth "hand-offs" between different cameras, creating videos from still images, performing stabiliza-

Figure 3: Environment map morphing. Two input views (left and right) and two virtual views showing the use of environment map morphing when camera motion is into the scene.

tion and compression of video, and visualizing objects during periods of obstruction.

2.2 Environment Map Morphing

An environment map is a representation of all the light that reaches a single point in space from every direction. *Environment map morphing* is a technique we are developing for view morphing between arbitrary environment maps. The reference environment maps can be acquired through panoramic mosaicing or wide-field-of-view cameras such as Columbia's OmniCamera. As with other view morphing techniques, the reference cameras can be uncalibrated and widely separated, and only a sparse set of point correspondences is necessary. The result is a virtual environment map as would be captured by a camera located on the line connecting the original two viewpoints. With several reference environment maps, collected from, say, several ground-based and aerial positions, virtual views from a large range of hard-to-reach viewpoints could be synthesized.

Environment map morphing removes the restriction inherent in static view morphing that the epipole not be visible in either view. In particular, view morphs can now portray motion into or out of a scene (see Figure 3). With dynamic view morphing, the new technique allows the vanishing points of scene objects to appear in the views.

Using environment maps overcomes limited field-of-view problems in conventional view morphing. With many more conjugate points available and with fewer non-overlapping areas, there is less distortion in the virtual views.

3 Voxel Coloring

We developed an approach called voxel coloring [Seitz and Dyer, 1997a, Seitz and Dyer, 1997b, Seitz, 1997] that reconstructs the visible surfaces in a scene by traversing the scene space in a particular order, based on the known camera positions, and colors the voxels on object surfaces. The original algorithm represented scene space as a uniform grid of voxels, and therefore the runtime of the algorithm was proportional to the number of voxels multiplied by the number of images. To make the algorithm fast enough for real-time interaction and to make the method viable for large outdoor scenes, time and space efficiency issues must be addressed. To this end, three methods have been investigated [Prock and Dyer, 1998], which are summarized below.

3.1 Texture-Mapping Hardware Support

We adapted the algorithm to exploit the texture mapping hardware that exists in many graphics workstations. By considering a layer of voxels as a plane, texture mapping can be used to rapidly project all the pixels in an image onto that layer.

3.2 Coarse-to-Fine Voxel Coloring

Another approach to speeding up voxel coloring is to take advantage of the spatial coherence of objects in the scene and the large areas of empty space in most scenes by using a multi-resolution, coarse-to-fine technique. First, a low resolution voxel space is used to quickly determine large areas of empty space. Only those voxels that may contain an object are considered at

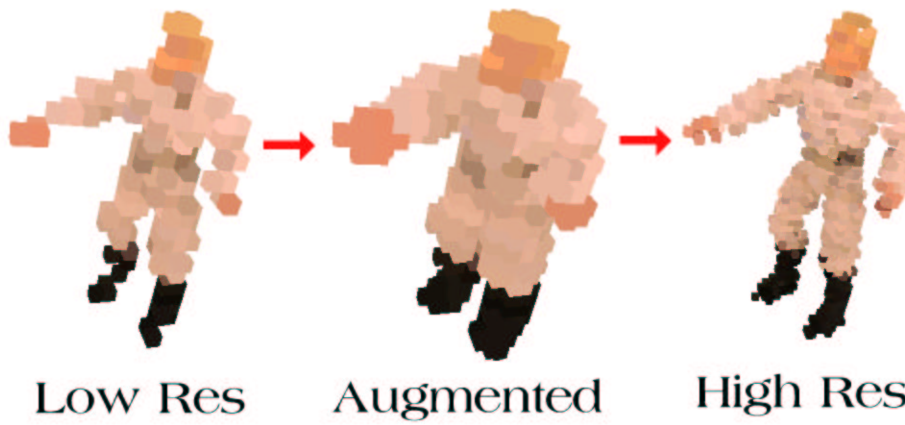

Figure 4: Multi-resolution voxel coloring. Left figure shows a set of voxels that have been colored at the previous level of resolution. Neighboring voxels are added in the center figure, which are then expanded at the next higher level of resolution. From this set of voxels a subset is determined to be colored (right figure).

the next higher level of resolution. In order to avoid missing fine details, i.e., missing small objects or surfaces within a coarse-resolution, uncolored voxel, a nearest neighbor search around the colored voxels at the next higher resolution is also performed. This process of coarse-to-fine voxel refinement is illustrated in Figure 4. In early experiments, performance was improved by 8 to 17 times for a scene containing 512 voxels on a side at the highest resolution.

3.3 Voxel Coloring Dynamic Scenes

Temporal coherence can be exploited when updating the voxel coloring for a dynamic scene. A low resolution coloring produced at the previous time step can be used as the starting point for coloring voxels based on images captured at the current time. A method for producing such a 3D "movie" has been implemented. To handle non-local temporal changes, we mark pixels or regions that have changed and then trace rays through voxel space from these pixels to identify those voxels that are candidates for coloring.

4 Projected Error Refinement for 3D Scene Reconstruction

We have developed a novel structure-from-motion technique for recovering (static) 3D scene structure and camera positions from a set of images [Bestor, 1998]. The approach overcomes some of the limitations of existing structure-from-motion methods by modeling perspective projection, allowing arbitrary camera positions, dealing with feature point outliers (i.e., errors in feature point correspondences and in feature point locations) and occlusion, and being computationally very efficient. The method is called *Projected Error Refinement* because it formulates the problem as optimizing the positions of the cameras and feature points so that the projectors (i.e., rays) of corresponding feature points come as close to intersecting as possible, defined by minimizing the angular projection error. An efficient nonlinear iterative refinement algorithm alternately refines the cameras' poses and the positions of the feature points. New feature points and images can be added or removed at any time during processing. The solution can be refined to an arbitrary precision, and the algorithm converges rapidly even when the initial estimate is poor.

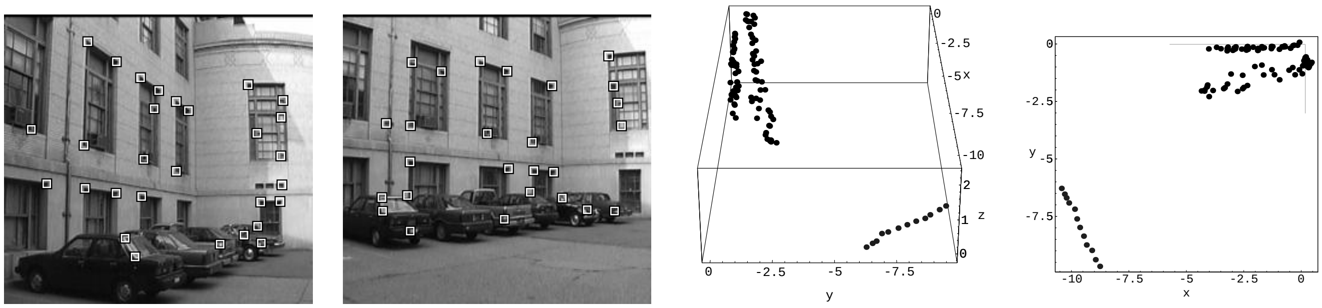

Figure 5: Structure from motion using projected error refinement. The left two images show two of the input views and detected feature points. The right two images show the result of the projected error refinement algorithm. Scene feature points are at the upper-left of the third figure and the upper-right of the right figure. The other points show the recovered camera positions.

Figure 5 shows two images of 12 taken of an outdoor scene containing significant perspective effects. 91 feature points were automatically extracted and tracked over the sequence of images, though most features were present in only a few frames. The results of the algorithm are shown in the two views in the right part of Figure 5.

References

[Bestor, 1998] G. S. Bestor. *Recovering Feature and Observer Position by Projected Error Refinement.* PhD thesis, Computer Sciences Department, University of Wisconsin-Madison, 1998.

[Manning and Dyer, 1998a] R. A. Manning and C. R. Dyer. Dynamic view morphing. Technical Report 1387, Computer Sciences Department, University of Wisconsin-Madison, 1998.

[Manning and Dyer, 1998b] R. A. Manning and C. R. Dyer. Interpolating view and scene motion by dynamic view morphing. In *Proc. 1998 Image Understanding Workshop*, 1998.

[Prock and Dyer, 1998] A. C. Prock and C. R. Dyer. Towards real-time voxel coloring. In *Proc. 1998 Image Understanding Workshop*, 1998.

[Seitz and Dyer, 1996a] S. M. Seitz and C. R. Dyer. Toward image-based scene representation using view morphing. In *Proc. 13th Int. Conf. on Pattern Recognition, Vol. I*, pages 84–89, 1996.

[Seitz and Dyer, 1996b] S. M. Seitz and C. R. Dyer. View morphing. In *Proc. SIGGRAPH 96*, pages 21–30, 1996.

[Seitz and Dyer, 1997a] S. M. Seitz and C. R. Dyer. Photorealistic scene reconstruction by voxel coloring. In *Proc. Computer Vision and Pattern Recognition Conf.*, pages 1067–1073, 1997.

[Seitz and Dyer, 1997b] S. M. Seitz and C. R. Dyer. Photorealistic scene reconstruction by voxel coloring. In *Proc. 1997 Image Understanding Workshop*, pages 935–942, 1997.

[Seitz and Dyer, 1997c] S. M. Seitz and C. R. Dyer. View Morphing: Uniquely predicting scene appearance from basis images. In *Proc. 1997 Image Understanding Workshop*, pages 881–887, 1997.

[Seitz, 1997] S. M. Seitz. *Image-Based Transformation of Viewpoint and Scene Appearance.* PhD thesis, Computer Sciences Department, University of Wisconsin-Madison, 1997.