Accelerated Pan and Zoom for Ray Traced Animation

Phil G. Gage and Sudhanhu K. Semwal

Department of Computer Science
University of Colorado, Colorado Springs
CO, 80933-7150
phil.gage@ngc.com | semwal@cs.uccs.edu

ABSTRACT

Ray tracing generates realistic images but is computationally intensive, especially when generating animation sequences. Many techniques have been developed, including exploitation of temporal coherence between successive frames to accelerate animation. This paper presents an efficient pan and zoom algorithm called the Panimation and RSVP algorithm, respectively. This new pan and zoom algorithm integrates the object animation acceleration technique of Jevans with the new pan and zoom algorithms and shows an order of magnitude performance improvement. Short animation sequences are presented to show our results.

Keywords

Ray tracing, animation, rendering.

1. INTRODUCTION

Many acceleration methods have been used to speed up ray tracing of still images, including bounding volumes, spatial subdivision and ray coherence. Jevans [22] presents an elegant method for accelerating ray traced animation using object space temporal coherence (called OSTC in this paper) to track the image areas that need to be updated for each frame. The new pan and zoom algorithms developed in this paper allow camera view changes. The new algorithms can be used separately or combined with other algorithms such as OSTC.

2 PREVIOUS WORK

Image-precision algorithms, such as ray tracing [1,5-8,11-13] have been the topic of much reseach for last several years. Figure 1 provides an example of a ray traced image rendered using our implementation. Spatial subdivision technuiques partition the scene into a grid of volumes called voxels and provide and reduce the rendering time for ray tracing. Some examples are: Glassner [8], Fujimoto [5], Semwal and Dauenhauer [13].

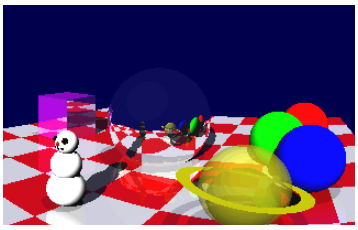


Figure 1. Example Ray Traced Image

Directional techniques [19] such as light buffers, ray coherence, ray classification and proximity clouds make use of direction during the preprocessing. Temporal coherence is discussed in [4] and object space temporal coherence in [9]. Coherence refers to local similarity, or smooth, gradual change in space or time. Chapman [4] presents an image space method similar to 2D morphing for accelerating ray traced animation using temporal coherence. Murakami [11] uses object space coherence with a static view to accelerate ray tracing by redrawing only the changed parts of the image. The ray tree for each pixel is saved, along with additional intersection data.

2.1 Object Space Temporal Coherence (OSTC)

Jevans [8] describes a simpler object space method using spatial subdivision and temporal coherence to accelerate animations with a static camera view. The OSTC method handles object animation well, however, if a light is moved or changes color, only the image areas that are in shadow from both the old and new light location will not be retraced. The shadow rays are kept in a separate bit set in each voxel to improve light animation performance. Briere [3] presents a more advanced approach that allows both color and geometry changes, but requires more complex data structures and algorithms.

3 The PanAnimation (Panimation) Algorithm

The obvious way to reuse previous frame data and accelerate pan is to shift the image and redraw only the newly exposed area. Typically, the new area scrolling into view is a small fraction of the image and only a few rows or columns of pixels need to be traced. To allow panning, pixels must cover an equal angle in both directions, implying a spherical projection with equal angle pixels in both azimuth and elevation.

Similar projections are widely used for geographic world maps [12], and several were studied for this application. The Cylindrical Equirectangular projection (Figure 2) was selected as a suitable technique. Using this cylindrical projection or a similar equal-angle method allows pan to be emulated by image shift, since each pixel represents an equal area of the viewing sphere centered at the observer. As the eye recedes an infinite distance from the image plane, both the cylindrical and standard projections approach the same orthographic projection with parallel rays. This indicates that the difference between the cylindrical and standard projections is indeed reduced by narrowing the view angle, which can be confirmed visually.

4 The Ray Sample Viewport (RSVP) Algorithm.

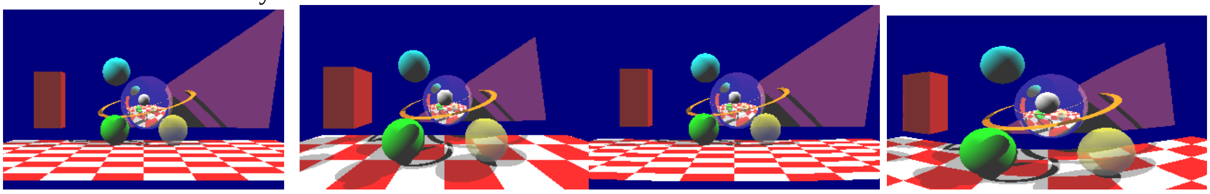
The Panimation algorithm does not support accelerated zoom, so we wanted to develop a better zoom algorithm to allow smooth zoom at any factor. A better approach for zooming out is to draw the new pixels around the edges of the image at the same resolution as the original image and resample the whole area down to a lower resolution for each frame. We call this approachh the Ray Sample Viewport (RSVP) algorithm.

Other useful effects are supported by the RSVP method. Rotate, scale, shear and more unusual changes can be produced by simple 2D transforms. Any pattern of pixels in the camera frame buffer can be resampled into the viewport image. This can accelerate pan, zoom and other transforms within the frame buffer view area.

5 Implementation

We implemented three algorithms in this paper, the Jevans OSTC animation acceleration method [9], the new Panimation algorithm and the new RSVP algorithm.

Integrating the OSTC and RSVP algorithms is simple. Instead of ray tracing the OSTC updated image areas, they are erased to the null color. When a viewport samples a null pixel, it is ray traced at that time. Since the tan function is used for the cylindrical projection, the arctan (or atan2) function can be used for the inverse transform, restoring the standard projection appearance. This fixes a major problem with the Panimation algorithm. Panimation provides fast large-scale pan but causes unwanted curvilinear distortion. Adding RSVP provides fast small-scale pan, zoom and other effects. The straight correction by our algorithm of the l checkerboard is shown in Figure 3. Java [2] was selected because of its OO features, graphic libraries and portability.



(a) Standard Projection 30° Angle b) Standard Projection 100° Angle c) Cylindrical Projection 30° Angle d) Cylindrical Projection 100° Angle

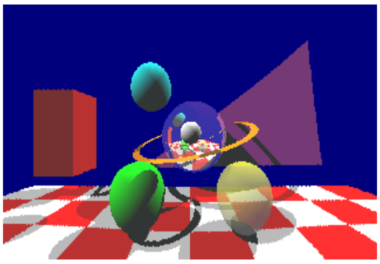
Figure 2. Standard and Cylindrical Projections

6 RESULTS AND DISCUSSION

The Test program generates a sequence of animation frame TGA files and provides Boolean flags to control animation and acceleration features. The OSTC method uses 16x16 screen regions with 256 bits in each bitset, the same as Jevans [9]. The scene contains a checkerboard ground plane, 12 polygons, 19 spheres and 2 light sources. A cylindrical camera is used with field of view 100 degrees.

Our results show that panning 3.6 degrees per frame was accelerated using Panimation by about a factor of 10. Zooming by a 2:1 ratio was accelerated using RSVP by about a factor of 10. Animation of a small bouncing ball in this scene was accelerated using OSTC by about a factor of 2. Simultaneous pan, zoom and animation caused complex view changes and was accelerated by about a factor of 2. These techniques can be used to accelerate pan, zoom and animation simultaneously. These results are very encouraging and indicate that even greater performance can be achieved for slower pan and zoom or longer animation sequences. The resulting 100 frame animation sequences were converted to animated GIF files and are available for review.

Figure 4 shows a test image where the black areas are rectangles erased to the null color by pan and object animation. The narrow black strips at the top and right are caused by the camera panning slightly up and right, shifting the image down and left. Only the black areas need to be ray traced, the majority of the image is reused from the previous frame, showing the potential for image space acceleration even with simultaneous pan and object animation. Figure 5 shows a camera image and two smaller viewport images extracted from the camera image. In the camera image, only the pixels sampled by the viewports were ray traced, the black areas are null pixels not traced.



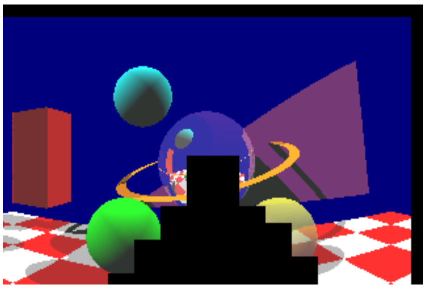
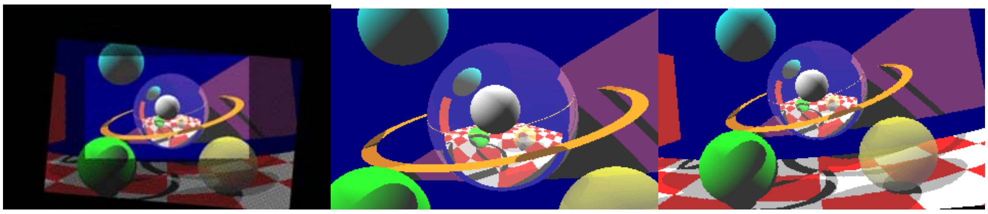


Figure 3: Corrected Cylindrical distortion (above)
Figure 4. Simultaneous Pan and Animation Image (right)



a.) 640x480 camera image with sampled pixels. b.) 320x200 sampled zoom viewport c.) 320x200 zoom and rotate viewport

Figure 5. Camera and Viewport Images

7 FUTURE WORK

This paper has focused on acceleration, but many other features could be added. Realism could be improved by interpolating sampled pixel colors, antialiasing, distribution ray tracing and improved texture mapping. It would be good to trigger auto-pan and auto-zoom events when a viewport hits a buffer edge or size limit. Other view projections such as Fisheye projections [14] would be interesting to implement.

8 CONCLUSION

This paper has presented the history and theory of ray tracing and animation acceleration. The static camera limitation of Jevans [9] and other earlier work was noted and animage space view acceleration algorithm was developed to accelerate pan, zoom and other effects from a fixed viewpoint. Fast pan is performed using image shifting with an equal-angle view projection. Fast zoom and other effects are performed using a sampling viewport and 2D transforms. Implementation details, more comparisons and rendering results of our implementation are in [15].

A Java [2] ray tracer was implemented to test the new algorithms both for animation and interaction. An object space temporal coherence animation acceleration method based on the Jevans [9] technique was also implemented and integrated with the new algorithms. The resulting system supports ray traced animation with acceleration for both object changes and camera field of view changes. A generalized animation methodology was developed that could be extended to include reprojection, saved ray trees and other accelerations.

The timing results show performance gains of an order of magnitude for the new pan and zoom algorithms. This work has been successful, personally rewarding, and has many applications and possibilities for future work. Possible uses include animation, interactive scene editing, virtual reality, video games, simulations, computer-aided design and 3D data visualization.

9 REFERENCES

[1] Amantides, J., Woo, A., “A Fast Voxel Traversal Algorithm for Ray Tracing,” Proceedings of Eurographics, 1987, pp. 3-10

[2] Arnold, K., Gosling, J., The Java Programming Language, Addison-Wesley, 1998

[3] Briere, N., Poulin, P., “Hierarchical View-dependent Structures for Interactive Scene Manipulation,” Computer Graphics Proceedings, 1996, pp. 83-90

[4] Chapman, J., Calvert, T., “Exploiting Temporal Coherence in Ray Tracing,” Proceedings of Graphics Interface 90, 1990, pp. 196-204

[5] Fujimoto, A., Tanaka, T., Iwata, K., “ARTS: Accelerated Ray Tracing System,” IEEE Computer Graphics & Applications, Apr 1986, pp. 16-26

[6] Gassner, A., “Space Subdivision for Fast Ray Tracing,” IEEE Computer Graphics & Applications, Oct 1984, Vol. 4, No. 10, pp. 15-22

[7] Glassner, A., An Introduction to Ray Tracing, Academic Press, 1989

[8] Jevans, D., “Adaptive Voxel Subdivision for Ray Tracing,” Proceedings of Graphics Interface 89, 1989, pp. 164-172

[9] Jevans, D., “Object Space Temporal Coherence for Ray Tracing,” Proceedings of Graphics Interface 92, May 1992, pp. 176-183

[10] Lext, J., Assarsson, U., Moller, T., “BART: A Benchmark for Animated Ray Tracing,” IEEE Computer Graphics & Applications, March 2001, Vol. 21, No. 2, pp. 22-31

[11] Murakami, K., Hirota, K., “Incremental Ray Tracing,” Proceedings of Eurographics Workshop on Photosimulation, June 1990, pp. 15-29

[12] Musgrave, K., “A Panoramic Virtual Screen for Ray Tracing,” Graphic Gems III, Academic Press, 1992, pp. 288-294

[13] Semwal, S. and Dauenhauer D., “Approximate Ray Tracing,“ Proceedings of the Graphics Interface, Halifax, Nova Scotia, pp. 75-82 (1990).

[14] Glaeser, G., Groller, E., “Fast Generation of Curved Perspectives for Ultra-Wide-Angle Lenses in VR Applications,” The Visual Computer, July 1999

[15] Phil Gage “View Coherecnce Acceleration for Ray Traced Animation,” MS Thesis, Univesity of Colorado, Colorado Springs, CO, US, directed by Professor Semwal, pp. 1-75. 2002.