Frameless Rendering: Double Buffering Considered Harmful

Gary Bishop, Henry Fuchs, Leonard McMillan, and Ellen J. Scher Zagier

Department of Computer Science, UNC Chapel Hill

Abstract

The use of double-buffered displays, in which the previous image is displayed until the next image is complete, can impair the interactivity of systems that require tight coupling between the human user and the computer. We are experimenting with an alternate rendering strategy that computes each pixel based on the most recent input (i.e., view and object positions) and immediately updates the pixel on the display. We avoid the image tearing normally associated with single-buffered displays by randomizing the order in which pixels are updated. The resulting image sequences give the impression of moving continuously, with a rough approximation of motion blur, rather than jerking between discrete positions.

We have demonstrated the effectiveness of this *frameless rendering* method with a simulation that shows conventional double-buffering side-by-side with frameless rendering. Both methods are allowed the same computation budget, but the double-buffered display only updates after all pixels are computed while the frameless rendering display updates pixels as they are computed. The frameless rendering display exhibits fluid motion while the double-buffered display jumps from frame to frame. The randomized sampling inherent in frameless rendering means that we cannot take advantage of image and object coherence properties that are important to current polygon renderers, but for renderers based on tracing independent rays the added cost is small.

CR Descriptors: I.3.1 [**Computer Graphics**]: Picture/image generation --- *Display algorithms*; I.3.6 [**Computer Graphics**]: Methodology and Techniques.

The Motivation: fast interaction

In conventional double-buffered systems each new image is one update interval old when it is first displayed, then it is held for another complete update interval while the next image is computed. For example, at an update rate of 5Hz an image will be 400 milliseconds old before it is replaced with a new image that is already 200 milliseconds old. In interactive systems, such large delays may cause users to repeatedly overshoot on the inputs resulting in operator induced oscillations. In head-mounted displays, delays cause the virtual world to appear unstable and to swim around as the user moves.

In a 1986 paper on Adaptive Refinement, Bergman, Fuchs, Grant, and Spach[1] suggest that the impact of these delays can be reduced by rapidly displaying a crude image when the scene is changing and progressing through higher quality renderings when the inputs are constant. This approach allows timely updates during changes and high-quality images as soon as possible. Their paper also suggests the possibility of a "golden thread", a single step that, if repeated a few times, will generate a crude image, and when repeated further will result in incrementally higher quality images.

The Idea: randomized immediate pixel update

We are experimenting with an approximation to that "golden thread" that computes a fraction of the pixels in the image and immediately updates them on the display, rather than computing a complete image of lower quality. The order of pixel update is randomized to avoid image tearing and to allow the appearance of simultaneous update everywhere in the image. For each pixel, the latest available input parameters are used so that each pixel accurately represents the time at which it is computed and displayed (this can only be approximated on current raster display systems because of their sequential scan of pixels).

Frames, as in movies and conventional computer displays, lose their meaning in this approach to rendering. When the scene is static, the image quickly converges to the current system state. When the view is changing or objects in the scene are moving, the image continuously blends toward the correct value for current time. At all times, the image on the display represents the most current system state with the available computing power. The image appears blurry, but with smooth and continuous motion. The blur produced is dependent on the rate of change and is a crude approximation to motion blur. Our frameless rendering method can be thought of as a grossly under-sampled version of Cook's Distributed Ray Tracing[2].

If the instantaneous changes in the scene are large (as in a cut from one scene to another) the image momentarily becomes a confusing mixture of past and present images. In order to compensate, the display system might switch to lower resolution or to double-buffering during rapid scene changes. We believe that such dramatic scene changes are rare in most interactive graphics applications and in synthetic environments.

Figure 1 illustrates frameless rendering with a sequence of still images. It should be noted that a user of a frameless rendering system would never see (for long) partially updated static images like those in the figure; instead, the partially updated images only occur when the scene is changing. When motion stops, the system would quickly update all the pixels with the newest values. The bottom row of the figure depicts the intermediate images shown by the frameless display at the times illustrated; the row above depicts the frames from a double-buffered display.

An Experiment: side-by-side comparison

In order to evaluate the visual effect of this method we implemented an experiment that shows a side-by-side comparison of conventional double-buffering with a simulation of frameless rendering. Each of the displays is given the same computation budget (each is allowed to compute pixels at the same rate). The double-buffered display only switches to the new image after all pixels have been updated; the frameless rendering simulation

mail: Sitterson Hall CB3175, Chapel Hill, NC 27599-3175; phone: (919) 962-1700; email: lastname@cs.unc.edu.

updates the pixels as soon as possible (though they only become visible at the next video refresh). Thus, the double-buffered display switches from frame to frame at a 5 Hz rate, while the frameless renderer updates about 16% of the pixels at each video frame.

This experiment was done as a proof of concept using mostly tools that were already in place. A set of complete frames were rendered and stored on disk using Rayshade[3], a public domain ray tracer. The disk files were then processed sequentially by replacing a randomly selected fraction of the pixels in the output image with pixels from the current input image file. We used a table to choose random pixels without replacement so update of all pixels was guaranteed. The resulting images were displayed on a conventional workstation using a program that sequentially displays precomputed images. The simulation of the double-buffered display was implemented by switching between the appropriate original images.

The demonstration clearly contrasted the jerky motion of the double-buffered display with the fluid motion of the frameless rendering. The frameless rendering appeared blurred, and careful examination showed that high-contrast edges were ragged. This effect may have been exaggerated by the small size (320 by 240) of the images used in the simulation to allow high update rates on the workstation display.

Conclusions

We have only begun to explore the potential of this rendering paradigm. An important next step will be to characterize the range of update rates and image rate of change over which the method is applicable. Another area for investigation will be display alternatives for situations in which the instantaneous display change is sufficiently large to produce confusing images.

It may be possible to extend the method to allow inclusion of antialiasing methods based on jittered sampling, thus allowing smooth motion during changes and high-quality images when stationary. This would be one step closer to the "golden thread" mentioned earlier.

It may also be possible to optimize image update based on how the view is changing and on how objects are moving. For example, the samples might be concentrated at moving edges in the image, or old pixels might be moved to new positions before they are combined with newly computed pixels.

Frameless rendering is a simple idea which can be applied to a variety of systems to produce an apparently dramatic improvement in the smoothness of interaction. We are encouraged by these preliminary results and plan to implement frameless rendering in head-mounted display systems for further evaluation.

Acknowledgments

Support for this research was provided in part by NSF/ARPA Science and Technology Center for Computer Graphics and Scientific Visualization, (NSF Cooperative Agreement #ASC 8920219) and by ARPA Contract DABT63-93-C-C048 "Enabling Technologies and Application Demonstrations for Synthetic Environments."

References

[1] Bergman, Larry, Henry Fuchs, Eric Grant, and Susan Spach. Image Rendering by Adaptive Refinement. Proceedings of SIGGRAPH '86. In *Computer Graphics* 20, 4, pp. 29-37, August 1986.

[2] Cook, Robert L., Thomas Porter, Loren Carpenter. Distributed Ray Tracing. Proceedings of SIGGRAPH '84. In *Computer Graphics* 18, 3, pp. 137-145, July 1984.

[3] Kolb, Craig E., *Rayshade User's Guide and Reference Manual,* Draft 0.4, January 10, 1992.

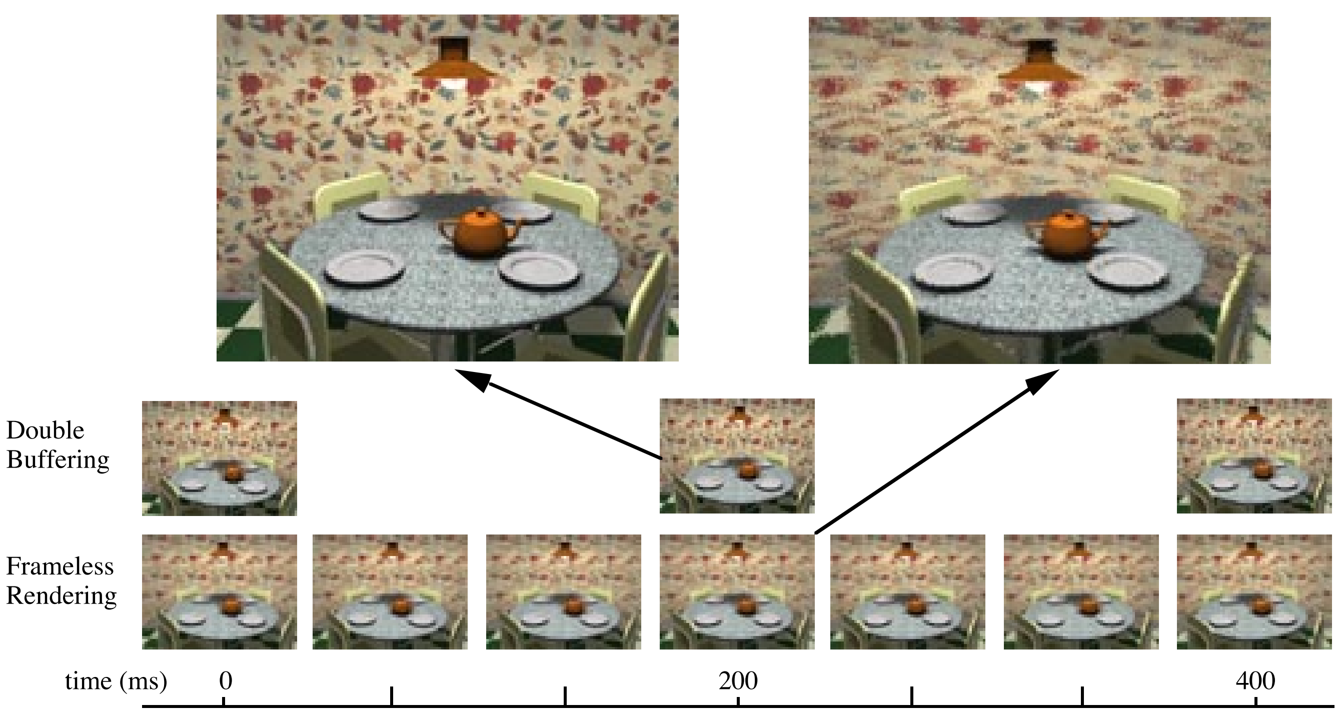

Figure 1. The bottom row shows 7 frames of a 15 Hz frameless rendering sequence with 33% of the pixels updated in each frame. The middle row shows 3 frames of a double-buffering sequence updated at 5 Hz.